DevSuite Admin Guide

Author: TechExcel co.Ltd

Date:

Table of Content

DevSuite Admin Guide 19
System Settings Administration 20
Chapter 1 System-Level User Administration and Li 20
1 Chapter 1- System-Level User Administration and License Management 20
1.1 Understanding System-Level User Administration 20
1.2 Administering User Licenses 20
1.2.1 Understanding License Edition 20
1.2.2 Understanding Site License 21
1.2.3 Understanding License Class 21
1.2.4 Understanding License Types 22
1.2.5 Loading DevSuite License File 22
1.2.6 Tracking DevSuite LIcenses 23
1.3 Administering User Accounts 24
1.3.1 Creating User Accounts 24
1.4 Administering User Logins 26
1.4.1 Viewing the Login Status Of Project Members 26
1.4.2 Managing Idle Timeout Settings 27
1.4.3 Viewing Login History Logs 27
1.5 Administering Divisions 28
1.5.1 Managing Divisions 29
Chapter 2 LDAP Directory Server Integration 30
2 Chapter 2- LDAP Directory Server Integration 30
2.1 Administering Directory Server Integration 30
2.2 Administering LDAP Directory Server Registration 31
2.3 Administering Field-Attribute Matching Identification 33
2.4 Administering Field and Attribute Mapping 33
2.5 Administering LDAP Directory Server Authentication 33
2.6 Importing User Accounts from a LDAP Server 33
2.6.1 Filtering User Accounts in the LDAP Import Wizard 34
Chapter 3 General System Settings Administration 35
3 Chapter 3- General System Settings Administration 35
3.1 Administering Standard Lookup Fields 35
3.1.1 Defining Standard Lookup Fields and Field Values 35
3.2 Multilingual Support and Localization Administration 36
3.3 Administering System Date and Time Settings 36
3.4 Administering Keyword Searching 38
3.5 Administering Time Categories 38
3.6 Administering LinkPlus Integration 39
3.7 Managing DevSuite Admin Reports 39
3.7.1 User Information Report 39
3.7.2 Project information Report 40
3.7.3 Admin Change Log Report 40
3.8 Administering Welcome and Warning System Messages 40
Chapter 4 Multiple Site Administration 41
4 Chapter 4- Multiple Site Administration 41
4.1 Understanding Site Administration 41
4.2 Administering DevSuite Sites 41
Chapter 5 DevSuite Web Administration 43
5 Chapter 5- DevSuite Web Administration 43
5.1 Understanding DevSuite Web Server Administration 44
5.2 Administering the DevSuite Web Server 44

5.2.1 Administering Multiple DevSuite Web Servers 45
5.3 DevTrack Web Administration 46
5.3.1 Defining DevTrack Web Connections 46
5.3.2 Defining DevTrack Web Issue List and Reporting Options 47
5.3.3 Administering Interproject Switching 47
5.3.4 Defining DevTrack Web Titles, Welcome Messages and Logout URL 47
5.4 KnowledgeWise Web Administration 48
5.5 Beta Customer Portal Administration 48
5.6 Administering Image and Multimedia Attachment Settings 48
5.6.1 Defining DevTrack Web Image Attachment Settings 48
5.6.2 Defining DevTrack Web Movies and Sound Attachments Settings 49
5.7 Administering DevTrack Web Authentication and Access Controls 49
5.7.1 Configuring DevTrack Login and Password Authentication 49
5.7.2 Configuring Windows NT User Authentication 49
5.7.3 Configuring Third Party Authentication—DevTrack Web SSO 50
Chapter 6 DevSuite Mail Service Administration 50
6 Chapter 6- DevSuite Mail Service Administration 50
6.1 Understanding DevSuite Mail Services 50
6.2 Administering DevSuite Mail Services 51
6.2.1 Installing and Configuring the Notification/ Retrieval Service 51
6.2.2 Administering E-mail Error Handling 52
6.2.2.1 Defining Email Trace Levels 53
6.2.2.2 Defining Mail Service Sensitivity Settings 53
6.2.2.3 Defining Email Error Log Files 53
6.2.2.4 Defining Email Service Error Reports 54
6.2.3 Troubleshooting DevTrack Notification Service Errors 54
Chapter 7 Product Release Management Administrati 55
7 Chapter 7- Product Release Management Administration 55
7.1 Understanding Produce Release Management Administration 55
7.2 Administering the Product Tree 56
7.2.1 Administering Product Catogories 56
7.2.2 Administering Products 57
7.2.3 Administering Versions 58
7.2.4 Administering Builds 59
7.3 Administering Product, Version, and Build Release Statuses 60
7.3.1 Defining the Product Release Lifecycle 61
7.3.2 Administering the Version Release Lifecycle 62
7.3.3 Administering Build Release Lifecycle 63
7.4 Administering Product Implementation Modules 64
7.4.1 Adding Product Implementation Modules 64
7.5 Administering Product Version Options 64
7.5.1 Defining Version Options 64
7.5.2 Applying Version Options to Versions 65
Chapter 8 Time and Cost Tracking Administration 65
8 Chapter 8- Time and Cost Tracking Administration 65
8.1 Understanding Time and Cost Administration 65
8.2 Administering Time Groups and Time Categories 65
8.2.1 Creating Time Groups 66
8.2.2 Editing Time Groups 66
8.2.3 Creating Time Categories 67
8.2.4 Editing Time Categories 67
8.3 Understanding Time and Cost Administration 68
8.4 Administering Time Groups and Time Categories 68
8.4.1 Creating Time Groups 68
8.4.2 Editing Time Groups 69
8.4.3 Creating Time Categories 69

8.4.4 Editing Time Categories 70
KnowledgeWise Projects Administration 71
Chapter 1 What's new 71
Chapter 2 KnowledgeWise Concepts 71
2 Chapter 1- KnowledgeWise Concepts 71
2.1 Understanding TechExcel KnowledgeWise and DevSuite 71
2.1.1 Understanding DevSuite ALM Solutions 71
2.1.2 Understanding TechExcel Project Hierarchy 71
2.2 Understanding DevSuite Administration 72
2.2.1 Understanding DevSuite Site (System-Level) Administration 72
2.2.2 Understanding Project Administration 73
2.3 Understanding DevSuite Sample Projects and Templates 73
2.3.1 Understanding Sample Projects 73
2.3.2 Understanding Project Templates 73
2.3.3 Understanding Solution Templates 73
Chapter 3 Project Administration 74
3 CHAPTER 2 Project Administration 74
3.1 Understanding KnowledgeWise Project Administration 74
3.1.1 Understanding Projects 74
3.2 Understanding the DevSuite Admin Workspace 74
3.2.1 Understanding Menu Bar Commands 74
3.2.2 Understanding the Tool Bar Commands 75
3.2.3 Logging into DevSuite Admin 75
3.2.4 Opening KnowledgeWise Projects 75
3.2.5 Closing DevTrack Projects 76
3.2.6 Understanding Keyboard Shortcuts 76
3.3 Administering KnowledgeWise Projects 76
3.3.1 Creating Knowledge Management Projects 77
3.3.2 Opening KnowledgeWise Projects 78
3.3.3 Backing Up KnowledgeWise Projects 78
3.3.4 Restoring KnowledgeWise Projects 79
3.3.5 Creating Projects Based from Backed Up Projects 79
3.3.6 Copying KnowledgeWise Project Settings 79
3.3.7 Understanding Active and Inactive Projects 80
3.3.7.1 Active Projects 80
3.3.7.2 Inactive Projects 80
3.3.8 Deleting KnowledgeWise Projects 80
3.3.9 Defining Project Names and Descriptions 80
3.3.10 Defining Knowledge Item ID Prefixes 81
3.3.11 Defining Projects as Active and Inactive 81
3.3.12 Defining Name Formatting 81
3.3.13 Defining Project Time Zone Settings 81
Chapter 4 Team Administration 81
4 CHAPTER 3 Team Administration 81
4.1 Understanding Team Administration 82
4.2 Administering Account Types, Access Controls, and Privileges 82
4.2.1 Defining Account Types 82
4.2.2 Administering Knowledge Management Privileges 82
4.2.3 Administering Invisible Pages and Fields 83
4.2.4 Administering Read-Only Pages and Fields 83
4.2.5 Administering Report Access Controls and Privileges 84
4.3 Administering Project Team Members 84
4.3.1 Adding User Accounts to Projects 84
4.3.2 Removing User Accounts from Projects 85
4.4 Administering Team Groups 85
4.4.1 Defining Team Groups 85

4.4.2 Adding and Removing Project Members to Team Groups 86
4.5 Administering Group Folders 86
4.5.1 Defining Group Folders 86
Chapter 5 Knowledge Folder Administration 87
5 CHAPTER 4 Knowledge Folder Administration 87
5.1 Administering Knowledge Item Folders 87
5.2 Administering Knowledge Folder Function Pages 87
5.2.1 Administering the Add Knowledge Folder Page 87
5.2.2 Administering the Edit Knowledge Folder Page 88
5.3 Administering Knowledge Folder System Pages 89
5.3.1 Defining System Page Titles 89
5.3.2 Customizing the Applicable Product System Page 89
5.3.3 Customizing the Knowledge Folder Applicable Owner Page 90
5.3.4 Customizing the Access Control System Page 90
5.3.5 Customizing the Folder Order System Page 90
5.4 Administering Knowledge Folder System Fields 91
5.4.1 Displaying Access Controls in Custom Pages 91
Chapter 6 Knowledge Item Administration abc 92
6 CHAPTER 5 Knowledge Item Administration abc 92
6.1 Understanding KnowledgeWise Knowledge Items 92
6.2 Administering Knowledge Item Function Pages 92
6.2.1 Administering the Add Knowledge Item Function Page 92
6.2.2 Administering the Edit Knowledge Item Function Page 93
6.2.3 Administering the Knowledge Item Detail Function Page 93
6.2.4 Administering the Forward Knowledge Item Function Page 93
6.3 Administering Knowledge Item System Pages 94
6.3.1 Administering the Knowledge History System Page 94
6.3.2 Administering the Version System Page 95
6.3.3 Administering the Knowledge Event System Page 95
6.3.4 Administering the Knowledge Item All Links System Page 95
6.3.5 Administering the Linked Knowledge System Page 95
6.4 Administering Knowledge Item Fields and Controls 95
6.4.1 Understanding Knowledge Item System Fields 95
6.5 Administering Knowledge Item List Columns 96
6.5.1 Displaying Knowledge System Fields in the Knowledge List Panel 96
6.5.2 Customizing Knowledge Item List Indicator Icons 96
6.5.3 Customizing Knowledge Item List Font Colors and Weights 97
6.6 Administering Knowledge Item-Level Knowledge Management 97
6.6.1 Enabling Spell Checking in the Description Control 98
6.6.2 Enabling HTML Formatting in the Description Control 98
Chapter 7 Secured Access Level Administration 99
7 CHAPTER 6 Secured Access Level Administration 99
7.1 Understanding Secured Knowledge Access Level Security 99
7.1.1 Understanding Access Level Conflict Resolution Rules 100
7.1.2 Comparison with Public and Protected Access Controls 100
7.2 Administering Knowledge Folder Secured Access Levels 100
7.2.1 Defining Knowledge Folder Access Levels 100
7.2.2 Defining User Account Access Levels 101
7.3 Administering Knowledge Item Access Levels 101
7.3.1 Defining Knowledge Folder Access Levels 101
7.3.2 Defining User Account Access Levels 102
7.4 Administering Non-Member Access Levels 102
Chapter 8 Page, Field, and Control Administration 102
8 CHAPTER 7 Page, Field, and Control Administration 103
8.1 Understanding KnowledgeWise Client GUI Customization 103
8.2 Administering System Pages 103

8.3 Administering Custom Pages 104
8.3.1 Creating Custom Pages 104
8.3.2 Adding Controls to Custom Pages 104
8.3.3 Removing Controls From Custom Pages 105
8.3.4 Ordering Controls in Custom Pages 105
8.3.5 Adding Static Labels 105
8.3.6 Understanding Custom Page Form Design Commands 105
8.3.7 Aligning Custom Controls and Labels 106
8.3.8 Resizing Controls and Labels 106
8.4 Administering System Controls 106
8.5 Administering Custom Fields and Controls 107
8.5.1 Understanding Control Types 107
8.5.2 Customizing Check Box Controls 109
8.5.3 Customizing Date-Time Controls 109
8.5.4 Customizing Short Text Box Controls 109
8.5.5 Customizing Multiple-line Text Box Controls 110
8.5.6 Customizing Composed Field Controls 110
8.6 Customizing Selection and Combination Box Controls 111
8.6.1 Defining Selection List Options (Manually) 111
8.6.2 Defining Selection List Options (From Lookup Fields) 111
8.6.3 Defining Parent-Child Relationships Between Controls 112
8.6.4 Customizing Autogrow Combo Box Controls 112
8.6.5 Customizing Combo Box Controls 113
8.6.6 Customizing Dropdown List Controls 113
8.6.7 Customizing Multiple Selection Controls 114
8.6.8 New Section 114
8.7 Administering Calculation Fields and Controls 114
8.7.1 Defining Numerical Calculation Controls 114
8.7.2 Defining Date Calculation Controls 116
8.7.3 Defining Selection Calculation Factors 116
Chapter 9 Workflow Administration 117
9 CHAPTER 8 Workflow Administration 117
9.1 Administering Knowledge Item Workflow 117
9.1.1 Enabling Support for Knowledge Workflow Rules 117
9.1.2 Defining Workflow States 118
9.1.3 Defining Workflow Transitions 118
9.1.4 Defining State-Based Access Controls 119
9.1.5 Defining Transition-Based Access Controls 119
9.1.6 Defining State-Based Applicable Owners and Default Owners 119
9.1.7 Defining Knowledge State Versioning Rules 120
9.1.8 Defining State-Based Read-Only Knowledge Item Fields 120
9.1.9 Defining State-Based Mandatory Knowledge Item Fields 121
9.1.10 Defining State-Based Invisible Knowledge Item Fields 121
9.2 Administering Knowledge Item State Groups 122
9.2.1 Enabling Knowledge Item State Group Support 122
9.2.2 Creating New Knowledge Item State Groups 122
9.2.3 Defining Knowledge Item State Groups 122
Chapter 10 Event Administration 122
10 CHAPTER 9 Event Administration 122
10.1 Administering KnowledgeWise Events 122
10.2 Administering Event Templates 123
10.2.1 Defining Default Event Titles and Descriptions 123
10.2.2 Defining Default Event Workflow States 123
10.2.3 Defining Default Event Owners 124
10.3 Administering Event Workflow 124
10.3.1 Creating Event States 125
10.3.2 Editing Event States 125

10.3.3 Deleting Event States 125
10.4 Administering Event Access Controls 126
DevSpec Projects Administration 127
Chapter 1 What's new 127
1 Full Unicode Support 127
1.1 New Section 127
Chapter 2 DevSpec Concepts 127
2 DevSpec Concepts 127
2.1 Understanding TechExcel DevSpec and DevSuite 127
2.1.1 Understanding DevSuite ALM Solutions 127
2.1.2 Understanding TechExcel Project Hierarchy 128
2.2 Understanding DevSuite Administration 129
2.2.1 Understanding DevSuite Site (System-Level) Administration 129
2.2.2 Understanding Project Administration 129
2.3 Understanding DevSuite Sample Projects and Templates 130
2.3.1 Understanding Sample Projects 130
2.3.2 Understanding Project Templates 130
2.3.3 Understanding Solution Templates 130
Chapter 3 Project Administration 130
3 Chapter 2- Project Administration 130
3.1 Understanding DevSpec Project Administration 130
3.1.1 Understanding Projects 131
3.2 Understanding the DevSuite Admin Workspace 131
3.2.1 Understanding Menu Bar Commands 131
3.2.2 Understanding the Tool Bar Commands 132
3.2.3 Logging into DevSuite Admin 132
3.2.4 Opening DevSpec Projects 133
3.2.5 Closing DevTrack Projects 133
3.2.6 Understanding Keyboard Shortcuts 133
3.3 Administering DevSpec Projects 134
3.3.1 Creating DevSpec Projects 134
3.3.2 Backing Up DevSpec Projects 135
3.3.3 Restoring DevSpec Projects 135
3.3.4 Creating Projects Based from Backed Up Projects 135
3.3.5 Copying DevTrack Project Settings 136
3.3.6 Understanding Active and Inactive Projects 136
3.3.6.1 Active Projects 136
3.3.6.2 Inactive Projects 137
3.3.7 Deleting DevTrack Projects 137
3.3.8 Logging into DevSuite Admin 137
3.3.9 Creating DevSpec Projects 137
Chapter 4 Team Administration 137
4 Chapter 3- Team Administration 137
4.1 Understanding Team Administration 137
4.2 Administering Account Types, Privileges, and Access Controls 138
4.2.1 Defining Account Types 138
4.2.2 Granting Privileges to an Account Type 139
4.2.3 Defining Page-Level and Field-Level Access Rights 139
4.2.4 Renaming Account Types 139
4.2.5 Deleting Account Types 139
4.2.6 Understanding Specification View Privileges and Access Controls 140
4.2.6.1 Privileges 140
4.2.6.2 Access Control 140
4.2.7 Understanding Requirement View Privileges and Access Controls 141
4.2.7.1 Privileges 141

4.2.7.2 Access Control 141
4.2.8 Understanding Change Request View Privileges 141
4.2.9 Understanding Report View Privileges 142
4.2.10 Understanding Invisible Pages and Fields 142
4.2.11 Understanding Read-Only Pages and Fields 142
4.3 Administering Secured Folder Access Levels 142
4.3.1 Understanding Secured Folder Access Level Conflict Resolution 143
4.3.1.1 Access type prioritization 143
4.3.1.2 User account prioritization 143
4.3.2 Defining Secured Work Item Access Levels 143
4.3.3 Defining Secured Folder Access Levels 144
4.3.4 Updating Secured Folder Access Levels for All Users 144
Chapter 5 Specification Administration 145
5 Chapter 4- Specification Administration 145
5.1 Understanding DevSpec Specifications 145
5.1.1 Understanding Specification Administration 145
5.1.2 Enabling Specifications 145
5.1.3 Defining Specification ID Prefixes 146
5.1.4 Defining Specification Starting Numbers 146
5.2 Administering Specification Folders 146
5.2.1 Customizing the Add Specification Folder Function Page 146
5.2.2 Administering the Edit Specification Folder Function Page 146
5.2.3 Administering Specification Folder System Pages 147
5.2.4 Customizing the Spec Folder Applicable Owner System Page 147
5.2.5 Customizing the Change Log System Page 148
5.2.6 Customizing the Folder Order System Page 148
5.2.7 Customizing the Access Control System Page 148
5.2.8 Displaying Access Controls in Custom Pages 148
5.3 Administering Specifications 149
5.3.1 Administering the Add Specification Function Page 149
5.3.2 Administering the Edit Specification Function Page 149
5.3.3 Administering the Specification Detail Function Page 150
5.3.4 Administering the Forward Specification Function Page 150
5.3.5 Administering the Version Function Page 150
5.3.6 Administering the Specification Template Function Page 150
5.3.7 Administering the Group Change Specification Function Page 151
5.3.8 Administering the Group Forward Specification Function Page 151
5.3.9 Administering the Specification Change Function Page 151
5.3.10 Administering the DevSpec User Function Page 152
Chapter 6 Change Requests Administration 152
6 Chapter 5- Change Requests Administration 152
6.1 Understanding DevSpec Change Request Administration 152
6.2 Administering Change Request Folders 152
6.2.1 Administering Change Request Folder Description Page 153
6.2.1.1 System Fields 153
6.3 Administering Change Requests 153
6.3.1 Administering Change Request System Pages 153
6.3.2 Administering the Edit Change Request Function Page 153
6.3.3 Administering the Change Request Detail Function Page 154
6.3.4 Administering the Forward Change Request Function Page 154
6.3.5 Administering the Applicable Product System Page 154
6.3.6 Administering the Change Request History System Page 154
6.4 Administering Change Request Workflow 155
6.4.1 Enabling Change Request Workflow Options 155
6.4.2 Creating Change Request Workflow States 156
6.4.3 Editing Change Request States 156
6.4.4 Deleting Change Request States 157

6.4.5 Defining Change Request State Transitions 157
6.4.6 Defining Change Request State Transition Rules 157
6.4.7 Defining Change Request State Applicable Owners 158
6.4.8 Defining Default Owners by Change Request State 158
6.4.9 Defining State-Based Read-Only Change Request Fields 159
6.4.10 Defining State-Based Mandatory Change Request Fields 159
6.4.11 Defining State-Based Invisible Change Request Fields 159
Chapter 7 DevSpec GUI Gustomization 160
7 Chapter 6- DevSpec GUI Gustomization 160
7.1 Understanding DevSpec GUI Customization 160
7.2 Administering System Pages 160
7.3 Administering Function Pages 160
7.4 Administering Custom Pages 161
7.4.1 Adding Custom Pages 161
7.4.2 Adding Controls to Custom Pages 161
7.4.3 Removing Controls From Custom Pages 162
7.4.4 Ordering Controls in Custom Pages 162
7.5 Administering System Fields and Controls 162
7.5.1 Understanding Specification Folder System Fields 162
7.5.2 Understanding Specification System Fields 163
7.5.3 Understanding Requirement Folder System Fields 163
7.5.4 Understanding Requirement System Fields 164
7.5.5 Understanding Change Request Folder System Fields 165
7.5.6 Understanding Change Request System Fields 165
7.5.7 Understanding Document and File Attachment System Controls 165
7.5.7.1 Document System Control (1105) 165
7.5.7.2 File Attachment System Control (1121) 165
7.6 Administering Custom Fields and Controls 165
7.6.1 Understanding Control Types 166
7.6.2 Customizing Check Box Controls 167
7.6.3 Customizing Date-Time Controls 167
Chapter 8 Event Administration 168
8 Chapter 7- Event Administration 168
8.1 Understanding DevSpec Events 168
8.2 Administering DevSpec Event Templates 168
8.2.1 Defining Event Templates 168
8.2.2 Defining Default Event Titles and Descriptions 169
8.2.3 Defining Default Event Workflow States 169
8.2.4 Defining Default Event Owners 169
8.2.5 Defining Event Template Scope 169
8.3 Administering Event Workflow 169
8.3.1 Defining Event States 170
8.4 Administering Event Management Privileges 171
8.4.1 Defining Event Management Privileges 171
Chapter 9 Implementation Integration Administrati 171
9 Chapter 8- Implementation Integration Administration 172
9.1 Understanding DevSpec Iterative Development Management Administration 172
9.2 Administering DevSuite Development Integration 172
9.2.1 Enabling Baseline Support 172
9.2.2 Defining Applicable Products and Versions 172
9.3 Administering Implementation Linking and Time Management 172
9.3.1 Enabling Implementation Link Time Management Tools 173
9.3.2 Defining Implementation Link Options 173
9.3.3 Defining Applicable Implementation Modules 173
9.4 Administering Specification Backlogs 174
9.4.1 Enabling Specification Backlog Management 174

9.4.2 Defining Spec Management Options for Child Development Projects 174
9.4.3 Defining State-Based Specification Assignment Rules 174
9.4.4 Defining State-Based Pending Specification Linking Rules 175
9.4.5 Defining Assigned Implementation Link Spec State Automation 175
9.4.6 Defining Closed Link Specification State Automation 175
9.4.7 Administering the Implementation Module Control 175
Chapter 10 Time and Cost Analysis Administration 175
10 Chapter 9- Time and Cost Analysis Administration 175
10.1 Understanding DevSpec Time, Cost, and Income Estimates 176
10.2 Administering Project, Time, Cost, and Income Estimation Settings 176
10.2.1 Enabling Time and Cost Analysis 176
10.2.2 Defining Time and Cost Analysis Preferences 176
10.2.3 Defining Time/Cost Tracking 177
10.2.4 Defining Time/Income Tracking 177
10.2.5 Selecting Applicable Time Categories 178
10.2.6 Updating Time Categories 178
10.3 Administrating Time, Cost, and Income Estimation Controls 179
10.3.1 Administering Requirement Estimation Controls 179
10.3.2 Administering Specification Estimation Controls 179
DevTrack Projects Administration 185
Chapter 1 What's new 185
Chapter 2 DevTrack Concepts 185
2 DevTrack Concepts 185
2.1 Understanding TechExcel DevTrack and DevSuite 185
2.1.1 Understanding DevSuite ALM Solutions 185
2.1.2 Understanding TechExcel Project Hierarchy 186
2.2 Understanding DevSuite Administration 186
2.2.1 Understanding DevSuite Site (System-Level) Administration 187
2.2.2 Understanding Project Administration 187
2.3 Understanding DevSuite Sample Projects and Templates 187
2.3.1 Understanding Sample Projects 187
2.3.2 Understanding Project Templates 188
2.3.3 Understanding Solution Templates 188
Chapter 3 Project Administration 188
3 Chapter 2- Project Administration 188
3.1 Understanding DevTrack Project Administration 188
3.1.1 Understanding Projects 188
3.2 Understanding the DevSuite Admin Workspace 189
3.2.1 Understanding Menu Bar Commands 189
3.2.2 Understanding the Tool Bar Commands 190
3.2.3 Logging into DevSuite Admin 190
3.2.4 Opening DevTrack Projects 190
3.2.5 Closing DevTrack Projects 191
3.2.6 Understanding Keyboard Shortcuts 191
3.3 Administering DevTrack Projects 192
3.3.1 Creating DevTrack Projects 192
3.3.2 Backing Up DevTrack Projects 193
3.3.3 Restoring DevTrack Projects 193
3.3.4 Creating Projects Based from Backed Up Projects 193
3.3.5 Copying DevTrack Project Settings 194
Chapter 4 Team Administration 194
4 Chapter 3- Team Administration 194
4.1 Understanding Team Administration 195
4.2 Administering Account Types, Access Controls, and Privileges 195
4.2.1 Defining Account Types 195

4.2.2 Enabling Advanced Access Control Support 196
4.2.3 Enabling the New Issue Deletion Option 196
4.2.4 Administering Issue Management Privileges 196
4.2.5 Administering Invisible Pages and Fields (Standard Access Controls) 197
4.2.6 Administering Read-Only Pages and Fields 198
4.2.7 Administering On Submission Access Controls 198
4.2.8 Administering On Existing Issues Access Controls 199
4.2.9 Administering DevTrack Knowledge Management Privileges 199
4.2.10 Administering Report Access Controls 199
4.2.10.1 Windows Client Reporting 200
4.2.10.2 Web Client Reporting 200
4.2.11 Administering Subproject Management Privileges 201
4.3 Administering Project Members 201
4.3.1 Adding User Accounts to Projects 201
4.3.2 Editing Project Member Account Types 201
4.3.3 Removing User Accounts from Projects 202
4.3.4 Viewing Project Member Information 202
Chapter 5 Development Issue Administration 202
5 Chapter 4- Development Issue Administration 202
5.1 Understanding DevTrack Client GUI Customization 202
5.2 Administering the Issue List Panel 203
5.2.1 Adding or Removing List Columns 203
5.2.2 Editing List Column Headers 203
5.2.3 Changing Column Order 203
5.2.4 Defining List Indicator Icons and Text Colors 203
5.2.5 Displaying Note and Event Attachment Icons 205
5.3 Administering the Issue Detail Panel 205
5.3.1 Defining Issue Detail Panel Page Order 205
5.3.2 Customizing the Issue Detail Panel in the Windows Client 206
5.3.3 Customizing the Issue Detail Page of the Web Client 206
5.3.3.1 Multiple page display 206
5.3.3.2 Single page display 206
5.4 Administering Function Pages 207
5.4.1 Administering the New Issue Window 207
5.4.1.1 system controls 207
5.4.2 Defining Manager Options in the Windows Client 208
5.4.2.1 Detailed action display 208
5.4.2.2 Simplified action display 208
5.4.3 Administering the Forward Issue Window 208
5.4.4 Administering the Close Issue Window 209
5.5 Administering System Pages 209
5.5.1 Administering the Issue Description System Page 209
5.5.2 Administering the Issue Detail System Page 209
5.5.3 Administering the Current Status System Page 210
5.5.4 Administering the Notes System Page 210
5.5.5 Administering the Link System Page 210
5.6 Administering Custom Pages 210
5.6.1 Creating Custom Pages 210
5.7 Administering Page Design 211
5.7.1 Adding Controls to Custom Pages 211
5.7.2 Ordering Controls in Custom Pages 211
5.7.3 Removing Controls From Custom Pages 211
Chapter 6 Development Issue Workflow Administrati 211
6 Chapter 5- Development Issue Workflow Administration 211
6.1 Administering Development Issue Workflow 212
6.2 Configuring Project Workflow 212
6.2.1 Enabling Transition-Based Workflow and Transition-Dependent Rules 213

6.2.2 Enabling State-Based Workflow and Workflow Rules 214
6.2.3 Enabling Transition Dependent Access Controls 215
6.3 Administering Issue Workflow States 215
6.3.1 Adding Workflow States 216
6.3.2 Deleting Issue Workflow States 216
6.3.3 Defining State-Dependent Workflow Rules 216
6.3.4 Defining State-Dependent Applicable Owner Access Controls 217
6.3.5 Defining State-Dependent Default Issue Owners 218
6.3.6 Defining Applicable Workflow States 218
6.3.7 Defining State-Dependent Mandatory Fields 218
6.3.8 Defining State-Dependent Read-Only Fields 218
6.3.9 Defining State-Dependent Invisible Fields 219
6.4 Administering Transitions 219
6.4.1 Adding Transitions 219
6.4.2 Editing Transitions 219
6.4.3 Deleting Transitions 219
6.4.4 Defining Transition-Dependent Workflow Rules 219
Chapter 7 Administering Advanced Features 219
7 Chapter 6- Administering Advanced Features 220
7.1 Administering DevTrack Development Issue Routing 220
7.1.1 Enabling Issue Routing 220
7.1.2 Defining Issue Routing Rule Criteria 220
7.1.3 Defining the Default Recipient of Routed Issues 221
7.1.4 Defining Issue Routing Rules (Criteria) 221
7.1.5 Defining Issue Routing Targets (Potential Owners) 221
7.2 Administering Personal Folders 221
7.2.1 Enabling Personal Folder Support 222
7.2.2 Defining Personal Folders 222
7.3 Administering Issue Templates 222
7.3.1 Defining Issue Template Options 222
7.4 Administering Link Types 223
7.4.1 Understanding Interproject Linking 223
7.4.2 Defining Referential Link Types 223
7.4.3 Defining Parent-Child Link Types 224
7.4.4 Deleting Link Types 225
7.5 Administering Custom Crystal Reports 225
7.5.1 Choosing DevTrack Reporting Platforms 225
7.5.2 Defining Custom Crystal Report Template Folders 225
7.5.3 Adding User-Defined Custom Reports 225
7.6 Managing Issue Priority 226
7.6.1 Defining Priority Order Management Settings 226
7.6.2 Defining Priority Value Management 227
7.7 Administering Status Groups 227
7.7.1 Enabling Status Group Support 228
7.7.2 Defining Status Groups 228
Chapter 8 Issue Notification & Escalation Adminis 229
8 Chapter 7- Issue Notification & Escalation Administration 229
8.1 Administering Issue Notifications and Escalations 229
8.2 Administering the DevTrack Mail Service 229
8.2.1 Managing Outgoing Email Server Settings 230
8.2.2 Defining Issue Notification Sending Addresses 230
8.2.3 Reloading Settings to DevTrack Mail Service 230
8.2.4 Defining Issue Notification Activity Dates 230
8.3 Administering Issue Notifications 230
8.3.1 Enabling Issue Notification in a DevTrack Project 231
8.3.2 Defining Issue Notification Rules 231
8.3.3 Administering Issue Notification Triggers 232

8.3.4 Defining Custom Issue Notification Status Change Triggers 232
8.3.5 Defining Custom Field Change Triggers 233
8.4 Administering Issue Escalations 234
8.4.1 Administering Issue Escalation Rules 235
8.4.2 Understanding Issue Escalation Triggers 236
8.4.3 Administering Issue Escalation Actions 236
8.4.4 Administering Issue Escalation Schedules 236
Chapter 9 Email Integration Administration 237
9 Chapter 8- Email Integration Administration 237
9.1 Administrating DevTrack Email Integration 237
9.2 Administrating the Email Retrieval Service 237
9.2.1 Defining Incoming Mail Server Settings 237
9.2.2 Defining Incoming Mail Server Properties 238
9.2.3 Defining Email Autosubmit Settings 238
9.3 Managing Issue Submission by Email 238
9.3.1 Understanding Advanced Issue Autosubmission 239
9.3.1.1 The [Issue Properties] tag 239
9.3.1.2 The [Description] tag 239
9.3.1.3 Custom tags for multiple lines of text 240
Chapter 10 Subproject Administration 240
10 Chapter 9- Subproject Administration 240
10.1 Understanding DevSuite Subprojects 240
10.1.1 Understanding Subprojects and the Beta Customer Portal 241
10.2 Administering Subproject Settings 241
10.2.1 Enabling Subproject Support 241
10.2.2 Defining Subproject Terminology 242
10.2.3 Defining Default Subprojects 242
10.2.4 Defining Applicable Issue Types 242
10.3 Administering Subproject Statuses 242
10.3.1 Defining Independent Subproject Statuses 242
10.3.2 Defining Subproject Status Based on Highest Ranking 243
10.3.3 Defining Subproject Status Based on Lowest Ranking 244
10.3.4 Defining Subproject Status Based on Issue Status 244
10.4 Administering Subproject Priorities 244
10.4.1 Creating Independent Subproject Priorities 245
10.4.2 Defining Highest Subproject Priorities 245
10.4.3 Defining Lowest Subproject Priorities 246
10.4.4 Defining Linked to Issue Subproject Priorities 246
Chapter 11 Time and Cost Tracking Administration 247
11 Chapter 10- Time and Cost Tracking Administration 247
11.1 Understanding DevSuite Development Time and Cost Tracking 247
11.1.1 Understanding DevSuite Time and Cost Tracking Administration 247
11.2 Administering Time Groups and Categories 247
11.2.1 Selecting Applicable Time Categories 248
11.2.2 Enabling Time Tracking 248
11.2.3 Enabling Cost Tracking 248
11.2.4 Defining Project Time Tracking Methods 248
11.2.5 Defining Hourly Rate by Account Type 249
11.2.6 Defining Hourly Rate by User Account 249
11.3 Administering Time and Cost Tracking Privileges 250
Chapter 12 Knowledge Management Administration 250
12 Chapter 11- Knowledge Management Administration 250
12.1 Understanding DevTrack Knowledge Management 251
12.2 Administering KnowledgeWise Integration 251
12.2.1 Enabling KnowledgeWise Integration 251
12.2.2 Enabling DevSpec Integration 251

12.2.3 Migrating Knowledge View Data 251
12.3 Administering Project-level Knowledge Management 252
12.3.1 Understanding KnowledgeWise 252
12.3.2 Understanding Knowledge Items 252
12.3.3 Understanding Knowledge View Access Controls 252
12.3.4 Enabling the Knowledge View 253
12.3.5 Selecting a Revision System 253
12.3.6 Defining the Maximum Number of Revisions 253
12.3.7 Customizing the Description Page 253
12.3.8 Customizing the History Page 254
12.3.9 Customizing the Resolution Page 254
12.3.10 Backing up the Document Files 254
12.4 Administering Issue-level Knowledge Management 254
12.4.1 Understanding the Notes Page 255
12.4.2 Enabling the Notes Page 255
12.4.3 Customizing Note Page List Columns 255
12.4.4 Understanding Memo Field Time Stamping 255
12.4.5 Managing Note Types 255
12.4.6 Managing Note Access Controls 256
DevTrack (Enterprise) 257
Chapter 1 DevTrack Concepts 257
1 Chapter 1- DevTrack Concepts 257
1.1 Understanding TechExcel DevTrack and DevSuite 257
1.1.1 Understanding DevSuite ALM Solutions 257
1.1.2 Understanding TechExcel Project Hierarchy 258
1.2 Understanding DevSuite Administration 258
1.2.1 Understanding DevSuite Site (System-Level) Administration 259
1.2.2 Understanding Project Administration 259
1.3 Understanding DevSuite Sample Projects and Templates 259
1.3.1 Understanding Sample Projects 259
1.3.2 Understanding Project Templates 260
1.3.3 Understanding Solution Templates 260
Chapter 2 Iterative Development Administration 260
2 Chapter 2 Iterative Development Administration 260
2.1 Understanding DevSuite Iterative Development 260
2.2 Administering Project-Level Iterative Development 260
2.2.1 Specifying Iterative Development Terminology 260
2.2.2 Defining Iteration Durations 261
2.3 Administering Development Backlogs 261
2.3.1 Specifying Backlog Terminology 261
Chapter 3 Enterprise Edition Features 261
3 Chapter 3 Enterprise Edition Features 261
3.1 Understanding DevTrack Enterprise Edition 262
3.2 Administering the DevTrack Search Engine 262
3.2.1 Installing the DevTrack Search Engine 262
3.2.2 Defining Search Engine Settings 263
3.2.3 Indexing the DevTrack Search Engine 263
3.3 Administering User Identity Authentication 263
Chapter 4 Issue Types and Multiple Workflow Admin 264
4 Chapter 4 Issue Types and Multiple Workflow Administration 264
4.1 Understanding Multiple Workflow and Issue Types 264
4.2 Administering Multiple Workflow and Workflow Assignment 265
4.2.1 Enabling Multiple Workflow 265
4.2.2 Defining Multiple Workflow Settings 265
4.2.3 Defining Subworkflows 266

4.2.4 Understanding Workflow Assignment Methods 267
4.2.5 Defining Issue Type Subworkflow Initialization 267
4.2.6 Granting Issue Type Assignment Privileges 267
4.3 Administering Issue Types 267
4.3.1 Defining Development Issue Type Terms 268
4.3.2 Linking Issue Type Property to Issue Property Fields 268
4.3.3 Adding Issue Types 268
4.3.4 Defining Default Issue Types 269
4.3.5 Defining Issue Types 269
Chapter 5 Development Event Administration 269
5 Chapter 5 Development Event Administration 269
5.1 Administering DevTrack Development Events 269
5.2 Administering Event Groups 269
5.2.1 Creating Event Groups 270
5.2.2 Ordering Event Groups 270
5.2.3 Ordering Event Templates within Event Groups 270
5.3 Administering Regular Development Event Templates 270
5.3.1 Defining Regular Event Templates 271
5.3.2 Editing Event Templates 271
5.3.3 Defining Default Event Titles and Descriptions 271
5.3.4 Defining Event Description Timestamping Rules 271
5.3.5 Defining Default Event Workflow States 272
5.3.6 Defining Default Event Owners 272
5.3.7 Defining Default Event Start and Due Dates 273
5.3.8 Defining Regular Development Event Attachment Rules 273
5.4 Administering Co-Owner Event Templates 274
5.4.1 Creating Co-Owner Event Templates 275
5.4.2 Administering Event Variables 276
5.4.3 Enabling Co-owner Events and Event Variables 276
5.4.4 Creating Unlinked Event Variables 276
5.4.5 Creating Linked Event Variables 277
5.4.6 Defining Co-Owner Event Attachment Rules 277
5.5 Administering Applicable Owners of Development Events 278
5.6 Administering Event Management Privileges and Access Controls 278
5.6.1 Defining Event Access Controls 278
5.6.2 Understanding Event Privileges 279
5.7 Administering Customer Access to Development Events 279
5.7.1 Understanding Event Customer Access Rights 279
5.7.2 Defining Event Customer Access Controls 279
5.8 Administering Event Workflow 279
Chapter 6 Event Notification & Escalation Adminis 280
6 Chapter 6 Event Notification & Escalation Administration 280
6.1 Administering Event Notifications and Escalations 280
6.1.1 Notification and escalation triggers 281
6.1.2 Push and pull notification subscriptions 281
6.2 Administering Event Notifications 281
6.2.1 Enabling Event Notification in a DevTrack Project 281
6.2.2 Defining Event Notification Rules 281
6.2.3 Administering Event Notification Triggers 282
6.3 Administering Event Escalations 283
6.3.1 Escalation triggers 283
6.3.2 Escalation conditions 283
6.3.3 Escalation actions 283
6.3.4 Subscribers 283
6.3.5 Administering Event Escalation Rules 283
6.3.6 Understanding Event Escalation Triggers 284
6.3.7 Administering Event Escalation Actions 284

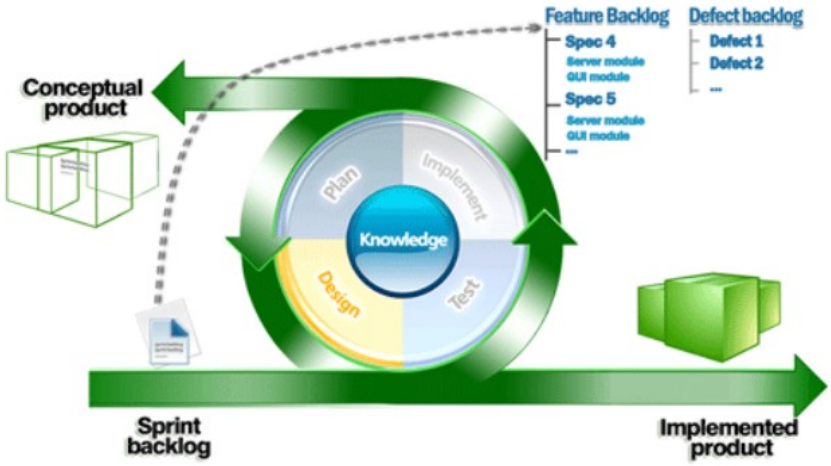
6.3.8 Escalation schedules 284
6.3.9 Notification methods 285
6.3.10 Event reassignment actions 285
6.3.11 Administering Event Escalation Schedules 285
6.3.12 Administering Event Reassignment Actions 285
6.4 Administering Event Notification and Escalation Conditions 286
6.5 Administering Event Notification and Escalation Actions 286
6.5.1 Identifying Notification Action Email Addresses 287
6.5.2 Defining Event Email Notifications 287
6.5.3 Defining Customer Email Notifications 288
6.5.4 Defining Pager and Mobile Phone Notifications 288
6.6 Administering Potential Recipients and Subscriptions 288
6.6.1 Identifying Potential Recipients of Notification Actions 289
6.6.2 Potential recipient variables 289
6.6.3 Customers as potential recipients 289
6.6.4 Defining the Scope of Potential Recipient Variables 290
Chapter 7 Beta Customer Portal Administration 290
7 CHAPTER 7 Beta Customer Portal Administration 290
7.1 Managing the DevTrack Beta Customer Portal 290
7.1.1 Understanding the Beta Customer Portal 291
7.1.2 Understanding Beta Customer Portal Security 291
7.1.3 Understanding the Customer Manager 291
7.2 Managing Customer Base Projects 292
7.2.1 Understanding Customer Base Projects 292
7.2.2 Understanding the Customer Base Settings Page 292
7.2.3 Associating Custom Base Projects with DevTrack Projects 293
7.2.4 Creating New Customer Base Projects 293
7.2.5 Defining Beta Customer Portal Login Settings 293
7.2.6 Defining Parent-Child Relationship Support 294
7.2.7 Defining Default Subprojects for Customer Base Projects 294
7.3 Customizing the Address Page of the Customer Manager 294
7.3.1 Customizing the Address Page 295
7.3.2 Defining Customer Types 296
7.3.3 Defining Project Linking Options 297
7.3.4 Customizing the Customer ID Control 297
7.4 Customizing Contact Info in the Customer Manager 298
7.4.1 Customizing the Contact Info Page 298
7.4.2 Defining Contact Access Types 299
7.4.3 Defining Access Type Privileges 300
7.4.4 Defining Project Access Controls 301
7.5 Managing DevTrack Customer Manager Privileges 301
7.5.1 Understanding Customer Manager Privileges 301
7.5.2 Defining Account Type Privileges 301
Chapter 8 Interproject Actions and Linking Admini 301
8 CHAPTER 8 Interproject Actions and Linking Administration 301
8.1 Administering Interproject Actions and Linking 301
8.2 Administering Standard Interproject Actions and Linking 302
8.2.1 Enabling Project-to-Project Integration 302
8.2.2 Enabling Standard Interproject Action Support 302
8.2.3 Enabling Interproject Link Support 302
8.2.4 Defining Interproject Standard Action Settings 303
8.2.5 Defining Interproject Editing Rules 303
8.2.6 Defining Interproject Field and Field Value Mapping 304
8.2.7 Defining Issue History for Linked Issues 305
8.2.8 Defining Standard Interproject State Automation Rules 305
8.2.9 Defining Interproject State Automation Rule Conditions 306
8.3 Administering System Interproject Actions 307

8.3.1 Defining Clone System Interproject Actions 307
8.3.2 Defining Submit System Interproject Actions 308
8.3.3 Defining Copy System Interproject Actions 308
8.4 Administering Custom Interproject Actions 308
8.4.1 Adding, Editing, and Deleting Custom Interproject Actions 309
8.4.2 Defining Custom Interproject Actions 309
8.4.3 Defining Custom Interproject Action Access Controls 310
8.4.4 Defining Custom Interproject Action Applicable States 310
8.5 Administering Custom Interproject Action Automations 310
8.5.1 Enabling Interproject Action Automation in Transition-Based Projects 311
8.5.2 Enabling Interproject Action Automation in State-Based Projects 311
8.5.3 Defining State-Dependent Interproject Automation Rules 311
8.5.4 Defining Transition-Dependent Interproject Automation Rules 312
8.5.5 Defining Interproject Action Automation Rule Conditions 312
Chapter 9 Product, Version, and Build Administrat 313
9 CHAPTER 9 Product, Version, and Build Administration 313
9.1 Understanding Product Version Tree Administration 314
9.2 Administering Milestone Date Types 314
9.2.1 Adding Milestone Date Types 314
9.2.2 Editing Milestone Date Types 315
9.2.3 Deleting Milestone Date Types 315
9.3 Administering the Product Tree 315
9.3.1 Administering Product, Version, and Build Release Cycles 316
9.3.2 Release status 316
9.4 Administering Applicable Products, Versions, and Builds 316
9.5 Administering Product Categories 317
9.6 Administering Products 317
9.6.1 Defining Products 318
9.6.2 Defining the Product Release Lifecycle 319
9.7 Administering Versions 320
9.7.1 Managing Version Milestones 320
9.7.2 Deleting Versions 322
9.7.3 Managing the Version Release Lifecycle 322
9.8 Administering Builds 323
9.8.1 Defining Builds 323
9.8.2 Managing Build Release Lifecycle 324
9.9 Administering Product Implementation Modules 325
Chapter 10 Subproject Administration 325
10 CHAPTER10 Subproject Administration 325
10.1 Understanding DevSuite Subprojects 325
10.1.1 Understanding the Subproject Tree Structure 325
10.1.2 Understanding Subprojects and the Beta Customer Portal 326
10.2 Administering Subproject Settings 326
10.2.1 Enabling Subproject Support 326
10.2.2 Defining Subproject Terminology 326
10.2.3 Defining Default Subprojects 327
10.2.4 Defining Applicable Issue Types 327
10.3 Administering Subproject Statuses 327
10.3.1 Defining Independent Subproject Statuses 327
10.3.2 Defining Subproject Status Based on Highest Ranking 328
10.3.3 Defining Subproject Status Based on Lowest Ranking 328
10.3.4 Defining Subproject Status Based on Issue Status 329
10.4 Administering Subproject Priorities 329
10.4.1 Creating Independent Subproject Priorities 330
10.4.2 Defining Highest Subproject Priorities 330
10.4.3 Defining Lowest Subproject Priorities 331
10.4.4 Defining Linked to Issue Subproject Priorities 331

Chapter 11 Branch Management Administration 332
11 CHAPTER11 Branch Management Administration 332
11.1 Understanding Multiple Branch Management 332
11.2 Administering Branch Issues 332
11.2.1 Enabling Branch Management 333
11.2.2 Defining Branch Issue Types 333
11.2.3 Adding, Editing, and Deleting Branch Issue Types 333
11.2.4 Defining Default Branch Issue Types 333
11.3 Administering Branch Link Types 333
11.3.1 Adding, Editing, and Deleting Branch Link Types 334
11.3.2 Defining Branch Link Type Properties 334
11.4 Administering Branch Actions 334
11.4.1 Adding, Editing, and Deleting Branch Actions 335
11.4.2 Defining Branch Actions 335
11.4.3 Defining Branch Action Privileges 336
11.4.4 Defining Branch Applicable States 336
Chapter 12 Branch Event Administration 336
12 CHAPTER12 Branch Event Administration 336
12.1 Understanding Branch Events 336
12.1.1 Enabling Branch Events 337
12.1.2 Enabling In-frame Editing of Branch Events 337
12.1.3 Defining Branch Event Templates 337
12.1.4 Adding, Editing, and Deleting Branch Event Templates 338
12.1.5 Defining Branch Event Template Attachment Rules 338
12.1.6 Defining Branch Event Workflow States 339
12.1.7 Managing Branch Event Access Rules 340
12.1.8 Administering Branch Event Workflow 340
12.1.9 Defining Branch Event Automatic Close Workflow Rules 340
12.2 Administering Advanced Branch Event Workflow 341
12.2.1 Disabling In-frame Editing 341
12.2.2 Defining branch event Advanced Workflow Rules 341

DevSuite Admin Guide

DevSuite is a complete system for development process management. From requirements to project planning, work item tracking to QA testing, DevSuite allows team to follow configurable, integrated processes.



Key Benefits

Create and enforce processes based on best practices and popular methodologies.
Improves productivity with a tightly integrated suite of best of breed tools.
Eliminate organizational silos.
Improve collaboration using shared knowledge.
Produce higher quality software.
Increase customer satisfaction.

Features Overview

Quickly design and implement processes based on industry-standard methodologies. Customize these processes to meet your unique development culture.
Support for Agile, Scrum, CMMI, Waterfall, XP, and many more.
Full traceability from designs to implementation.
Traceability the entire development lifecycle and get feedback at every step of the way.
Integrated, role-based tools for Requirements Management, Project Planning, Task tracking, and QA testing.
Synchronizes with all popular SCM tools for complete tracking of tasks and their source code deliverables.
Designed for distributed, multi-site teams as well as collocated groups.
Scalable to thousands of users through web-service technology.
100% configurable user interface and attributes for requirements, specifications, work items, and test plans.
Support for both manual and automated testing of requirements.
Hundreds of reports, metrics, and burndowns to clearly communicate critical data.

System Settings Administration

Chapter 1 System-Level User Administration and Li

1 Chapter 1- System-Level User Administration and License Management

Wiki Summary.

1.1 Understanding System-Level User Administration

System-level user administrative tasks include license management and allocation, user account creation and definition, and the management of project administrators. Optional system-level administrative tasks include multiple site management, division management, and LDAP directory server integration and administration.

1.2 Administering User Licenses

TechExcel offers many different varietiesoflicenses to development organizations that purchase DevSuitesoftware, and the licenses purchased by these development organizations determine the features available within the DevSuiteimplementation.

DevSuitelicenses are divided into four different categories:

License Edition: The License Edition defines the version of the DevSuite software purchased. DevSuite Solution Pack is available for either Team Edition or Enterprise Edition.

Site License: The Site License defines whether Multiple Site support is available in the DevSuite implementation. Multiple Site licenses are only available to users of the DevSuite Enterprise Edition.

License Class: The License class defines the duration of the license. TechExcel offers three classes of licenses: evaluation licenses, permanent purchased licenses, and limited-time purchased licenses.

License Type: License types represent the number of seats that are available in a DevSuite site. DevSuite supports three types of licenses: Fixed licenses, Floating licenses, and Light licenses.

1.2.1 Understanding License Edition

The License Edition defines the version of the DevTrack software purchased. DevSuite is available in two editions: DevSuite Team Edition and DevSuite Enterprise Edition.

A DevSuite Team Edition licenseenables development teams to use all the standard DevSuite features. DevSuite Enterprise Edition features (Subprojects, Events, Beta Customer Portal, and so on) are not available.

Development teams may purchase any number of fixed licenses and floating licenses.

Development teams may also purchase support for*licensed modules*.

Although many DevSuite Enterprise Edition configuration tools are displayed in DevSuite Admin, these features cannot be enabled or configured in the DevSuite Team Edition. Purchasers of the DevSuite Team Edition may purchase*licensed modules*. Every DevTrack Standard Edition license includes information about the modules purchased. If the modules are not enabled in the license, these features cannot be accessed using that license.

DevTrack Web

DevTrack Web for Customers

DevTrack QA Test Management

DevTrack Offline Support

Version Control Support

Power Beta Customer Support

Search Engine Support

LDAP Support

A DevSuite Enterprise Edition licenseenables development teams to use all of the features in the DevSuite Standard Edition plus a number of features that are only available in the Enterprise Edition.

DevSuite Enterprise Edition features include:

Advanced access controls

Subprojects

Personal folders

Event management

Event notification

Event escalation

Beta Customer Portal

DevTrack Search Engine

DevTrack Multiple Sites

Multiple workflows

Purchasers of the DevSuite Enterprise Edition may purchase any number of fixed licenses, floating licenses, and light licenses. Light licenses enable customers to access DevTrack projects through the Beta Customer Portal. TechExcel provides 100 light licenses with every purchase of the DevTrack Enterprise Edition. Purchasers of the DevTrack Enterprise Edition may purchase*licensed modules*. Every DevSuite Enterprise Edition license includes information about the modules purchased. If the modules are not enabled in the license, these features cannot be accessed using that license.

1.2.2 Understanding Site License

TechExcel offers purchasers of the DevSuite Enterprise Edition two types of site licenses: Multiple Site licenses and Single Site licenses.

Multiple Site License: Multiple Site licenses enable development organizations to create site families and to share licenses across multiple DevSuiteimplementations (sites).Each site belongs to one of three site types: System master sites, work sites, and stand-alone sites.

Single Site License: Single Site licenses do not support site families. All DevSuiteimplementations using a Single Site license are stand-alone sites.

1.2.3 Understanding License Class

The License class defines the duration of the license. TechExcel offers three classes of licenses: evaluation licenses, permanent purchased licenses, and limited-time purchased licenses.

Evaluation License: An evaluation license is a free license that enables the users to evaluate DevSuitefor a short time period: 15 days, 30 days, or 60 days. Evaluation licenses are available for DevSuiteEnterprise Edition.

Limited-Purchased License: Alimited-purchased license enables the users to work in DevSuiteapplications for alonger limitedtime period.Limited-purchased license is ideal for customers to use in the test server, where new build can be tested

before rolling out to the production server. Limited-purchasedlicenses are available for the DevSuite Team Edition and the DevSuiteEnterprise Edition.

Purchased License: A purchased license enables the users to work in DevSuiteapplications for an unlimited time period. Permanent licenses are available for the DevSuite Team Edition as well as the DevSuite Enterprise Edition.

1.2.4 Understanding License Types

License types represent the number of seats that are available in a DevSuite site.

DevSuitesupports three types of licenses: Fixed licenses, Floating licenses, and Light licenses.

Fixed License: Fixed licenses are assigned to a single user. A fixed license enables a project member to log into any site in a site family.

Floating License: Floating licenses are not assigned to any one user, but are assigned to a particular DevTrack site. Floating licenses regulate the number of users who may be logged into a DevTrack site at one time. A single floating license may be assigned to multiple users, but only a certain number of users, which is the total number of floating licenses, may log into DevTrack at one time.

Light License: Light licenses enable beta customers to log into DevTrack projects through the Beta Customer Portal. Light licenses are only available in the DevSuiteEnterprise Edition.

Every DevTrack project member is assigned either a fixed license or a floating license.For more information see ?reating User Accounts?SPAN style="mso-fareast-font-family: 'MS Mincho'; mso-fareast-language: JA">

1.2.5 Loading DevSuite License File

The DevSuite License Manager enables system administrator to load,allocate, and track the licenses that belong to a DevTrack site.

Every DevSuite user must have a license

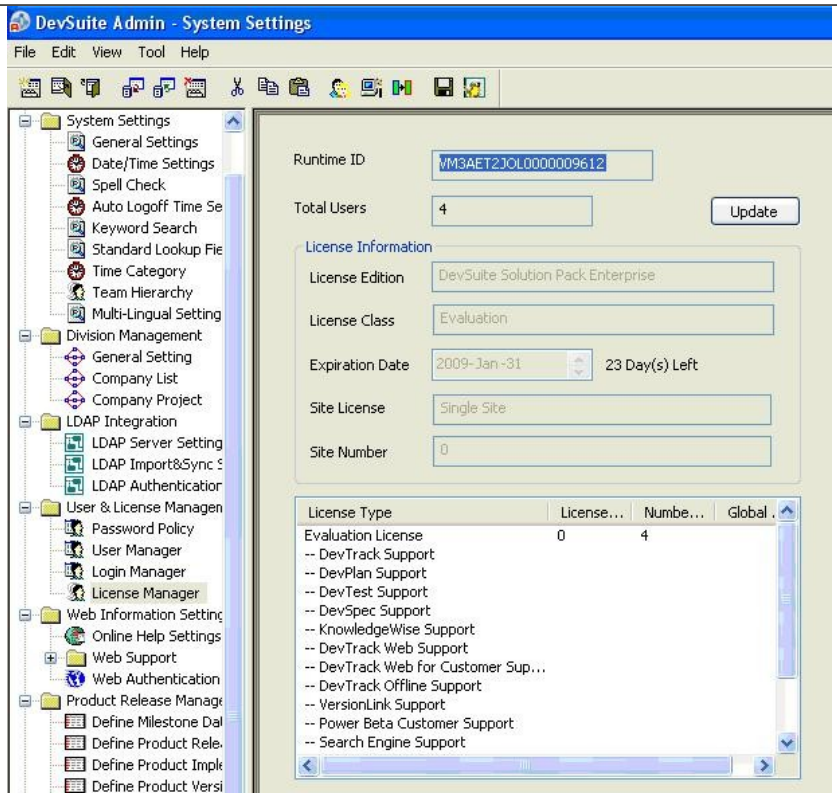
Customers may purchase floating licenses and fixed license for project members.

Customers may purchase light licenses for their beta customers.

To load user license:

1 Save the license file (.lic) to a directory on the local machine.

2 Open the License Manager. The License Manager may be accessed by two methods:
?Select the License Manager page in the User and License Management folder of the system settings project.
?Select the License Manager command in the tool menu. The License Manager appears.



3Click the Update button. The Open file browser appears.

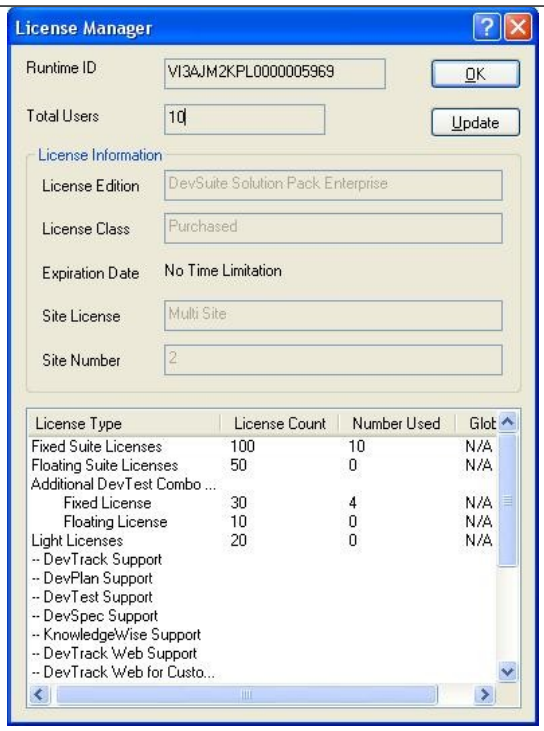
4Locate the license file on the local directory.

5Click the Open button. The licenses are loaded and automatically allocated to the current site.

Note: Please restart the application server after loading the license in order for it to take effect.To restart the DevSuite Application Server, go to Control Panel --> Administrative Tools --> Services. Highlight DevSuite Application Server. Right click on it and choose 'Restart'.

1.2.6 Tracking DevSuite LIcenses

The License Manager shows detailed information about the licenses that have been allocated to a project including the license edition, license class, and available products, modules and features.



The Runtime ID: The Runtime ID number identifies the unique license assigned to the site.

Total Users: The Total Users identifies the number of licensed users.

License Edition: The License Edition identifies the license edition: DevTrack SBE, DevTrack Standard Edition, DevTrack Enterprise Edition, or DevSuite Solution Pack.

Expiration Date: The Expiration Date identifies the date that the current Subscription and Evaluation license expires. Normally there is no time limitation for purchased license.

Site License: The Site License identifies the number of licensed sites --singlesiteor multiple sites.

For each license type (suite licenses, knowledge licenses, DevTest combo licenses, and light licenses), the License Manager shows the number of licenses available, the number of licensed used, and the number of global licenses used.

1.3 Administering User Accounts

In DevSuite, every user account is defined by a unique loginID, password, user profile, and contact information?sually a phone number and email account.

All work projects in a DevSuite site share a common group of licensed users that may belong to multiple projects?evTrack, DevSpec, DevTest. A user account may be assigned a different account type in each work project.

The DevSuite Admin User Manager is the primary tool for managing and tracking user contact information, account types, project membership, administrative privileges, and licensing information.

1.3.1 Creating User Accounts

In DevSuite, a user account identifies a user that may access and work in a DevSuite project?evTrack project, KnowledgeWise projectorDevSpec project.Every project in a DevSuite site sharesa common set of user accounts.

The basic user account properties displayed in the User Information tab include:

User Name: The User Name property represents the name that the project member uses to log into the application.

First Name: The First Name property stores the first name of project members.

Last Name: The Last Name property stores the last name of project members.

Password: The Password property stores the password that project members use to log into the DevTrack project.

Auto Login Account: The Auto Login Account field stores the NT username that a project member may use to log into development projects through DevTrack Web. The auto login account is only used if the system administrator has enabled Windows NT Authentication in the DevSuite site. For more information see ?onfiguring Windows NT Authentication?

Phone (W): The Phone (W) property may be used to store the work phone number of project members.

Phone (H): The Phone (H) property may be used to store the work phone number of project members.

E-mail: The E-mail property may be used to store the e-mail address of project members.

Address: The Address property may be used to store the mailing address of project members.

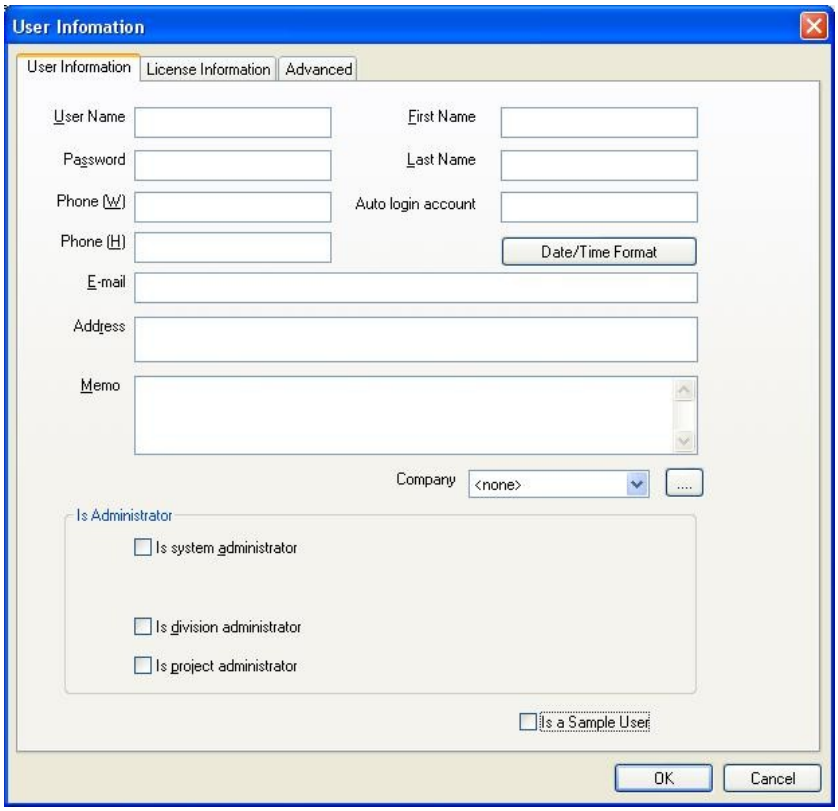
Memo: The Memo property may store short notes about project members.

Is System Administrator: The Is System Administrator property defines whether a user account has system-level administrative privileges. If a user is marked as a system administrator, he or she is atutomatically a division and project administrator.

Is Division Administrator: The Is Division Administrator property defines whether a user account has division-level administrative privileges. If a user is marked as a division administrator, he or she is automatically a project administrator.

Is Project Administrator: The Is Project Administrator property defines whether a user account has project-level administrative privileges.

Isa SampleUser: The IsA SampleUser property defines whether a user account is created for testing and evaluation in the sample projects. No license is needed for sample users.



To create a user account:

1Click the User Manager button in the tool bar. The User Manager appears.

2Click the New button. The User Information dialog box appears.

3Define basic user account properties in the User Information tab.

4 Optional:To designate the user as a project, division, or system administrator, select the appropriate check box in the Is Administrator area of the User Manager.

5 Optional:To assign a license to the user, select the check box in the License Information tab of the User Manager. A user account must be assigned a license to access a project except for sample users.

6 Optional:Define pager and mobile phone account settings in the Advanced tab.

?Mobile phone account settings define contact information that enables users to be contacted by mobile phone.

?Pager account settings define contact information that enables users to be contacted by pager or PDA.

7Click the OK button. The user account is displayed in the User Manager list.

1.4 Administering User Logins

The Login Manager displays high-level informationaboutthe current status and login history of DevSuite project members including login names, login and logout times, machine names, and license types.

The Login Manager consists of two tabs: the Login Status tab and the Login History tab.

The Login Status tabdisplays high-level information about the user accounts logged into each DevSuite project.
The Login History tabdisplays high-level information about each user session during an administrator-defined time period.

The Login Manager command may be invoked by two methods:

?Select the Login Manager command in the Tool menu.
?Click the Login Manager Button in the tool bar.

1.4.1 Viewing the Login Status Of Project Members

The Login Status list displays high-level information about the project members that are logged into a project. The following columns may be displayed in the Login Status list of the Login Manager. List columns may be added or removed from Login Status list.

Login Name: The Login Name column shows the name of the project member that is logged into the project.

Login Time: The Login Time column shows the time that the project member logged into the project.

Project: The Login Project column shows the name of the project.

Application: The Application column shows the client that was used to access the project?eb or Window Client.

Source: The Machine column shows the name or IP address of the machine that was used to log into the project.

License Type: The License Type column shows the type of license that is assigned to the project member.

Last Active Time:

Lockup Time:

Session ID:

To view the login status for DevSuiteprojects:
1.Open the Login Manager.
The Login Manager command may be invoked by two methods:

Select the Login Manager command in the Tool menu.
Click the Login Manager button in the tool bar.

2.Select the Login Status tab. The Login Status tab displays high-level information about the user accounts logged into each DevSuite project.

3.Select a project from the Project dropdown list. The Login Status list displays information about every project member that is currently logged into the selected project.

4.Optional: To refresh the login status list click the Refresh button.

5.Optional: To log a user off the system, highlight the user and click the Log Off button.

6.Optional: Tocustomizethecolumn displayed in the Login Status list click theCustomizebutton.Move columns that would like to be displayed to the right box.

1.4.2 Managing Idle Timeout Settings

In DevSuite, administrator-defined timeout settings control access to projects and manage the distribution of floating licenses to project team members. DevSuiteclient users may be automatically logged out of DevSuiteprojects if they remain inactive for a period that exceeds the idle timeout period.

Idle timeout settings define the amount of time that must pass (in minutes) between actions in client before a user is automatically logged out of the project. The minimum idle timeout setting is 15 minutes.

System administrators may disable DevTrack idle timeout controls by entering 0 in the Idle Timeout control of the Login Manager. If zero 0 is entered in the Idle Timeout control, idle users are not logged out of DevTrack projects.

To define system idle timeout settings:
1.Open the Login Manager. The Login Manager command may be opened by two methods:

Select the Login Manager page in the User & License Management folder of the tree panel.

Click the Login Manager button in the tool bar. The Login Manager appears.

2.Select the Login Status tab.

3.Enter 0 or a number greater than 15 in the Idle Timeout to Log Off the Client (in Minutes) field.

Note: Idel timeout can also be defined in theAuto Logoff Time Settingspage under System Settings Folder.

1.4.3 Viewing Login History Logs

The Login History list displays high-level information about the past DevSuite sessions of project members. The following columns may be displayed in the Login History list of the Login Manager:

Login Name: The Login Name column shows the name of the project member that is logged into the project.

Status: The Status column shows whether the user is currently on or off the application.

Login Time: The Login Time column shows the time that the project member logged into the project.

Logoff Time: The Logoff Time column shows the time that the project member logged off the project.

Project: The Login Project column shows the name of the project.

Application: The Application column shows the client that was used to access the project?eb or Window Client.

Source: The Machine column shows the name or IP address of the machine that was used to log into the project.

License Type: The License Type column shows the type of license that is assigned to the project member.

Session ID:

To view the login status for DevSuiteprojects:
1.Open the Login Manager.
The Login Manager command may be invoked by two methods:

Select the Login Manager command in the Tool menu.
Click the Login Manager button in the tool bar.
2.Select the Login History tab. The Login History tab displays high-level information about the user accounts logged into each DevSuite project.

3.Optional: To define the date range returned in the list, select the beginning date in the From Date control and an ending date in the To Date control.

4.Optional: To refresh the login history click the Refresh button.

5.Optional: Toclear the login history logclick theClear Logbutton.

6.Optional: Tocustomizethecolumn displayed in the History Logclick theCustomizebutton.Move columns that would like to be displayed to the right box.

1.5 Administering Divisions

In DevSuite, a*division*is a group of project administrators and project members that ?wn?one or more projects. Each division is defined by an address, a set of applicable work projects, a set of project administrators, and a group of project members.

Divisions enable development organizations greater control to define and manage project-level access rights and access privileges. The ability of project administrators to manage work projects may be restricted based on division membership.

A DevSuite site may be managed by three types of administrators: system administrators, project administrators, and division administrators. Division administrators are project administrators whose access rights are limited to thoseprojects that are applicable to their division.

Division management is done in theDivision Management folderwithin the System Settings Project.

In theGeneral Setting page, system administrators can enable division management, customize division terminology, enable cross Division user management, enable division administrator to view system settings, and enable cross-Division inter-project copy.

Enabling Division Management
To enable support for divisions in a DevSuite site, select the Enable Division Management check box in the General Settings page of the Division Management folder of a system settings project.

Customizing Division Terminology
To customize the term that is used to refer to ?ivisions?in a DevTrack site, input a term in the Refer Division As text box in the General Settings page of the Division Management folder of a system settings project.

Enabling Cross-Division User Management
To enable project administrators to act as division administrators for multiple divisions in a DevSuite site, select the Cross-Division User Management check box in the General Settings page of the Division Management folder of a system settings

project.

Enabling Division Admins to View System Settings
To enable division administrators to view system settings in a DevTrack site, select the Enable Division Administrators to View System Settings check box in the General Settings page of the Division Management folder of a system settings project.

Enabling Cross-Division Interproject Copying
To enable project members to copy issues between projects belonging to different divisions, select the Enable Cross-Division Interproject Copy check box in the General Settings page of the Division Management folder of a system settings project.

1.5.1 Managing Divisions

System Administrators can manage divisions in theDivision List page.System administrators can add a new division, define the properties and settings of the division using the corresponding four tabs (General Information, Properties, Administrators, and Members).

To add a division

1Click the Add button in the Division List page of the Division Management folder in the System Settings project. The Add New Division dialog box appears.
2Enter the name of the division in the Name text box.
3Click the OK button. The Add New Division dialog box closes. The division is displayed in the division list.

To define the general information of a division:

1Select a division in the division list of the Division List page of the Division Management folder in the System Settings project.
2Select the General Information tab.
3Define division properties in the General tab. Division properties include the Name, Address, City, State. Country, Zip, Phone, Fax, and Description.

To assign projects to a division:

1Select a division in the division list of the Division List page of the Division Management folder in the System Settings project.
2Select the Projects tab.
4 Click the Change Projects button in the Projects tab. The Division Project dialog box appears.

3Select the project which would like to be assigned to this division.
4Click the Change Division button or double click on the target project. The Change Division dialog box appears.
5Select the division from the Division dropdown list.
6Click the OK button. The Change Division dialog box closes.
7Click the Close button. The Division Project dialog box closes.

To assign division administrators to a division:

1.Select a division in the division list of the Division List page of the Division Management folder in the System Settings project.
2.Select the Administrators tab.
3.Assign or remove division administrators to a division. The Available Division Administrator list displays the names of every project administrator that has been defined as a division administrator in the User Manager.

?To add a division administrator, select a division administrator in the Available Division Administrator list and click the Right arrow button.
?To remove a division administrator, select a division administrator in the Division Administrator list and click the Left arrow button.

To make a user a division administrator:

1Open the User Manager. The User Manager may be accessed by the following methods:

?Select the User Manager command in the Tool menu.
?Click the User Manager button in the tool bar.

2Select a project member in the user list of the User Manager.
3Click the Edit button. The User Information property sheet appears.
4Select the Is Division Administrator check box in theUser Manager tab.
5Click the OK button. The User Information property sheet closes.
6Click the Close button. The User Manager closes.

To assign regular users to a division:

1. Select a division in the division list of the Division List page of the Division Management folder in the System Settings project.
2.Select the Members tab. All members belonging to this division are shown here.

3. To assign user accounts to a division, open the User Manager. The User Manager may be accessed by the following methods:

?Select the User Manager command in the Tool menu.
?Click the User Manager button in the tool bar.

2Select a project member in the user list of the User Manager.
3Click the Edit button. The User Information property sheet appears.
4Select a division in the Division dropdown list of the User Information tab.

System Administrators can also rename a division or delete it.

To rename a division:

1Select a division in the division list of the Division List page of the Division Management folder in the System Settings project.
2Click the Edit button. The Edit Division dialog box appears.
3Update the name of the division in the Division Name edit box.
4Click the OK button. The Edit Division dialog box closes.

To delete a division:

1Select a division in the division list of the Division List page of the Division Management folder in the System Settings project.
2Click the Delete button. A confirmation dialog box appears.
3Click the Yes button.

Chapter 2 LDAP Directory Server Integration

2 Chapter 2- LDAP Directory Server Integration

Wiki Summary.

2.1 Administering Directory Server Integration

A directory service stores and organizes information about the resources on a network ?ncluding e-mail addresses, computers, and peripheral devices such as printers. The directory service provides a centralized database that enables network administrators to manage access to these resources by applications and users.

A directory service organizes content in a directory server into a logical and accessible structure and acts as a central authority that can securely authenticate resources and manage identities and relationships between them.

DevSuite-directory server integration enables administrators to manage user data (user names, phone numbers, passwords, and so on) in a directory server rather than in individual DevSuite projects or sites. Once the system administrator has integrated the DevSuite site with their directory servers,they may use LDAP to manage user accounts and logins:

?System administrators may import user data from an LDAP server into a DevSuite site.
?System administrators may enable LDAP directory server authentication for all projects managed in a site. An LDAP server is usedtoauthenticate users logging into DevSuite projects usingdomain user accounts.

Understanding Integration Tasks

In TechExcel DevSuite, all LDAP directory server integration and authentication are defined at the system-level in the DevSuite Admin client. System administrators are responsible for enabling and defining DevSuite-LDAP directory server integration settings, enabling and definingDevSuite authentication settings, and mapping DevSuite user account properties to LDAP user attributes.

All system-level LDAP directory server integration tasks are performed in theLDAP Integration folderof the System Settings project in the DevSuite Admin client. DevSuite-directory server integration is a four-step process:

1Register one or more LDAP directory servers.

Every integrated directory service is defined by its name and directory server type.

2Define DevSuite field-LDAP attribute matching identifiers.

3Map DevSuite fields to LDAP attributes.

4Define LDAP directory server user authentication settings.

An LDAP directory server may be used to authenticate users logging into projects using Windows-based clients.

2.2 Administering LDAP Directory Server Registration

DevSuite enables DevSuite site integration with multiple LDAP directory servers. As a first step towards integration, every directory server must be ?egistered?in the System Settings project.

Directory server registration defines the directory server type, and connection settings. Using controls, in the LDAP ServerSetting page, system administrators may define the list of directory services that are used in the DevSuite site.

DevSuite supports integration with three types of LDAP directory services: MS ADAM, MS Directory, and Other Directory Servers.

MS ADAM: Microsoft ADAM (Active Directory Application Mode) is a flexible LDAP directory service that runs as a user service, rather than as a system service.

MS Directory: Microsoft Active Directory is a network operating system directory that enables administrators to manage enterprise-wide information from a centralized repository.

Other Directory Servers: In DevSuite, the term*Other LDAP Servers*is used as a catchall term to refer to non-Active Directory server types?he directory services software and standards that define an LDAP server.

System administrators must identify LDAP servers and define the LDAP server name, the server type before they can define the LDAP directory service settings.

To register an LDAP directory server:

1Select the LDAP Server Setting page in the LDAP Integration folder of the System Settings project. All LDAP integration tasks are performed in the System Settings project in the DevSuite Admin client.

2Click the New button. The New LDAP Server dialog box appears.

3Define the name and server type.

The server name may be any name that enables the administrator to identify the LDAP server within DevSuite. DevSuite supports Microsoft ADAM, Microsoft Active Directory, and other LDAP server types.

4Click OK. The LDAP server is displayed in the LDAP Server List of the LDAP Server Setting manager.
After selecting the LDAP server and defining its name and server type, the administrator may define the LDAP server connections.

ToDefineConnections toanLDAP Directory Server

LDAP directory serversettings identify the location of the directory service, connection settings (including username and passwords), and the search filter that selects the data to be imported from the directory server.

On the LDAP Server Setting page, highlighting the LDAP server previously defined and clicking the Server Info button,system administrators may define, update, and test LDAP server connections, define a DN path and password to access appropriate data, and define a search filter to limit the scope of LDAP data accessed.

Server Name: The Server Name is a name that identifies the registered LDAP server in the DevSuite system.

Host Name: The hostname identifies the server machine that hosts a particular LDAP directory server. The devices on a network are uniquely identified by their hostnames.

Port: A port number is a specific number that is used to map data to a particular process running on a computer. In a DevSuite-directory server integration, the port number identifies the channel over which data is communicated between the DevSuite and the directory server. For example, port 389.

User DN and Password: A distinguished name (DN) is a collection of attributes (relative distinguished names) that uniquely reference an object in the directory tree?uch as a user account. The DN may consist of a common name (CN) and one or more domain name components(DC). For example, CN=users,DC=mydomain,DC=com The user DN identifies the DN that is used to access LDAP directory server data. The user DN must be accompanied by the corresponding password.

Server Type: The Server Type identifies the LDAP directory service application that represented by the integrated directory server.

Search Filter: The search filter identifies the attribute to be imported from the LDAP directory server. For example, objectClass=user

Attribute Name of the User ID/GUID: The User ID/GUID identifies the LDAP attribute that corresponds to the DevSuite User ID property.For example, SAMAccountName

Steps to define a connection to an LDAP directory server:

1.Select a LDAP directory server in the LDAP Server List. System administrators must register an LDAP directory server before they may define LDAP directory server connection settings.

2.Click the Server Info button. The LDAP Server Information Setting manager appears.

3.Define or update the server name and hostname in the Server Setting tab.

?The server name is any name that identifies the LDAP server within DevSuite Admin.
?The host name is the name of the machine hosting the LDAP directory server.

4.Define the port number in the Port text box.

5.Optional: To test the connection to the LDAP server, click the Test button. The LDAP Server Information manager tests whether DevSuitemayaccess the LDAP server through the defined host and port number.

6.Define theUser DNaccount and password: The sync user DN identifies the DN that that is used to access and synchronize LDAP directory server data. The sync user DN must be accompanied by the corresponding password.An example of the User DN can beCN=users,DC=mydomain,DC=com

7.Define or update the server type by selecting and option from the Server Type dropdown list. The Server Type dropdown list displays the server type selected when the LDAP directory server was registered.

8.Define a search filter.The Search Filter identifies the LDAP directory field that is to be imported.An example of a search filter would beobjectClass=user.
Always use a search filter. If no search filter is defined, DevSuite imports all data contained in the root directory.

9.Optional: To confirm that DevSuite can connect with the LDAP server,select theAttribute Name of theUserID/GUID control. The data displayed in the Attribute Name of the UserID/GUID control is drawn from the LDAP server.

If no data is displayed in the UserID/GUID control,that meansDevSuite cannot connect to the LDAP server. Confirmif the UserDN and password are correct.

10. Define an attribute name for the User ID/GUID. For example,SAMAccountName.The User ID/GUID identifies the attribute that corresponds to the DevSuite User ID property.

11. Click the OK button.

2.3 Administering Field-Attribute Matching Identification

Matching identifiers are sets of mapped DevSuite user account properties and LDAP attributes that are used by the DevSuite to identify mapped pairs of DevSuite user accounts and LDAP user accounts.

Defining the Search Root

The search root for user accounts is a DN (distinguished name) that identifies the root directory for team member accounts in the directory server. A distinguished name consists of container name (CN) and one or more domain controls (DC). For example, CN=users,DC=mydomain,DC=com.

The root directory represents the LDAP directory that contains project member information. DevSuite searches all subfolders within the root directory for user data matching the search criteria defined in the Search Filter control of the Server List page.

To define the search root:

1Select a registered LDAP server in the LDAP Server List of the LDAP Servers Setting for User Import & Sync page.

2Click the Setting button. The LDAP Servers Setting for User Import & Sync manager appears.

3Select the General Setting for Import and Sync tab

4Define the root for user account data in the Search Root for Team Members text box.

The root directory represents the LDAP directory that contains user information.
DevSuite searches all subfolders within the folder for user data matching the search criteria defined in the search filter

2.4 Administering Field and Attribute Mapping

Please input description.

2.5 Administering LDAP Directory Server Authentication

2.6 Importing User Accounts from a LDAP Server

DevSuite-LDAP directory server integration enables administrators to manage user data (user names, phone numbers, and so on) in a directory server instead of their DevSuite sites.

System administrators may import user account attributes from an integrated LDAP server into a DevSuite system settings project using the LDAP Import wizard.

The LDAP Import wizard enables administrators to quickly create or update project user accounts by importing data from an integrated LDAP server. While importing user data, the administrator may also assign a license to the user.

Note: System administrators that plan to useLDAPto authenticate user logins into the DevSuite clients must redefine the passwords of user accounts imported from an LDAP directory server. LDAP passwords are encrypted and cannot be imported into DevSuite system settings projects from LDAP directory servers.

To import user account attributes:

1Click theUserManager icon in the tool bar. TheUserManager appears.

2Click the Import button. The Select a User Import Source dialog box appears.

3Select the Import from LDAP option. The Select Support Members wizard appears.

4Select an LDAP Server from the LDAP Server dropdown list. DevSuite must be integrated with an LDAP server, otherwise no optionswould bedisplayed in the LDAP Server dropdown list.

5 Optional: To search for user accounts based on the first name of the user, enter a query string in the First Name control. To filter user accounts based the first name of the user, the appropriate LDAP attribute must be mapped to the DevSuite First Name field. For example, the DevSuite First Name field may be mapped to the LDAP givenName attribute. System administrator may use the asterisk * wild card character to search for text strings in the First Name control.

6 Optional: To search for user accounts based on the last name of the user, enter a query string in the Last Name control. To search for user accounts based on the last name of the user, the appropriate LDAP attribute must be mapped to the DevSuite Last Name field. For example, the Last Name field may be mapped to the sn attribute. System administrator may use the asterisk * wild card character to search for text strings in the Last Name control.

7 Optional: To search for user accounts based on any LDAP user account attribute, enter a query string in the Search Filter control.

The query string entered in the Search Filter must identify both the*attribute*and the*attribute value*. For example, the query sn=smith returns all LDAP user accounts with the last name ?mith? System administrators may define complex queries using four logical operators in the Search Filter control: The asterisk wildcard character, the ampersand & AND operator, the pipe | OR operator, and the exclamation point! NOT operator.

8Click the Search button. Data matching the search parameter defined is displayed in the LDAP Users list.

9Add or remove users to the Selected Users list.

?Click the Add button to move users from the LDAP Users list to the Selected Users list.
?Click the Remove button to move users from the Selected Users list to the LDAP Users list.

10Click the Next button. The Select Users to be Overwritten page appears.

11 Optional: To overwrite properties of existing user accounts with their mapped LDAP attributes, select one or more user accounts. Any LDAP user accounts that match existing DevSuite user accounts based on administrator-defined matching identifiers is displayed in the Select Users to be Overwritten page.

12Click the Next button. The License Assignment page appears.

13Select one or more user accounts in the User list and click the Assign License button. The Select a License Type dialog box appears.

14Assign appropriate licenses to the user.

15Click the OK button. The Select a License Type dialog box closes.

16Click the Finish button. The user accounts are imported into the DevSuite system settings project.

2.6.1 Filtering User Accounts in the LDAP Import Wizard

The LDAP Import wizard enables system administrators to search for and import LDAP user accounts from a directory server and into the system settings project of a DevSuite site.

System administrators may filter the user accounts displayed in the LDAP Users list by defining search criteria based on LDAP user account attributes in the First Name control, Last Name control, and Search Filter control in the LDAP Import wizard
The First Name and Last Name controls enable administrators to define simple queries based on the name of the user. To search for LDAP user accounts by name, the appropriate LDAP user account attributes must be mapped to the DevSuiteFirst Name and Last Name fields.

The Search Filter control enables administrators to define complex queries based on any user account LDAP attribute.

System administrators may define complex search criteria in the Search Filter control based on user account attributes. System administrators may use four characters to define user account search parameters:

Asterisk *:The asterisk * character represents the wildcard character and may be used to search for text strings in the First Name, Last Name, and Search Filter controls.

To search for all user accounts that have a last name (surname) that begins with the letter*A*, enter the query string sn=a* in the Search Filter control.

Ampersand &: The ampersand & character represents the AND operator and may be used to define complex queries in the Search Filter control.

To search for user accounts that have a organization attribute value of ?BCSoftware?and a country attribute value of*USA*, enter the query string (&(o=ABCSoftware) (co=usa)) in the Search Filter control. The query string returns only those LDAP user accounts that match both organization attribute and the country attribute criteria.

Pipe |: The pipe | character represents the OR operator and may be used to define complex queries in the Search Filter control.

To search for user accounts that have a organization attribute value of ?BCSoftware?or a country attribute value of ?SA? enter the query string (|(o=ABCSoftware)(co=usa) in the Search Filter control. The query string returns all LDAP user accounts that match either of these criteria. All users in theUnited Statesor users working for ABCSoftware regardless of their country.

Exclamation!: The exclamation point! character represents the NOT operator and may be used to define complex queries in the Search Filter control.

To search for all user accounts that do not have a StreetAddress attribute, enter the query string(!(STREE=*)) in the Search Filter control.

Chapter 3 General System Settings Administration

3 Chapter 3- General System Settings Administration

Wiki Summary.

3.1 Administering Standard Lookup Fields

In DevSuite, a standard lookup field is an administrator-defined custom field that may be used to define and track business object data in projects throughout a DevSuite site.

Each standard lookup field defines a set of field values, which may potentially defined that field in the project.In the DevSuite clients, the field is represented by a multiple selection control (such as a dropdown list) and the field values are displayed as options within that control.

Using controls in the Standard Lookup Field page, system administrators may define any number of standard lookup fields and standard lookup field values.

A standard lookup field is defined by its field name, field description, field scope, and one or more field values. Each field value may be defined by a field description.

Field Name: The field name enables project administrators to identify the field in each project.

Field Description:The field description provides project administrators with information about the data tracked in the field.

Field Scope: The field scope defines the range of projects in which the standard lookup field may be employed. All standard lookup fields defined in the system settings project are {System} fields.

Field Value:A field value is a field property definition that may define a field property. Each standard lookup field defines one or more field values which constitute of options.

3.1.1 Defining Standard Lookup Fields and Field Values

To add a standard lookup field:

1Select the Standard Lookup Field page in the System Settings folder. All standard lookup fields are defined in the System Settings project.

2Click the New button in the Lookup Field area. The Add Standard Lookup Field dialog box appears.

3Enter a descriptive title for the lookup field in the Lookup Field Name field.

4Select a project from the Project dropdown list.

5Enter a brief description of the data values contained by the lookup field in the Description text area.

6Click the OK button. The Add Standard Lookup Field dialog box closes.

To edit a standard lookup field:

1Select a standard lookup field in the Standard Lookup Field page.

2Click the Edit button in the Lookup Field area. The Edit Standard Lookup Field dialog box appears.

3Edit standard lookup field definitions.

4Click the OK button. The Edit Standard Lookup Field dialog box closes.

To delete a standard lookup field:
1Select a standard lookup field in the Standard Lookup Field page.

2Click the Edit button in the Lookup Field area. A warning dialog box appears.

3Click the Yes button. The standard lookup field and all standard lookup field values are deleted.

To add a lookup field value:
1Select a standard lookup field in the Standard Lookup Field page.

2Click the Add button in the Lookup Field area. The Add Choice Value dialog box appears.

3Define the name and description of the standard lookup field value.

4Click the OK button. The Add Choice Value dialog box closes.

To edit a lookup field value:
1Select a standard lookup field in the Standard Lookup Field page.

2Select a standard lookup field value in the Lookup Value list.

3Click the Edit button in the Lookup Field area. The Edit Choice Value dialog box appears.

4Define the name and description of the standard lookup field value.

5Click the OK button. The Edit Choice Value dialog box closes.

To delete a lookup field value:
1Select a standard lookup field in the Standard Lookup Field page.

2Select a standard lookup field value in the Lookup Value list.

3Click the Delete button in the Lookup Field area. A confirmation dialog box appears.

4Click the Yes button. The standard lookup field value is deleted.

3.2 Multilingual Support and Localization Administration

DevSuite solutions support the tracking of data in multiple different languages including languages that use multi-byte encodings such as Chinese, Japanese, or Arabic. All field labels, system fields and custom fields in DevSuite solutions may track multi-byte characters.

To enable multilingual support in the DevSuite site, please contact TechExcel support for further assistance.

3.3 Administering System Date and Time Settings

DevSuite enables organizations to define standard date and time settings and formatting at the system-level. A system-defined time zone, date and time formats may be enforced globally to every project in the site or designated as default settings which may be overridden at the project or user-level.

All DevSuite clients connect to the DevSuite Application Server to get the current date and time. Date and time synchronization ensures that all users connecting to a DevTrack Database are using the same time clock standard.

To define a standard time zone: To define the standard time zone for a DevSuite system, select a time zone in the System

Time Zone dropdown list in the Date/Time Settings folder.

To define a standard time zone rule: To define rules for implementing the standard system time zone, select an option in the Time Zone area of the the System Time Zone dropdown list in the Date/Time Settings folder.

To define rules for implementing the standard system time zone, select an option in the Time Zone area of the Date/Time Settings page. The Time Zone area displays four mutually exclusive options:

Default: The standard time zone corresponds to the default time zone of the application server machine.

System Level: The standard time zone is enforced throughout the site. Every project in the site stamps all actions and entries based on the system time zone.

Project Level: The standard system time zone may be overridden at the project level. Project administrators may accept the standard system time zone or define a standard project time zone.

User Preference: The standard time zone may be overridden at the user- level. Project members may define the time zone of all dates and times displayed in their clients.

Defining System Date Formats

To define the standard date format for a DevSuite system, select a date format in the Date Format dropdown list in the Date/Time Settings folder.

The Date Format dropdown list displays 13 date formats:

M/d/yy
MM/dd/yyyy
yy/MM/dd
yyyy/MM/dd
yyyy-MM-dd
dd-MMM-yy
dd/MMM/yyyy
dd/MM/yy
dd/MM/yyyy
dd.MM.yy
dd.MM.yyyy
dd-MM-yy
dd-MM-yyyy

Defining System Time Formats

To define the standard time format for a DevSuite system, select a time format in the Time Format dropdown list in the Date/Time Settings folder.

The Time Format dropdown list displays four date formats:

h:mm:ss am/pm
hh:mm:ss am/pm
H:mm:ss
HH:mm:ss

To Define the Date/Time Format Rules

To define rules for implementing the standard system date and time, select an option in the Date/Time Format area of the Date/Time Settings page.

The Date/Time Format area displays three options:

System Defined: The standard date/time format is enforced throughout the site. Every project in the site stamps all actions and entries based on the system time zone.

Project Defined: The standard system date/time format may be overridden at the project level. Project administrators may accept the standard system time zone or define a standard project time zone.

User Defined: The standard time zone may be overridden at the user-level. Project members may define dates and times displayed in their clients.

Displaying Date and Times in Reports

To ensure that the date and time is displayed in the list panels and reports, click the Show Time Info check box.

3.4 Administering Keyword Searching

In DevSuite projects, the user may search for work items based on property definitions, workflow states, or the ?eywords?that are recorded in fields. A keyword is a term that captures the essence of a topic and so is likely to be used in the description of a knowledge item or work item making it an effective search condition.

DevTrack enables project members to search for keywords in the following fields: the Issue Title, Issue Description, History, Notes Title, Notes Description, Link Comments, Event Title, and Event Description. Using controls in the Keyword Search page of the System Settings folder, system administrators may define how precisely the DevTrack Search Engine interprets keyword searches:

Exact Keyword Search: If the exact keyword search option is selected, the search engine returns only those text strings that exactly match the keyword queried. Characters immediately preceding or following the keyword string ensure that the string is not returned as a match.

A keyword search that uses the keyword ?pp?returns work items that have the text string*app*in the fields queried. The keyword search does not return issues that include the words*apparel*,*appear*,*apple*, or*application*.

Keyword Search Includes Wildcard Characters:If the loose keyword search option is selected, the search engine returns all text strings match the keyword queried. Characters immediately preceding or following the keyword string are ignored.

A keyword search that uses the keyword ?pp?returns work items that include the words*apparel*,*appear*,*apple*, or*application*as well as the text string*app*.

3.5 Administering Time Categories

DevSuite timeand price analysis features enable development organizations to estimate and track the cost developing new features and the income generated from those features.

A time category is a system-level grouping of work activities that is defined by a default hourly rate. This hourly rate may represent the cost or income of development activities.

System administrators can create different time groups, and under each time group, multiple time categories can be defined. In DevSuite, all time categories are defined at the system-level and are shared across all projects in a site.

Defining Time Groups

To add a time group:

1 Click the New Time Group button in the Time Category page. The New Time Group dialog box appears.

2 Define the name of the time group, such as Development Time, QA Time and etc.

3 Select one of the system time type from the dropdown list.
Available system time types are: Design Time, Development Time, Test Time and Others.

4 Click the OK button. A new time group has been successfully created.

To edit or delate a time group:

1 Highlight the target time group and click the Edit or Delete button.

Defining Time Categories

To add a time category:

1Highlight the time group to which you wish to add a new time category.

2Click the New Time Category button in the Time Category page. The Add Time Category dialog box appears.

2Define the name of the time category in the Time Category text box.

3Define the hourly rate in the Default Hourly Rate text box.

4Click the OK button. The Add Time Category dialog box closes.

To edit a time category:
1Select a time category in the Time Category list of Time Category page.

2Click the Edit button. The Edit Time Category dialog box appears.

3Edit the name of the time category in the Time Category text box.

4Edit the hourly rate in the Default Hourly Rate text box.

5Click the OK button. The Edit Time Category dialog box closes.

To delete a time category:
1Select a time category in the Time Category list of Time Category page.

2Click the Delete button. A confirmation dialog box appears.

3Click the Yes button. The time category is removed from the Time Category list.

3.6 Administering LinkPlus Integration

TechExcel LinkPlus enables administrators to integrate DevSuite projects with existing third-party software. In order for DevTrack to recognize a ?ink?to LinkPlus, the Admin must first be configured to search for the specific application you have written.

There is no limit to the number of links you can create to a single DevTrack server. When creating or editing a Link, you must select the project to which this link will apply. From there, you specify the linked system and project ID that has been programmatically assigned through your LinkPlus application.

3.7 Managing DevSuite Admin Reports

DevSuite Admin reports enable system administrators to track system-level and project-level information about the projects and users in a DevSuitesite

DevSuitefeatures three DevSuite Admin reports:

User Information Report: The User Information Report displays detailed information about DevTrack users in a printable format.

Project Information Report: The Project Information Report displays detailed information about DevTrack projects in a printable format.

Admin Change Log Report: The Admin Change Log Report displays all of the changes made to the DevSuitesystem in a printable format.

3.7.1 User Information Report

The User Information Report enables system administrators to view high-level information about the project members of a specific project.

Report data includes the activity status (active or inactive) of each user, their license type, and account type in the selected project.

To define a User Account report:
1Select the UserInformationReport page in the Admin Report Settings folder.

2Take advantage of the User Selection dropdown list to filter users. System administrators can choose to show active users,

inactive users or both.

3Choose one of the two display styles using the Display Style dropdown list. Two styles are available: user information style and user list style.

4Optional: To print the report, click the Print button.

3.7.2 Project information Report

The Project Information report enables system administrators to view high-level information about a specific project including its status (active or inactive)and ID number. The Project Information report also shows the user name of all project administrators and project members as well as the privileges granted to each user based on their account type.

The Project Information Report provides administrators a detailed report about five project-specific areas:

Project Overview: The Project Overview area shows the project name, project status, project ID, starting number, and project description.

Project Administrators: The Project Administrators area lists all project administrators for the project.

Account Type Privileges: The Account Types Privileges Members area shows which project members are members of each account type and the privileges granted to each account type.

Groups: The Groups area shows which project members are members of each team group.

Folders: The Folders area shows all folders, folder owner groups, and the View, Put, and Get access controls assigned to each account type.

To define a Project Informationreport:

1Select the Project InformationReport page in the Admin Report Settings folder.

2Select a project in the Project dropdown list. The Project dropdown list displays the names of every project in theDevSuitesite.

3 Optional: To print the report, click the Print button.

3.7.3 Admin Change Log Report

The Admin Change Log Report enables system administrators to create printable reports that show every change made to a system or individual projects during a specified time period.

The report displays high-level information about multiple changes including the time and date of the change and the user account that made the change in a tabular format of rows and intersecting columns.

To define an Admin Change Log report:
1Select the Admin Change Log Report page in the Admin Report Settings folder.

2To define the scope of the report, select an option in the Project dropdown list. The Project dropdown list displays the names of every project in the siteas well asthe{Whole System} variable.

3Define the beginning and ending date of the report.

4Optional: To print the report, click the Print button.

3.8 Administering Welcome and Warning System Messages

Administrators may create two different types of system messages:

Welcome Message: Welcome messages are displayed in the Project to Login dialog box and may be used to provide project members with system-level information. Administrator-defined welcome message is the default message in the login

screen.

Warning Message: Warning messages are also displayed in the Project to Login dialog box and override the welcome message. Each warning message has a broadcast period and automatically overrides the welcome message and appears in the login screen for the duration of its broadcast period.

Both system welcome messages and system warning messages appear in the same text area of the login screen. Whenever a warning message is created, that message automatically overrides the welcome message and appears in the login screen for the duration of its broadcast period.

Chapter 4 Multiple Site Administration

4 Chapter 4- Multiple Site Administration

4.1 Understanding Site Administration

A DevSuite implementation?alled a*site*?onsists of a system settings project and one or more work projects. Every work project in a DevSuite work site shares a common database, knowledge base, and other system-level configurations that are defined in the DevSuite Admin client. Development organizations that operate multiple sites may centralize the management and control of DevSuite licenses using a*site family.*A site family consists of a system master site and one or more work sites.

DevTrack multiple site management enables development organizations to manage user licenses in a central repository so that licensed users world wide may access and work in any project in the site family.

Master Site: The master site stores and manages all user licenses in the site family. System administrators may use controls in the master site to generate authentication codes that enable work sites to join the site family.

Work Site: A work site runs its own DevSuite Database Server, DevSuite Application Server, DevSuite Document Server, and other DevTrack services. Every site may create and manage any number of KnowledgeWise, DevSpec, and DevTrack projects.

Although managed in a single site, DevTrack licenses are shared across the entire site family:

Fixed License: Fixed licenses managed in the master site enable users from any work site to access any project belonging to any site in the site family. Travelling users may use their license to log project belonging to other DevTrack sites.

Floating License: Floating licenses, however, are managed on a site-by-site basis. Each work site manages its own floating licenses. Users may not log into projects using floating licenses once the maximum number of users isreached.

DevTrack implementations that do not belong to a site family and which manage their own licenses are called*stand-alone sties*.

4.2 Administering DevSuite Sites

System administrators can administer the multi-site settings on the Site Settings page of the Multi Sites folder.

Creating a stand alone site

When first clicking on the Site Settings page, a dialogue box for asking to create a stand alone site will be displayed. This is the first step towards multi-site management.

To create a stand alone site, system administrators need to specify the site name and the web service URL for the current site. Make sure that server name is correctly specified.

To edit site information:

1Click the Change button in the Site Settings page of the Multi Site folder in the System Settings project. The Edit Site dialog box appears.

2Input the name of the site in the Site Name edit box.

3Input the web service URL in the Site Web Service URL edit box.

4 Optional: To test the connection to the web service, click the Test button.

5Click the OK button. The Edit Site dialog box closes.

Creating a Site Family

After establishing the stand alone site, system administrators can create a site family using controls and buttons on the Site Settings page.

To create a site family:

1Click the Create a New Multiple Site Family button in the Site Settings page of the Multi Site folder in the System Settings project. A confirmation dialog box appears.

2Click the Yes button. The confirmation dialog box closes. The Create a Site Family dialog box appears.

3Input the name of the site in the Site Name edit box. The default name is Master Site.

4Input the web service URL in the Site Web Service URL edit box.Make sure the server name is correctly specified.

5Click the OK button. The Create a Site Family dialog box closes. The site is displayed in the Site Family list of the Site Settings page.

Adding Work Sites to a Site Family

System administrators of the master site may create a family site and add work sites to a family site. Generating anauthentication code enables administrators of work sites to join the family site.

To add a work site to a site family:

1Click the Add button in the Site Settings manager. The Add a Regular Work Site dialog box appears.

2Enter the name of the work site in the Site field.

3Enter the WebService URLof the work site. Make sure the server name is correctly specified.

4Click the Generate buttontogetan authentication code.An authentication code is required to join a site family. Each site is identified by a unique ten digit alphanumeric authentication key.

5Copythe authentication codefor later use. The work site must use the authentication key to join the site family.

6Select the application type of the work site. Select the DevTrack radio button if the work site is also a DevSuite site, and select the DevTest radio button if it's a DevTest site.

7Click the OK button. The Add a Regular Work Site dialog box closes. The work site is added to the Site Family list in the Site Settings page.

To edit work site settings:

1Select a work site in the Site Family list of the Site Settings page.

2Click the Edit button in the Site Settings manager. The Edit a Regular Work Site dialog box appears.

3Editthe name of the work site in the Site field.

4EdittheWeb ServiceURLof the work site.

5Click the OK button. The Edit a Regular Work Site dialog box closes. The work site is added to the Site Family list in the Site Settings page.

Allocating Floating Licenses to Work Sites

To allocate floating licenses to work sites:
1Select the Site Settings page in the Multi Site folder of a system settings project.

2Click the Allocate Site Licenses button. The Multiple Site License Allocation dialog box appears.

3Select a site in the site list. The site list displays the name of every work site added to the site family and the number of floating licenses currently allocated to that site.

Note: By default all floating licenses are initially allocated to the master site. To allocate floating licenses to work sites, one or more floating licenses must be reallocated from the master site.

4Click the Edit button. The Update Floating License dialog box appears.

5Input the number of floating licenses to be allocated to the selected site in the Floating License Number edit box.

6Click the OK button. The Update Floating License dialog box closes.

Changing a System Master Site to a Standalone Site

System administrators of work sites or system master sites may use theChange to Stand Alone Site control buttonto change their sites into a stand alone site.

System master sites cannot be changed to a stand alone site as long as other sites belong to the site family. The system master site must first transfer the site family to another system master site, or all of the work sites must leave the site family before the site can be changed to a stand alone site.

Transferring System Master to Another Site

TheTransfer System Master Site buttonenables administrators to convert a system master site into a work siteand to convert a work site into the new system master sitewithin the site family

To transfer system master to another site:
1Click the Transfer System Master Site button in the Site Settings page. A warning dialog box appears.

2Click the Yes button. The Transfer System Master Site dialog box appears.

3Select a regular work site in the Select Regular Work Site list.

4Click the Next button. The Transfer System Master Site dialog box appears.

5Click the Transfer button.

Chapter 5 DevSuite Web Administration

5 Chapter 5- DevSuite Web Administration

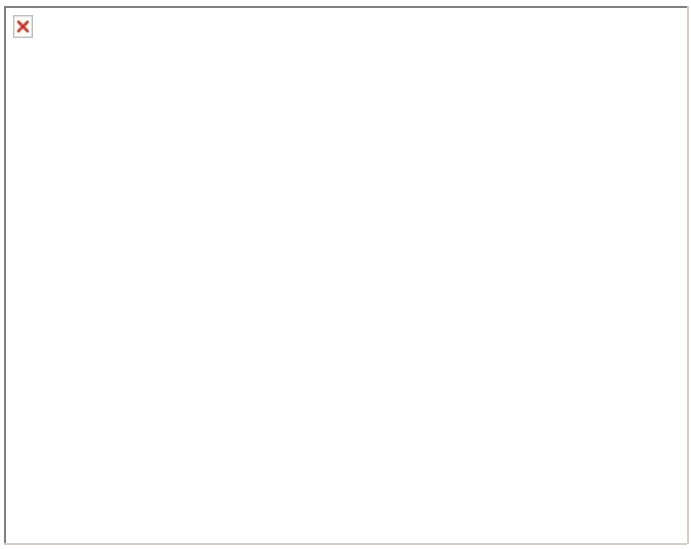
DevSuite Web administration includes the definition and configuration of DevSuite Web Server settings for every DevTrack project in the site, including DevTrack Web authentication, system runtime keys,

In this chapter:

- Understanding the DevSuite Web Server
- Administering the DevSuite Web Server
- Administering Image and Multimedia Attachment Settings
- Administering DevTrack Web Authentication and Access Controls

5.1 Understanding DevSuite Web Server Administration

The DevSuite Web Server is an optional DevSuite component that enables distributed development teams to access to development and knowledge management projects over the Internet.



All data is completely synchronized in real time to the site database.The DevSuite Web Server connects to the DevSuite Application Server to retrieve database access information and log on to the DevSuite Database Server through an ODBC connection.

The DevSuite Web Server makes DevTrack development projects, KnowledgeWise knowledge management projects, and the Beta Customer Portal available to project stakeholders anywhere in the world." All that is required to access a project is a user name, password, and a standard web browser (Firefox or Internet Explorer).

The DevSuite Web Server is an optional client. The DevSuite Web Server needs to be installedon the serverif userswould like toaccess DevTrack through the Web.

Note: TechExcel recommends installing the DevSuite Admin module on the DevSuite Web Server computer so that changes made to the DevSuite Web Server can be quickly modified in DevSuite Admin.

5.2 Administering the DevSuite Web Server

Once the DevSuite Web Server is installed on the machine, system administrators can configure the web server settings using controls available in the DevSuite Web sub-folders in the Start Menu.

Configuring the DevSuite Web Server

The DevSuite Web Server connects to the DevSuite Application Server to retrieve database access information and log on to the DevSuite Database Server through an ODBC connection.

Connections to the DevSuite Application Server and DevSuite Database Server are automatically configured when DevSuite is installed. Connection settings may be changed or updated using controls in the DevSuite Web Server Configuration window.

To configure or update DevSuite Web Server:

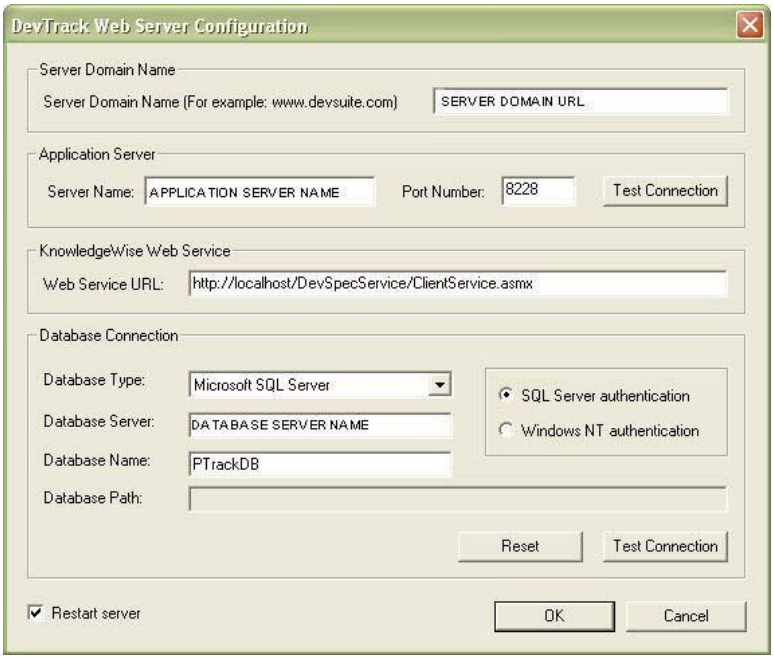
1Select Start > All Programs > TechExcel DevSuite > DevSuite Web > Configure DevSuite Web.
The DevSuite Web Server Configuration shortcut is created in the DevSuite Web Server folder by default when the DevSuite Web Server is installed. The DevSuite Web Server Configuration window appears.

2To define the connection to the DevSuite Application Server, enter the server machinethathoststhe DevSuite Application Server and port number.

3To define the connection to the DevSuite Database Server, enter the DBMS, the server machine hosting the DevSuite Database Serverandthe database name

4Define the authentication method.
To use Windows NT authentication, select the Windows NT Authentication option button.
To use SQL Server authentication, select the SQL Server Authentication option button.

5Click the OK button.



5.2.1 Administering Multiple DevSuite Web Servers

Whenever a DevSuite site is implemented across wide area networks, remote users may sometimes report performance issues that are due to
(1) latency across long distances and (2)lack of bandwidth at remote or home offices.

TechExcel recommends that development organizations that have implemented their DevSuite site across multiple offices install one or more regional web servers to distribute the load across the regional web servers, reduce the processing required of the primary web server, and to eliminate regional bottlenecks.

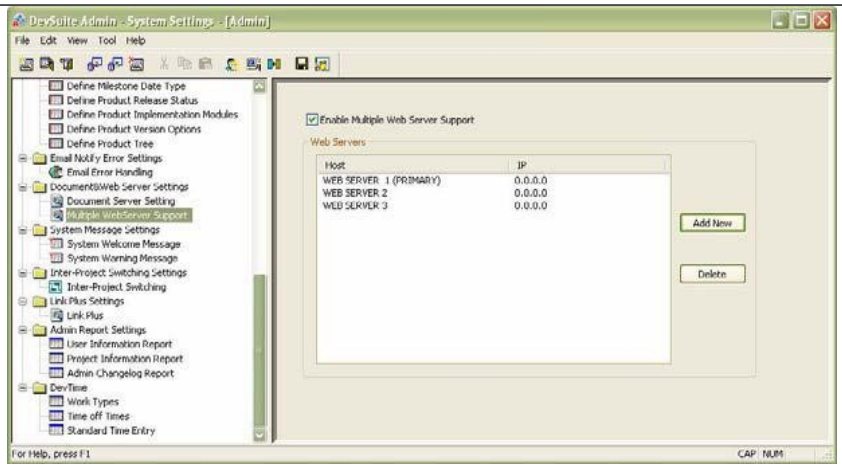
Installing a regional web server also guarantees performance by establishing a singlelink between the two offices, while cutting down on traffic going overseas. Since bandwidth is guaranteed between the database server and the regional web server, data transfer is constant and rapid.

If there is no dedicated bandwidth between the remote DevTrack Web server and the central DB server, installing a remote DevTrack Web server may not result in faster performance.

Install a regional web server if:

A centralized web server is geographically distant from some offices
Clients must connect from various local offices within a single region
At least 2MB of dedicated bandwidth exists between the regional web server and the central DevTrack database server

Using controls in the Multiple Web Server Support page in the Document & Web Server Settings folder, system administrators may enable support for multiple web servers, define connections to each DevSuite Web Server, and designate the primary DevSuite Web Server.



To define connections to multiple DevSuite Web Servers:

1Go tothe Multiple Web Server Support page in the Document & Web Server Settings folder of the system settings project.

2Click the Add New button. The Add Web Server dialog box appears.

3Define the host name and IP address of the machine running the DevSuite Web Server.

4To define the DevSuite Web Server as the primary web server, select the Is Primary Web Server option box.

5Click the OK button. The Add Web Server dialog box closes.

5.3 DevTrack Web Administration

The DevTrack web client is a “thin client” that enables project team members to view, manage, and track incidents, events, and customers through the Internet using a web browser.

Development organizations may deploy DevTrack using any combination of Windows and web clients. DevTrack Web may be used as a stand-alone product or in tandem with the DevTrack Windows client uniting internal and external development teams. All data is completely synchronized in real time to the central DevTrack database.

5.3.1 Defining DevTrack Web Connections

When DevTrack Web Server is installed and the web clients are accessed by users, the DevTrack web URL needs to be correctly specified in the DevSuite Admin. In this way, changes made through the Admin Client can be pushed and updated to the Web Client.

To define a DevTrack Web connection:

1Select Web Information Settings > Web Support > General Settings.

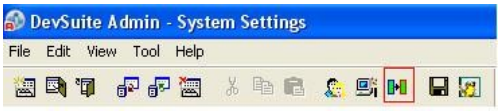
2Define the DevTrack Web URL for in the DevSuite Web Server Path text box. The default URL isHTTP://SERVERNAME/SCRIPTS/TEXCEL/DEVTRACK/Please replace SERVERNAME with the correct host name.

3 Optional: To test the connection, click the Connection Test button.

Note: Starting and Stopping the DevSuite Web Server

In DevSuite, project-level configurations and customizations may not be immediately displayed to users accessing those projects through the DevSuite Web Server.

To ensure that all changes are properly displayed, system administrators must stop and restart the DevSuite Web Server. To stop and start the DevSuite Web Server, click the Reload Web Settings button in the tool bar of the DevSuite Admin client.



5.3.2 Defining DevTrack Web Issue List and Reporting Options

In DevTrack web client, issue list and reports can be set to perform either auto refresh or manual refresh.System administrators can define the refresh mode in the DevSuite Admin Client.

To define DevTrack Web issue list options:

1Select Web Information Settings > Web Support > General Settings.

2Select an option in the Issue List And Report Refresh Option area:

- **User definable refresh:**individual users would be able to define the refresh mode in the web client based on their own preference.
- **Manual refresh only:** issue lists and reports would refresh only when the refresh is demanded by users.
- **Auto refresh only:**system would auto refresh both the issue list and the reports.

5.3.3 Administering Interproject Switching

In an integrated TechExcel DevSuite-Service Suite environment, interproject switching enables users of the DevTrack Web client to quickly access TechExcel ServiceWise or TechExcel CustomerWise projects using links in the DevTrack Web Home page. The Home page shows high-level information about a user's projects including the project name, description, and the number of issues (or incidents) that the user has open in each project. The Home page may display links to multiple DevTrack, ServiceWise, and CustomerWise projects.

To enable interproject switching between DevTrack and CustomerWise

1Select the Interproject Switching page in the Interproject Switching Setting folder.

2Define the path to the DevTrack Web Server in the DevTrack Server edit box.

3To enable switching between DevTrack and TechExcel CustomerWise
- Click the Switching between DevTrack and TechExcel CustomerWise Project check box.
- Define the path of a TechExcel CustomerWise Web Server in the TechExcel CustomerWise Web Server edit box.

4 Optional: To test the connection, click the Connection Test button.

To enable interproject switching between DevTrack and ServiceWise

1Select the Interproject Switching page in the Interproject Switching Setting folder.

2Define the path to the DevTrack Web Server in the DevTrack Server edit box.

3To enable switching between DevTrack and TechExcel ServiceWise
- Click the Switching between DevTrack and TechExcel CustomerWise Project check box.
- Define the path of a TechExcel ServiceWise Web Server in the TechExcel CustomerWise Web Server edit box.

4 Optional: To test the connection, click the Connection Test button.

5.3.4 Defining DevTrack Web Titles, Welcome Messages and Logout URL

System administrators can define the DevTrack Web Titles and welcome messages that they would like the users to see when accessing DevTrack Web Client. Meanwhile, system administrators can also define the web page users would be taken to after they log out of the web client.

To define DevTrack login title and welcome messages:

1Select Web Information Settings > Web Support > General Settings.

2Define the title of the DevTrack Web login page in the Web Login Title text box.

3Define the message displayed in the DevTrack Web login page in the Web Login Welcome Message text box.
- To edit the message using a WYSIWYG text editor, click the Edit button.
- To edit the message using HTML markup tags, click the Edit Source button.

To define DevTrack Web Logout URL:

1Select Web Information Settings > Web Support > General Settings.

2Define the Web URL that users would be taken to after logging out in the Web Logout URL text box.
3To test the connection, click the Connection Test button.

5.4 KnowledgeWise Web Administration

KnowledgeWise is the digital repository of DevSuite where documents, images and all sorts of digital assets are stored and shared. When KnowledgeWise Web is deployed, the correct web URL as well as the web service URL need to be specified in the DevSuite Admin Client. In this way, changes made in the Admin Client can be successfully pushed and updated to the KnowledgeWise Web Client.

To define KnowledgeWise Web connection:
1Select Web Information Settings > Web Support > General Settings.

2Define the KnowledgeWise Web URL in the KnowledgeWise Web Server Path text box.
The default URL isHTTP://SERVERNAME/KNOWLEDGEWISEWEB/LOGIN.ASPX
Make sure that the correct host name is specified.

3Define the KnowledgeWise Web Service URL in the KnowledgeWise Web Service Path text box.
The default URL isHTTP://SERVERNAME/DEVSPECSERVICE/CLIENTSERVICE.ASMX
Make sure that the correct host name is specified.

4 **Optional**: To test the connection, click the Connection Test button.

5.5 Beta Customer Portal Administration

5.6 Administering Image and Multimedia Attachment Settings

In DevSuite, project team members may add attachments whenever they submit or update a project item. Attachments may include documents, worksheets, screen captures, of any file type.

Using controls in the Web Image Settings page, system administrators may enable image, movie, and sound attachments to be displayed in the DevTrack Web client, define the size of thumbnail, small, medium, and large image previews, and define the maximum size of multimedia attachments.

5.6.1 Defining DevTrack Web Image Attachment Settings

Project team members accessing DevTrack development projects through the DevTrack Web client may choose to view attached images as links or as previews in the web browser. Image previews are defined by administrator-defined sizes: thumbnail, small, medium, or large. If the image preview is displayed directly in the client, the user may click the preview to view the image at its original size and resolution.

To define image attachment support:
1Select Web Information Settings > Web Image Settings > Web Image Settings.

2Select the Support Directly Display Image Attachments check box.

3 **Optional**: To define the maximum size of thumbnail images, enter the height and width of a thumbnail image in pixels.

4 **Optional**: To enable support for small images, select the Small Image check box and enter the height and width of a small image in pixels.

5 **Optional**: To enable support for medium images, select the Medium Image check box and enter the height and width of a medium image in pixels.

6 **Optional**: To enable support for large images, select the Large Image check box and enter the height and width of a large image in pixels.

5.6.2 Defining DevTrack Web Movies and Sound Attachments Settings

System administrators can enable the attachments of multimedia, including movies and attachments, in the web client using controls on the Web Image Settings page.

To enabl attachment of movies and sound:

1Select Web Information Settings > Web Image Settings > Web Image Settings.

2Select the Support Movies and Sound Attachments check box.

3To define the maximum size of movies or sounds, enter the maxium file size in the text box.

5.7 Administering DevTrack Web Authentication and Access Controls

Authentication is the process by which DevSuite verifies the identity of users and controls access to DevTrack projects.

DevSuite supports three methods for authenticating the identity of users logging into projects through the DevTrack Web Server: basic authentication, Windows NT Server authentication, and third-party single sign on authentication. Using controls in the Web Authentication page of the Web Information Settings folder in the system settings project, system administrators may determine the method used to authenticate project members accessing DevTrack projects through the DevTrack Web Server.

Configuring DevTrack Login and Password Authentication

If the DevTrack login and password authentication method is selected, project members may use their DevSuite user name and password to log into DevTrack development projects through DevTrack Web. DevTrack Login and Password Authentication Project access is usually controlled by means of a unique username and password that is assigned to each project member. This method, called*basic authentication*, requires that users establish their identity by supplying the correct username and password every time they log into a DevSuite project.

Windows NT User Authentication

If this option is selected the users need only enter their Windows NT username and password the first time they log into a project through DevTrack Web. The user is automatically logged into the project for all future sessions.

Third Party Authentication (DevTrack Web SSO)

Single sign on (SSO) enables a user to employ a single login ID and password to log into multiple applications within an enterprise. Businesses that employ third-party SSO authentication solutions may use those tools to authenticate DevTrack Web users and log them into development projects.

Note: DevSuite supports LDAP authentication for logging into development projects through the DevTrack Windows client. LDAP directory server authentication is not supported for DevTrack Web.

5.7.1 Configuring DevTrack Login and Password Authentication

If the DevTrack login and password authentication method is selected, project members may use their DevSuite user name and password to log into DevTrack development projects through DevTrack Web.

To configure DevTrack Web login and password authentication, select theDevTrack Login and Password Authenticationoption in the Web Authentication page of the Web Information Settings folder. Users must establish their identity by supplying the correct username and password every time they log into a development project through DevTrack Web.

5.7.2 Configuring Windows NT User Authentication

If the Windows NT authentication method is selected, project members may use their Windows NT user name and password to log into DevTrack development projects through DevTrack Web.

The first time the project member accesses a project through the DevTrack Web Server, they must authenticate their identity using their DevTrack user name and password. DevSuite stores their username and password and links their DevTrack and

Windows NT usernames and passwords. In all future DevTrack Web sessions, the project member is automatically authenticated based on their Windows NT username and bypasses the DevTrack Web Login page.

To configure DevTrack Windows NT User Authentication:

1Configure IIS to support basic authentication in the Authentication Methods window of the Directory Security tab in the DevTrack Properties manager.

2Select theWindows NT User Authenticationoption in the Web Authentication page of the Web Information Settings folder. The first time that the user logs into a development project using their NT user name and password, the user's NT username is saved is saved in the User Manager as that user's auto login account.

5.7.3 Configuring Third Party Authentication—DevTrack Web SSO

Single sign on (SSO) enables a user to employ a single login ID and password to log into multiple applications within an enterprise. Businesses that employ third-party SSO authentication solutions may use those tools to authenticate DevTrack Web users and log them into development projects.

DevTrack Web SSO authentication enables project members to bypass the DevTrack Web login page and automatically access DevTrack development projects once that user has logged into their system using their SSO login ID and password.

For example, an organization may employ a single sign on (SSO) portal to authenticate the identity of users for multiple applications using basic authentication, digital signatures, PINs generated by a security token, or any other authentication method.

If the SSO portal successfully authenticates the user, the page may then display URLs to multiple applications enabling the user to bypass the login page in each. The link to the DevTrack Web project must pass the following authentication credentials: the DevTrack project ID, the user's username, and the user's password to the DevTrack Web login page.

http://WebServerName/scripts/texcel/devtrack/devtrack.dll?switch-prod?projectid=#&login=XX&pwd=yy

So long as the authentication portal passes the correct authentication credentials to the DevTrack Web login page—a project ID, username, and password that is stored in the DevSuite database, the user is logged into the development project.

DevTrack Web SSO is particularly useful to organizations that have integrated DevTrack Web or the Beta Customer Portal with their existing intranet or support web site and wish to enable project members or customers to bypass the DevTrack Web Login page.

To enable DevTrack Web third party authentication integration, select the Third Party Software Authenticationoption in the Web Authentication page of the Web Information Settings folder. The rest of the configuration is done in the authentication application. No configuration is required within DevSuite.

Chapter 6 DevSuite Mail Service Administration

6 Chapter 6- DevSuite Mail Service Administration

This chapter describes the configuration and administration of the DevSuite Mail Service.

In this chapter:

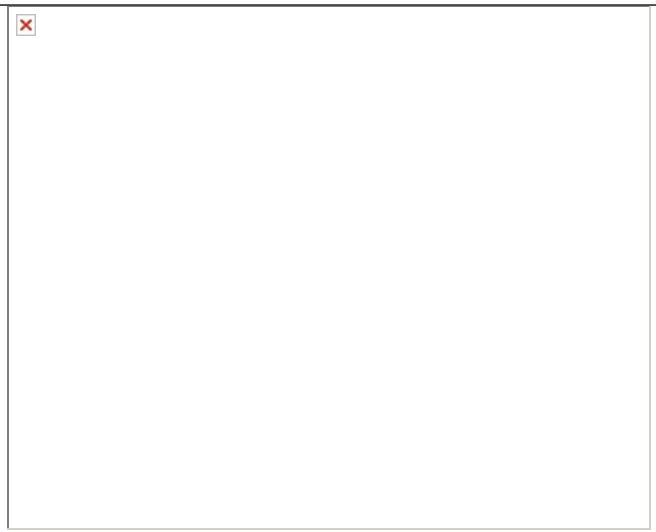
- Understanding DevSuite Mail Services
- Administering DevSuite Mail Services

6.1 Understanding DevSuite Mail Services

DevSuite Mail Services enable development organizations to integrate DevSuite workflow, event scheduling, and issue escalation with the convenience of e-mail.

DevSuite supports multiple e-mail based services including e-mail notification, e-mail escalation, issue submission by e-mail, and event scheduling.

The DevSuite Mail Services (Notification Service and Retrieval Service) connect clients to the DevSuite Database:



The DevSuite Mail Service is a Windows NT service that runs as a mail agent and works with any MAPI- or SMTP-compliant mail server." All e-mail delivery is achieved through an e-mail server using a POP3 account.

The DevSuite Email Service communicates with the DevSuite Database Server and email server to find and send email messages generated within a DevTrack project. The DevSuite Mail Service retrieves the user name and password needed to access to the DevSuite Database Server from the DevSuite Application Server.

Using a username and password retrieved from the DevSuite Application Server, the DevSuite Mail Service talks to the DevSuite Database Server and queries the database for unsent email.

If unsent email is found, the DevSuite Mail Service talks to the email server and sends out all email messages. After successfully sending email messages, the DevSuite Mail Service resets flags in the DevSuite database.

6.2 Administering DevSuite Mail Services

DevSuite Mail Services enable development organizations to integrate DevSuite workflow, event scheduling, and issue escalation with the convenience of e-mail.

Although all email notification, escalation, and issue submission rules are defined on a project-by-project basis, the DevSuite system administrator is responsible for installing the installing the DevSuite Mail Services and configuring the DevSuite environment to operate properly.

System administrators are also responsible for defining e-mail error handling rules. DevSuite e-mail error handling enables administrators to track and diagnose problems with DevSuite e-mail services.

E-mail error handling administrative tasks include:

- Installing and Configuring the Notification/ Retrieval Service
- Administering E-mail Error Handling
- Defining Email Trace Levels
- Defining Mail Service Settings
- Defining Email Error Log Files
- Define Mail Service Messages
- Defining E-mail Service Error Reports

6.2.1 Installing and Configuring the Notification/ Retrieval Service

DevSuite email notifications and escalations are handled by the DevTrack Mail Service, and DevSuite issue submission by email is handled by the DevTrack Mail Retrieval Service. System administrators need to install either or both services based on the business needs.

To install the DevTrack Mail Service

1Click the DevTrack Mail Service installer.

2The install shield appears and may take a few moments to load. The Welcome to the Install Wizard for DevTrack Mail

Service window appears. Click the Next button. The Destination Folder window appears.

3Click the Next button. The License Agreement window appears.

4Read the license agreement and select the I accept the terms in the license agreement option. Click the Next button. The Destination Folder window appears.

5Select a destination directory. By default, the DevTrack Mail Service is installed to C:\PROGRAM FILES\TECHEXCEL\DTSERVER|DEVTRACKMAIL\

- To accept the default destination directory, click the Next button.
- To change the destination directory, click the Change button and navigate to a directory, and then click the Next button.

The Ready to Install Program window appears.

6Click the Next button. The DevTrack Mail Service Configuration window appears.

7Review and update the Server configuration settings as necessary.

- To test the connection to the application server, click the Test Connection button in the Application Server area
- To test the connection to the DevSuite Database Server, click the Test Connection button in the Database Connection area.

8Click the OK button. The Installing DevTrack Mail Service window appears.

9Click the Next button. The DevTrack Mail Service Installation Completed page appears

10Click the Finish button.

To install the DevTrack Mail Retrieval Service

1Click the DevTrack Mail Retrieval Service installer.

2The install shield appears and may take a few moments to load. The Welcome to the Install Wizard for DevTrack Mail Retrieval Service window appears. Click the Next button. The Destination Folder window appears.

3Click the Next button. The License Agreement window appears.

4Read the license agreement and select the I accept the terms in the license agreement option. Click the Next button. The Destination Folder window appears.

5Select a destination directory. By default, the DevTrack Mail Retrieval Service is installed to C:\PROGRAM FILES\TECHEXCEL\DTSERVER|DTMAIL\RECEIVE

- To accept the default destination directory, click the Next button.
- To change the destination directory, click the Change button and navigate to a directory, and then click the Next button.

The Ready to Install Program window appears.

6Click the Next button. The DevTrack Mail Retrieval Service Configuration window appears.

7Review and update the configuration settings as necessary.

- To test the connection to the application server, click the Test Connection button in the Application Server area
- To test the connection to the DevSuite Database Server, click the Test Connection button in the Database Connection area.

8Click the OK button. The Installing DevTrack Mail Retrieval Service window appears.

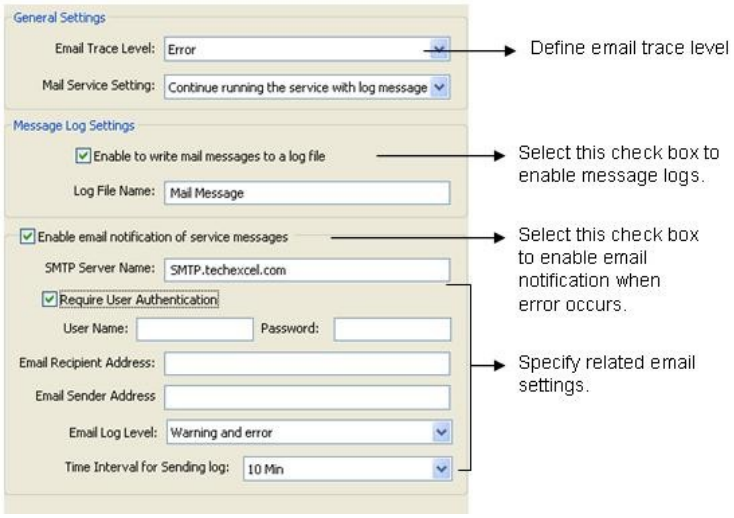
9Click the Next button. The DevTrack Mail Retrieval Service Installation Completed page appears

10Click the Finish button.

6.2.2 Administering E-mail Error Handling

DevSuite e-mail error handling enables system administrators to track and diagnose problems with incoming and outgoing e-mail notifications. All e-mail error handing settings are defined in the System Settings project.

System-level settings apply to all projects in the DevSuite site. Using the Email Error Handling page, system administrators may define general settings including trace levels for e-mail notification errors, create error logs, and define e-mail notification service messages.



6.2.2.1 Defining Email Trace Levels

In DevSuite, email trace levels define the type of messages that are logged when a DevTrack Email Service error occurs.

The DevTrack Email Service may log three different types of messages: info messages, warning messages, and error trace messages.

To define email trace settings:

1Select Email Error Handling page in the Email Notify Error Message folder.

2Select an option in the E-mail Trace Level dropdown list.

The E-mail Trace Level dropdown list displays three options:

- To log all Info, warning, and error messages, select the Info, Warning, and Error option.
- To log warning and error messages, select the Warning and Error option.
- To log Error messages only, select the Error option.

6.2.2.2 Defining Mail Service Sensitivity Settings

Administrator-defined email service settings determine the sensitivity of DevTrack Email Service.

To define email service settings:

1Select Email Error Handling page in the Email Notify Error Message folder.

2Select an option in the E-mail Service Setting dropdown list. The E-mail Service Level dropdown list displays two options:

- TheContinue Running the Service with Log Messagesservice level: If the Continue Running the Service option is selected, the DevTrack Email Service continues to run after an error occurs.
- TheStop Service when Error Occursservice level: If the Stop Service when Error Occurs option is selected, the DevTrack Email Service immediately stops after an error occurs

6.2.2.3 Defining Email Error Log Files

System administrators may turn on the email error logs on the Email Error Handling page. This would allow the system to write error messages to the log file defined.

Error logs are extremely helpful for diagnosis when DevTrack Mail Service runs into issues.

To define email error log files:

1Select Email Error Handling page in the Email Notify Error Message folder.

2Select the Enable to Write Error Messages to Log File check box.

3Define the name of the log file in the Log File Name text box.

6.2.2.4 Defining Email Service Error Reports

Administrators may use the Email Notification Error Service Handling Setting manager to define e-mail notification error reports.

If an e-mail notification message is not delivered to any selected recipients, the sender automatically receives an error report e-mail and e-mail attachment that explains why the e-mail notification was not delivered.

System administrators define the recipient, sender, the level of detail, and the time interval of the e-mail service error report e-mail in the Email Error Handling page in the Email Notify Error Settings folder.

To define e-mail service messages:

1Select Email Error Handling page in the Email Notify Error Message folder.

2Select theEnable Email Notification of Service Messagescheck box.

3Define the name of the SMTP server in the SMTP Server Name text box.

4 Optional: To require user authentication, select theRequire User Authenticationcheck box and enter the user name and password.

5Define the e-mail address for service message recipients in the E-mail Recipients text box.

6Define the e-mail address for service message senders in the E-mail Senders text box.

7Select an option from the E-mail Log Level dropdown list. The E-mail Log Level dropdown list displays two options:

- Warning and Error
- Error

8Definethe time interval for sending logs.Select an option from the Time Interval for Sending Log dropdown list. The E-mail Log Level dropdown list displays 12 options:

10 Min	20 Min	30 Min
40 Min	50 Min	1 Hour
2 Hour	3 Hour	4 Hour
5 Hour	8 Hour	12 Hour

6.2.3 Troubleshooting DevTrack Notification Service Errors

When the DevTrack Mail Service is not working properly, running a trace version will help system administrators to troubleshoot the service. A trace is a detailed log that records the current action and response of the mail service.

To run a trace version of the DevTrack E-mail service, use thetrace.exe under the DevTrack Mail installation directory. The default installation folder is
C:\PROGRAM FILES\TECHEXCEL\DTSERVER|DEVTRACKMAIL\

Administrators may also send detailed information collected in the trace log to TechExcel support.

To run the trace version of DevTrack Mail Service:

1Run the Trace.exe.

2The trace window appears.

3Click the archive button, type in a file name to save the trace message—for example, trace.txt.

4Create a new issue in your DevTrack, or perform any actions that would trigger the email rules.

5Start your DevTrack mail service from control panel > Services. The trace message is displayed in the dbtrace window—for example,init DB.

6Wait five minutes, click Archive button again (this will close the trace message file), and then send that file to TechExcel.

TechExcel DevTrack uses the EasyMail application API to handle the DevTrack client e-mail processing.

Error Code	Error Description
1	An unknown error has occurred
2	An error has resulted because there was no message specified

3	The process has run out of memory.
4	An error has occurred due to a problem with the message body or attachments.
5	There was a problem initiating the conversation with the mail server. Ensure the setting of the Domain property is correct.
6	There was an error terminating the conversation with the SMTP mail server.
7	The from address was not formatted correctly or was rejected by the SMTP mail server. Some SMTP mail servers will only accept mail from particular addresses or domains. SMTP mail servers may also reject a from address if the server can not successfully do a reverse lookup on the from address.
8	An error was reported in response to a recipient address. The SMTP server may refuse to handle mail for unknown recipients.
9	There was an error connecting to the SMTP mail server.
10	There was an error opening a file. If you have specified file attachments, ensure that they exist and you have access to them.
11	There was an error reading a file. If you have specified file attachments, ensure that they exist and you have access to them.
12	There was an error writing to a file. Ensure that you have sufficient disk space.
15	No mail server specified.
16	There was a problem with the connection and a socket error occurred
17	Could not resolve host.
18	Connected but server sent back bad response.
19	Could not create thread
20	Cancelled as a result of calling the Cancel() method
21	The operation timed-out while the host was being resolved
22	The operation timed-out while connecting
24	ESMTP Authentication failed.
25	The selected ESMTP Authentication mode is not supported by the server.
26	ESMTP Authentication protocol error
27	Socket Time-out Error

Chapter 7 Product Release Management Administrati

7 Chapter 7- Product Release Management Administration

This chapter describes the product tree and how the product, version, and build structures defined within the product tree may be used to define iterative development and subproject structures in a DevSuite site.

In this chapter:

- Understanding Product Release Management Administration
- Administering the Product Tree
- Administering Product Implementation Modules
- Administering Product Version Options

7.1 Understanding Produce Release Management Administration

DevSuite product release management structures define and organize the product releases—products, versions, and builds—under development in a DevSuite site and define a framework for controlling how these releases are designed and implemented.

The*product tree*is a hierarchical structure consisting of folders and subfolders that organize the products, versions, and builds under development in a site. This structure enables project administrators to define and control the scope of the work carried out in individualDevSpec, DevPlan and DevTrackprojects.

The product tree defines the scope of the product releases that*may be*managed in a DevSuite site. The product tree enables project administrators to define the scope of the development tasks managed in their projects—to manage a product release in specific project, an appropriate product, version, or implementation modulemust be defined “applicable” to the project.

The product tree also enables project administrators to control how product releases are managed*within*a project by defining the scope of project management folders and subfolders.

- In a DevSpec requirements management project, the product t tree provides project managers a framework for defining the scope of the specifications managed in each specification folder.
- In a DevTrack development project, the product tree defines the framework in which development issues and branch

issues are managed in the project. Each subproject in a DevTrack project is defined as a release folder and linked to a specific product, version, or build.

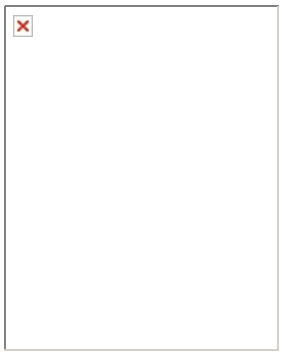
System-level product release management is the process of defining and organizing the products, versions, and builds under development in a DevSuite site and for defining the*implementation modules*that are applicable to each product and version.

System administrators can mange all the aspects of the product release in theProduct Release Managementfolder in the System Settings project.

7.2 Administering the Product Tree

In DevSuite, the product tree is a hierarchical tree structure that organizes and represents the product categories, products, versions, and builds under development in a DevSuite site.

Each product, version, and build represents a distinct deliverable that is managed and tracked in DevSuite. The product tree defines parent-child relationships between project deliverables:



Product tree parent-child relationships enable project administrators to quickly identify and define appropriate product categories, products, versions, and builds as applicable to their development projects."

Every product release deliverable is identified by the color of its folder and its relationship to the other deliverables in the product tree:

Product Category:A product category is any grouping of products—such as a product line—that a business uses to organize the products that it develops." A product category may be the parent to one or more products or child product categories.

Product:A product represents the release of a specific product that is under development in a DevSuite site." A product may be the parent of many different versions of that product. Every product is defined by its release status and one or more applicable implementation modules.

Version:A version (release version) represents the release of a version of a product." Every version is the child of a single product and may be the parent to one or more builds. Every version is defined by its release status, a set of milestones, a set of version options, and one or more applicable implementation modules.

Build:A build represents a specific build of a product version." Every build is the child of a single version. Every build is defined by its release status.

All products, versions, and builds are defined by their*release status*.A release status is an administrator-defined workflow state that identifies the progress of that deliverable within its development life cycle and determines the work that may be done on product deliverables in that state. For more information on the releases status of products, versions, and builds see “Administering Product, Version, and Build Release Statuses”.

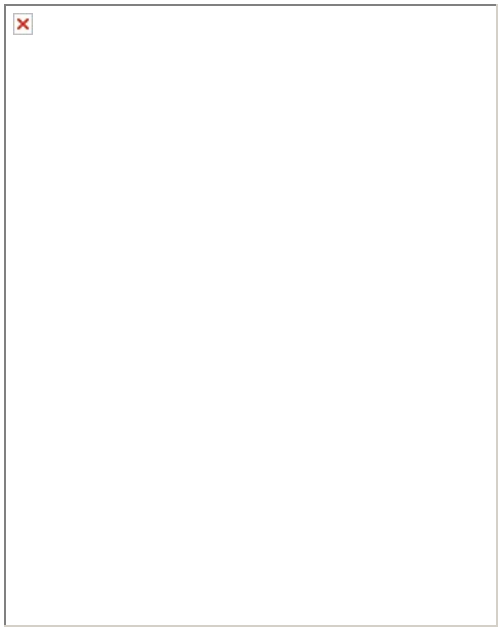
All products and versions are defined by their*implementation modules*. In DevSuite, a product implementation module is a tool for organizing a distinct area of development—such as a platform (Windows, Linux), an application (thick client, thin client, smart client), plug-in, tool, widget, or any other development item. For more information on the implementation modules of products and versions see “Administering Product Implementation Modules”.

7.2.1 Administering Product Catogories

A product category is any grouping of products that a business uses to organize the products that it develops. A product category may be the parent to one or more products or child product categories.

A product category is a tool for organizing the products under development in a DevSuite site. A product category may represent a product line, a suite of applications, or another other grouping of products that makes sense for the business.

The scope of development activities in DevSpec and DevTrack projects may be defined by product category.



To add a product category:
1Select Product Release Management > Define Product Tree in the tree panel."

2Select the root folder or a product category subfolder in the Product Tree. A product category may be the child of another product category or of the root folder in the product tree.

3Click the New Category button. The New Product Category dialog box appears.

4Define the title and description of the product category.

5Click the OK button. The product category folder is displayed in the product tree panel.

To define a product category:
1Select Product Release Management > Define Product Tree in the tree panel.

2Select a product category in the product tree.

3Define product properties in the Product Properties tab. Every product category is defined by its title and description.

To delete a product:
1Select Product Release Management > Define Product Tree in the tree panel.

2Select a product category in the product tree.

3Click the Delete button. A confirmation dialog box appears.

4Click the Yes button.

7.2.2 Administering Products

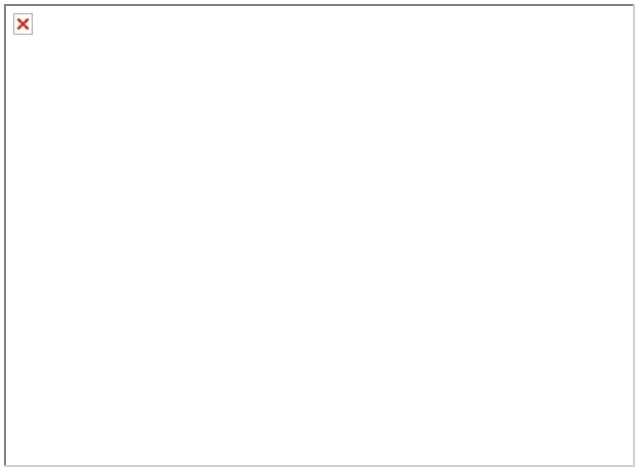
A product represents the release of a specific product that is under development in a DevSuite site. A product may be the parent of many different versions of that product.

Every product is defined by its product category, release status, and one or more applicable implementation modules:

Release Status: A release status is an administrator-defined workflow state that identifies the progress of that deliverable within its development life cycle and determines the work that may be done on product deliverables in that state.

Implementation Module: An implementation module is a tool for organizing a distinct area of development—such as a platform (Windows, Linux), an application (thick client, thin client, smart client), plug-in, tool, widget, or any other development item.

The scope of development activities in DevSpec and DevTrack projects may be defined by product.



To add a product:
1Select Product Release Management > Define Product Tree in the tree panel."
2Select a product category subfolder in the Product Tree. All product category subfolders in the Product Tree are blue. Every product is the child of a product category.

3Click the New Product button. The New Product dialog box appears.

4Define a unique name for the product in the Product Name control.

5Click the OK button. The New Product dialog box closes.

To define a product:
1Select Product Release Management > Define Product Tree in the tree panel.

2Select a product subfolder in the product tree. All product subfolders in the Product Tree aregreen.

3Define product properties in the Product Properties tab. A product is defined by its title, description, and release status.

4Identify the implementation modules that are applicable to the product in the Implementation Module tab.

To delete a product:
1Select Product Release Management > Define Product Tree in the tree panel.

2Select a product subfolder in the product tree.

3Click the Delete button. A confirmation dialog box appears.

4Click the Yes button.

7.2.3 Administering Versions

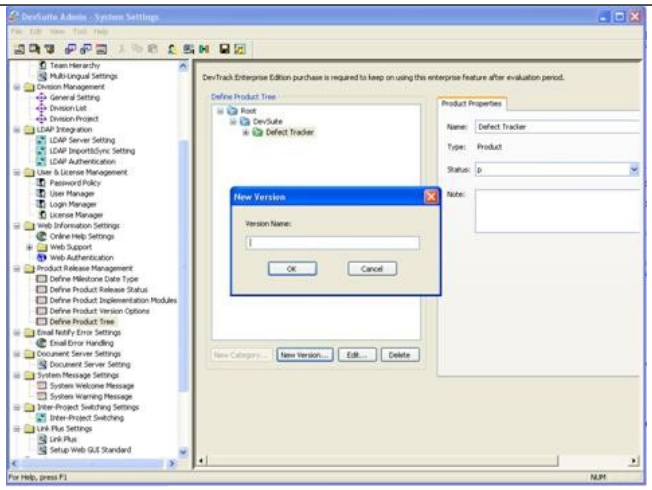
A version represents the release of a version of a product. Every version is the child of a single product and may be the parent to one or more builds.

Every version is defined by its parent product, a release status, a set of milestones, a set of version options, and one or more applicable implementation modules.

Release Status:A release status is an administrator-defined workflow state that identifies the progress of that deliverable within its development life cycle and determines the work that may be done on product deliverables in that state.

Version Milestone:A milestone is a point in time that represents an important intermediate event in a project. Milestones commonly represent the completion of key components, reporting requirements, and the beginning or end of project phases. Every version milestone is based on a milestone date type.

The scope of development activities in DevSpec and DevTrack projects may be defined by version.



To add a version:
1Select Product Release Management > Define Product Tree in the tree panel.

2Select a product subfolder in the Product Tree. All product subfolders in the Product Tree are green. Every version is the child of a product.

3Click the New Version button. The New Version dialog box appears.

4Define a unique name for the product in the Version Name text box.

5Click the OK button. The New Version dialog box closes and a version folder is displayed as the child of the select product in the product tree.

To define a version:
1Select Product Release Management > Define Product Tree in the tree panel.

2Select a version subfolder in the Product Tree. All version subfolders in the Product Tree are gray.

3Define version properties in the Version Properties tab. A version is defined by its title, description, and release status.

4 Optional: To define version milestones, select the Select Applicable Dates button and select one or more milestone date types in the dialog box.

To delete a version:
1Select Product Release Management > Define Product Tree in the tree panel.

2Select a version subfolder in the product tree.Click the Delete button.A confirmation dialog box appears.

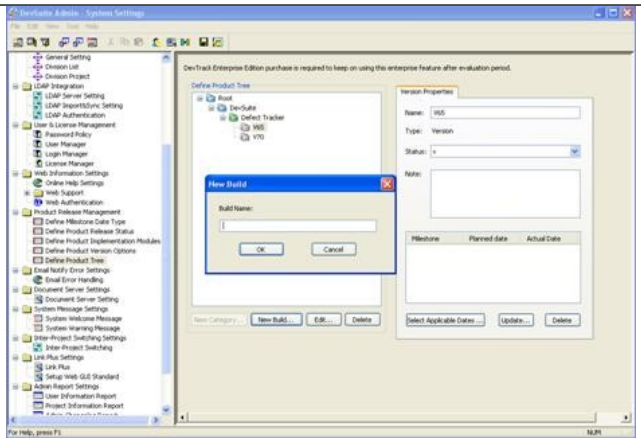
3Click the Yes button.

7.2.4 Administering Builds

Every build in the product tree represents a build of a version in development in a DevTrack project. Each build is the child of a single version.

Every build is defined by its parent version and its release status.A release status is an administrator-defined workflow state that identifies the progress of that deliverable within its development life cycle and determines the work that may be done on product deliverables in that state.

The scope of development activities in DevSpec and DevTrack projects may be defined by build.



To add a build:

1Select Product Release Management > Define Product Tree in the tree panel.

2Select a version subfolder in the Product Tree. All version subfolders in the Product Tree are gray. Every build is the child of a version.

3Click the New Build button. The New Build dialog box appears.

4Define a unique name for the product in the Build Name text box.

5Click the OK button.
The New Build dialog box closes and a build folder is displayed as the child of theselected version in the product tree.

To define a build:

1Select Product Release Management > Define Product Tree in the tree panel.

2Select a build subfolder in the Product Tree. All build subfolders in the Product Tree are gold.

3Define build properties in the Build Properties tab. A build is defined by its title, description, and release status.

To delete a build:

1Select Product Release Management > Define Product Tree in the tree panel.

2Select a build subfolder in the product tree.Click the Delete button.A confirmation dialog box appears.

3Click the Yes button.

7.3 Administering Product, Version, and Build Release Statuses

In DevSuite, the development of products, versions, and builds are defined by a unique lifecycle consisting of multiple*release statuses.*

A release status is an administrator-defined workflow state that identifies the progress of that release within its development life cycle. Administrator-defined workflow rules determine how a release item (product, version, or build) may be managed within each release state in its lifecycle.

The privileges available to project administrators and project members in each release depend on the release type: product, version, or build.

Release	Product Privileges	Client Privileges
Product	Add new versions	Add new issues
	Add new build	Add new issues
Version	Add new build	Add new issues
Build		Add new issues

Every release status is defined as an open or closed. Products, versions, and builds in open states (that is, release states with an open release status) are managed in workflow based on administrator-defined rules. Work items in closed states must be reopened to be managed in release management workflow.

Open Release Status: An open release status indicates that the product, version, or build is currently under development. Changes may be made to releases with an open release status.

Closed Release Status:A closed release status indicates that the product, version, or build is currently not under development. The product, version, or build may be completed or may still be in the future. Changes may not be made to releases with a closed release status.

Open project are definable, and the system administrator may enable certain privileges (at either administrative or client level) and restrict others. Closed releases cannot be changed at either level.

System administrators can define the release lifecycles of products, versions and builds using the tabs in the Define Product Release Status page.

7.3.1 Defining the Product Release Lifecycle

In DevSuite, a product is a representation of a development item. Every DevSuite product is the child of a product category and the parent to one or more versions.

The development of products is defined by the product lifecycle. The product lifecycle is a set of release states and workflow rules for managing products within each release state that enables the development organization to manage and track the development of that product in workflow.

Using controls in the Product Release Status page, the system administrator may define multiple product release states and define rules for administering and managing products within that state.

Figure 10-2: Product Release Status Page

The lifecycle of a product may consist of one or more product release states." Each release state defines a set of rules for managing products when they are in that particular state.

Each product release state is defined by three rules.

Add New Versions: The Add New Versions privilege enables project managers to create new child version subprojects under the product subproject.

Add New Builds: The Add New Builds privileges enables project managers to create new child build subprojects under the product subproject.

Add New Issues: The Add New Issues privilege enables project members to create new issues in a release subproject.

To add a product state:

1. Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

   The Release Status manager appears.

2. Click the Add button in the Product tab.

   The New Product Status dialog box appears.

3. Define the name of the product status in the Product Status Name control.

4. Click the OK button.

   The New Product Status dialog box closes.

5. Define Product Status properties.

6. Click the OK button.

To edit a product state:

1. Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

   The Release Status manager appears.

2. Select a product release status in the Product list of the Product tab.

3. Click the Edit button.

   The Edit Product Status dialog box appears.

4. Update the name of the product status in the Product Status Name control.

5. Click the OK button.

   The Edit Product Status dialog box closes.

6. Update Product Status properties in the Product Properties tab.

7. Click the OK button.

To delete a product state:

1. Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

   The Release Status manager appears.

2. Select a product release status in the Product list of the Product tab.

3 Click the Delete button.

A DevTrack dialog box appears.

4 Click the Yes button.

7.3.2 Administering the Version Release Lifecycle

In DevSuite, a version is a representation of a version of a product release. Every DevSuite version is the child of a product and the parent to one or more builds.

The development of product versions is defined by the version lifecycle. The version lifecycle is a set of release states and workflow rules for managing versions within each release state that enables the development organization to manage and track the development of that version in workflow.

The lifecycle of a product may consist of one or more product release states. Each release state defines a set of rules for managing products when they are in that particular state.

Add New Builds: The Add New Builds privileges enables project managers to create new child build subprojects under the version subproject.

Add New Issues: The Add New Issues privilege enables project members to create new issues in a version subproject.

To add a version state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Click the Add button in the Version tab.

The New Version Status dialog box appears.

3 Define the name of the version status in the Version Status Name control.

4 Click the OK button.

The New Version Status dialog box closes.

5 Define Version Status properties.

6 Click the OK button.

To define version status properties:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 To define the version status as an open or closed status, select an option in the Product Properties tab.

- Select the Open
- Select the Closed

3 To enable, select the Can Add New Build check box.

4 To enable, select the Can Add New Issues check box.

5 Click the OK button.

To edit a version state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a version release status in the Product list of the Product tab.

3 Click the Edit button.

The Edit Version Status dialog box appears.

4 Update the name of the product status in the Version Status Name control.

5 Click the OK button.

The Edit Version Status dialog box closes.

6 Update Product Status properties in the Version Properties tab.

7 Click the OK button.

To delete a version state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a version release status in the Version list of the Version tab.

3 Click the Delete button.

A DevTrack dialog box appears.

4 Click the Yes button.

7.3.3 Administering Build Release Lifecycle

In DevSuite, a build represents a specific build of a product version. Every build is the child of a version.

Every build is defined by its release status. The release status indicates the current status of the build in the DevTrack site. Every build is assigned either an open release status or a closed release status.

Open releases statuses are definable, and the system administrator may enable or disable the open status privilege: The Add New Issues privilege enables project members to create new issues in the version subproject.

To add a build state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Click the Add button in the Build tab.

The New Build Status dialog box appears.

3 Define the name of the build status in the Product Status Name control.

4 Click the OK button.

The New Build Status dialog box closes.

5 Define Build Status properties.

6 Click the OK button.

To define build release state properties:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Click the Add button in the Build tab.

3 To define the build status as an open or closed status, select an option in the Product Properties tab.

- Select the Open
- Select the Closed

4 To enable, select the Can Add New Test check box.

5 Click the OK button.

To edit a build state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a product release status in the Product list of the Product tab.

3 Click the Edit button.

The Edit Product Status dialog box appears.

4 Update the name of the product status in the Product Status Name control.

5 Click the OK button.

The Edit Product Status dialog box closes.

6 Update Product Status properties in the Product Properties tab.

7 Click the OK button.

To delete a build state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a product release status in the Product list of the Product tab.

3 Click the Delete button.

A DevTrack dialog box appears.

4 Click the Yes button.

7.4 Administering Product Implementation Modules

In DevSuite, a product implementation module is a tool for organizing a distinct area of development—such as a platform (Windows, Linux), an application (thick client, thin client, smart client), plug-in, tool, widget, or any other development item.

An implementation module may represent any category that is useful to the development organization and helps them to better manage and track work.

Implementation modules enable development organizations to control which areas of development may be worked on in DevSpec requirements management projects and DevTrack development projects.

- The scope of the specifications managed in a DevSpec requirements management project is defined by the implementation modules that are “applicable” to that project.
- The scope of development activities managed in a DevTrack project is defined by the implementation modules that are “applicable” to that project.

The product tree enables project administrators to define the scope of the development tasks managed in their projects—to manage a product release in specific project, an appropriate product, version, or implementation module must be defined “applicable” to the project.

To work on an implementation module in a project, the implementation module must be applicable to the products and versions managed in that project.

7.4.1 Adding Product Implementation Modules

To add a product implementation module:

1. Select Product Release Management > Define Product Implementation Modules in the tree panel.
2. Click the Add button.

   The Add a New Implementation Module dialog box appears.
3. Define the name of the implementation module in the Module Name text box.
4. Click the OK button.

   The implementation module is added to the Implementation Module list.

7.5 Administering Product Version Options

In DevSuite, a*version option*represents a set of requirements or specifications that may be implemented in a release of a product version. Each version may be defined by multiple version options.

Using controls in the Product Version Options page, system administrators may add edit and delete version options and identify which version options are applicable to each version.

Figure 10-3: Product Version Options Page

Version options are represented by gold folders in the product tree in the Product Version Options page."

7.5.1 Defining Version Options

To add a version option:

1. Select Product Release Management > Define Product Version in the tree panel.
2. Select a version folder in the product tree.

   Versions are represented by green folders.
3. Click the Select Applicable Option button.

The Applicable Option window appears.

4 Click the New button.

The New Option dialog box appears.

5 Define the title and description of the version option.

6 Click the OK button.

The version option is displayed in the Available Option list.

7.5.2 Applying Version Options to Versions

Every version option may be defined as applicable to multiple versions. Many different versions may share the same version options.

To add a version option:

1 Select Product Release Management > Define Product Version in the tree panel.

2 Select a version folder in the product tree.

Versions are represented by green folders.

3 Click the Select Applicable Option button.

The Applicable Option window appears.

4 Add or remove version options to the version.

- To apply a version option to a version, select a version option in the Available Options list and click the Right Arrow button.
- To remove a version option from a version, select a version option in the Selected Option list and click the Left Arrow button.

Chapter 8 Time and Cost Tracking Administration

8 Chapter 8- Time and Cost Tracking Administration

Time and cost tracking administration defines the project management tools that enable a business to measure and track project resources.

In this chapter:

- Managing Time and Cost Tracking
- Managing Issue-Level Time and Cost Tracking
- Managing Project-Level Time Tracking

8.1 Understanding Time and Cost Administration

DevTrack enables development organizations to track the time that project members spend working in projects as well as on individual development issues. This feature helps development organizations review the length of each issue lifecycle, and thus further analyze development and user trends.

Project-Level Time Tracking:Project-leveltime tracking enables development organizations to track the time that project members spend working in individual projects.

Issue-Level Time Tracking:Issue-level time tracking enables development organizations to track the time that project members spend working on individual development issues.

Time tracking is designed to be used with DevTrack's another innovate feature –issue cost tracking, which provides an easy way for development organizations to calculate the cost of the development efforts. Development organizations can also generate time/cost tracking reports for business purposes.

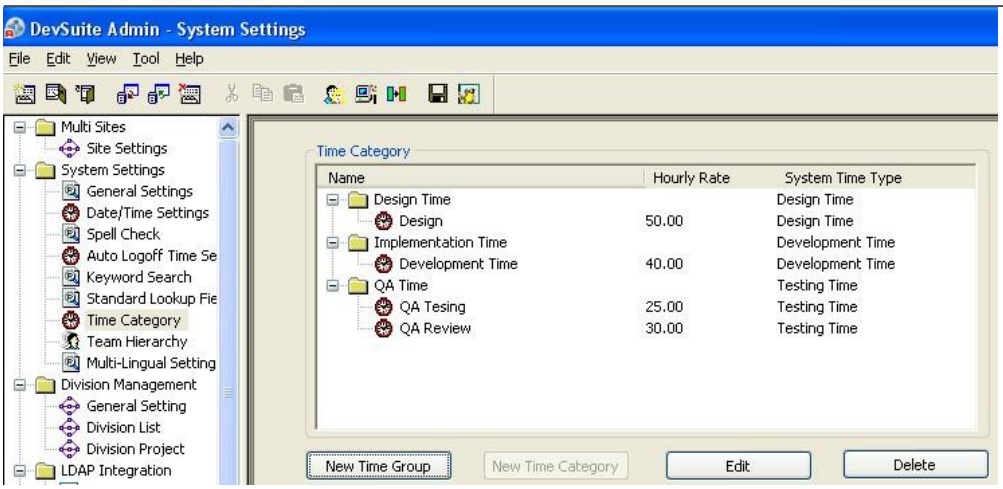
Issue-Level Cost Tracking: Issue-level cost tracking enables development organizations to calculate the cost of their development efforts.

Note:DevTrack issue-time tracking is not an automatic time-recording system. Instead, team members will be prompted with a dialog box when forwarding an issue (or under other scenarios defined by the project administrator). In the dialog box, members can manually input the hours they spend working on the issue.

8.2 Administering Time Groups and Time Categories

In the older version of DevTrack (prior to DevSuite 7), issue time/cost tracking is defined and managed at project level. Project administrators needed to create various work types to represent the work done by the team members in each of the project.

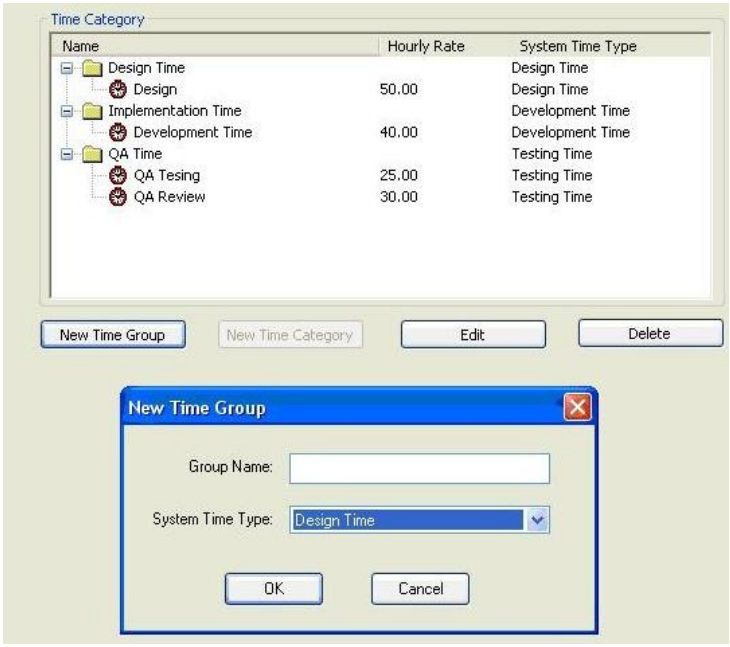
Starting in DevSuite 7, issue time/cost tracking has presented itself as a global concept----various time groups and time categories can be defined at system level and implemented flexibly at project level. All projects across the DevSuite site can share and use the time categories defined in the System Settings Projects.



8.2.1 Creating Time Groups

In DevSuite, a time group is defined by its title and its*time type*. Every time group belongs to one of four time types: Design Time time type, Development Time time type, Testing Time time type, and Other time type.

Using controls in the Time Category page, system administrators may create, edit or delete time groups.



To create a time group:

1Select Time Category Page under the System Settings folder.

2Click the New Time Group button.

The Add Time Group dialog box appears.

3 Define the name of the time group that best illustrates the time categories it is going to contain.

4 Select one of the system time types from the dropdown list that matches best with the to-be created time group. It can be one of the followings: Design Time time type, Development Time time type, Testing Time time type, and Other time type.

5 Click the OK button.

The Add Time Group dialog box closes and the time group is displayed in the Time Category tree as a folder that can contain various time categories.

8.2.2 Editing Time Groups

In DevSuite, a time group is defined by its title and its*time type*. Every time group0 belongs to one of four time types: Design Time time type, Development Time time type, Testing Time time type, and Other time type.

A time group may be the parent of multiple time categories.

Using controls in the Select Time Category window, project administrators may identify applicable time categories.

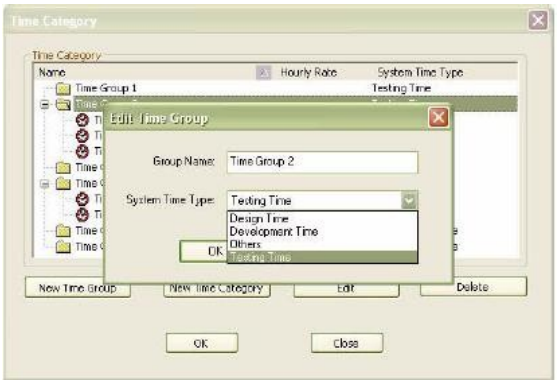


Figure 11-2:

8.2.3 Creating Time Categories

In DevSuite, a time category is defined by its title, hourly rate, and parent time group. Using controls in the Select Time Category window, project administrators may identify applicable time categories.

Figure 11-3:

To create a time categories:

1. Select Advanced Features > Time and Price Analysis.
2. Click the Select button in the Time Category area.

   The Select Time Category window appears.
3. Click the Setup button.

   The Time Category Manager appears.
4. Select a time group folder in the time group tree.
5. Click the New Time Category.

   The Add Time Category dialog box appears.
6. Define the name of the time category.
7. Define the default hourly rate.
8. Click the OK button.

   The Add Time Category dialog box closes and the time category is displayed in the time category tree as the child of the selected time group folder.

8.2.4 Editing Time Categories

Using controls in the Select Time Category window, project administrators may identify applicable time categories.

Figure 11-4:

8.3 Understanding Time and Cost Administration

DevTrack enables development organizations to track the time that project members spend working in projects and on individual development issues.

Issue-Level Time Tracking:Issue-level time tracking enables development organizations to track the time that project members spend working on individual development issues.

Issue-Level Cost Tracking: Issue-level cost tracking enables development organizations to calculate the cost of their development efforts.

Project-level and issue-level time tracking data may be viewed and analyzed in DevTrack reports.

8.4 Administering Time Groups and Time Categories

8.4.1 Creating Time Groups

In DevSuite, a time group is defined by its title and its*time type*. Every time group0 belongs to one of four time types: Design Time time type, Development Time time type, Testing Time time type, and Other time type.

Using controls in the Select Time Category window, project administrators may identify applicable time categories.

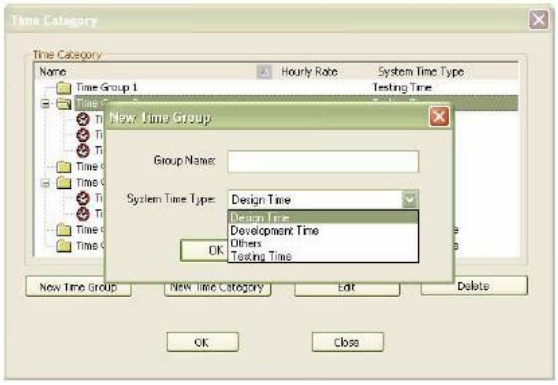


Figure 11-1:

To create a time group:

1. Select Advanced Features > Time and Price Analysis.
2. Click the Select button in the Time Category area.

   The Select Time Category window appears.
3. Click the Setup button.

   The Time Category Manager appears.
4. Select a time group folder in the time group tree.
5. Click the New Time Category.

   The Add Time Category dialog box appears.
6. Define the name of the time category.
7. Define the default hourly rate.

8 Click the OK button.

The Add Time Category dialog box closes and the time category is displayed in the time category tree as the child of the selected time group folder.

8.4.2 Editing Time Groups

In DevSuite, a time group is defined by its title and its*time type*. Every time group0 belongs to one of four time types: Design Time time type, Development Time time type, Testing Time time type, and Other time type.

A time group may be the parent of multiple time categories.

Using controls in the Select Time Category window, project administrators may identify applicable time categories.

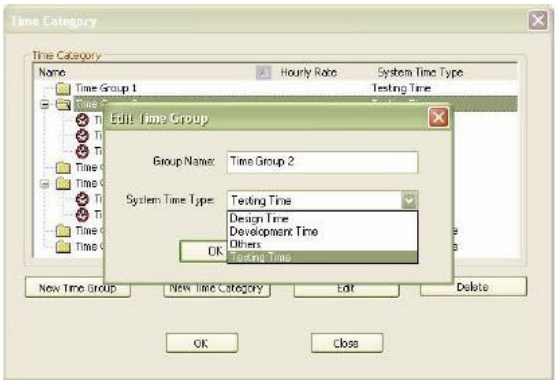


Figure 11-2:

8.4.3 Creating Time Categories

In DevSuite, a time category is defined by its title, hourly rate, and parent time group. Using controls in the Select Time Category window, project administrators may identify applicable time categories.

Figure 11-3:

To create a time categories:

1 Select Advanced Features > Time and Price Analysis.

2 Click the Select button in the Time Category area.

The Select Time Category window appears.

3 Click the Setup button.

The Time Category Manager appears.

4 Select a time group folder in the time group tree.

5 Click the New Time Category.

The Add Time Category dialog box appears.

6 Define the name of the time category.

7 Define the default hourly rate.

8 Click the OK button.

The Add Time Category dialog box closes and the time category is displayed in the time category tree as the child of the selected time group folder.

8.4.4 Editing Time Categories

Using controls in the Select Time Category window, project administrators may identify applicable time categories.

Figure 11-4:

KnowledgeWise Projects Administration

Chapter 1 What's new

Chapter 2 KnowledgeWise Concepts

2 Chapter 1- KnowledgeWise Concepts

This chapter provides a broad overview of DevSuite features and modules describing how these tools may be used to enhance application life cycle management processes.

In this chapter:

- Understanding TechExcel KnowledgeWise and DevSuite
- Understanding DevSuite Administration
- Understanding DevSuite Sample Projects and Templates

2.1 Understanding TechExcel KnowledgeWise and DevSuite

TechExcel DevSuite is a family of integrated application lifecycle management (ALM) tools that place knowledge management —from ideas, to formal specifications, to competitive information to issue resolution and customer insight—at the core of any product development initiative.

The DevSuite knowledge-centric strategy enables improve communication, keep up-to-date on changes, and reduce the development cycles so that the business may deliver the right products for the right markets in the shortest possible time.

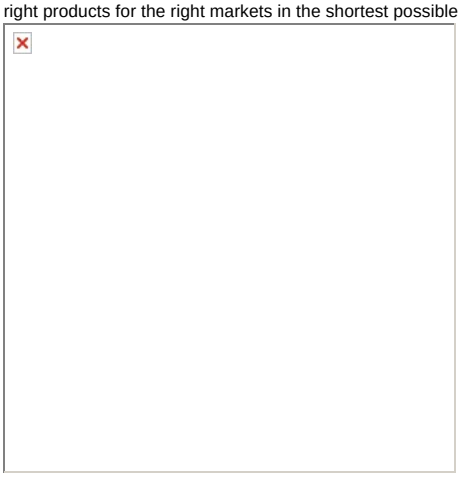


Figure 1-1: DevSuite Knowledge-Centric Development

DevSuite places knowledge management at the core of all development processes." TechExcel KnowledgeWise provides for the easy and efficient collection and organization of informal ideas, gathered from a wide variety of sources, that area shared across multiple DevSpec, DevTrack, and DevTest projects.

KnowledgeWise projects provide controlled access to documents, improve communication and coordination between distributed development teams, and facilitate the management and sharing of information between development teams and project stakeholders.

TechExcel DevSuite leverages intellectual assets with KnowledgeWise, communicating a clear product vision and tactical execution strategy by linking ideas and customer feedback, specifications, requirements, designs, prototypes, and other documents to specific areas of work.

2.1.1 Understanding DevSuite ALM Solutions

TechExcel DevSuite features five ALM solutions that operate in an n-tier architecture: TechExcel KnowledgeWise, TechExcel DevTrack, TechExcel DevPlan, TechExcel DevSpec, and TechExcel DevTest.

All DevSuite products share the same core architecture—the DevSuite Database Server, DevSuite Application Server, DevSuite Document Server, and DevSuite Web Services—and are fully integrated enabling every branch of the development organization to communicate and work together.

While each element is valuable even when functioning independently, the DevSuite architecture is founded upon the assumption of future scaling and is capable of advanced inter-module integration." Whether used as separate components or as an incorporated set of applications, the DevSuite solutions are ideal for companies at any stage in their product development.

2.1.2 Understanding TechExcel Project Hierarchy

At the core of every DevSuite implementation—a*DevSuite site*— is one or more KnowledgeWise knowledge management projects that organize, manage, and track the knowledge collected in multiple DevSpec, DevTrack, and DevTest projects.

All projects in a DevSuite site are organized into a hierarchical structure that defines the relationships between the projects in a DevSuite site. Every DevSpec or DevTrack project is the child of a parent KnowledgeWise knowledge management project and relies on that project to manage project intelligence.

Figure 1-2: DevSuite Project Hierarchy

KnowledgeWise enables development organizations to build a knowledge base of shared knowledge in many different media." All knowledge items may be linked with different areas of work so that internal teams and external customers can search the knowledge base for self-service content based on account type-based privileges and access controls.

KnowledgeWise projects may be accessed by three methods: (1) the KnowledgeWise Web client, (2) the knowledge view of a child DevSpec project; (2) the knowledge view of a child DevTrack project.

2.2 Understanding DevSuite Administration

In DevSuite, administrative tasks are managed at two levels: system administration and project administration.

Site Administration: DevSuite system administration is the process of configuring, administering, and maintaining a DevSuite site. System-level administration activities include maintaining the implementation, creating and defining new projects, managing user accounts, managing licenses, and configuring the DevSuite Server.

Project Administration: DevSuite project administration is the process of defining project-level settings and business objects, customizing project GUIs, configuring workflow rules, managing project team representation and access controls, and integrating add-on modules.

All DevSuite site and project administrative tasks are performed in the DevSuite Admin client. The DevSuite Admin client provides DevTrack, DevPlan, DevSpec, and KnowledgeWise system and project administrators with a single interface for defining, configuring, and administering TechExcel sites and projects.

All DevSuite site and system-level administrative tasks are performed in a System Settings project. Every DevSuite site consists of a single System Settings project. Settings defined in a system settings project apply to every work project in that site.

All project-level administrative tasks are defined on a project-by-project basis and are only applicable to a single work project. In DevSuite 1, project administrators may define four types of projects: KnowledgeWise projects, DevSpec projects, DevTrack projects, and DevTime projects.

Note: DevTest projects may be defined, configured, and administered in the DevTest Admin client. All other DevSuite projects are managed in the DevSuite Admin client.

2.2.1 Understanding DevSuite Site (System-Level) Administration

DevSuite system administration is the process of configuring, administering, and maintaining a DevSuite site. A site consists of a system settings project and one or more work projects. Every work project in a DevSuite site shares a common database, knowledge base, and other system-level configurations that are defined in the DevSuite Admin client.

In DevSuite, system administration is the process of defining system, user, and project settings that define work in a DevSuite site.

A DevSuite site is an implementation of a DevSuite server and may include multiple DevTrack, DevSpec, DevPlan, and DevSpec projects. System-level settings are settings that apply to every project in the DevSuite site. DevSuite system administration tasks fall into seven general areas:

System Administration: System-level settings administrative tasks include the configuration of the DevSuite Server.

User Administration: System-level user administrative tasks include the definition and management of user accounts and project administrators. Optional system-level administrative tasks include multiple site management, division management, and LDAP directory server integration and administration.

Multiple Site Administration: Multiple site administration includes the definition and management of a site family and management of user licenses for all member sites.

License Management: System-level license administrative tasks include license management and allocation.

Web Administration: Web administration includes the definition and configuration of DevTrack Web Server settings for every DevTrack project in the site.

E-mail Integration: E-mail integration and administration is the foundation of DevTrack, DevPlan, and DevSpec e-mail notification and escalation.

Branch Development: Release management administration defines the product structures that are shared across DevTrack, DevSpec, and DevPlan project providing a common framework for organizing and managing the development of products, versions, and builds.

Time And Cost Tracking Administration: Time and cost tracking administration defines the project management tools that enable a business to measure and track project resources.

System-level user administrative tasks include license management and allocation, user account creation and definition, and the management of project administrators. Optional system-level administrative tasks include multiple site management, division management, and LDAP directory server integration and administration.

2.2.2 Understanding Project Administration

DevSuite project administration is the process of defining project-level settings and business objects, customizing project GUIs, configuring workflow rules, managing project team representation and access controls, and integrating add-on modules.

Project administrators define what is tracked in the project, how it is tracked, and who may track the data. All project-level configurations are project-specific and apply to a single KnowledgeWise, DevSpec, DevTrack, or DevTime project.

Project administration tasks may be broken into five inter-related areas:

- Project Definition and Configuration
- Team Representation Administration
- Work Item Definition and GUI Customization
- Workflow Administration
- Add-on Module Integration

2.3 Understanding DevSuite Sample Projects and Templates

DevSuite builds on TechExcel's commitment to the rapid deployment of its ALM by providing development organizations with the tools they need to quickly configure and implement projects that enforce good development practices and drive the quality of releases.

To facilitate the configuration of a fully integrated site and projects that support your development processes, DevSuite provides developers with tools for evaluating and testing DevSuite features.

Sample Projects: A sample project is a project that provides development organizations with a sandbox for configuring and evaluating DevSuite features and integrations in a risk free environment.

Project Templates: A project template is a blueprint for creating a project of a specific type.

Solution Templates: A solution template is a preconfigured project that facilitates the definition of DevSpec projects that implement five proven development methodologies: CMMI, Scrum, SpecDD, TDD, and Iterative Development.

TechExcel strongly recommends that development organizations deploy and test DevSuite in a test environment prior to your DevSuite implementation. A test environment enables development organizations to test project configurations and to train users in new processes prior to the “go live” date in a production environment.

2.3.1 Understanding Sample Projects

In DevSuite, a sample project is a project that provides the development organization with a sandbox for configuring and evaluating features and integrations in a risk free environment. DevSuite features ten predefined projects that provide development organizations with a sandbox to experiment with KnowledgeWise, DevSpec, and DevTrack tools and features.

All sample projects include sample data, sample project team members, and fully configured workflow rules. Project administrators may freely configure sample projects to represent their development processes.

- Project configurations and settings tried and tested in sample projects may be imported into newly created “live projects”.
- Sample projects provide development organizations with a tool for training new users or to introduce existing users to new features or changes in businesses processes.

Using controls in the DevSuite Admin client, administrators may create additional sample KnowledgeWise, DevSpec, and DevTrack projects.

2.3.2 Understanding Project Templates

In DevSuite, a project template is a blueprint for creating a project of a specific type— KnowledgeWise knowledge management projects, DevSpec requirements management projects, or DevTrack development projects.

Each project template defines a complete set of project-level settings including the definition of all project business objects, workflow rules, team representation, and project integrations.

Project templates may be used to quickly configure new projects whenever they are created in a DevSuite site.

2.3.3 Understanding Solution Templates

TechExcel DevSpec, an integrated requirements management solution, features a complete library of preconfigured solution templates that facilitate the implementation of today's most popular and proven development methodologies.

Each solution template defines DevSuite project settings based on software development best practices so that development organizations may get up and running quickly. Once these business rules are in place, development teams may subsequently customize the project settings to suit their individual needs using TechExcel's award-winning project customization tools.

TechExcel provides complete documentation for implementing each methodology that describes how DevSuite workflow and automation may be used to implement and enforce best practices.

Chapter 3 Project Administration

3 CHAPTER 2 Project Administration

In this chapter:

- Understanding KnowledgeWise Project Administration
- Understanding the DevSuite Admin Workspace
- Administering KnowledgeWise Projects

3.1 Understanding KnowledgeWise Project Administration

In DevSuite, administrative tasks are managed at two levels: the system-level and the project-level.

This book describes project-level administrative tasks that apply to TechExcel DevTrack work projects.

In DevSuite, the configuration and customization of work projects is the responsibility of a project administrator. Project administrative tasks include the creation and definition of projects, the definition of project settings and workflow, and team management structures.

3.1.1 Understanding Projects

In DevSuite, a project is a tool for storing, organizing, and managing incident, event, employee, asset, knowledge, and project team data.

Every DevSuite implementation—called a DevSuite site—may consist of multiple types of work projects including DevTrack development projects, DevSpec requirements management projects, KnowledgeWise knowledge management projects, and DevTest QA management projects.

A work project is a tool for storing and managing work item and event data. The work items managed and tracked in a project are project-specific: development issues in DevTrack, test templates and test tasks in DevTest, and specifications, requirements, and change requests in DevTrack. Every work project represents a distinct area of work and is defined by unique team structures (account types, team groups, and group folders) and workflow rules.

3.2 Understanding the DevSuite Admin Workspace

In DevSuite, all system- and project-level administrative tasks are performed in the DevSuite Admin client.

When the System Settings project is opened, the DevSuite Admin client displays tools for configuring and administering an entire DevSuite site.

Figure 2-1: DevSuite Admin Client

The DevSuite Admin client organizes the work area into two bars and two panels." Each page is designed to enable an administrator to configure and administer system-level and project-level features.

The DevSuite Admin client is organized into four sections:

Menu Bar: The menu bar displays control buttons that enable the administrator to perform system-level and project-level administrative tasks. Menus include the File, Edit, View, Tool, and Help menus.

Tool Bar:The tool bar displays controls that enable administrators to.perform system-level and project-level administrative tasks.

Tree Panel: The Tree Panel organizes administration tools into a hierarchical structure consisting of folders and folders.

Page Panel: The Page Panel displays form-based administrative tools that enable the administrator to configure and maintain system-level settings.

3.2.1 Understanding Menu Bar Commands

The menu bar displays control buttons that enable the administrator to perform system-level and project-level administrative tasks. Menus include the File, Edit, View, Tool, and Help menus.

The majority of the command buttons displayed in the DevSuite Admin tool bar are project-specific commands. Of particular importance to system administrators are the commands displayed in the Tool menu.

File: The File menu displays commands that enable the administrator to create, open, back up, restore work projects.

Edit: The Edit menu displays commands that enable the administrator to manage text and GUI customization tasks.

View: The View menu displays commands that enable the administrator to display or hide the tool bar, status bar, and alignment bar in the Admin client.

Tool: The Tool menu displays commands that enable the administrator to perform a range of system-level administrative tasks including user, login, and license administration.

Help: The Help menu displays commands that enable the administrator to access online help systems and information about the current build

3.2.2 Understanding the Tool Bar Commands

The tool bar displays controls that enable administrators to.perform system-level and project-level administrative tasks.

The majority of the command buttons displayed in the DevSuite Admin tool bar are project-specific commands. Of particular importance to system administrators are the User Manager button, License Manager button, and the Reload Web Settings buttons.

New Project button	The New Project button enables the administrator to create a new work project.
Open Project button	The Open Project button enables the administrator to open an existing work project.
Close Project button	The Close Project button enables the administrator to close a work project.
Back Up Project button	The Back Up button enables the administrator to back up an existing work project.
Restore Project button	The Restore Project button enables the administrator to restore a closed work project.
Delete Project button	The Delete Project button enables the administrator to delete a work project.
User Manager button	The User Manager button enables the administrator to open the User Manager.
Login Manager button	The Login Manager button enables the administrator to open the Login Manager.
Reload Web Settings button	The Reload Web Settings button enables the administrator to stop and restart the DevSuite Web Server ensuring that updated configurations are available to web-based clients.
Save Project button	The Save Project button enables the administrator to save changes to project-level configurations.
Upload to Web Server	The Upload to Web Server button enables the administrator to upload system-level and project-level configurations and settings to the DevSuite Web Server. The DevSuite Web Server must stopped and restarted for administrative changes to be visible to web clients.

3.2.3 Logging into DevSuite Admin

To log into DevSuite Admin:

1Select the DevSuite Admin icon in the Windows Start menu.

By default, the DevSuite Admin icon is added to the DevSuite Admin subfolder within the TechExcel DevSuite folder. The DevSuite Admin Login dialog box appears

2Enter a user name and password.

Note: DevSuite Admin is delivered with a default system administrator defined (User Name: Admin, Password: (blank)). The password is case-sensitive, however the user name is not case-sensitive. For security reasons TechExcel recommends changing the default login setting as soon as possible

3To change web services, select an option from the Web Service dropdown list.

The DevSuite Admin client connects to the DevSuite Application Server through the DevSuite Web Service. Changing the web service enables the client to connect to a different DevSuite Application Server and manage a different DevSuite site.

4Click the OK button.

The DevSuite Admin client opens. By default, the client displays the Project Setup page. To administer a project, open or create a project using tools in the File menu.

3.2.4 Opening KnowledgeWise Projects

The Open Project dialog box displays the System Settings project as well as all live and sample DevTrack, DevSpec, DevPlan, DevTime and KnowledgeWise projects that are available in the DevSuite site.

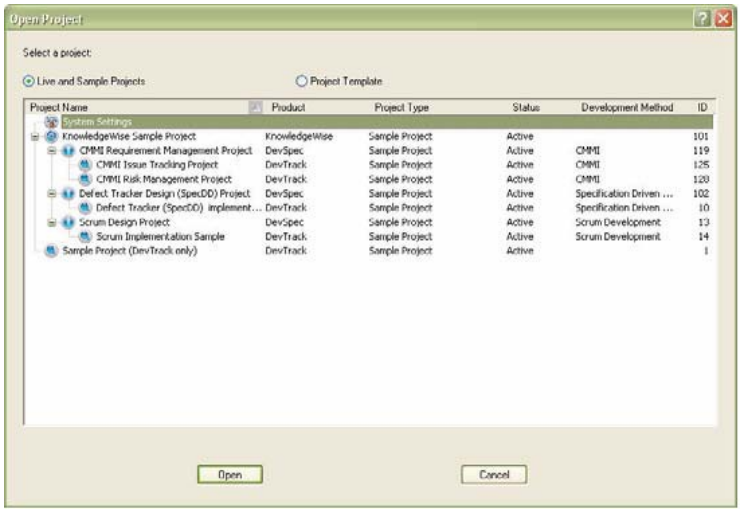


Figure 2-2: Open Project Dialog Box

For information on opening and administering KnowledgeWise, DevSpec, and DevTrack projects see the appropriate project administration guide.

To open a development project:

1 Select the Open Project command.

The DevSuite Admin client provides two options foropening projects:

- Select the Open project command in the File menu.
- Press CTRL + O.

The Open Project dialog box appears.

2 Select the DevTrack Project in the project list.

3 Click the OK button.

The System Settings project is displayed in the workspace.

3.2.5 Closing DevTrack Projects

Administrators may use the Close Project command to close DevTrack projects. The Close Project command may be invoked by two methods:

- Select File > Close Project in the menu bar.
- Press CTRL + S.

3.2.6 Understanding Keyboard Shortcuts

Keyboard shortcuts are combinations of keystrokes that perform predefined functions. Keyboard shortcuts consist of a modifier key and a hot key:

CTRL + X	Cut
CTRL + C	Copy
CTRL + V	Paste
CTRL + N	New Project
CTRL + O	Open Project
CTRL + S	Close Project
CTRL + B	Backup Project
CTRL + R	Restore Project
CTRL + D	Delete Project

3.3 Administering KnowledgeWise Projects

In DevSuite, a project is a tool for storing, organizing, and managing incident, event, employee, asset, knowledge, and project team data.

Every DevSuite implementation—called a*DevSuite site*—may consist of multiple types of work projects including development projects, timesheet projects, KnowledgeWise projects, and DevTest projects.

A work project is a tool for storing and managing work item and event data. The work items managed and tracked in a project are project-specific: development issues in DevTrack, test templates and test tasks in DevTest, and specifications, requirements, and change requests in DevSpec. Every work project represents a distinct area of work and is defined by unique team structures (account types, team groups, and group folders) and workflow rules.

Generally, the configuration and customization of work projects is the responsibility of a project administrator. However, many project administrative tasks are the responsibility of the system administrator. System administrators may create, delete, open, close, back up, and restore work projects and work projects in the General System Settings folder of the DevSuite Admin client.

3.3.1 Creating Knowledge Management Projects

The New Project dialog box enables administrators to define the project title, the project type, and to import project settings from previously-defined projects.

Using controls in the New Project Manager, project Administrators may create new projects, sample projects, and project templates.

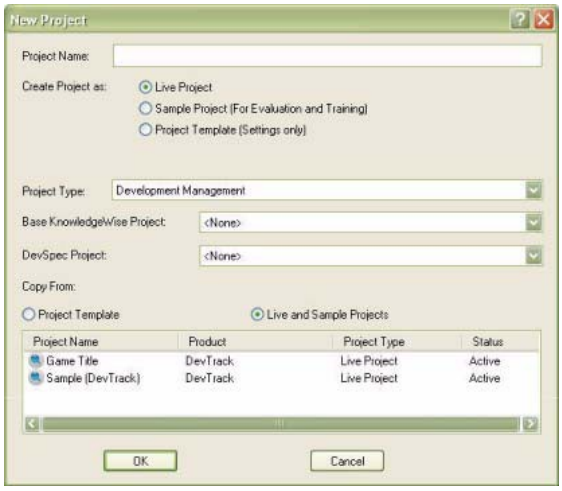


Figure 2-3: Open Project Manager

A knowledge base is a KnowledgeWise knowledge management project that is shared by multiple DevSuite projects. Every project is defined by one knowledge base—once defined, the knowledge base cannot be changed.

New projects may be created as live projects, sample projects, or project templates.

Live Project:A live project may be used to manage and track knowledge for multiple live DevSpec requirements management projects and live DevTrack development projects.

Sample Project:A sample project provides an organization with a sandbox for evaluating and testing project configurations.

Project Template:A project template is a blueprint for creating a specific type of project that defines project-level business rules, workflow, and team representation configurations. Project templates may be used to quickly define new projects.

When creating KnowledgeWise projects administrators may create projects from scratch or they may import project properties from previously defined KnowledgeWise projects. For information on importing project properties from other KnowledgeWise projects see “Copying KnowledgeWise Project Settings”.

To create a knowledge management project:

1Invoke the New Project command.

The New Project command may be invoked by three methods:

- Select the New Project command in the File menu.
- Press CTRL + N.
- Click the New Project button in the tool bar.

The New Project manager appears.

2Select a project creation option in the Create Project As area:

- To create a live project, select the Live Project option button.
- To create a sample project, select the Sample Project (For Evaluation and Training) option button.
- To create a project template, select the Project Template (Settings Only) option button.

3Enter a project title in the Project Name text box.

Every project is identified by an administrator-defined name. The project name enables administrators to distinguish between projects of the same project type. Do not use the following characters in the name of the project:

/ \ : ? “ < > |

4Select the KnowledgeWise Project project type in the Project Type dropdown list.

The Project Type dropdown list displays four options: Development Management (DevTest), Time Sheet Management, KnowledgeWise Project, DevSpec Project.

5 Optional: To copy project settings from a project template or an existing live or same project select an option in the Copy Settings From area.

- To copy project settings from an existing project template, select the Project Template option button and select a project template in the project list.
- To copy project settings from a live or sample project, select the Live or Sample Project option button and select a project template in the project list.

6 Click the OK button. The new project is created.

3.3.2 Opening KnowledgeWise Projects

To open a KnowledgeWise project:

1 Invoke the Open Project command."

The Open Project command may be invoked by three methods:

- Select the Open Project command in the File menu.
- Press CTRL + O.
- Click the Open Project button in the tool bar.

The Open Project dialog box appears.

2 Select a KnowledgeWise project in the Project Name list.

The Project Name list displays all active and inactive KnowledgeWise, DevTrack, DevPlan, and DevTime projects in the TechExcel DevSuite site.

3 Click the OK button.

The project opens.

4 Select a project type in the Project Type dropdown list.

The Project Type dropdown list displays four options:

- Development Management
- Timesheet Management
- KnowledgeWise Project
- KnowledgeWise Project

5 Optional: Select a project knowledge base in the Base KnowledgeWise Project dropdown list.

Every KnowledgeWise development project and KnowledgeWise project is the child of a single KnowledgeWise project.

6 Optional: Select a KnowledgeWise project in the KnowledgeWise Project dropdown list.
Every KnowledgeWise development project is the child of a single KnowledgeWise project.

7 Optional: Copy project settings from an existing project or project template.

- To copy project settings from a project template, select the Project Template option button and select the project template in the project list.
- To copy project settings from an existing live or sample project of the same project type, select the Live or Sample Project option button and select the project in the project list.

8 Click the OK button.

3.3.3 Backing Up KnowledgeWise Projects

Administrators may use the Backup Project command and the Back Up Project dialog box to back up individual KnowledgeWise projects or the entire KnowledgeWise database.

KnowledgeWise stores backups of KnowledgeWise projects in a proprietary DTK file format. Once the administrator define the project information, the administrator may back up either the entire database or only selected projects. KnowledgeWise Admin creates the backup file with the selected project data.

KnowledgeWise Admin stores backed up projects in a proprietary DTK file format.

Note: KnowledgeWise backup feature is a good way to save project properties, but cannot be considered a substitute for regular database server backups.

To backup KnowledgeWise projects:

1 Invoke the Back Up Project command.

- Select File > Back Up Project in the menu bar.
- Press CTRL + B.

The Back Up Project dialog box appears.

2 Click the Ellipsis button to navigate to the location of the backup file.

The Backup dialog box opens.

3 Navigate to the location of the backup files.

4 Enter a name for the DTK file in the File Name field.

5 Select the KnowledgeWise Backup File (DTK) option in the Save as Type dropdown list.

6 Click the Save button.

The Backup dialog box closes.

7 Select a back up option:

- Click the Whole Database radio button to backup all projects in a KnowledgeWise implementation.
- Click the Selected Projects radio button to backup some, but not all projects in a KnowledgeWise implementation. Highlight the projects to be backed up.

8 Click the OK button.

The project is backed up.

3.3.4 Restoring KnowledgeWise Projects

Administrators may use the Restore Project command to restore backed up projects.

KnowledgeWise stores backups of KnowledgeWise projects in a proprietary DTK file format. Administrators may choose to back up selected projects or the entire database.

When restoring backed up KnowledgeWise projects administrators may overwrite existing projects or create new projects based on the backed up projects.

To restore KnowledgeWise projects:

1 Invoke the Restore command.

- Select File > Restore in the menu bar.
- Press CTRL + R.

The Restore Project dialog box appears.

2 Locate the backed up project.

KnowledgeWise stores backups of KnowledgeWise projects in a proprietary DTK file format.

3 Click the Open button.

The KnowledgeWise Restore II dialog box appears. The dialog box displays the Project ID number of the project to be restored.

4 Select the Overwrite radio button in the Options area.

5 Click the OK button.

The project is restored. If the DTK file selected contains multiple backed up, the KnowledgeWise Restore II dialog box appears for each backed up project.

3.3.5 Creating Projects Based from Backed Up Projects

Administrators may use the Restore command to create new KnowledgeWise projects from backed up copies of existing projects.

Whenever an administrator backs up a KnowledgeWise project, KnowledgeWise saves a copy of that project in a proprietary DTK file format.

Administrators may create a new KnowledgeWise project from DTK files.

To create a new project from a backed up project:

1 Invoke the Restore command.

- Select File > Restore in the menu bar.
- Press CTRL + R.

The Restore Project dialog box appears.

2 Locate the backed up project.

KnowledgeWise stores backups of KnowledgeWise projects in a proprietary DTK file format.

3 Click the Open button.

The KnowledgeWise Restore II dialog box appears.

4 Select the Create New project radio button in the Options area.

5 Enter a title in the New Project Name field.

6 Click the OK button.

The new project is created and the Open Project dialog box appears.

3.3.6 Copying KnowledgeWise Project Settings

Project administrators may use the New Project dialog box to create new KnowledgeWise projects based on previously-defined KnowledgeWise projects.

The Copy Settings from an Existing Project command enables administrators to import project properties into the new project. Copying KnowledgeWise project settings cuts down the time required to configure a KnowledgeWise project if the different development projects share similar information.

Project administrators may copy any combination of the following settings to the new project:

- Project members, groups and group folders
- Workflow settings
- Time Tracking settings
- Search and E-mail queries
- Incident routing (autorouting)
- Incident escalation (autoescalation)
- Knowledge view settings
- Event workflow settings
- Interproject settings
- Perforce integration settings
- Issue Template settings

To create KnowledgeWise projects based on existing projects:

1 Select File > New Project.

2 The New Project dialog box appears.

3 Enter a project title in the Project Name field.

Do not use the following characters in the name of the project:

/ \ : ? " < |

4 Select a project type from the project dropdown list.

5 Select the Copy Settings from an Existing Project command.

A list of all existing projects is displayed in the window, and you may select the project with the settings you want to import.

6 Select a project in the Project list.

7 Click the OK button.

3.3.7 Understanding Active and Inactive Projects

Every KnowledgeWise project has either an active or inactive status. The status of a project determines the tasks that an administrator may perform on that project.

Administrators define the status of a project by selecting the Active Project check box in the Overview page.

3.3.7.1 Active Projects

Administrators may perform the following tasks using active KnowledgeWise projects: • Administrators may open active projects in KnowledgeWise Admin and configure project settings.

- Administrators may use the User Manager to add user accounts as project members to active projects.
- Administrators may import project properties from Active projects.

3.3.7.2 Inactive Projects

Administrators may perform the following tasks using inactive KnowledgeWise projects:

- Inactive KnowledgeWise projects may be deleted using the Delete Project command. Active projects cannot be deleted.

3.3.8 Deleting KnowledgeWise Projects

Project administrators may use the Delete command to delete inactive KnowledgeWise projects.

Administrators may only delete projects that have an inactive status. For instructions on making a project inactive see “Defining Projects as Active and Inactive”.

Note: Deleting a project from the system is permanent. Administrators must perform all appropriate back ups before deleting a project.

To delete a project:

1 Select File > Delete.

2 The Delete Project dialog box appears.

The Delete Project dialog box displays all of the projects that may be deleted. Only those projects that have an inactive status may be deleted.

3 Select a project.

4 Click the Delete button.

3.3.9 Defining Project Names and Descriptions

Every project is identified by an administrator-defined name. The project name enables administrators to distinguish between projects of the same project type. The Description property enables administrators to define a general description of the current project.

To define KnowledgeWise project name and descriptions:

1. Select the Overview page in the Project Settings folder.

   The Project Settings page appears.

2. Input a name in the Name text box.

   The roject Name property defines the name of the project and is used to identify the project in the Open Project dialog box.

3. Input a brief description in the Description multiple-line text box.

3.3.10 Defining Knowledge Item ID Prefixes

A requirement ID prefix is an administrator-defined text string that is automatically concatenated to the front of each knowledge item ID number. All knowledge item ID prefixes are defined on a project-by-project basis.

If the text string SMB- is defined as the knowledge item ID prefix in a KnowledgeWise project, every requirement created in that project is identified by an ID number that begins with that text string. For example,SMB-100, SMB-101, SMB-102 and so on.

knowledge item ID prefixes enable project teams to quickly identify and sort requirements based on the project that was used to create them.

3.3.11 Defining Projects as Active and Inactive

In TechExcel DevSuite, every project is defined by its activity status. KnowledgeWise projects may be active projects or inactive projects.

- Active projects may opened and managed in the DevSuite Admin or KnowledgeWise Windows clients.
- Inactive projects may be deleted.

To define the status of a KnowledgeWise project:

1. Open a KnowledgeWise project.
2. Select the Overview page in the Project Settings folder of the tree panel.
3. Define the activity status of the project:
   - To define the project as inactive, deselect the Active Project check box.
   - To define the project as active, select the Active Project check box.

3.3.12 Defining Name Formatting

Name formatting enables project administrators to determine how the names of project members are displayed in the KnowledgeWise clients.

The names of project members may be displayed using two formats:[First Name] [Last Name] and [Last Name], [First Name].

To define the status of a KnowledgeWise project:

1. Select the Overview page in the Project Settings folder of the tree panel.
2. Select the format used to display names in the project:
   - To display project members[First Name] [Last Name], select the First_Name Last_Name option button. For example,*John Smith.*
   - To display project members [Last Name], [First Name], select the Last_Name, First_Name option button. For example,*Smith, John.*

3.3.13 Defining Project Time Zone Settings

To define project time zone settings:

1. Click the Time/Date Format button in the Overview page of the Project Settings folder.

   The Select System Time/Date manager appears.

2. Select a time zone option from the System Time Zone dropdown list.

   The System Time Zone dropdown list displays 24 time zones.

3. Define time zone rules in the Time Zone area.

The time zones that is used to tag project data in the KnowledgeWise clients may be defined at the system-level, the project-level

Chapter 4 Team Administration

4 CHAPTER 3 Team Administration

In this chapter:

- Understanding Team Administration
- Administering Account Types, Access Controls, and Privileges
- Administering Project Team Members
- Administering Team Groups
- Administering Group Folders

4.1 Understanding Team Administration

The KnowledgeWise clients provide project team members with*controlled*access to all development, facilitate communication and workflow, and improve the productivity and effectiveness of the organization.

All KnowledgeWise development activities are controlled by account typed-based privileges, workflow rules, and access controls. The ability of a KnowledgeWise user account to access projects, views, folders, and pages depends on the “role” (account type) assigned to that user in the project.

The account type concept provides administrators with the flexibility to customize project workflow and security to support their unique business requirements. KnowledgeWise team representation structures enable development organizations to define sophisticated and flexible workflow rules for managing project data and security.

Project-level team administration is the process of defining team management structures (account types, team groups, and group folders), adding user accounts to the project, and assigning appropriate role and responsibilities to each project member.

- An account type is a set of access rights, privileges, and responsibilities that define the role that a user account plays in a project. Every user account is assigned one account type in each project.
- A team group is a collection of project team members that do not necessarily share the same account type but who share common responsibilities in the project.
- A group folder is a mechanism for managing the distribution of knowledge items in a project.

4.2 Administering Account Types, Access Controls, and Privileges

In KnowledgeWise, privileges and access rights are not granted to user accounts directly, but assigned to administrator-defined*account types*.

Users are assigned an account type when they join a project and assume the rights and privileges granted to that account type—when a user’s responsibility changes, he or she may be assigned a different account type.

An account type defines the role assigned to a user account in a project and determines the projects, views, data, and reports that are available to a project team team member through the KnowledgeWise clients.

In each project, the project administrator is responsible for creating all account types, granting privileges and access rights to those account types, and assigning these account types to each project members based on the role and responsibilities that user performs in the project.

4.2.1 Defining Account Types

KnowledgeWise team representation provides development organizations with maximum flexibility to manage work items in workflow, enforce security, and ensure accountability.

All account types are project-specific and defined by the project administrator. In KnowledgeWise, organizations are free to define the account types that best represent their business processes.

For example, a typical development project might feature five different account types: the Manager account type, the System Architect account type, the Programmer account type, the Tech Support account type, and the QA account type.

Figure 3-1: Typical KnowledgeWise Project Account Types

To create an account type:

1 Select Project Settings > Account Type in the tree panel."

2 Click the New button in the Account Types area of the Account Type page.

The New Account Type dialog box appears.

3 Enter a name in the Account Type Name text box.

4 Optional: To copy privilege settings from another account type:

- Select the Copy Account Type Setting From check box.
- Select an account type from the dropdown list.

5 Click the OK button.

To rename an account type:

To rename an account type, select the account type in the Existing Accounts list and click the Edit button.

To delete an account type.:

To delete an account type, select the account type in the Existing Accounts list and click the Delete button.

4.2.2 Administering Knowledge Management Privileges

In KnowledgeWise, a privilege is the right to perform a specific task such as creating, editing, or deleting a work items such as a report, event, or knowledge item. In KnowledgeWise, privileges and access rights are not granted to user accounts directly, but assigned to administrator-defined*account types*.

Users are assigned an account type when they join a project and assume the rights and privileges granted to that account type—when a user's responsibility changes, he or she may be assigned a different account type.

Using controls in the Privileges tab of the Account Type page, project members may grant knowledge item management privileges to project members of each account type in a KnowledgeWise project.

Can access view:The Can Access View privilege enables the project member to access the knowledge view.

Can manage all knowledge folder:The Can Manage All Knowledge Folder privilege enables the project member to create, edit, and delete knowledge folders.

Can setup access control:The Can Setup Access Control enables the project member to define knowledge folder access controls.

Can unlock documents checked out by other users:The Can Unlock Documents Checked Out By Other Users enables the project member to unlock knowledge items checked out by other project team members.

Can perform group change:The Can Perform Group Change privilege enables the project member to edit multiple knowledge items simultaneously using a group action.

Can define public query: The Can Define Public Query privilege enables the project member to create a public query.

Can manually create knowledge version: The Can Manually Create Knowledge Version enables the project member to manually create a version of a knowledge item.

Can delete knowledge version: The Can Delete Knowledge Version enables the project member to delete a version of a knowledge item.

Can perform rollback: The Can Perform Rollback enables the project member to rollback a knowledge item to a previous version.

Can create default value knowledge: The Can Create Default Value Knowledge enables the project member to define default value knowledge.

Can import knowledge (including knowledge folder): The Can Import Knowledge (Including Knowledge Folder) enables the project member to import knowledge items or knowledge folders.

Can access user preference: The Can Access User Preference enables the project member to personalize the KnowledgeWise client workspace.

Can push user preference settings to other user: The Can Push User Preference Settings To Other User enables the project member to define personalization settings for other project team members.

4.2.3 Administering Invisible Pages and Fields

Invisible page and field access controls enable project administrators to control which project team members may view project data based on their account type.

Pages or data-entry controls that track confidential data may be displayed to project members of one account type—for example, management—and hidden from other account types.

Using controls in the Account Type page, project administrators may define selected fields (data-entry controls) or entire pages as invisible.

Figure 3-2: Invisible Tab in Account Type Page (Detail)

Invisible page and field access controls are account type-based." Project administrators may also define workflow-based invisible field controls in which individual data-entry controls are hidden in particular workflow states.

To define invisible pages and fields:

1 Select Project Settings > Account Type page in the tree panel.

2 Select an account type in the Account Types list. Invisible pages and fields may be uniquely defined for each account type.

3 Select the Invisible tab. The Invisible tab is only displayed in the Account Type page if invisible access controls are enabled in the project.

4 Select a page or field in the Invisible tab. The Invisible tab displays all pages and fields in a hierarchical tree structure.

- An invisible field is a data-entry control that cannot be viewed or edited in the client.Data-entry controls may be defined as invisible fields based on account type-based access controls or state-based workflow rules.
- An invisible page is a system or custom page that cannot be viewed or edited in the client. Any system or custom page except the Description page may be defined as invisible to project account types. Administrators may, however, make all the individual fields in the Description page invisible.

4.2.4 Administering Read-Only Pages and Fields

Read-only page and field access controls enable project administrators to control which project team members may update project data based on their account type.

Pages or data-entry controls that track confidential data may be editable to project members of one account type—for example, management—and read-only to project members belonging to other account types.

Using controls in the Account Type page, project administrators may define selected fields (data-entry controls) or entire pages as read-only.

Figure 3-3: Read-Only Tab in Account Type Page (Detail)

Read-only page and field access controls are account type-based." Project administrators may also define workflow-based read-only field controls in which individual data-entry controls cannot be edited in particular workflow states. Also, all custom data-entry controls may be designated as read-only in the client.

Note: The invisible access control overrides read-only access control. If a page or field has been defined as invisible to an account type, administrators may not define the page or field as read-only.

To define read-only pages and fields:

1 Select Project Settings > Account Type page in the tree panel.

2 Select an account type in the Account Types list. Read-only pages and fields may be uniquely defined for each account type.

3 Select the Read-Only tab. The Read-Only tab is only displayed in the Account Type page if read-only access controls are enabled in the project.

4 Select a page or field in the Read-only tab. The Read-Only tab displays all pages and fields in a hierarchical tree structure.

- A read-only field is a data-entry control that may be viewed, but not edited in the client. Data-entry controls may be defined as read-only fields based on account type-based access controls or state-based workflow rules.
- A read-only page is a system or custom page that cannot be viewed or edited in the client.

4.2.5 Administering Report Access Controls and Privileges

KnowledgeWise uses access controls and privilege controls to manage workflow and implement security in projects. An access control whether a project member may view or edit a page or data-entry control in the client.

Report access controls define which project team members may access reports in the report view of the KnowledgeWise clients. Access to each report type may be granted to members

Using controls in the Report Access tab of the Account Type page, project administrators may define which reports project members may access based on their account type. Distinct access controls may be defined for each report in the KnowledgeWise Windows client and the KnowledgeWise Web client.

Can Access Knowledge Report View: The Can Access Knowledge Report View privilege enables the user to access the knowledge report view.

Can Create Public Knowledge Report: The Can Create Public Knowledge Report privilege enables the user to create public knowledge reports

Can Create Private Knowledge Report: The Can Create Private Knowledge Report privilege enables the user to create private knowledge reports.

Can Delete Public Knowledge Report:The Can Delete Public Knowledge Report privilege enables the user to delete public knowledge reports

Can Enable Web Query Report: The Can Enable Web Query Report privilege enables the user to create web query reports.

4.3 Administering Project Team Members

In DevSuite, the administration of user accounts and project members is divided between system administrators and project administrators.

- System administrators are responsible for creating user accounts, defining user account properties, and assigning licenses to DevSuite users.
- Project administrators may add user accounts to their projects and assign an account type to that user account.

In KnowledgeWise, all development and project management tasks are managed and tracked by a project team consisting of multiple*project team members.*A project team member is a licensed DevSuite user that has been assigned an account type.

A DevSuite user account cannot access a KnowledgeWise project unless they have been assigned a KnowledgeWise (or DevSuite) license by the system administrator and a project account type by the KnowledgeWise project administrator.

Using controls in the Project Member page, project members may add user accounts to KnowledgeWise projects, assign account types to user accounts, and view project member information.

4.3.1 Adding User Accounts to Projects

To add user accounts to a project:

1 Select Project Settings > Project Member in the tree panel.

2 Select a user from the Available Members list. Administrators may use the SHIFT and CTRL keys to select multiple users accounts.

3 Click the Add button. The Account Type dialog box appears.

4 Select an account type option from the Account Type dropdown list.

5 Click the OK button. The user account appears in the Members in Project list. Inactive user accounts are displayed with the word Inactive to show they are inactive.

4.3.2 Removing User Accounts from Projects

To remove a user account from a project:

1 Select Project Settings > Project Member in the tree panel.

2 Select a user from the Members in Project list.

3 Click the Remove button. The user account disappears from the Members in Project list and appears in the Available Members list.

4.4 Administering Team Groups

In KnowledgeWise, a team group is an administrator-defined collection of project members—who do not necessarily share the same account type, but who share common responsibilities in a project.

Team groups organize the project team into smaller, more useful entities that facilitate the assignment and control of knowledge.

Team groups are fully customizable. Project administrators create team groups to represent teams of project members that share the same manager, are working on the same product or component, share the same shift, work from the same office, or any other grouping that makes sense to the organization.

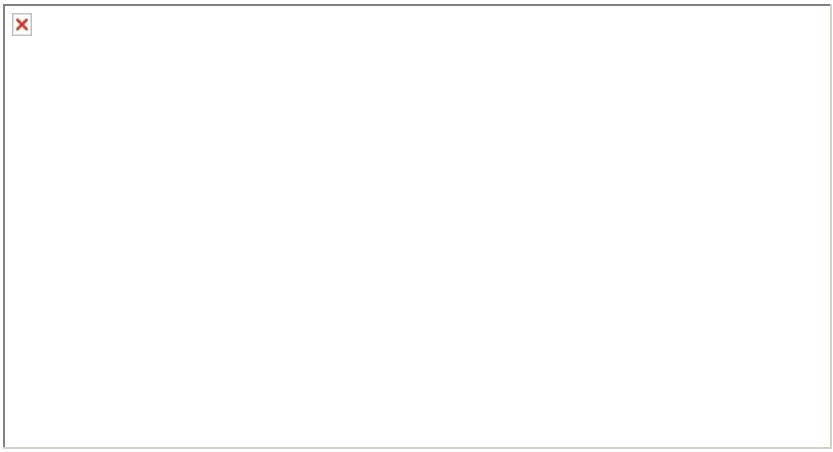


Figure 3-4: Project Members and Groups

Team groups cannot “own” knowledge items or knowledge folders." But project administrators may use team groups to determine which project team members may own knowledge items in particular workflow states.

Team groups are particularly useful to project administrators when they define workflow and routing rules, privileges, and access controls.

Workflow:Team groups may be defined as an*applicable owner*of items in each workflow state. Only those project members that are members of an applicable team group may be assigned knowledge items in that state.

Secured Access Control: Secured access controls may be account type-based, user account-based, or team group-based.

In the KnowledgeWise client, knowledge items may be filtered in the knowledge item list panel by team group. Moreover, report authors may run reports that return project data grouped by team group.

4.4.1 Defining Team Groups

Using controls in the Team Groups page, project administrators may create any number of team groups and assign any number of project team members to each team group. A project member may belong to multiple team groups.

To create a team group:

1 Select Project Settings > Team Groups in the tree panel. The Team Groups page appears.

2 Click The New button. The New Team Group dialog box appears.

3 Enter a name in the In The Group Name field.

4 Click the OK button. The New Team Group dialog box closes and the group appears in the Existing Groups list of the Groups page.

5 Select the group in the in Existing Groups list.

6 Select project members in the Groups Member Selection area. A check mark appears next to the name of the project member when they have been added to the group.

4.4.2 Adding and Removing Project Members to Team Groups

To add project members to a team group:

1 Select Project Settings > Team Groups in the tree panel. The Groups page appears.

2 Select a team group in the Existing Groups list.

3 Add or remove project team members to the team group.

- To add a project member, select the check box next the user's name in the Group Member Selection list.
- To remove a project member, deselect the check box next the user's name in the Group Member Selection list. A check mark appears next to the members name when they have been added to the group.

4.5 Administering Group Folders

Group folders are an optional KnowledgeWise feature that enables development organizations to more effectively implement issue-level security and manage the distribution of the work.

Any work item that may be assigned to a project team member may also be assigned to a group folder. For example, a group folders may be defined as an*applicable owner*of knowledge items in a specific workflow state enabling project members to "forward" knowledge items to that group folder as if it were any other project member.

Used in this way, group folders serve as a "buffer" between team members and their workload. Knowledge items may be temporarily stored in a group folder before they are redistributed to the appropriate project members based on project member availability, workload, expertise, or any other criteria.

Access to the knowledge items "owned" by a group folder is controlled by team group and account type-based*group folder access controls*.

- Every group folder is owned by a single team group. Access to the group folder is limited to members of the owning team group.
- Group folder access controls defined which project members may view or edit the knowledge items contained within the group folder. To edit knowledge items a team member must belong to an account type that has been granted the appropriate access rights.

Group folder administrative tasks include the creation and definition of one or more group folders and the definition of group folder access controls on a folder-by-folder basis.

4.5.1 Defining Group Folders

A group folder is a mechanism for storing a pool of common knowledge items that serves as a "buffer" between project members and their workload.

Every group folder is defined by its name, an owning team group, and access controls. The ability to view and edit knowledge items "owned" by a group folder is controlled by team membership and account type-based access controls.

Using controls in the Group Folder page, project administrators may create group folders, define group folder membership, and group folder access controls.

Figure 3-5: Group Folder Page

In KnowledgeWise, every group folder is "owned" by an administrator-defined team group or by the{All Members} user variable." Access to the group folder is limited to members of the owning team group unless the folder is owned by the {All Members} user variable.

Account type-based access controls enforce folder-level security. To edit the knowledge items assigned to a group folder, the user must be long to an account type that has been granted the Edit access rights for that folder.

Any number of group folders may be created in a project.

To create a group folder:

1 Select Project Settings > Group Folder in the tree panel. The Group Folder page appears.

2 Click the New button. The New Group Folder dialog box appears.

3 Define a unique name and description of the group folder. The name of the group folder is displayed in the Team Member dropdown list of the KnowledgeWise clients.

4 Select a team group from the Owner Group dropdown list. The Owner Group dropdown list displays all team groups defined in the project as well as the All Members user variable.

5 Click the OK button. The New Group Folder dialog box closes and the group folder is displayed in the Group Folder list.

6 To grant folder access rights to an account type, select the View, Put, and Get controls in the Folder Access Control list. A red check mark indicates that the access right is granted to all project members belonging to that account type.

To edit a group folder:

To edit a group folder, select a group folder in the group folder list and click the Edit button. The name and owning team group may be redefined.

To delete a group folder:

To delete a group folder, select a group folder in the group folder list and click the Delete button.

Chapter 5 Knowledge Folder Administration

5 CHAPTER 4 Knowledge Folder Administration

In this chapter:

- Administering Knowledge Item Folders
- Administering Knowledge Folder Function Pages
- Administering Knowledge Folder System Pages
- Administering Knowledge Folder System Fields

5.1 Administering Knowledge Item Folders

In the KnowledgeWise client, knowledge items are managed in a hierarchical tree structure—the knowledge item tree—that organizes product knowledge items in the workspace, implements security, and enforces good development practices.

Every area of development within the knowledge item tree framework is represented by a knowledge item folder. A knowledge item folder is a discreet area of work that corresponds to a specific product or version.

Every knowledge item folder is defined by its folder type—protected, public, or secured—and a set of folder access and knowledge access controls.

5.2 Administering Knowledge Folder Function Pages

The function page acts as a frame for displaying the custom and system pages that enable project team members to manage work items.Project customization consists of creating custom pages and controls and adding those pages to function page so that they are displayed in the client.

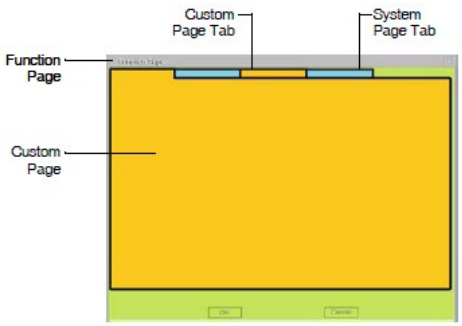


Figure 4-1: Function Page Diagram

KnowledgeWise enables project administrators to customize two knowledge folder function pages: the Add Knowledge Folder function page and the Edit Knowledge Folder function page.

5.2.1 Administering the Add Knowledge Folder Page

In the KnowledgeWise client, project members may create knowledge item folders and define knowledge item folder access controls, applicable owner rules, and applicable product rules using the Add Knowledge Item Folder window.

Figure 4-2: Add Knowledge Item Folder Window

The Add Knowledge Item Folder window displayed in the KnowledgeWise client is a manifestation of a customized Add Knowledge Item Folder function page." Add Knowledge Item Folder page customizations include the display of multiple system pages and custom pages.

Using controls in the Knowledge Item GUI folder, project administrators may define which system pages and custom pages are displayed in the client. Project administrators may add up to three system pages and any number of custom pages to the Add Specification Folder function page.

Folder Description: The Folder Description system page enables project members to define core knowledge folder properties including its title, workflow state, and owner.

Applicable Product: The Applicable Product system page enables project members to identify the products and versions that are applicable to the knowledge item folder.

Applicable Owner: The Applicable Owner system page enables project members to identify the user accounts, team groups, group folders, account types, and user variables that may manage the knowledge items in the folder.

Access Control: The Access Control system page enables project members to define the folder type, knowledge item access level, and folder access level of each knowledge item folder.

To customize the Add Knowledge Folder function page:

1Select Knowledge Folder GUI Settings > Function Pages > Submission Pages in the tree panel.

2Add or remove system pages and custom to the function page.

The Available Pages list displays the Applicable Product, Applicable Owner, and Access Control system pages and all administrator-defined custom pages.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

3 Optional: To customize the title displayed in the KnowledgeWise client, input text in the Page Display Name text box.

5.2.2 Administering the Edit Knowledge Folder Page

In the KnowledgeWise client, project members may define knowledge item folder access controls, applicable owner rules, applicable product rules, and track knowledge item folder properties using the Edit Knowledge Item Folder window.

Figure 4-3: Edit Knowledge Item Folder Window

The Edit Knowledge Item Folder window displayed in the KnowledgeWise client is a manifestation of a customized Edit Knowledge Item Folder function page." Edit Knowledge Item Folder page customizations include the display of multiple system and custom pages.

Using controls in the Knowledge Item GUI folder, project administrators may define which pages—system pages and custom pages—are displayed in the client. Project administrators may add up to six system pages and any number of custom pages to the Edit Knowledge Item Folder function page:

Folder Description: The Folder Description system page enables project members to define core knowledge folder properties including its title, workflow state, and owner.

Applicable Product: The Applicable Product system page enables project members to identify the products and versions that are applicable to the knowledge item folder.

Access Control: The Access Control system page enables project members to define folder types—public, protected, or secured, knowledge item access levels, and folder access levels of to each folder.

Applicable Owner: The Applicable Owner system page enables project members to identify the user accounts, account types, team groups, group folders, and user variables that may own the knowledge items in the folder.

Folder Order: The Folder Order system page enables project members to manage the order of the child subfolder displayed within a parent folder in the knowledge item tree panel.

To customize the Edit Knowledge Folder function page:

1Select Knowledge Folder GUI Settings > Function Pages > Submission Pages in the tree panel.

2Add or remove system pages and custom to the function page.

The Available Pages list displays the Applicable Product, Applicable Owner, Access Control, Change Log, Folder Order, and Time Report system pages and all administrator-defined custom pages.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

3 Optional: To customize the title of the page displayed in the KnowledgeWise client, input text in the Page Display Name text box.

5.3 Administering Knowledge Folder System Pages

In KnowledgeWise, a system page is a system-defined form that enables KnowledgeWise users to collect, manage, and track information in the client. System pages feature the built-in functionality of system controls. Unlike custom controls which are fully customizable and may be used to track any type of data, each system control is specifically designed to track a particular set of data that is linked to other controls and functionalities in the KnowledgeWise project.

KnowledgeWise features five predefined knowledge item folder system pages: the Change Log, Applicable Product, Access Control, Applicable Owner, Folder Order, and Time Report knowledge item folder system pages.

Folder Description: The Folder Description system page enables project members to define core knowledge folder properties including its title, workflow state, and owner.

Applicable Product: The Applicable Product system page enables project members to identify the products and versions that are applicable to the knowledge item folder.

Access Control: The Access Control system page enables project members to define folder types—public, protected, or secured, knowledge item access levels, and folder access levels of to each folder.

Applicable Owner: The Applicable Owner system page enables project members to identify the user accounts, account types, team groups, group folders, and user variables that may own the knowledge items in the folder.

Folder Order: The Folder Order system page enables project members to manage the order of the child subfolder displayed within a parent folder in the knowledge item tree panel.

Knowledge Item Folder system pages may be displayed in the Add Knowledge Item Folder window and the Edit Knowledge Item Folder window of the KnowledgeWise client:

Knowledge Item Folder System Pages	Knowledge Item Folder Function Pages	
	Add Knowledge Item Folder	Edit Knowledge Item Folder
Folder Description	X	X
Applicable Product/Version Page	X	X
Applicable Owner Page	X	X
Access Control Page	X	X
Folder Order Page		X

Table 3-1: Knowledge Item Folder Function Page

5.3.1 Defining System Page Titles

To define the title of a system page:

1Select Knowledge Folder GUI Settings > System Pages > [system page] in the tree panel.

KnowledgeWise features five predefined knowledge item folder system pages: Applicable Product, Applicable Owner, Access Control, Change Log, Folder Order, and Time Report system pages.

2 Optional: To customize the title of the page displayed in the KnowledgeWise client, input a title in the Page Display Name text box.

5.3.2 Customizing the Applicable Product System Page

In KnowledgeWise, a knowledge item may define a function in a particular product or version only if that product or version has been defined as applicable to the knowledge item folder containing the knowledge item.

The Applicable Product system page may be displayed in the Add Knowledge Items Folder page and the Edit Knowledge Item Folder page in the KnowledgeWise client.

Using controls in the Applicable Product system page, project members may identify the products and versions that are applicable to the knowledge item folder and, thereby, define how the knowledge items in that folder are to be used in the project.

To define the title displayed in the KnowledgeWise client:

1Select Knowledge Folder GUI Settings > System Pages > Applicable Product in the tree panel.

2 Optional: To customize the title of the page displayed in the KnowledgeWise client, input a title in the Page Display Name text box.

5.3.3 Customizing the Knowledge Folder Applicable Owner Page

In KnowledgeWise, project members may restrict user access to knowledge items on a folder-by-folder basis using folder-level access controls and knowledge item folder applicable owner rules.

An applicable owner is a project member, team group, or group folder that may own a work item—in this case, a knowledge item—based on administrator-defined rules.

Using controls in the Applicable Owner page, project members may designate a project member as an applicable owner of the knowledge items contained in that folder based on their user account, account type, team group, group folder, or user variable.

user account:A user account represents an individual KnowledgeWise user that has been assigned a license and account type that enables that user to work in the project.

account type: An account type represents a set of privileges and responsibilities that may be granted to a user account in a KnowledgeWise project. Every user account is defined by its account type.

team group:A team group is an administrator-defined collection of project members—who do not necessarily share the same account type—that share common responsibilities in the project. Team groups organize support teams into smaller, more useful entities that facilitate the assignment and control of items and events.

group folder:A group folder represents a kind of buffer between the team group and the items assigned to that group. Each group folder is owned by a particular group and privilege protected by three access controls: View, Put, and Get. Project members that have been assigned the appropriate account type may view items assigned to a group folder, add items to a group folder, or forward those items to another project member or group.

The Applicable Owner system page may be displayed in the Add Knowledge Item Folder page and the Edit Knowledge Item Folder page in the KnowledgeWise client.

5.3.4 Customizing the Access Control System Page

In the KnowledgeWise client, the Access Control system page enables project members to define folder-level access controls. Access controls determine which project members may access and manage the knowledge items and knowledge item subfolders managed in each knowledge item folder.

Using controls in the Access Control page, a change manager may designate the folder type of each knowledge item folder—protected, public, or secured—and define folder-level access controls by account type, team group, and user account.

Folder Access Level:A folder access level defines which project members may access, create, manage, and delete knowledge item and knowledge item subfolders. Every knowledge item folder and knowledge item folder in a KnowledgeWise project is assigned a folder access level, which defines the access rights of each user.

Knowledge Item Access Level:A knowledge item access level defines which project members may access, view, edit, create, and delete the knowledge items in a knowledge item folder.

The Access Control system page may be displayed in the Add Knowledge Item Folder page and Edit Knowledge Item Folder page in the KnowledgeWise client. Alternately, the folder access level and knowledge item access level controls that are usually displayed in the Access Control system page may be displayed in knowledge item folder custom pages. For more information see See “Displaying Access Controls in Custom Pages”.

To customize the Access Control system page:

1Select Knowledge Folder GUI Settings > System Pages > Access Control in the tree panel.

2 Optional: To customize the page title, input a title in the Page Display Name text box.

3Select an access control display preference:

- To display the Access Control page as a distinct tabbed page, select the Separate Page option button.
- To enable Access Control page data-entry controls to be added to knowledge item folder custom pages, select the Separate Fields option button.

Note: If the Separate Fields option is selected, the Access Control page may not be added to the Add Knowledge Item Folder or Edit Knowledge Item Folder function page. The Folder Access Level system control and the Specification Access Level system control may be added to any specification folder custom page.

5.3.5 Customizing the Folder Order System Page

In the KnowledgeWise client, the Folder Order page enables project members to manage the order of the subfolders displayed within a folder in the knowledge item tree panel.

Knowledge Item folders are displayed in a hierarchical tree structure of folders and subfolders called the knowledge item tree. Every folder within the root knowledge item folder may contain multiple child subfolders. Folder-level access controls and other folder properties may cascade down the hierarchy.

The Folder Order page may be displayed in the Edit Knowledge Item Folder page in the KnowledgeWise client.

To define the title displayed in the KnowledgeWise client:

1Select Knowledge Folder GUI Settings > System Pages > Folder Order in the tree panel.

2 Optional: To customize the title of the page displayed in the KnowledgeWise client, input a title in the Page Display Name text box.

5.4 Administering Knowledge Folder System Fields

System fields are system-defined folder property fields that have a designated use in the KnowledgeWise project— the data integrated is other TechExcel KnowledgeWise features for knowledge item management or reporting.

Each knowledge item folder system field is represented by a data-entry control in the KnowledgeWise client that enables project members to manage and track data. Project administrators may make only limited customizations to system controls.

- The title of the corresponding data-entry control may be customized.
- The Folder Description field may be customized to support spell checking, time stamping, and HTML tags.

The values displayed as options in these system controls are drawn from the KnowledgeWise database and cannot be customized by the administrator.

1001	Folder Name	Long (<=255) text (edit box)	The Folder Name field tracks the userdefined name of the knowledge item folder.
1002	Folder Description	Multiple-line Edit Box	The Folder Description field tracks a short description of the knowledge item folder and its use in the project. Project administrators may enable spell checking, time stamping, and HTML tags.
1003	Folder Status	Drop Down List	The Folder Name field tracks the userdefined name of the knowledge item folder.
1004	Is Read-only	Check Box	The Is Read-only check box designates a knowledge folder page as read-only.
1005	Default Value	Drop Down List	The Default Value field tracks the userdefined name of the knowledge item folder.
1006	Folder Identifier	Short (<=50) text (edit box)	The Folder Identifier field tracks a system-defined folder identifier number of the knowledge item folder.
1007	Folder Access Control	Access Control	The Folder Access Control system field defines and controls access to knowledge folders by user account, account type, and team group.
1008	Knowledge Access Control	Access Control	The Knowledge Access Control system field defines and controls access to knowledge items by user account, account type, and team group.

5.4.1 Displaying Access Controls in Custom Pages

KnowledgeWise knowledge item folder-level access controls may be defined using the folder access level and knowledge item access level controls. By default, these controls are displayed in the Access Control system page that is in the Add Knowledge Item Folder and Edit Knowledge Item Folder windows.

The Folder Access Level and Knowledge Item Access Level controls may be added to knowledge item folder custom pages as system controls.If this option is selected, the Access Control page is not displayed in the client.

To display access controls in custom pages:

1 Select the Access Control page in the System Pages subfolder in the Knowledge Item Folder GUI folder.

2 Optional: To enable Access Control page data-entry controls to be added to knowledge item folder custom pages, select the Separate Fields option button.

If this option is selected, the Access Control page may not be added to the Add Knowledge Item Folder or Edit Knowledge Item Folder function page. The Folder Access Level system control and the Specification Access Level system control may be added to any specification folder custom page.

3 Select a custom page in the Custom Pages folder.

4 Right-click in the page and select the Add Control command in the shortcut menu.

The Add New Field window appears.

5 Select the Folder Access Control or Knowledge Item Access Control in the System Controls list.

The Add New Field window closes.

6Click the OK button.

Chapter 6 Knowledge Item Administration abc

6 CHAPTER 5 Knowledge Item Administration abc

In this chapter:

- Understanding KnowledgeWise Knowledge Items
- Administering Knowledge Item Function Pages
- Administering Knowledge Item System Pages
- Administering Knowledge Item Fields and Controls
- Administering Knowledge Item List Columns
- Administering Knowledge Item-Level Knowledge Management

6.1 Understanding KnowledgeWise Knowledge Items

In KnowledgeWise, knowledge items administration is the process of defining structures for recording, managing, and tracking knowledge items in a project. Knowledge Item management structures include knowledge item folders, knowledge items, and knowledge item workflow. All knowledge items are managed in the knowledge item view of the KnowledgeWise client.

Project administrators may define the knowledge item object, define workflow rules, and customize the client GUI to enable project members to effectively manage those objects.

6.2 Administering Knowledge Item Function Pages

The function page acts as a frame for displaying the custom and system pages that enable project team members to manage work items.Project customization consists of creating custom pages and controls and adding those pages to function page so that they are displayed in the client.

The function page acts as a frame for displaying the custom and system pages that enable project team members to manage work items.Project customization consists of creating custom pages and controls and adding those pages to function page so that they are displayed in the client.

KnowledgeWise enables project administrators to customize five knowledge item function pages: the Add Knowledge function page, Edit Knowledge function page, Forward Folder function page, Version function page, Knowledge Template function page, Group Change function page, Group Forward function page, KW Search function page, KW S page, and KW Search function page.

Add Knowledge: The Add Knowledge function page enables the user to define core knowledge item properties and to submit new knowledge items to the project.

Edit Knowledge: The Edit Knowledge function page enables the user to edit core knowledge item properties.

Detail Knowledge: The Detail Knowledge function page enables the user to view knowledge item details.

Forward Knowledge;The Forward function page enables the user to update core knowledge item properties and to forward a knowledge item to another user or workflow state.

KW Search for KnowledgeWise Project Team Members:The KW Search for KnowledgeWise Project Team Members h function page enables KnowledgeWise project team members to search for knowledge items in the KnowledgeWise project.

KW Search for Non- KnowledgeWise Project Team Members: The KW Search for Non-KnowledgeWise Project Team Members function page enables DevTrack and DevSpec project team members to search for knowledge items in the KnowledgeWise project.

KW Search. for Non- Project Team Members: The KW Search. for Non- Project Team Member function page enables non-project members (such as customers) to search for knowledge items in the KnowledgeWise project.

6.2.1 Administering the Add Knowledge Item Function Page

In the KnowledgeWise client, project members may create new knowledge items and define knowledge item properties using the Add Knowledge Item window—a multiple-page tabbed form.

The Add Knowledge Item window displayed in the KnowledgeWise client represents administrator-defined Add Knowledge Item function page.

The Add Knowledge Item function page may display the Knowledge Description system page and all custom pages.

To customize the Add Knowledge Item function page:

1Select Knowledge GUI Settings > Function Pages > Submission Pages in the tree panel.

2Add or remove system pages and custom to the function page.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.

- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

3 Optional: To define the title displayed in the KnowledgeWise client, input text in the Page Display Name text box.

6.2.2 Administering the Edit Knowledge Item Function Page

In the KnowledgeWise client, project members may define or update knowledge item properties in the Edit Knowledge Item window—a multiple-page form that may display the All Links, Linked Knowledge, and Events pages as well as any number of custom pages.

The Edit Knowledge Item window displayed in the KnowledgeWise client represents administrator-defined Edit Knowledge Item function page. The Edit Knowledge Item function page may three system pages and any number of custom pages:

Knowledge DescriptionP: The Knowledge Description system page enables project members to define core knowledge item properties including its title, workflow state, and owner.

All Links: The Knowledge Item All Links system page enables project members to track, link, and manage the knowledge items, knowledge items, and change requests that are associated with a knowledge item.

Linked Knowledge: The Linked Knowledge system page enables project members to track, link, and manage the knowledge items, and knowledge items that are associated with a knowledge item.

Events: The Knowledge Item Events page enables project members to create, edit, delete, and track the events.

Version: The Version system page enables project members to manage and track multiple versions of knowledge items.

To customize the Edit Knowledge Item function page:

1Select Knowledge GUI Settings > Function Pages > Editing Pages in the tree panel.

2Add or remove system pages and custom to the function page.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

3 Optional: To define the title displayed in the KnowledgeWise client, input text in the Page Display Name text box.

6.2.3 Administering the Knowledge Item Detail Function Page

In the KnowledgeWise client, project members may manage and track knowledge item properties in the knowledge item detail panel of the knowledge item view. The knowledge item detail panel shows detailed information about a single knowledge item in multiple tabbed pages including the All Links, Linked Knowledge, History, and Events pages as well as any number of custom pages.

Using controls in the Knowledge Item GUI folder, project administrators may add or remove system pages and custom pages to the Knowledge Item Detail function page:

Knowledge Description: The Knowledge Description system page enables project members to define core knowledge item properties including its title, workflow state, and owner.

All Links: The Knowledge Item All Links system page enables project members to track, link, and manage the knowledge items, knowledge items, and change requests that are associated with a knowledge item.

Linked Knowledge: The Linked Knowledge system page enables project members to track, link, and manage the knowledge items, and knowledge items that are associated with a knowledge item.

Version: The Version system page enables project members to manage and track multiple versions of knowledge items.

Events: The Knowledge Item Events page enables project members to create, edit, delete, and track the events that are associated with a knowledge item

History: The History system page provides project members with a read-only audit log of a knowledge item in four standard reports—the Change Log report, the Owner State Change History report, the Summary report, the Tracking History report—and any number of custom reports.

To customize the Detail Knowledge Item function page:

1Select Knowledge GUI Settings > Function Pages > Detail Pages in the tree panel.

2Add or remove system pages and custom to the function page.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

3 Optional: To define the title displayed in the KnowledgeWise client, input text in the Page Display Name text box.

6.2.4 Administering the Forward Knowledge Item Function Page

In the KnowledgeWise client, project members may reassign knowledge items to other project members in workflow using the Forward Knowledge Item window—a multiple-page form that may any number of custom pages.

The Forward Knowledge Item window displayed in the KnowledgeWise client represents administrator-defined Forward Knowledge Item function page. No system pages are displayed in the Forward Knowledge Item page.

The Forward Knowledge Item function page may display the Knowledge Description system page and all administrator-defined custom pages.

To customize the Forward Knowledge Item function page:

1Select Knowledge GUI Settings > Function Pages > Forward Pages in the tree panel.

2Add or remove system pages and custom to the function page.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

3 Optional: To define the title displayed in the KnowledgeWise client, input text in the Page Display Name text box.

6.3 Administering Knowledge Item System Pages

In KnowledgeWise, a system page is a system-defined form that enables KnowledgeWise users to perform tasks in the client. Each system page generally displays one or more system controls that enable project members to manage and track a special set of data.

Unlike custom pages which are fully customizable, system pages are specifically designed to perform a task. System page customization is generally limited to the ability to customizing page and data-entry control titles.

KnowledgeWise features six knowledge system pages: the History system page, Version system page, Event system page, All Links system page, and Linked Knowledge system page.

Description: The Description system page enables project members to define core knowledge item properties including its title, workflow state, and owner.

History: The History system page displays a read-only audit log of the changes made to a knowledge item in four standard reports: the Change Log report, the Owner State Change History report, the Summary report, the Tracking History report.

Version: The Version system page enables project members to manage and track multiple versions of knowledge items. Event The Knowledge Events page enables project members to create, edit, delete, and track the events that are associated with a knowledge item.

All Links: The All Links system page enables development organizations to associate knowledge items with KnowledgeWise requirements and specifications so that related work items may be tracked and managed together in workflow

Linked Knowledge: The Linked Knowledge system page enables development organizations to link and track associated knowledge items

Using controls in DevSuite Admin client, project administrators may add knowledge item system pages and knowledge item custom pages to knowledge item function page.s.

However, not every system page may be added to every function page. Only those system pages that are defined as applicable to a function page may be added. Administrator-defined custom pages may be added to any function page.:

Function Pages		Add	Edit	Detail	Forward	Version	Template	Group Forward	Group Change
Custom Pages		X	X	X	X	X	X	X	X
System Pages	History			X					
	Version		X	X					
	Event		X	X					
	All Links		X	X					
	Linked Knowledge		X	X					
	Description	X	X	X	X	X	X	X	X

6.3.1 Administering the Knowledge History System Page

The History system page displays a read-only audit log of the changes made to a knowledge item in four standard reports.

Four system-defined reports may be displayed in the History page. Custom reports based on custom pages may also be displayed in the page.

Change Log: The Change Log report displays the complete record of every update made to a knowledge item in chronological order.

Owner State Change History: The Owner and State Change report displays the history of ownership and state changes throughout the entire life cycle of a knowledge item.

Summary: The Summary report displays a summary of key knowledge item properties including he knowledge item ID, title, description, the knowledge item owner, submitter, and current state.

Tracking History: The Tracking History report displays detailed information about knowledge item updates in a tabular format.

Custom: Custom reports display data that is managed and tracked in an administrator-defined custom page. Every administrator-defined page may be represented by a report in the History page.

The History system page may be displayed in the Knowledge Item Detail function page, Knowledge Item Edit function page, and the Knowledge Item Change function page.

To add or remove a report to the History system page:

1Select Knowledge GUI System Settings > System Pages > History page.

2Add or remove reports to the function page.

- To add a report to the function page, select a page in the Available Sections list and click the Right Arrow button.
- To remove a page from the function page, select a page in the History Info list and click the Left Arrow button.

3 Optional: To define the title displayed in the KnowledgeWise client, input text in the Page Display Name text box.

6.3.2 Administering the Version System Page

The Version system page enables project members to manage and track multiple versions of knowledge items.

The Version system page may be displayed in the Edit Knowledge Item function page or the Knowledge Detail function page.

The title of the All Links system page may be customized in each project.

6.3.3 Administering the Knowledge Event System Page

The Knowledge Events page enables project members to create, edit, delete, and track the events that are associated with a knowledge item.

The Event system page may be displayed in the Edit Knowledge Item function page or the Knowledge Detail function page.

The title of the All Links system page may be customized in each project.

6.3.4 Administering the Knowledge Item All Links System Page

The All Links system page enables development organizations to associate knowledge items with KnowledgeWise requirements and specifications so that related work items may be tracked and managed together in workflow

The All Links system page may be displayed in the Edit Knowledge Item function page or the Knowledge Detail function page. The title of the All Links system page may be customized in each project.

6.3.5 Administering the Linked Knowledge System Page

The Linked Knowledge system page enables development organizations to link and track associated knowledge items.Administering Knowledge Item Custom Pages

In KnowledgeWise, a custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data.

Each custom page may display system controls and custom controls. Using custom pages, system controls, and custom controls, a business may create forms that are uniquely designed to manage and track data that is specific to their environment and business.

Like system pages, custom pages may be added to function pages to build the multiple-page forms that enable project members to manage requirements, specifications, and change requests in project views.

Project administrators may add and remove system, custom, and multiple-selection controls to custom pages, customize the layout of controls in the page, and define their the tab order.

6.4 Administering Knowledge Item Fields and Controls

Project administrators may determine which knowledge item fields may be displayed in the knowledge item list panel and define identifying icons for knowledge item workflow states and owners.

The List View Graphic Indicator Setting manager enables project administrators to customize the appearance of the knowledge items displayed in the knowledge items list panel based on knowledge item properties. Project administrators may customize the following features in the knowledge item list panel:

- Customize the font weight of text of issues in the Issue list based on issue property values.
- Customize the color of text of the issue in the Issue list based on issue property values..
- Customize the icons that identify issue state based on issue property values.

6.4.1 Understanding Knowledge Item System Fields

System fields are system-defined data-entry controls that have a designated use in the KnowledgeWise project and are integrated with other TechExcel KnowledgeWise features for knowledge item management or reporting.

Project administrators may add, remove, or customize twenty system fields to system or custom pages.

Project administrators may make only limited customizations to system controls. The values displayed as options in these system controls are drawn from the KnowledgeWise database and cannot be customized by the administrator.

1101	Title	Long (<=255) Text (Edit Box)

1102	Description	Multiple-line Edit Box
1103	Knowledge Folder Name	Long (<=255) Text (Edit Box)
1104	Knowledge Folder Path	Long (<=255) Text (Edit Box)
1105	File Attachment	File Attachment
1106	Assigned by	Short (<=50) Text (Edit Box)
1107	Date Assigned	Date-Time
1108	Last Modified by	Short (<=50) Text (Edit Box)
1109	Date Last Modified	Date-Time
1110	Created by	Short (<=50) Text (Edit Box)
1111	Date Created	Date-Time
1112	Knowledge Item ID	Short (<=50) Text (Edit Box)
1113	Knowledge Owner	Drop Down List
1114	Knowledge State	Drop Down List
1115	Knowledge Substate	Drop Down List
1119	Product/Version	Product/Version
1122	File Attachment	File Attachment
1123	External ID	Long (<=255) Text (Edit Box)

6.5 Administering Knowledge Item List Columns

The knowledge item list panel in the KnowledgeWise client enables project members to view high-level information about multiple knowledge items in a tabular list of rows and intersecting columns.

Using controls in the Knowledge List Columns page, project administrators may add or remove columns, define the display order of those columns, and define options for the rows the list panel.

6.5.1 Displaying Knowledge System Fields in the Knowledge List Panel

Up to 14 system fields may be displayed in the knowledge list panel.

ID	Title	Knowledge State
Knowledge Owner	Assigned by	Date Assigned
Created by	Date Created	Date Last Modified
Knowledge Folder Name	Knowledge Folder Path	Knowledge Substate
Last Modified by	External ID	

To add knowledge fields to the knowledge item list panel:

1. Select Knowledge GUI Settings > Field Design List > Knowledge Item List Column.
2. Add or remove knowledge item fields to the knowledge item list:
   - To add a knowledge column to the knowledge item list, select a knowledge system field in the Available Data Fields list and click the Right Arrow button.
   - To remove a knowledge column from the knowledge item list, select a knowledge system field in the Selected List Columns list and click the Left Arrow button.

6.5.2 Customizing Knowledge Item List Indicator Icons

Project administrators may choose to display a list indicator icon in the knowledge item list panel to indicate the current status of knowledge items based on a*single*knowledge field.

Using controls in the Select an Icon dialog box, project administrators may associate icons to one of two fields: the workflow state field or the owner field.

If the administrator associates an icon for each knowledge item workflow state (for example, Urgent, High, Medium, Low), issues in the knowledge item list panel are identified by the appropriate icon.

Project administrators may also use colored and boldface text to identify specific issue property values.

Note: Indicator icons may be defined for *one and only one* issue property at a time. Indicator icons (and indicator text) identifying the value assigned to an issue based on a single issue property. Defining indicator icons for other issue properties immediately deletes all settings defined for the previous issue property.

To define knowledge item field icons:

1. Select the Knowledge Item List Column page in the System Pages subfolder in the Knowledge Item GUI folder.
2. Click the List View Graphic Design button.

   The List View Graphic Indicator Setting manager appears.
3. Select a knowledge item property in the Indicator Field Selection dropdown list.

   The Field Selection dropdown list displays the Knowledge Item Owner and Knowledge Item State fields. The knowledge item property selected determines the options that are displayed in the Available Choices list.
4. Select a field option in the Available Choices list.

   The Available Choices list displays all administrator-defined knowledge item owners or knowledge item workflow states.

   - If the Knowledge Item Owner property was selected in the Field Selection dropdown list, the Available Choices list displays every user account, account type, and team group that may own knowledge items in the project.
   - If the Knowledge Item State property was selected in the Field Selection dropdown list, the Available Choices list displays every administrator-defined knowledge item workflow state.
5. Click the Select Icon button. The Select an Icon dialog box appears.
6. Choose an icon option from the Icon Field area.
7. Click the OK button.

   The Select an Icon dialog box closes. The icon appears next to the option selected in the Available Options list.
8. Click the OK button.

6.5.3 Customizing Knowledge Item List Font Colors and Weights

To customize knowledge item list font colors and weights:

1. Select the Issue List in the Main View GUI folder of the Admin Tree window.

   The Issue List manage appears.
2. Select the List View Graphic Display Settings button.

   The List View Graphic Indicator Setting manager appears.
3. Select an option from the Indicator Field Selection dropdown list.

   The Field Selection dropdown list displays multiple issue properties. The options displayed in the Available Choices list depend on the issue property option selected in the Field Selection dropdown list.
4. Select a field option in the Available Choices list.

   The Available Choices list displays the options available for the selected issue property as well as multiple system-defined objects. Project administrators may not customize the icon displayed for system-defined objects.
5. **Optional**: To set the issue in boldface type select the Text Bolded check box.
6. **Optional**: To define the color of issue text select an option from the Text Color dropdown list.
7. Click the OK button.

6.6 Administering Knowledge Item-Level Knowledge Management

In the KnowledgeWise client, project members may define and track knowledge item details in the Description system field. The Description field is displayed as a multiple-line text edit control.

Text may be entered into the Description control whenever a project member submits, updates, forwards, closes, or reopens a knowledge item.

The Description control may feature additional tools that enable project members to better describe development tasks:

- **Time stamping**: The Description control supports time stamping of all data entered in the field. A time stamp is a signature that is concatenated to text entered in the data-entry field that logs the exact date and time of the entry.
- **Spell checking**: The Description control supports spell checking. Spell checking is an optional feature that must be enabled by a project administrator in a DevTrack project.
- **HTML Text Editor**: The Description control displays tools that enable project members to format the text displayed in the memo field control.

6.6.1 Enabling Spell Checking in the Description Control

To enable spell checking in the knowledge item Description control:

1. Select the Description field (1102) in the System Fields list of the Knowledge Item Field page in the Field Design subfolder in the Knowledge Item GUI folder.
2. Click the Edit button. The Edit System Field dialog box appears.
3. Select the Enable Spell Checking check box.
4. Click the OK button.

   The Edit System Field dialog box closes.

6.6.2 Enabling HTML Formatting in the Description Control

To enable time stamping in the knowledge item Description control:

1. Select the Description field (1102) in the System Fields list of the Knowledge Item Field page in the Field Design subfolder in the Knowledge Item GUI folder.
2. Click the Edit button.

   The Edit System Field dialog box appears.
3. Select the Support HTML Format check box.
4. Choose an HTML text editor option:
   - To support editing within the Description control and in a pop-up text editor, select the Inline and Dialog Editing option button.
   - To support editing within the Description control and in a pop-up text editor, Dialog Editing option button.
5. Click the OK button.

   The Edit System Field dialog box closes.

Figure 5-6: Custom Page

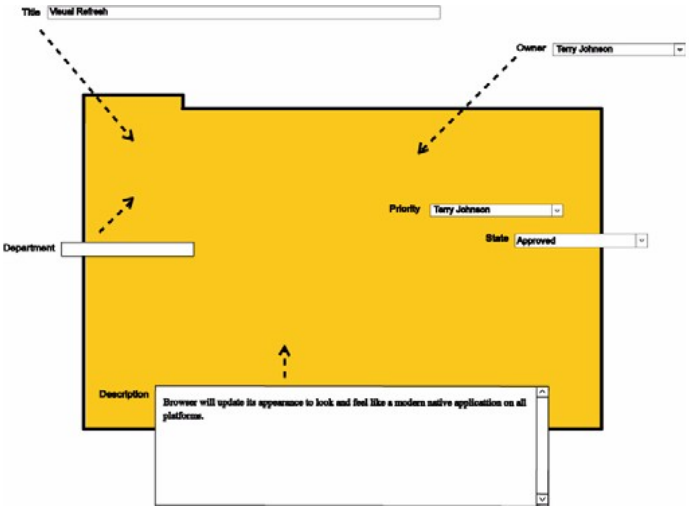


Figure 5-5: Linked Knowledge Page in KnowledgeWise Web Client

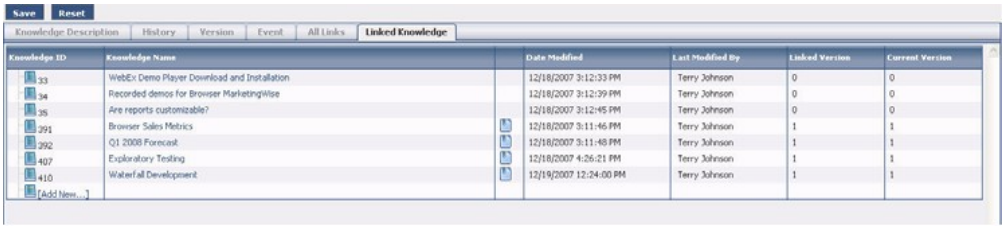


Figure 5-4: Event Page in the KnowledgeWise Client

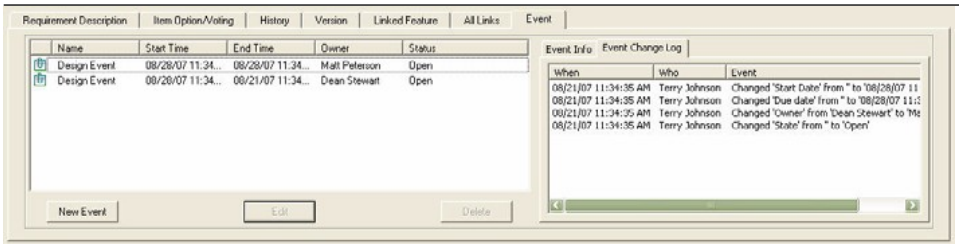


Figure 5-3: Version Page in the KonwledgeWise Client

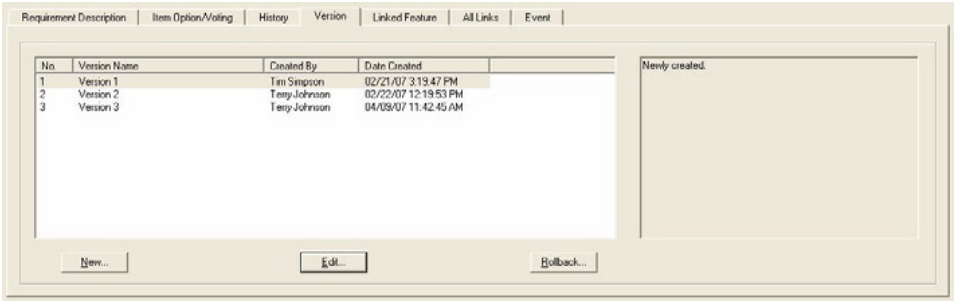


Figure 5-2: History Page in the KnowledgeWise Client

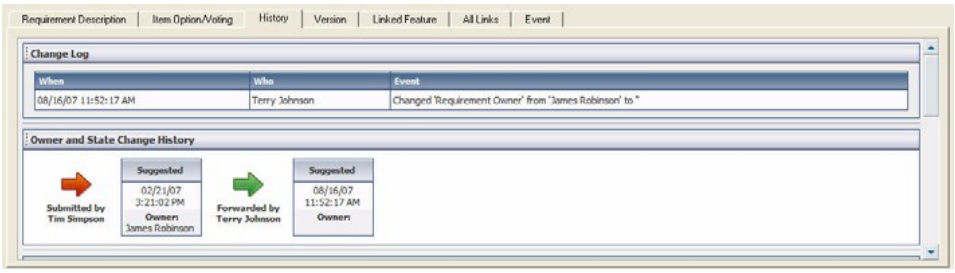
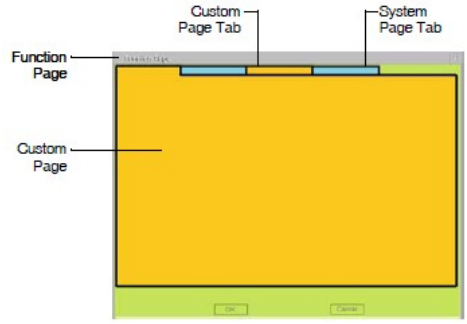


Table 4-1: Knowledge Item Function Pages

Figure 5-1: Function Page Diagram



Chapter 7 Secured Access Level Administration

7 CHAPTER 6 Secured Access Level Administration

In this chapter:

- Understanding Secured Knowledge Access Level Security
- Administering Knowledge Folder Secured Access Levels
- Administering Knowledge Item Access Levels
- Administering Non-Member Access Levels

7.1 Understanding Secured Knowledge Access Level Security

Secured access levels are an optional DevSuite Enterprise Edition feature that enable development organizations to define flexible folder-level and knowledge item-level access controls.

In KnowledgeWise, every secured knowledge folder and secured knowledge item is defined by its name, status, and one or more access levels. An access level is a set of

administrator-defined access controls that define which project members may view, create, edit, and delete a secured knowledge folder or secured knowledge item.

In the KnowledgeWise, knowledge folders may be defined as public, protected, or secured. If the knowledge folder is a secured folder, it is defined by one or more access levels.

- Secured folder access controls and secured knowledge item access controls are defined on a folder-by-folder basis. Each secured folder may be defined by one or more access levels.
- Distinct access levels may be assigned to a secured knowledge folder and to the knowledge items it contains.

Project Members may only manage a secured knowledge folder or secured knowledge item if they have been granted access rights within an access level that defines that knowledge folder or knowledge item.

Secured knowledge access level administration is the process of creating knowledge folder and knowledge item access levels, and of assigning access types to project team members based on their account type, team group, or user account.

7.1.1 Understanding Access Level Conflict Resolution Rules

Unlike public and protected access controls, which are strictly account type-based,*secured access level*access controls are defined by account type, team group,*and*user account.

As a result, a project member may be granted three or more sets of access rights— called*access types*—within the same access level: one for their account type, one for their user account, and one for each team group membership.

Secured access level rules ensure that each project member is granted maximum access within each access level based on the access types assigned to them unless their access rights are assigned by name.

For example,

- If Terry Johnson is assigned two access types—one for his*account type*and one for his*team group*, the access type that provides the greatest access defines his access rights in that access level.
- However, if Terry Johnson is assigned an access type based on his user account, he is assigned the access rights granted by that access type regardless of all other considerations.

The access type assigned to a project team member by name—that is, based on his or her user account—always determines which access rights are granted to that user. If no access type is granted to the user by name, the access type that provides the project member with greater access is assigned to that project team member.

7.1.2 Comparison with Public and Protected Access Controls

In KnowledgeWise, project member access to knowledge folders and knowledge items may be controlled using account-type based*public access controls*,*protected access controls*, or*secured access controls*.

All public and protected access controls are defined by the project administrator and are implemented uniformly throughout the project—all public folders are defined by the same access controls and all protected knowledge items are defined by the same access controls. Project team members may designate a knowledge folder as public or protected, but they cannot define access rights that are tailored to specific folders or knowledge items.

Using secured knowledge access controls, project members may define the level of access (access level) that is required to view, create, edit, or delete the folders and knowledge items within a secured folder.

7.2 Administering Knowledge Folder Secured Access Levels

Secured folder access levels are an optional DevSuite Enterprise Edition feature that enable development organizations to define flexible folder-level access controls.

In KnowledgeWise, every secured knowledge folder is defined by its name, status, and one or more*access levels*. An access level is a set of administrator-defined access controls that define which project members may view, create, edit, and delete a*secured knowledge folder*.

The access level required to manage secured knowledge folders (and their contents) is defined on a folder-by-folder basis. A project member may view and manage a secured knowledge folder only if are granted access rights by an access level that is applicable to that secured folder.

Within each access level, access rights are assigned to project members based their account type, user account, or team group membership.

7.2.1 Defining Knowledge Folder Access Levels

Five distinct access rights may be granted to project members: No Access, Read Only, Can Edit, Can Create, and Can Delete: These access rights are grouped together into five*access types*:

No access:The No Access control prevents the project member from accessing knowledge folders or knowledge items.

Read only:The Read-Only control enables the project member to view but not edit knowledge folders or knowledge items.

Can edit:The Can Edit control enables the project member to view and edit but not create he knowledge folder or knowledge item.

Can create and edit:The Can Create and Edit control enables the project member to view, edit, and create knowledge folders or knowledge items.

Can delete, create and edit:The Can Delete, Create, and Edit control enables the project member to view, edit, create, and delete knowledge folders or knowledge items.

To define a knowledge folder access level:

1 Select Knowledge Access Control > Folder Access Control in the tree panel.

2 Click the New button. The New Folder Access Level dialog box appears.

3 Define the name of the folder access level and click the OK button. The New Folder Access Level dialog box closes.

4 Select one or more account types, team groups, or user accounts in the User list and click the Change Select Selected button. The Access Type dialog box appears.

5 Select an option in the Access Type dialog box. The Access Type dialog box displays five options:

- No Access
- Read Only
- Can Edit
- Can Create and Edit
- Can Delete, Create and Edit

6 Click the OK button. The Access Type dialog box closes and the access type is assigned to the selected account type, team group, or user account.

7.2.2 Defining User Account Access Levels

To add a user account to an access level:

1 Select Knowledge Access Control > Folder Access Control in the tree panel.

2 Select an access level in the Access Level list.

3 Click the Add User button. The Add User dialog box appears.

4 Select one or more user accounts in the user list. The list displays the name and account type of every project team member.

5 Select an access type in the Access dropdown list. The Access Type dialog box displays five options:

- No Access
- Read Only
- Can Edit
- Can Create and Edit
- Can Delete, Create and Edit

6 Click the OK button. The Add User dialog box closes and the access type is assigned to the selected user account.

To remove a user account from an access level:

1 Select Knowledge Access Control > Folder Access Control in the tree panel.

2 Select an access level in the Access Level list.

3 Select a user account in the User list.

4 Click the Remove User button. A confirmation dialog box appears.

5 Click the Yes button.

7.3 Administering Knowledge Item Access Levels

Secured folder access level security is an optional, DevSuite Enterprise Edition feature that enable development organizations to define powerful folder- and knowledge item-level access controls. Project members may choose to define which levels of access are required to view, create, edit, and delete folder and knowledge items within a secured folder.

Secured access controls are based on administrator-defined*access levels*. In KnowledgeWise, an access level is a set of access rights that are granted to project team members based on their account type, team group, or user account. Five distinct access rights may be granted to project members:

No access:The No Access control prevents the project member from accessing knowledge folders or knowledge items.

Read only:The Read-Only control enables the project member to view but not edit knowledge folders or knowledge items.

Can edit:The Can Edit control enables the project member to view and edit but not create he knowledge folder or knowledge item.

Can create and edit:The Can Create and Edit control enables the project member to view, edit, and create knowledge folders or knowledge items.

Can delete, create and edit:The Can Delete, Create, and Edit control enables the project member to view, edit, create, and delete knowledge folders or knowledge items.

7.3.1 Defining Knowledge Folder Access Levels

To define a knowledge folder access level:

1 Select Knowledge Access Control > Folder Access Control in the tree panel.

2 Click the New button. The New Folder Access Level dialog box appears.

3 Define the name of the folder access level and click the OK button. The New Folder Access Level dialog box closes.

4 Select one or more account types, team groups, or user accounts in the User list and click the Change Select Selected button. The Access Type dialog box appears.

5 Select an option in the Access Type dialog box. The Access Type dialog box displays five options:

- No Access
- Read Only
- Can Edit
- Can Create and Edit
- Can Delete, Create and Edit

6 Click the OK button. The Access Type dialog box closes and the access type is assigned to the selected account type, team group, or user account.

7.3.2 Defining User Account Access Levels

To add a user account to an access level:

1 Select Knowledge Access Control > Folder Access Control in the tree panel.

2 Select an access level in the Access Level list.

3 Click the Add User button. The Add User dialog box appears.

4 Select one or more user accounts in the user list. The list displays the name and account type of every project team member.

5 Select an access type in the Access dropdown list. The Access Type dialog box displays five options:

- No Access
- Read Only
- Can Edit
- Can Create and Edit
- Can Delete, Create and Edit

6 Click the OK button. The Add User dialog box closes and the access type is assigned to the selected user account.

To remove a user account from an access level:

1 Select Knowledge Access Control > Folder Access Control in the tree panel.

2 Select an access level in the Access Level list.

3 Select a user account in the User list.

4 Click the Remove User button. A confirmation dialog box appears.

5 Click the Yes button.

7.4 Administering Non-Member Access Levels

In KnowledgeWise, knowledge items may be organized in a hierarchical tree structure consisting of multiple folders and subfolders. This structure, called the knowledge tree, is displayed in the knowledge tree panel of the KnowledgeWise client.

Every knowledge folder in that structure may be designated as a public knowledge folder, protected knowledge folder, or secured knowledge folder. Distinct access controls may be defined for each folder type (public, protected, and secured) enabling development organizations to implement folder-level security throughout the project.

- Public and protected knowledge access controls are account type-based and implemented uniformly throughout the project. Every public knowledge folder is defined by the same set of access controls.
- Secured knowledge access controls are based on administrator-defined*access levels*and may be defined on a folder-by-folder basis. Secured access controls are an optional DevSuite feature that must be enabled in each project. For more information see “Understanding Secured Knowledge Access Level Security”

To grant public knowledge access rights:

1 Select Access Control > Non-Member Access Control in the tree panel.

2 Select the Can View Public Knowledge check box.

To grant protected knowledge access rights:

1 Select Access Control > Non-Member Access Control in the tree panel.

2 Select the Can View Protected Knowledge check box.

To define knowledge folder access levels:

1 Select Access Control > Non-Member Access Control in the tree panel.

2 Select the Can View Knowledge with Defined Access check box.

3 Select an option in the Knowledge Folder Access Level list. The Knowledge Access Level list displays all administrator-defined knowledge folder access levels.

To define knowledge item access levels:

1 Select Access Control > Non-Member Access Control in the tree panel.

2 Select the Can View Knowledge with Defined Access check box.

3 Select an option in the Knowledge Access Level list.

The Knowledge Access Level list displays all administrator-defined knowledge item access levels.

Chapter 8 Page, Field, and Control Administration

8 CHAPTER 7 Page, Field, and Control Administration

In this chapter:

- Administering System Pages
- Administering Custom Pages
- Administering System Controls
- Administering Custom Fields and Controls
- Customizing Selection and Combination Box Controls
- Administering Calculation Fields and Controls

8.1 Understanding KnowledgeWise Client GUI Customization

KnowledgeWise project administration is largely the process of defining structures for recording, managing, and tracking knowledge items in workflow and maintaining multiple versions of each knowledge item.

Customized project management structures are manifested in the KnowledgeWise client as views, panels, folders, and pages.

A view is an interface that displays and organizes data in the client workspace. Views enable project members to manage and track different types of data using structures that are specifically designed to manage that data. The building blocks of each view— folders, pages, and controls—may be customized to support custom business processes.

knowledge tree panel: The tree panel is a hierarchal structure composed of folders and subfolders that organizes knowledge items into distinct areas of work.

knowledge list panel: The list panel shows high-level information about multiple knowledge items in a tabular format of rows and columns.

knowledge detail panel: The detail panel displays detailed about a single knowledge item or knowledge folder in one or more form pages.

The building blocks of client interface include: controls, folder pages, and form pages.

Control:A control is a GUI element displayed in a folder page or custom page that enables the user to define, update, and track data. KnowledgeWise supports system controls and custom controls. Controls are displayed in system pages and custom pages. Each controls represents a knowledge item field.

System Page — A system page is a predefined form that enables project members to collect, manage, and track project data. A system page displays a predefined set of system controls.

Custom Page — A custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data. A custom page may display system controls and custom controls.

Function Page — A function page is a compound page composed of multiple custom pages and system pages. In the KnowledgeWise client, function pages are displayed as multiple-page forms that enable the user to perform a particular function such as adding or updating a knowledge folder or a knowledge item.

The function page acts as a frame for displaying the custom and system pages that enable project team members to manage work items.Project customization consists of creating custom pages and controls and adding those pages to function page so that they are displayed in the client.

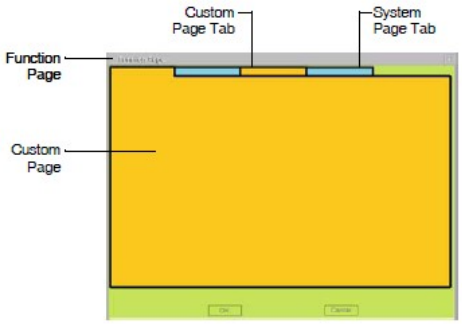


Figure 7-1: Function Page Diagram

8.2 Administering System Pages

In KnowledgeWise, a system page is a predefined page that enables project members to perform a specific task—such as collecting, managing, and tracking project data.

Each system pages features one or more system controls. A system control is a system-defined control that represents a predefined system field and features control-specific functionality. System fields represent core object properties—such as workflow states and object statuses—and the data tracked in these fields is frequently forms the foundation of TechExcel businesses rules.

Like custom pages, system pages may be added to function pages to build the multiple-page forms that enable project members to manage knowledge items in project view.

Knowledge folder system pages enable a user to manage the knowledge folders in the knowledge tree panel of the knowledge view.

8.3 Administering Custom Pages

In KnowledgeWise, a custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data.

Each custom page may display system controls and custom controls. Using custom pages, system controls, and custom controls, a business may create forms that are uniquely designed to manage and track data that is specific to their environment and business.

Figure 7-2: Custom Page

Like system pages, custom pages may be added to function pages to build the multiple-page forms that enable project members to manage knowledge items in project views."

Project administrators may add and remove system, custom, and multiple-selection controls to custom pages, customize the layout of controls in the page, and define their the tab order.

8.3.1 Creating Custom Pages

A custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data.

To create a custom page:

1 Right-click a Custom Pages folder and select the Add New Page command in the shortcut menu.

Custom pages may be created for knowledge folders or knowledge items.

The custom page is displayed in the workspace.

2 Input the title of the custom page in the Page Display Name text box.

The display name defines the text that identifies the tabbed page in the client. By default, the custom page is given the title New Page.

3 Add custom controls or system controls.

4 Define the tab order of the controls displayed in the custom page.

8.3.2 Adding Controls to Custom Pages

To add a control to a custom page:

1 Right-click in the custom page and select the Add Control command in the shortcut menu.

The Add New Field manager appears. The Add New Field manager displays every custom controls and system controls that may be added to the custom page.

2 Optional: To filter controls by control type, select an option from the Field Type dropdown list.

The Field Type dropdown list displays multiple control types including the dropdown list control type, short text box control type, and the field label control type. KnowledgeWise enables administrators to define controls based on 17 different control types.

3 Optional: To define how the control is presented in the page, select an option from the Label Alignment dropdown list.

The Label Alignment dropdown list displays three options:

- Right Aligned
- Left Aligned
- Center Aligned

4 Optional: To define control as read only in the page, select the Add It as Read Only check box.

A read-only control is a control that is displayed, but cannot be edited.

5Click the OK button.

8.3.3 Removing Controls From Custom Pages

To remove a control to a custom page:

1Select a control in the custom page.

2Right-click and select the Delete Control command in the shortcut menu.

Note: The control is not deleted, but simply removed from the page. The control may subsequently be added to the page or another page.

A confirmation dialog box appears.

3Click the Yes button.

8.3.4 Ordering Controls in Custom Pages

The control with the lowest tab order number is the control that is immediately active when the page is accessed in the client.

To define the tab order of controls in a custom page:

1Select a custom page in a Custom Pages folder.

2Click the Define Field Tab Order button in the custom page.

3The Field Tab Order dialog box appears.

The Field Tab Order dialog box displays a list of the custom controls and system controls that have been added to the custom page.

4Define the tab order of each control in the page relative to the other controls displayed in the page.

- To move the control up the list, select the control and click the Up arrow. The lower the tab order number, the earlier the control may be accessed.
- To move the control down the list, select the control and click the Down arrow. The greater the tab order number, the later the control may be accessed.

5Click the OK button.

8.3.5 Adding Static Labels

Project administrators may use the Add Static Label command to add a static label to custom pages.

Static labels enable administrators to add descriptive or instructive text to custom pages.

To add static labels:

1Select the custom page in the tree panel.

2Right-click the custom control and select the Add Static Label command n the shortcut menu.

The Add New Static Label dialog box appears.

3Enter text in the Label Text box.

4If you want the a group box to contain the static label, select the Add a Group Box check box.

5If you want the text to appear in boldface, select the Bold Font check box.

6Click the OK button.

8.3.6 Understanding Custom Page Form Design Commands

The Form Design tools enable administrators to quickly create forms that are aesthetically pleasing and useful.

- Project administrators may align two or more form elements with one another along a left, right, top or bottom margin.
- Project administrators may resize two or more form elements so that they share the same width and height.

Effective form pages are easy to understand layout. They should be intuitive and easy to follow. One way to give your custom forms a clean, professional appearance is to ensure that all of the labels and controls on the page are uniformly sized and aligned.

To use the Form Design tools administrators must select two or more form elements. DevSuite Admin applies the style of the upper form element (the form element closest to the top of the page) to the lower form element (the one closest to the bottom of the page).

Left Align Icon	Aligns two or more form elements along the left edge.
Right Align icon	Aligns two or more form elements along the right edge.
Bottom Align icon	Aligns two or more form elements along the bottom edge.
Top Align icon	Aligns two or more form elements along the top edge.
Resize Width icon	Resizes two or more form elements to the same width.
Resize Height icon	Resizes two or more form elements to the same height.
Resize Height-Width icon	Resizes two or more form elements to the same height and width.

Table 6-1: Alignment Command Icons

8.3.7 Aligning Custom Controls and Labels

Project administrators may use form design tools to customize the layout of form elements (controls and labels) in custom pages

To add form elements to custom pages:

1. Select the form elements you want to align in the custom page.

   When selected a white border appears around the form element. The Alignment tools appear at the bottom of the Form window only when two or more form elements are selected.

2. Select an Alignment tool.

   - To right align the form elements select the Right Align icon
   - To left align the form elements select the Left Align icon.
   - To align the form elements at the top select the Top Align icon.
   - To align the form elements a the bottom select the Bottom Align icon.

   The form elements are immediately aligned.

8.3.8 Resizing Controls and Labels

Project administrators may use the Horizontal, Vertical, and Horizontal-Vertical alignment tools to resize two or more form elements (controls and labels).

To resize controls and labels:

1. Select the form elements you want to align in the custom page.

   When selected a white border appears around the form element. The Alignment tools appear at the bottom of the Form window only when two or more form elements are selected.

2. Select an Alignment tool.

   - To make the form elements the same width, select the Horizontal icon.
   - To make the form elements the same height, select the Vertical icon.
   - To make the form elements the same height and width, select the Horizontal- Vertical icon.

   The form elements immediately resized.

8.4 Administering System Controls

A control is a graphic element displayed in a window, panel, or dialog box that enables the user to define, update, and track data. Each control represents a property—system field or custom field—of a business object. KnowledgeWise controls enable project members to manage folders and knowledge items.

A system control is a system-defined control that represents a predefined system field and features control-specific functionality. System fields represent core object properties—such as workflow states and object statuses—and the data tracked in these fields is frequently forms the foundation of TechExcel businesses rules.

Consequently, system controls are designed for a specific form of interaction and customizations to these controls is limited.

Every system field is defined by four properties: the Field ID, Field Name, Control Title, and Control Type.

Field ID: The Field ID is an identification number uniquely identifies the system field in the project.

Field Name: The Field Name is a unique title that identifies the system field in the project.

Control Title: The Control Title is a customizable field label that identifies the system control in the client GUI

Control Type: The Control Type defines the functionality of a control—its appearance and operation.

The field ID and field name of system fields are fixed in KnowledgeWise and cannot be changed. These properties enable administrators to identify fields across projects and facilitate integration and interproject configurations.

8.5 Administering Custom Fields and Controls

A control is a graphic element displayed in a window, panel, or dialog box that enables the user to define, update, and track data. Each control represents a property—system field or custom field—of a business object. In KnowledgeWise, controls enable project member to manage knowledge items.

Custom controls are administrator-defined data-entry controls that may be added to system pages and custom pages. Project administrators may define appropriate field labels, control types, and predefined values for each control.

Data collected in custom fields may be imported or exported, searched, and used to run reports.

8.5.1 Understanding Control Types

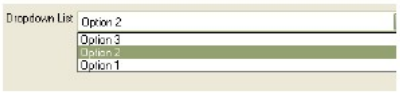
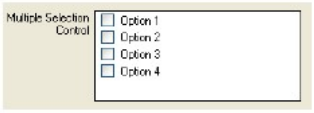
In KnowledgeWise, every control is defined by its control type and represents the property—system field or custom field—of a business object.

A control type defines the appearance and operation of a control and defines how the graphic element may be used to manage project data. Common control types include text boxes, dropdown lists, and check boxes.

All custom controls are fully customizable. The project administrator may define the title of the control. The number of customizations that may be made to a control depend on its control type.

KnowledgeWise supports 16 different control types: combo boxes, date-time controls, dropdown lists, edit boxes, multiple selection controls, and multiple-line edit boxes.

Autogrow Combo Box	An autogrow combo box combines a dropdown list with a short text box to enable users to choose between inputting a value directly into the control or selecting a value from an existing list of options. Every value entered into the text box is added to the list of options that are available to subsequent users. Compare with Combo Box.	
Calculation Field	A calculation control is a read-only control that displays a *calculation value*—the sum, difference, product, or quotient of two or more factors based on an administrator-defined formula. The calculation value displayed in the calculation control is defined by the calculation factors, the calculation formula, and the calculation format	
Check Box	A check box enables the user to choose between two mutually exclusive options: selected (true) or clear (false).	

Combo Box	A combo box combines a dropdown list with a text box to enable users to choose between inputting a value directly into the control or selecting a value from an existing list of options. Values entered into the text box are *not* added to the list of options that are available to subsequent users. Compare with Autogrow Combo Box.	
Composed Field	A composed field is a read-only control that may display a combination of text and field values from one or more linked system fields. The field values displayed in the control are determined dynamically in the client.	
Date-Time Control	A date-time control enables the user to select a specific date in a popup calendar.	
Decimal Edit Box		
Definable Combo Box		
Dropdown List	A dropdown list enables the user to select a value from a list of predefined options.	
HTML Text Message		
Integer Edit Box		
Long (<=255) Text Box		
Multiple Selection Control	A multiple selection control enables the user to select multiple values in a predefined list of options.	
Multiple-line Edit Box	A multiple-line text box enables the user to input multiple-lines of text. In KnowledgeWise, the control may support HTML tags, time stamping, and spell checking.	
Short (<=50) Edit Box	A short text box enables the user to input a single line of text.	
Static Label	A static label is a read-only control that enables the administrator to provide supplementary instructions or information.	
Voting Grid	A voting grid tracks data in a tabular list of rows and columns and enables the user to enter integer, decimal, and currency values.	

8.5.2 Customizing Check Box Controls

A check box control enables the user to choose between two mutually exclusive options: selected (true) or clear (false).

The field label of the check box is customizable.

To customize a check box control:

1. Select a Field Design page in the DevSuite Admin client.
2. Create a custom control or edit an existing control.
   - To create a new custom control, click the New button.
   - To edit an existing custom control, select the control in the Custom Fields list and click the Edit button.

   The Custom Control dialog box appears.
3. Select the Check Box option from the Control Type dropdown list.
4. Define the title of the control in the Field Label text box.
5. Click the OK button.

   The Custom Control dialog box closes and the custom control is displayed in the Custom Fields list.

8.5.3 Customizing Date-Time Controls

A date-time control enables the user to select a date (month, date, and year) and time in a popup calendar.

Figure 7-3: Date-Time Control in the KnowledgeWise Client

In its inactive state the control is displayed as a text box with a small button (the ellipsis button)." Clicking the ellipsis button displays the popup calendar that enables the user to browse and select the date.

To customize a date-time control:

1. Select a Field Design page in the DevSuite Admin client.
2. Create a custom control or edit an existing control.
   - To create a new custom control, click the New button.
   - To edit an existing custom control, select the control in the Custom Fields list.and click the Edit button.

   The Custom Control dialog box appears.
3. Select the Date-Time option in the Control Type dropdown list.
4. Define the title of the control in the Field Label text box.
5. Define the scope of the control. Date-Time controls may track dates or dates and times.
   - To enable the control to track the date only, select the Date option button.
   - To enable the control to track the date and time, select the Date and Time option button.
6. Click the OK button.

   The Custom Control dialog box closes and the custom control is displayed in the Custom Fields list.

8.5.4 Customizing Short Text Box Controls

A short text box enables a user to input a single line of text.

The field label of the single line text box is customizable.

To customize short text box control:

1. Select a Field Design page in the DevSuite Admin client.
2. Create a custom control or edit an existing control.

- To create a new custom control, click the New button.
- To edit an existing custom control, select the control in the Custom Fields list.and click the Edit button.

The Custom Control dialog box appears.

3 Select the Single (<=50) Text (Edit Box) option from the Control Type dropdown list.

4 Define the title of the control in the Field Label text box.

5 Click the OK button.

The Custom Control dialog box closes and the custom control is displayed in the Custom Fields list.

8.5.5 Customizing Multiple-line Text Box Controls

A multiple-line text box enables a user to input multiple-lines of text.

Multiple-line text box controls, also called *memo fields*, are data-entry controls that enable organizations to tracked detailed discussions about project knowledge.

Multiple-line text box controls support HTML tags, time stamping, and spell checking.

To define a multiple-line text box:

1 Select a Field Design page in the DevSuite Admin client.

2 Create a custom control or edit an existing control.

- To create a new custom control, click the New button.
- To edit an existing custom control, select the control in the Custom Fields list.and click the Edit button.

The Custom Control dialog box appears.

3 Select the Multiple-line Edit Box option from the Control Type dropdown list.

4 Define the title of the control in the Field Label text box.

5 Optional: To enable spell checking in the custom control, select the Enable Spell Checking check box.

6 Optional: To enable HTML support in the custom control, select the Support HTML Format check box.

7 Optional: To enable time stamping in the custom control, select the Time-stamped check box.

A time stamp is a signature that is concatenated to text entered in the data-entry field. KnowledgeWise time stamps log the exact date and time of the entry, the name of the project member entering the data, and the action that accompanied the work description entry.

8 Click the OK button.

The Custom Control dialog box closes and the custom control is displayed in the Custom Fields list.

8.5.6 Customizing Composed Field Controls

A composed field is a read-only control that displays text and field values drawn from one or more linked system fields. The field values displayed in the control are determined dynamically in the client.

A composed field control may be used to create custom IDs composed of multiple values. For example, Template ID values, Test Task ID values, and the Task Owner ID values managed in system controls may be displayed in a single customized composed field control.

All composed fields are custom-defined. KnowledgeWise offers no system-defined composed fields.

To create composed field custom controls:

1 Select a Field Design page in the DevSuite Admin client.

2 Create a custom control or edit an existing control.

- To create a new custom control, click the New button.
- To edit an existing custom control, select the control in the Custom Fields list.and click the Edit button.

The Custom Control dialog box appears.

3 Select the Composed Field option from the Control Type dropdown list.

4 Enter a title for the custom control in the Field Label text box.

The Field Content text box displays the content of the composed field custom control

5 Optional: To add text to the custom composed field, enter text in the Field Content text box.

6 Optional: To add system fields to the Field Content text box, select the Add Field Content button.

The Add a Field for Composed Field dialog box appears.

7 Select a system field from the System Field list.

8 Click the OK button.

The system field appears in the Field Content text box.

9 Click the Finish button.

8.6 Customizing Selection and Combination Box Controls

A selection control is a graphic element that enables the user to choose an option from a predefined list of options. The options displayed in a selection control may be defined by the administrator (definable list) or drawn from an existing field (lookup field).

KnowledgeWise features four selection control types: autogrow combo boxes, combo boxes, dropdown lists, and multiple selection controls.

Autogrow Combo Box: An autogrow combo box combines a dropdown list with a text box to enable users to choose between inputting a value directly into the control or selecting a value from an existing list of options.

Combo Box: A combo box combines a dropdown list with a text box to enable users to choose between inputting a value directly into the control or selecting a value from an existing list of options.

Dropdown List: A dropdown list enables the user to select a value from a list of predefined options.

Multiple Selection List: A multiple selection control is GUI element control type that enables the user to select multiple values from a predefined list of options.

The autogrow combo box nd definable combo box control types are combination controls that enable the user to choose between selecting a predefined option and inputting a value into the text box.

In KnowledgeWise, parent-child relationships may be defined between two selection controls. In a parent-child relationship, the option selected in the parent selection control determines the options that are available for selection in the child selection control.

8.6.1 Defining Selection List Options (Manually)

To manually define a selection list option:

1 Create or edit a customized selection list control.

Selection controls include autogrow combo boxes, combo boxes, definable combo boxes, dropdown lists, and multiple selection controls.

2 Click the Next button in the Custom Control wizard.

The Define Selection List page appears in the Custom Control wizard.

3 Select the User-defined Selection List option button.

The options displayed in a selection list may be manually defined by the user or drawn from an existing lookup field. For step-by-step instructions see See “Defining Selection List Options (From Lookup Fields)”.

4 Click the Next button in the Custom Control wizard.

The Define Parent-Child Relationships page appears in the Custom Control wizard.

5 Define selection list options in the List Items area.

- To add an option, click the New button and define the name of the option in the dialog box.
- To edit an option, select the option in the Option list, click the Edit button and define the name of the option in the dialog box.

6 Click the Finish button.

The Custom Control wizard closes and the custom control is displayed in the Custom Fields list.

8.6.2 Defining Selection List Options (From Lookup Fields)

The Lookup Field Project dropdown list displays all the KnowledgeWise projects in the current site, the {Standard Lookup -Standard} option, and the {Standard Lookup - KnowledgeWise} option,

To link selection list to a lookup field:

1 Create or edit a customized selection list control.

Selection controls include autogrow combo boxes, combo boxes, definable combo boxes, dropdown lists, and multiple selection controls.

2 Click the Next button in the Custom Control wizard.

The Define Selection List page appears in the Custom Control wizard.

3 Select the Link Selection List option button.

The options displayed in a selection list may be drawn from an existing lookup field or manually defined by the user or. For step-by-step instructions see See “Defining Selection List Options (Manually)”.

4 Select the source of the lookup field in the Lookup Field Project dropdown list.

The Lookup Field Project dropdown list displays all the KnowledgeWise projects in the current site, the{Standard Lookup -Standard} option, and the {Standard Lookup - KnowledgeWise} option,

5 Select the lookup field in the Lookup Field dropdown list.

The Option list displays the selection list in the options.

6 Click the Next button in the Custom Control wizard.

The Define Parent-Child Relationships page appears in the Custom Control wizard.

7 Select an option in the Parent {Master} dropdown list.

The Parent {Master} dropdown list displays the lookup field selected in the previous page of the wizard.

8 Select an option in the Parent Selection dropdown list.

The Parent Selection dropdown list displays all of the options that are displayed in the selection list of the parent master. The option selected is the*active parent option*.

9 Select one or more options in the Available Choices list.

10 Click the Finish button.

The Custom Control wizard closes and the custom control is displayed in the Custom Fields list.

8.6.3 Defining Parent-Child Relationships Between Controls

In a parent-child relationship, the option selected in the parent control determines the options that are displayed in the child control.

In KnowledgeWise, parent-child relationships may be defined between any two customized selection controls including autogrow combo boxes, combo boxes, definable combo boxes, dropdown lists, and multiple selection controls.

To define a parent-child relationship between two controls:

1 Select a control from the Parent Control dropdown list in the Custom Control wizard.

The Parent Control dropdown list displays every customized selection controls that has been defined in the project. A selection control may belong to the autogrow combo box, combo box, definable combo box, dropdown list, and multiple selection list control types.

2 Select an option in the Parent Selection dropdown list.

The Parent Selection dropdown list displays all of the options that are displayed in the selection list of the parent control. The option selected is the*active parent option*.

3 Define the options to be displayed in the selection list of the child control when the*active parent option*is selected in the selection list of the parent control.

- To manually define a child selection list options, click the New button and define the name of the option in the dialog box.
- To import child selection list options, click the Import button and define the source of the list options—a lookup field or structured text file.

8.6.4 Customizing Autogrow Combo Box Controls

An autogrow combo box combines a dropdown list with a text box to enable the user to choose between typing a value directly into the text box or to select a value from an existing list of options.

Every value entered into the text box is added to the dropdown list and displayed to subsequent users.

Figure 7-4: Autogrow Combo Box control

No system fields may be defined using the autogrow combo box control type." All autogrow combo boxes are custom controls.

All selection controls—dropdown lists, autogrow combo boxes, combo boxes, and multiple selection controls—support parent-child relationships. In a parent-child relationship, the option selected in the parent control determines the options that are displayed in the child control. For more information see See “Defining Parent-Child Relationships Between Controls”.

To customize an autogrow combo box control:

1 Select a Field Design page in the DevSuite Admin client.

2 Create a custom control or edit an existing control.

- To create a new custom control, click the New button.
- To edit an existing custom control, select the control in the Custom Fields list.and click the Edit button.

The Custom Control wizard appears.

3 Select the Auto-Grow Combo Box option in the Control Type dropdown list.

4 Define the title of the control in the Field Label text box.

5 Click the Next button.

6 Define the source of the options displayed in the autogrow combo box selection list:

- To define the options displayed in the list, select the User-defined Selection List option button and define the options in the next page of the Custom Control wizard.
- To pull the selection list options from a lookup field, select the Link Selection List option button and define the source field.

7 Click the Next button.

8 Optional: Define a parent-child relationship between the control and an existing control.

In a parent-child relationship, the option selected in the parent control determines the options that are displayed in the child control.

9 Optional: Define selection list options.

10 Click the Finish button.

The Custom Control wizard closes and the custom control is displayed in the Custom Fields list.

8.6.5 Customizing Combo Box Controls

A combo box combines a dropdown list with a short text box to enable the user to choose between inputting a value directly into the control or to select a value from an existing list of options.

Figure 7-5: Combo Box Control

Unlike the autogrow combo box control type, values entered into the text box of a combo box control are not added to the selection list displayed to subsequent users."

All selection controls—dropdown lists, autogrow combo boxes, combo boxes, and multiple selection controls—support parent-child relationships. In a parent-child relationship, the option selected in the parent control determines the options that are displayed in the child control. For more information see See “Defining Parent-Child Relationships Between Controls”.

To add or remove options to selection controls:

1 Select a selection control in the System Fields list of the System Field Design page in the Field Design subfolder of the Incident View folder.

2 Click the Design button.

The KnowledgeWise dialog box appears.

3 Click the Setup button.

The List Setup dialog box appears.

4 Add or remove field values to the selection control.

- To add field values to the selection control, select the field value in the Inactive Selection list and click the Left Arrow button.
- To remove field values from the selection control, select the field value in the Active Selection list and click the Right Arrow button.

5 Click the OK button.

The List Setup dialog box closes.

6 Click the OK button.

The KnowledgeWise dialog box closes.

8.6.6 Customizing Dropdown List Controls

A dropdown list enables the user to select a value from a list of predefined options.

Figure 7-6: Drodown List Control

All selection controls—dropdown lists, autogrow combo boxes, combo boxes, and multiple selection controls—support parent-child relationships." In a parent-child relationship, the option selected in the parent control determines the options that are displayed in the child control. For more information see See “Defining Parent-Child Relationships Between Controls”.

To add or remove options to selection controls:

1 Select a selection control in the System Fields list of the System Field Design page in the Field Design subfolder of the Incident View folder.

2 Click the Design button.

The KnowledgeWise dialog box appears.

3 Click the Setup button.

The List Setup dialog box appears.

4Add or remove field values to the selection control.

- To add field values to the selection control, select the field value in the Inactive Selection list and click the Left Arrow button.
- To remove field values from the selection control, select the field value in the Active Selection list and click the Right Arrow button.

5Click the OK button.

The List Setup dialog box closes.

6Click the OK button.

The KnowledgeWise dialog box closes.

8.6.7 Customizing Multiple Selection Controls

A multiple selection control is GUI element control type that enables the user to select multiple values from a predefined list of options.

The multiple selection control enables organizations to track information about a work item where the object is defined by multiple properties that are not exclusive.

Figure 7-7: Multiple Selection Control

All selection controls—dropdown lists, autogrow combo boxes, combo boxes, and multiple selection controls—support parent-child relationships." In a parent-child relationship, the option selected in the parent control determines the options that are displayed in the child control. For more information see See “Defining Parent-Child Relationships Between Controls”.

To create a multiple selection control:

1Click the Add button in the Multiple Selection Control page in the Field Design subfolder of the Incident View folder.

The Add Multiple Selection Control dialog box appears.

2Input the title of the control in the Display Name text box.

3Click the OK button.

The Add Multiple Selection Control dialog box closes.

The multiple selection control is displayed in the Multiple Selection Control list.

8.6.8 New Section

8.7 Administering Calculation Fields and Controls

In KnowledgeWise, a calculation control is a read-only control that displays a*calculation value*—the sum, difference, product, or quotient of two or more factors based on an administrator-defined formula.

KnowledgeWisesupports three types of calculation controls based on the factors thatcontribute to the calculation value: numerical calculation controls,date calculation controls, selection calculation controls.

Numerical Calculation Control:A numerical calculation control displays a*calculation value*—the sum, difference, product, or quotient of two or more factors based on an administrator-defined formula.

Date Calculation Control: Adate calculation control displays the difference—in hours, days, orweeks— between the date or date-time values tracked in two Date-Timecontrols or between a single Date-Time control and the{Now} system variable.

Selection Calculation Control:A numerical calculation control displays a*calculation value*—the sum, difference, product, or quotient of t factors dynamically drawn from other controls.

Selectioncalculation controls are particularly useful to organizations that needto knowledge items based on a combination of factors.

Allcustom calculation controls are read-only in the KnowledgeWise client.Project team members may view, but not change the calculation valuedisplayed

8.7.1 Defining Numerical Calculation Controls

Thecalculation value displayed in the calculation control is defined bythe calculation factors, the calculation formula, and the calculationformat.

Factors: The calculation factors

The custom calculation control supports factor field types:

• Numerical field type

• Selection field type

• Date field type

Formula: Thecalculation formula defines how the calculation factors are added,subtracted, multiplied, or divided to determine the calculation value.

Administrators may define the formula using four operators:

• Addition (+)

• Subtraction (-)

• Multiplication (*)

• Division (/)

Format: The format defines the appearance of the calculation value in the custom calculation control.

Administrators may choose between two options:

• Whole Number

• Precision option.

To define a custom calculation control:

1Select a Field Design page in the DevSuite Admin client.

2Create a custom control or edit an existing control.

• To create a new custom control, click the New button.

• To edit an existing custom control, select the control in the Custom Fields list.and click the Edit button.

The Custom Control dialog box appears.

3Select the Calculation Field option from the Control Type dropdown list.

4Define the title of the calculation field in the Field Label text box.

5Click the Add button.

The Add New Factor dialog box appears.

6Define the calculation factor field type.

Everyfactor is defined its field type. The calculation field supports threefactor field types: numerical field type, selection field type, and thedate field type.

•To define numerical calculation factors, select the Numerical optionbutton. For more information see See “Defining Numerical CalculationControls”.

•To define selection calculation factors, select the Date option button.For more information see See “Defining Date Calculation Controls”.

•To define date calculation factors, select the Selection option button.For more information see “Defining Selection Calculation Factors”

Once defined, the calculation factor is displayed in the Factor list and assigned a alphabetical character.

7Define the calculation formula.

Thecalculation formula defines how the calculation factors are added,subtracted, multiplied, or divided to determine the calculation value.

The calculation field supports four operators:

+ The addition sign adds two factors together.

- The subtraction sign subtracts the factor on the right from the one on the left.

* The multiplication sign multiplies two factors together.

/ The division sign divides the factor on the left by the factor on the left.

8Define the calculation format.

Numerical and selection calculation values may be formatted.

• To round up the calculation value, select the Whole Number option button.

•To define the number of decimal points displayed in the calculationvalue, select the Precision option button and enter an integer in thetext box.

9Click the Finish button.

The Custom Control dialog box closes and the custom control is displayed in the Custom Fields list.

8.7.2 Defining Date Calculation Controls

InKnowledgeWise, a date calculation control displays the difference—inhours, days, or weeks— between the date or date-time values tracked intwo Date-Time controls or between a single Date-Time control and the{Now} system variable.

Adate calculation control is a custom control based on the calculationfield control type in which the calculation factor is defined by thedate field type. The calculation field control type supports threefield types: the numerical field type, selection field type, and thedate field type.

To define date calculation factors:

1Select a Field Design page in the DevSuite Admin client.

2Create a custom control or edit an existing control.

- To create a new custom control, click the New button.
- To edit an existing custom control, select the control in the Custom Fields list.and click the Edit button.

The Custom Control dialog box appears.

3Select the Calculation Field option from the Control Type dropdown list.

4Define the title of the calculation field in the Field Label text box.

5Click the Add button.

The Add New Factor dialog box appears.

6Select the Date option button. Every factor is defined its field type.

The calculation field supports three factor field types: numerical field type, selection field type, and the date field type.

7Define the calculation formula (minuend - subtrahend = difference).

The dropdown lists display the name of every administrator-defined date-time control and the{Now} system variable.

- To define the minuend, select an option from the dropdown list on theleft. The minuend is the number from which another number is to besubtracted.
- To define the subtrahend, select an option from the dropdown list onthe right. The subtrahend is the number that is to be deducted fromanother number.

8Select a time unit from the Time Unit dropdown list.

The option selected defines the time units displayed in the calculation field. The Time dropdown list displays three options:

- Hours
- Days
- Weeks

9Click the Finish button.

The Add a New Factor dialog box closes.

8.7.3 Defining Selection Calculation Factors

In KnowledgeWise, a selection calculation control displays a*calculation value*—the sum, difference, product, or quotient of two or more factors based on an administrator-defined formula.

Selectioncalculation controls are particularly useful to organizations that needto prioritize items—requirements, specifications, and changerequests—based on a combination of factors.

To customize selection calculation controls:

1Select a Field Design page in the DevSuite Admin client.

2Create a custom control or edit an existing control.

- To create a new custom control, click the New button.
- To edit an existing custom control, select the control in the Custom Fields list.and click the Edit button.

The Custom Control dialog box appears.

3Select the Calculation Field option from the Control Type dropdown list.

4Define the title of the calculation field in the Field Label text box.

5Click the Add button.

The Add New Factor dialog box appears.

6Select the Selection option button.

Everyfactor is defined its field type. The calculation field supports threefactor field types: numerical field type, selection field type, and thedate field type.

7 Select an option from the Selection dropdown list.

The Selection dropdown list displays multiple system controls.

8 Select an option in the Field Choice area.

The Field Choice list displays the field values that define the selected system field.

9 Define a numerical value for the selected field value.

10 Click the OK button.

The Add a New Factor dialog box closes.

Chapter 9 Workflow Administration

9 CHAPTER 8 Workflow Administration

In this chapter:

- Administering Knowledge Item Workflow
- AdministeringKnowledgeItemStateGroups

9.1 Administering Knowledge Item Workflow

Workflow is a business process that may be used to manage the tasks that compose a project. Workflow rules enable project managers to define and track the flow of work between individuals and teams. A well-defined set of workflow rules enables an organization to perform tasks in an efficient and repeatable manner.

In KnowledgeWise, workflow defines the life cycle of the knowledge items managed in the project. The knowledge item lifecycle is comprised of multiple*workflow states*— each workflow state represents a specific stage in the knowledge item lifecycle and defines rules for managing the knowledge item at that stage.

For example, project workflow might consists of six workflow states: {New}, Proposed, Approved, Draft, Finalized, and In Design.

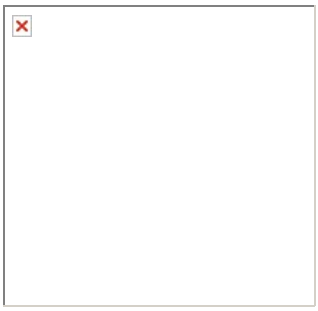


Figure 8-1: Knowledge Item Workflow Transitions

Each node represents a knowledge item workflow state." The arrows between each node represent the transition between those workflow states. The closed states are darkened.

KnowledgeWise workflow is completely customizable. The lifecycle of project knowledge may consists of as many workflow states as are needed to represent the needs of a business.

Every workflow state is as an*open workflow state*or a*closed workflow state*. The status assigned to a workflow state determines how the knowledge item is managed in that state. The knowledge item life cycle consists of at least two workflow states: an open workflow state and an closed workflow state. By default, all new knowledge items start in theNew workflow state, an open state.

Workflow administration is the task of defining workflow states and the transitions between those states, defining workflow-based access controls, and knowledge-item versioning rules.

9.1.1 Enabling Support for Knowledge Workflow Rules

For every workflow state, the project administrator may defines rules for managing the knowledge item at that stage in the knowledge lifecycle.

All KnowledgeWise workflow rules are optional and must be enabled by the project administrator before they may defined.

Optional workflow rules include support state-to-state transitions, applicable owner rules, state-based invisible, read-only and mandatory fields, automatic versioning, and a choice between state-based or transition-based access controls:

Using controls in the Workflow Option page, project administrators may enable KnowledgeWise workflow rules in a project.

To enable workflow access control support:

1 Select Workflow > Workflow Options in the tree panel.

2 Optional: To enable support for state-to-state transitions, select the State-to-State check box. State-to-state transition rules define all possible transitions between workflow states. A knowledge item may pass from one workflow state to next only if a transition exists between those to workflow states.

3 Optional: To enable support for state-based applicable owner rules, select the Enable Applicable Owners check box. Applicable Owners rules define which project members may own a knowledge item in each workflow state based on their account type, team group, group folder, system variable, or user account.

4 Optional: To enable support for field-level access controls, select check box next to the appropriate field access control. KnowledgeWise supports workflow-based invisible fields, read-only fields, and mandatory fields.

- Read-only Fields
- Invisible Fields
- Mandatory Fields

5 Optional: To define state-based or transition-based access controls, select an option in the Who Can Change State dropdown list.

- Transition-based access controls
- State-based access controls

WhoCanChangeStateaccess controls must be defined if the state-to-state workflow rule is enabled in the project.

6 Optional: To enable support for automatic versioning, select the Enable Automatic Version Increase check box and one or more knowledge item fields in the Version Increase list. Automatic Version Increase rules define which knowledge item properties may be automatically updated during knowledge item workflow.

9.1.2 Defining Workflow States

In KnowledgeWise, workflow defines the lifecycle of the knowledge items managed in the project The knowledge item lifecycle is comprised of multiple, administrator-defined*workflow states.*A workflow state represents a stage in the lifecycle of a knowledge item—from its initial creation to its closure and one or more intermediary steps.

In KnowledgeWise, every workflow state is defined its name, its status (open or closed), and the*editabilty*of knowledge items in that state—workflow states may be defined as*read-only state*s.

To add a workflow state:

1 Select Workflow > Workflow Options page in the tree panel.

2 Click the New button. TheAddNewStatedialog box appears.

3 Define the name of the workflow state in the State Name text box.

4 Define the status (open or closed) of the workflow state. Every workflow state is as an open workflow state or a closed workflow state. The status assigned to a workflow state determines how the knowledge item is managed in that state.

- To define the state as an open state, select the Open option button.
- To define the state as an open state, select the Closed option button.

5 Optional: To define the knowledge item as read-only in the workflow state, select the Read-Only check box. In read-only workflow states, knowledge item properties may be viewed, but not updated. Compare with read-only access controls, which enable the administrator to defined specific fields as invisible in each state.

For more information see “Defining State-Based Read-Only Knowledge Item Fields”

6 Click the OK button. TheAddNewStatedialog box closes.

To edit a workflow state:

To edit a workflow state, select a workflow state in the KnowledgeWise States list and click the Edit button. All workflow state properties are editable.

To delete a workflow state:

To delete a workflow state, select a workflow state in the KnowledgeWise States list and click the Delete button.

9.1.3 Defining Workflow Transitions

State transition rules enable organizations to control the knowledge item life cycle by defining possible workflow paths—called*transitions*. A knowledge item may pass from one workflow state to next only if a transition exists between those to workflow states.

Using controls in the State Transition tab, project administrators may create transitions between workflow states. If state-to-state transition rules are not enabled and defined, a knowledge item may freely pass between any two workflow states without restriction.

To define a workflow transition:

1 Select Workflow > Workflow Settings.

2 Select the State Transition tab.

The State Transition tab is only displayed in the Workflow Settings page if state-to-state transition rules are enabled in the project. For step-by-step instructions see See “Enabling Support for Knowledge Workflow Rules”.

3 Select a workflow state in the Knowledge Item States list.

The selected knowledge item workflow state is the beginning state. Transitions enable a knowledge item to pass from the selected state to an ending state, but not the other way around.

4 To define ending states, select one or more workflow states in theNextStatelist.

TheNextStatelist displays the name of every workflow state defined in the project. A check mark indicates that a transition exists between the beginning and ending workflow states.

9.1.4 Defining State-Based Access Controls

State-based access controls define which project members may forward knowledge issues on a state-by-state basis.A single state-based rule defines who can change the workflow status of knowledge items in a particular workflow state.

Using controls in the Who Can Change tab, project administrators may define workflow-based access controls for each workflow state.

Figure 7-2: Who Can Change Tab (State-Based)

Permissions may be granted to project team members based on account type or team group membership, their relationship with the knowledge item (as the owner or submitter), or by user account.

To define state-based access controls:

1 Select Workflow > Workflow Settings.

2 Select the Who Can Change tab. The state-based Who Can Change tab is displayed in the Workflow Settings page state-based access controls are enabled in the project. For step-by-step instructions see See “Enabling Support for Knowledge Workflow Rules”

3 Select a workflow state in the Knowledge Item States list.

4 Select one or more user accounts, account types, team groups, or user variables in the Who Can Change list. The Who Can Change list displays the name of all user accounts, account types, and team groups defined in the project as well as the {Submitter} and {Current Owner} user variables.

User Account: The selected project member may forward the knowledge item in its current workflow state to another workflow state.

Account Type: Every project member belonging to the selected account type may forward the knowledge item in its current workflow state to another workflow state.

Team Group: Every project member belonging to the selected team group may forward the knowledge item in its current workflow state to another workflow state.

{Submitter}The project member that initially submitted the knowledge item may forward the knowledge item in its current workflow state to another workflow state.

{Current Owner}: The project member that currently owns the knowledge item may forward the knowledge item in its current workflow state to another workflow state.

9.1.5 Defining Transition-Based Access Controls

Transition-based access controls define which project members may forward knowledge issues from one workflow state to another workflow state on a transition-by-transition basis.

Using controls in the Who Can Change tab, project administrators may define workflow-based access controls for each transition between a selected*beginning state*and every other workflow state connected to that state by a transition. Different access controls may be defined for each transition: beginning state-ending state pair.

Permissions may be granted to project team members based on account type or team group membership, their relationship with the knowledge item (as the owner or submitter), or by user account.

To define transition-based access controls:

1 Select Workflow > Workflow Settings.

2 Select the Who Can Change tab. The transition-based Who Can Change tab is displayed in the Workflow Settings page transition-based access controls are enabled in the project. For step-by-step instructions see See “Enabling Support for Knowledge Workflow Rules”.

3 Select a workflow state in the Knowledge Item States list. The selected workflow state is the beginning state.

4 Select an ending state in theNextStatelist. TheNextStatelist displays the name of every workflow state defined in the project. The selected workflow state is the ending state. Each beginning state-ending state pair represents a transition.

5 Select one or more user accounts, account types, team groups, or user variables in the Who Can Change list.

The Who Can Change list displays the name of all user accounts, account types, and team groups defined in the project as well as the{Submitter} and {Current Owner} user variables.

User Account: The selected project member may forward the knowledge item in its current workflow state to another workflow state.

Account Type: Every project member belonging to the selected account type may forward the knowledge item in its current workflow state to another workflow state.

Team Group: Every project member belonging to the selected team group may forward the knowledge item in its current workflow state to another workflow state.

{Submitter}The project member that initially submitted the knowledge item may forward the knowledge item in its current workflow state to another workflow state.

{Current Owner}: The project member that currently owns the knowledge item may forward the knowledge item in its current workflow state to another workflow state.

9.1.6 Defining State-Based Applicable Owners and Default Owners

In DevSuite, an applicable owner is a project team member or group folder that may own a work item (such as an issue, requirement, or knowledge item) in a particular workflow state.

State-based applicable owner rules restrict the ownership of knowledge items in each state based by user account, account type, team group, or user variable. Project administrators may optionally designate one project team member, group folder, or user variables as the*default owner*of knowledge items in each workflow state.

The default owner is merely the project team member or group folder that is the default option in the Owner dropdown list control in the client. The user need not update the field if the knowledge item is to be forwarded to the default owner.

To define applicable owners by workflow state:

1 Select Workflow > Workflow Settings.

2 Select the Applicable Owners tab. The Applicable Owners tab is displayed in the Workflow Settings page if applicable owner rules are enabled in the project. For step-by-step instructions see See “Enabling Support for Knowledge Workflow Rules”.

3 Select a workflow state in the Knowledge Item States list. The selected workflow state is the beginning state.

4 Select one or more user accounts, account types, team groups, or user variables in the Who Can Change list.
The Who Can Change list displays the name of all user accounts, account types, and team groups defined in the project as well as the{Submitter} and {Current Owner} user variables.

User Account: The selected project member is designated as an applicable owner of knowledge items in the workflow state.

Account Type: Every project member belonging to the selected account type is designated as an applicable owner of knowledge items in the workflow state.

Team Group:Every project member belonging to the selected team group may is designated as an applicable owner of knowledge items in the workflow state.

Group Folder: The selected group folder is designated as an applicable owner of knowledge items in the workflow state.

{Submitter}: The project member that initially submitted the knowledge item is designated as an applicable owner of knowledge items in the workflow state.

{Current Owner}: The project member that currently owns the knowledge item is designated as an applicable owner of knowledge items in the workflow state.

{Previous Owner}: Any project team member that owned the knowledge item prior to the current owner is designated as an applicable owner of knowledge items in the workflow state.

5 Optional: To define the default owner, select a user account, team group, or user variable in the Default Owner dropdown list. The Default Owner dropdown list displays only the names of the user accounts, team groups, and user variables that have been designated as applicable owners for issues in the current workflow state.

9.1.7 Defining Knowledge State Versioning Rules

In KnowledgeWise, project members may create and manage multiple versions of knowledge items. A version of a knowledge item is a snapshot of that knowledge item—its property definitions, file attachment, and linked knowledge—at a point in time.

Change management teams may roll back to previous versions of the software at any time. State-based knowledge item versioning rules determine whether new versions of knowledge items may be created in a knowledge item state and how those new versions are created and managed.

Each knowledge item state is defined by one of four knowledge item versioning rules:

Freely Change With Version Increase:The Editable Knowledge Item rule enables project members to freely update a knowledge item in the knowledge item state. New versions are allowed, but not required.

Version Increase Required:The Version Increase Required rule mandates that a new version of a knowledge item is created for project members to update a knowledge item in the knowledge item state.

Change Request Required: The Change Request Required rule mandates that a new version of a knowledge item may be created if a knowledge item change request is submitted in the knowledge item state.

No Version Increase Allowed:The No Versions Allowed rule prevents project members from creating new versions of a knowledge item in a knowledge item in the knowledge item state.

To define knowledge item automatic versioning rules:

1 Enable state-based applicable owner workflow rules. State-based knowledge item state applicable owner rules are an optional feature that may be enabled and defined in each KnowledgeWise project. For step-by-step instructions see See “Enabling Support for Knowledge Workflow Rules”.

2 Select a knowledge item workflow state in the Knowledge Item States list in the Knowledge Item Workflow Settings page of the Knowledge Item folder in the Workflow folder. TheKnowledgeItemStatelist displays all administrator-defined knowledge item states.

3 Select the Automatic Version tab.

4 Select a knowledge item versioning option:

- To enable project members to edit knowledge item properties without creating a new version of the knowledge item, select the Editable Knowledge Item option button.
- To mandate that project members create a new version of a knowledge item to edit knowledge item properties, select the Version Increase Required option button.
- To mandate that project members submit a change request to create a new version of a knowledge item, select the Change Request Required option button.
- To prevent project members from creating versions of knowledge item, select the No Versions Allowed option button.

9.1.8 Defining State-Based Read-Only Knowledge Item Fields

In KnowledgeWise, a read-only field is a data-entry control that is displayed in the KnowledgeWise client, but cannot be edited.

State-based read-only field rules define which knowledge item properties project members may view and update in each knowledge item workflow state. Knowledge Item controls defined as read-only may not be updated by any project member regardless of that user’s account type or team group as long as the knowledge item is in that knowledge item state.

Six system-defined knowledge item fields may be defined as read-only in each knowledge item state

- Product/Version
- File Attachment
- Description
- Knowledge Owner
- KnowledgeState
- Title

To define state-based read-only fields:

1 Enable read-only rules. Knowledge Item state applicable owner rules are an optional feature that may be enabled and defined in each KnowledgeWise project. For step-by-step instructions see See “Enabling Support for Knowledge Workflow Rules”

2 Select the Knowledge Item Workflow Settings page in the Knowledge Item folder of the Workflow folder.

3 Select a knowledge item workflow state in the Knowledge Item States list.

4 Select the Read-Only tab. The Applicable Owners tab is only displayed if the state-based read-only field rules are enabled.

5 To define a knowledge item state as read-only in the selected knowledge item state, select the check box next to the name of the knowledge item property in the Read-Only Field list. Six system-defined knowledge item fields may be defined as read-only in each knowledge item state

- Product/Version
- File Attachment
- Description
- Knowledge Owner
- KnowledgeState
- Title

9.1.9 Defining State-Based Mandatory Knowledge Item Fields

In KnowledgeWise, a mandatory field is a data-entry control that must be defined before a knowledge item may progress to the next stage in the knowledge item life cycle. State-based mandatory fields rules identify the knowledge item properties that must be defined in a knowledge item workflow state before that knowledge item may progress to another workflow state in the knowledge item life cycle.

Six system-defined knowledge item fields may be defined as mandatory in each knowledge item state

- Product/Version
- File Attachment
- Description
- Knowledge Owner
- KnowledgeState
- Title

To define state-based mandatory fields:

1 Enable mandatory rules. Knowledge Item state applicable owner rules are an optional feature that may be enabled and defined in each KnowledgeWise project. For step-by-step instructions see See “Enabling Support for Knowledge Workflow Rules”.

2 Select the Knowledge Item Workflow Settings page in the Knowledge Item folder of the Workflow folder.

3 Select a knowledge item workflow state in the Knowledge Item States list.

4 Select the Mandatory tab. The Applicable Owners tab is only displayed if the state-based mandatory field rules are enabled.

5 Optional: To define a knowledge item state as mandatory in the selected knowledge item state, select the check box next to the name of the knowledge item property in the Mandatory Field list.

Six system-defined knowledge item fields may be defined as mandatory in each knowledge item state

- Product/Version
- File Attachment
- Description
- Knowledge Owner
- KnowledgeState
- Title

9.1.10 Defining State-Based Invisible Knowledge Item Fields

In KnowledgeWise, an invisible field is a data-entry control that is hidden from certain project members based on their account type or team group.

State-based invisible field rules define which knowledge item properties are displayed in each knowledge item workflow state. Knowledge Item controls defined as invisible may not be viewed or updated any project member regardless of that user’s account type or team group as long as the knowledge item is in that knowledge item state.

Six system-defined knowledge item fields may be defined as invisible in each knowledge item state

- Product/Version
- File Attachment
- Description
- Knowledge Owner
- KnowledgeState
- Title

To define state-based invisible fields:

1 Enable invisible rules. Knowledge Item state applicable owner rules are an optional feature that may be enabled and defined in each KnowledgeWise project. For step-by-step instructions see See “Enabling Support for Knowledge Workflow Rules”.

2 Select the Knowledge Item Workflow Settings page in the Knowledge Item folder of the Workflow folder.

3 Select a knowledge item workflow state in the Knowledge Item States list.

4 Select the Invisible tab. The Applicable Owners tab is only displayed if the state-based invisible field rules are enabled.

5 To define a knowledge item state as invisible in the selected knowledge item state, select the check box next to the name of the knowledge item property in the Invisible

Field list.

Six system-defined knowledge item fields may be defined as invisible in each knowledge item state

- Product/Version
- File Attachment
- Description
- Knowledge Owner
- KnowledgeState
- Title

9.2 Administering Knowledge Item State Groups

In KnowledgeWise, a knowledge item state group is a set of knowledge item workflow states that have been grouped together for reporting purposes. In the current version of KnowledgeWise states are mutually exclusive and can only belong to one knowledge item state group.

This may not be true of subsequent versions of KnowledgeWise. Knowledge Item state group administrative tasks include:

- Enabling Knowledge Item State Group Support
- Creating NewKnowledgeItemStateGroups
- DefiningKnowledgeItemStateGroups

9.2.1 Enabling Knowledge Item State Group Support

To enable knowledge item state group support, select the Enable Supporting Knowledge Item State Group check box in the Knowledge Item State Group page in the Advanced Features folder.

9.2.2 Creating New Knowledge Item State Groups

To create a new knowledge item state group:

1 Click the New button in the Knowledge Item State Group page of the Advanced Features folder. The Add a New Item dialog box appears.

2 Input the title of the knowledge item state group in the Name text box.

3 Click the OK button. The Add a New Item dialog box closes. The new knowledge item state group appears in the Knowledge Item State Group list.

9.2.3 Defining Knowledge Item State Groups

To define a knowledge item state group:

1 Select a knowledge item state group in the Knowledge Item State Group list. of the Knowledge Item State Group page in the Advanced Features folder.

2 Select one or more knowledge item states in the Status Selection list.

Chapter 10 Event Administration

10 CHAPTER 9 Event Administration

In this chapter:

- Administering KnowledgeWise Events
- Administering Event Templates
- Administering Event Workflow
- Administering Event Access Controls

10.1 Administering KnowledgeWise Events

Throughout the project planning process, project managers may schedule and manage team meetings as events. An event is a management task that facilitates communication and collaboration between project stakeholders. Typical events include brainstorming sessions, team meetings, design reviews, management reviews, presentations, and product demos.

In KnowledgeWise, events are the vehicle that drive the planning process and ensure the quality of the end product. Regular team meetings scheduled within KnowledgeWise enable project managers ensure that all project team members are working together to meet project objectives.

Every KnowledgeWise event is based on an administrator-defined*event template*. An event template is a blueprint for creating a specific type of event—such as a meeting, presentation, or demo—that defines the business rules that determine how that event is managed in the project.

Event administration, therefore, is the largely a matter of creating event templates, defining the rules that manage events based on that template, and determining when events based on that event template may be created in the project.

10.2 Administering Event Templates

An event template defines the business rules that are used to manage the events based on that template including the applicable owners, account type-based access controls, and workflow settings.

Using controls in the Event Template page, project administrators may create and define multiple event templates.

For each event template created in the Event Template page, the project administrator must define appropriate properties, access controls, an event group, and event workflow rules.

Creating Event Templates

Each event template represents a specific type of event—such as a meeting, presentation, or demo—and defines who may own the event, who may attend the event, and how the event is managed in workflow.

To create an event template:

1. Click the New button in the Event Template page of the Event subfolder.

   The Add a New Event Template dialog box appears.

2. Define a unique, descriptive name for the event template in the Name text box control.

   The title of the event template is inherited by every event based on that template and is the default title for those event instances. For more information see “Defining Default Event Titles and Descriptions”.

3. Define a brief description of the event in the Description field.

   Event template descriptions enable project manager to describe the purpose and functionality of an event template.

4. Click the OK button. The Add a New Event Template dialog box closes.

   The event template is displayed in the Events folder of the Event Templates tree control.

Editing Event Templates

Project administrators may edit basic event template properties in the Properties tab of the Event Templates page in the Events folder.

The Event Type property cannot be updated.

Both event template properties are mandatory and can only be defined when the project administrator initially creates the event template.

Deleting Event Templates

Project administrators delete event templates in the Event Template page of the Event subfolder.

To delete event templates:

1. Select the Event Template page in the Admin Tree window.
2. Select an event template in the Event Templates list.
3. Click the Delete button in the Event Templates area.

   A confirmation dialog box appears.

4. Click the Yes button.

   The event template disappears from the Event Templates tree control.

10.2.1 Defining Default Event Titles and Descriptions

Every event is defined by a title and a description of the event. The title and description of events are tracked in two data-entry controls: the Title control and the Description control. Project administrators may define default event titles and event descriptions for each event template. Every event that is based on that event template inherits the predefined title and description.

To define default event titles and descriptions:

1. Select an event template in the Event Templates tree list control of the Event Templates page.
2. Define the title of the event in the Title control.

   Every event based on the event template is automatically given the title defined in the event template.

3. Define the description of the event in the Description control.

   Every event based on the event template automatically inherits the description defined in the event template.

4. Click the OK button.

10.2.2 Defining Default Event Workflow States

In the KnowledgeWise client, project members may track the progress of a event in the State dropdown list. Each event state represents a particular stage of the life cycle that defines an event from initial creation to its closure and one or more intermediary steps.

The default workflow state is the event state that automatically defines newly created events. Project administrators may define default event workflow states for each event

template

Every event that is based on that event template inherits all administrator-defined event settings including the default event state. Project administrators may customize the title of data-entry control, define a default state, and define the options that are available in the State dropdown list in each event template.

10.2.3 Defining Default Event Owners

In the KnowledgeWise client, project members may track the current owner of events in the Owner dropdown list control. The default event owner is the project member that is automatically assigned ownership of the event as soon as it is created.

Project administrators may define default event owners for each event template. Every event that is based on that event template inherits all administrator-defined event settings including the default event owner from that template.

To define a default event owner:

1. Select an event template in the Event Templates tree list control of the Event Templates page.
2. Double-click the Owner control in the Properties tab.

   The Event Templates dialog box appears.
3. Select an option from the Default Owner dropdown list.

   The Default Owner dropdown list display all project members and administrator-defined group folders, as well as the {Parent Issue Owner} and {Parent Issue Submitter} system variables.
4. Click the OK button.

   The Event Templates dialog box closes.

10.3 Administering Event Workflow

Workflow is a business process used to manage tasks within a project. Workflow rules enable project managers to define and track the flow of work between individuals and teams. KnowledgeWise event workflow enable administrators to define sophisticated workflow rules to manage and track issue subtasks. Event workflow defines the life cycle of a event.

Event workflow states represent the various stages of the event life cycle from its initial creation to its closure and one or more intermediary steps.

Event workflow enables organizations to independently manage the various subtasks that must be completed before a development issue can be forwarded from one issue workflow state to the next.

Event workflow rules enable project teams to manage and track development subtasks independently of one another, assign those tasks to different owners, and define separate schedules (start and due dates) for each event.

Like development issues, events are managed within a set of administrator-defined workflow rules. The life cycle of a event is composed of open and closed workflow states.

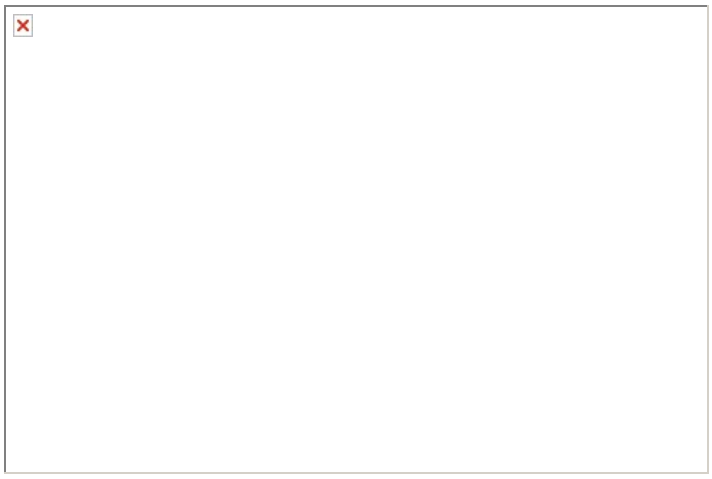


Figure 8-1: Event Workflow within Development Issue Workflow

The workflow of events is typically much simpler than that of their parent development issues."

For example, the life cycle of a event frequently consists of no more than two event workflow states: an open state and a closed state. In this scenario, the event merely tracks whether a specific subtask has been completed.

Alternatively, event workflow may consist of an opening state and multiple closed states to indicate the result of the subtask.

The event has three workflow states:

- To be tested (open)
- Test passed (closed)
- Test failed (closed)

The life cycle of a event may consist of multiple open and closed events." But no transition rules may be defined for event workflow. Transitions exists between every event workflow state in event workflow and consequently an event may be forwarded to any event state within event workflow.

All event workflow states are defined on a template-by-template basis." The event life cycle and event workflow rules used to manage and track a event instance are determined by event template used to create that event.

10.3.1 Creating Event States

Project administrators may define the event life cycle of an event template in the State tab of the Event Template page in the Events subfolder.

Project administrators may enable and define two levels of event workflow rules in KnowledgeWise projects: event workflow and advanced event workflow.

To create event states:

1Select an event template in the Template Tree of the Event Template page.

2Click the New button in the State tab.

TheEventStatedialog box appears.

3Define a unique name for the event state.

4Define the event state status.

5Every event state has one event state status:

- To define an event state as an open event, select the Open status option.
- To define an event state as a successfully closed event, select the Closed Successfully status option.
- To define an event state as a failed closed event, select the Closed Failed status option.

6Click the OK button.

10.3.2 Editing Event States

To edit event states:

1Select an event template in the Template Tree of the Event Template page.

2Select a event state in the State tab.

3Click the Edit button.

TheEventStatedialog box appears.

4Update the name for the event state.

5Define the event state status.

Every event state has one event state status:

- To define an event state as an open event, select the Open status option.
- To define an event state as a successfully closed event, select the Closed Successfully status option.
- To define an event state as a failed closed event, select the Closed Failed status option.

6Click the OK button.

10.3.3 Deleting Event States

To delete event states.:

1Select an event template in the Template Tree of the Event Template page.

2Select a event state in the State tab.

3Click the Delete button.

A confirmation dialog box appears.

4Click the Yes button.

10.4 Administering Event Access Controls

Event access controls define which project members may view, submit, edit, and delete events in a KnowledgeWise project. Access controls enable project administrators to permit or deny a project member to perform a certain task based on their role in the project.

The ability of project members to perform tasks in KnowledgeWise projects is usually determined by their*account type*, but the privilege to specific perform tasks may also be defined by team group membership or relationship between the project member and the event or issue in question.

All event access controls are defined on a template-by-template basis. Every event instance inherits the event access control settings from the event template that was used to create it.

Defining Event Access Controls

Event access controls determine which project members may manage events based two factors: the event template that was used to create the event and the role of that project member in the project.

All event access controls are defined on a template-by-template basis. Project administrators may define event access controls for every event template in the project. Event instances inherit access control settings from the template that was used to create them.

The foundation of all KnowledgeWise access controls are account types. The account type defines the role (responsibilities and privileges) that are granted to a project member in a project. Project administrators may grant event access controls to project members based on their account type, team group membership, or one of four system variables:

{Event Submitter}: The {Event Submitter} variable represents the project member that submitted the event.

{Event Owner}: The {Event Owner} variable represents the project member that currently owns the event.

Project administrators may define event access controls for each event template in the User Access tab of the Event Template page in the Events subfolder.

To define event access controls:

1Select an event template in the Event Templates list.

2Select the User Access tab.

3Select an access control option in the Applicable Account Types list of the User Access tab.

4Select an option in the in the Applicable Account Types list.

- The Select the All Account Types radio button enables administrators to select all of the account types displayed in the list.
- The Select the Selected Account Types option radio button enables administrators to select only some of the account types displayed in the list.

Understanding Event Access Controls

All event access controls are defined in the User Access tab of the Event Template page in the Events subfolder.

Each event template is defined by five event access controls. Project members may view, submit, edit, or delete events based on the privileges that are granted to their account type in the event template that was used to create the event.

Can Submit privilege: The Can Submit privilege enables project members to submit events based on an event template.

Can Edit privilege: The Can Edit Open Event privilege enables project members to edit closed events based on an event template.

Can Delete privilege: The Can Delete Event privilege enables project members to delete events based on an event template.

DevSpec Projects Administration

Chapter 1 What's new

1 Full Unicode Support

Wiki Summary.

1.1 New Section

Please input description.

Chapter 2 DevSpec Concepts

2 DevSpec Concepts

This chapter provides a broad overview of DevSuite features and modules describing how these tools may be used to enhance application life cycle management processes.

In this chapter:

- Understanding TechExcel DevSpec and DevSuite
- Understanding DevSuite Administration
- Understanding DevSuite Sample Projects and Templates

2.1 Understanding TechExcel DevSpec and DevSuite

TechExcel DevSuite is a family of integrated application lifecycle management (ALM) tools that place knowledge management —from ideas, to formal specifications, to competitive information to issue resolution and customer insight—at the core of any product development initiative.

The DevSuite knowledge-centric strategy enables improve communication, keep up-to-date on changes, and reduce the development cycles so that the business may deliver the right products for the right markets in the shortest possible time.

Figure 1-1: DevSuite Knowledge-Centric Development

DevSuite places knowledge management at the core of all development processes." TechExcel KnowledgeWise provides for the easy and efficient collection and organization of informal ideas, gathered from a wide variety of sources, that area shared across multiple DevSpec, DevTrack, and DevTest projects.

KnowledgeWise projects provide controlled access to documents, improve communication and coordination between distributed development teams, and facilitate the management and sharing of information between development teams and project stakeholders.

TechExcel DevSuite leverages intellectual assets with KnowledgeWise, communicating a clear product vision and tactical execution strategy by linking ideas and customer feedback, specifications, requirements, designs, prototypes, and other documents to specific areas of work.

2.1.1 Understanding DevSuite ALM Solutions

TechExcel DevSuite features five ALM solutions that operate in an n-tier architecture: TechExcel KnowledgeWise, TechExcel DevTrack, TechExcel DevPlan, TechExcel DevSpec, and TechExcel DevTest.

All DevSuite products share the same core architecture—the DevSuite Database Server, DevSuite Application Server, DevSuite Document Server, and DevSuite Web Services—and are fully integrated enabling every branch of the development organization to communicate and work together.

KnowledgeWise	TechExcel DevSuite leverages intellectual assets with KnowledgeWise, communicating a clear product vision and tactical execution strategy by linking ideas and customer feedback, specifications, requirements, designs, prototypes, and other documents to specific areas of work during a development project. Documents are shared with all resources involved in the execution of the project allowing for an uncompromised vision to direct the path of any development project.
DevPlan	TechExcel DevPlan manages the transformation of concepts into formal strategic plans. DevPlan offers an intuitive planning hierarchy to formalize scope and optimize resource usage, team-based planning and calendaring capabilities. These features enable complete control over all product development projects from design planning to implementation and enables increased team efficiency and collaboration.
DevSpec	TechExcel DevSpec is an integrated requirements management solution that is specifically designed to provide visibility, traceability and validation of your product or project requirements. DevSpec provides a frame work to create new requirements, specifications and features that can be linked to development and testing implementation projects.
DevTrack	TechExcel DevTrack enables development teams to manage every aspect of the development process including issue management, team management, and communications management.
DevTest	TechExcel DevTest is a test management solution that enables test organizations to manage every stage of testing life cycle—from test case design, to test execution, to test analysis. DevTest provides testing groups with the tools they need work more effectively and efficiently, hold down costs, and to deliver higher quality products. While each element is valuable even when functioning independently, the DevSuite architecture is founded upon the assumption of future scaling and is capable of advanced inter-module integration. Whether used as separate components or as an incorporated set of applications, the DevSuite solutions are ideal for companies at any stage in their product development.

2.1.2 Understanding TechExcel Project Hierarchy

At the core of every DevSuite implementation—a*DevSuite site*— is one or more KnowledgeWise knowledge management projects that organize, manage, and track the knowledge collected in multiple DevSpec, DevTrack, and DevTest projects.

At the core of every DevSuite implementation—a*DevSuite site*— is one or more KnowledgeWise knowledge management projects that organize, manage, and track the knowledge collected in multiple DevSpec, DevTrack, and DevTest projects.

All projects in a DevSuite site are organized into a hierarchical structure that defines the relationships between the projects in a DevSuite site. Every DevSpec or DevTrack project is the child of a parent KnowledgeWise knowledge management project and relies on that project to manage project intelligence.

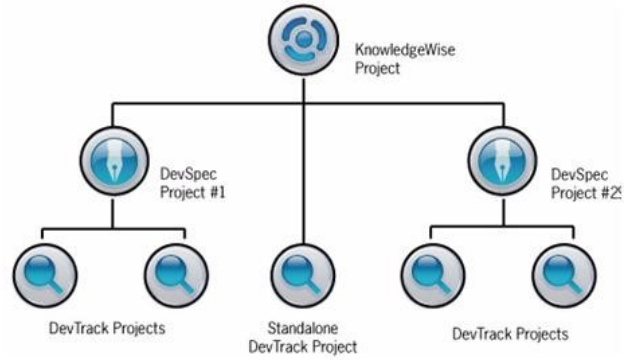


Figure 1-2: DevSuite Project Hierarchy

KnowledgeWise enables development organizations to build a knowledge base of shared knowledge in many different media. All knowledge items may be linked with different areas of work so that internal teams and external customers can search the knowledge base for self-service content based on account type-based privileges and access controls.

KnowledgeWise projects may be accessed by three methods: (1) the KnowledgeWise Web client, (2) the knowledge view of a child DevSpec project; (2) the knowledge view of a child DevTrack project.

2.2 Understanding DevSuite Administration

In DevSuite, administrative tasks are managed at two levels: system administration and project administration.

Site Administration: DevSuite system administration is the process of configuring, administering, and maintaining a DevSuite site. System-level administration activities include maintaining the implementation, creating and defining new projects, managing user accounts, managing licenses, and configuring the DevSuite Server.

Project Administration: DevSuite project administration is the process of defining project-level settings and business objects, customizing project GUIs, configuring workflow rules, managing project team representation and access controls, and integrating add-on modules.

All DevSuite site and project administrative tasks are performed in the DevSuite Admin client. The DevSuite Admin client provides DevTrack, DevPlan, DevSpec, and KnowledgeWise system and project administrators with a single interface for defining, configuring, and administering TechExcel sites and projects.

All DevSuite site and system-level administrative tasks are performed in a System Settings project. Every DevSuite site consists of a single System Settings project. Settings defined in a system settings project apply to every work project in that site.

All project-level administrative tasks are defined on a project-by-project basis and are only applicable to a single work project. In DevSuite 1, project administrators may define four types of projects: KnowledgeWise projects, DevSpec projects, DevTrack projects, and DevTime projects.

Note:DevTest projects may be defined, configured, and administered in the DevTest Admin client. All other DevSuite projects are managed in the DevSuite Admin client.

2.2.1 Understanding DevSuite Site (System-Level) Administration

DevSuite system administration is the process of configuring, administering, and maintaining a DevSuite site. A site consists of a system settings project and one or more work projects. Every work project in a DevSuite site shares a common database, knowledge base, and other system-level configurations that are defined in the DevSuite Admin client.

System-level configurations define the DevSuite site and the integration of every project within that site.

In DevSuite, system administration is the process of defining system, user, and project settings that define work in a DevSuite site. A DevSuite site is an implementation of a DevSuite server and may include multiple DevTrack, DevSpec, DevPlan, and DevSpec projects. System-level settings are settings that apply to every project in the DevSuite site.

DevSuite system administration tasks fall into seven general areas:

System Administration: System-level settings administrative tasks include the configuration of the DevSuite Server.

User Administration: System-level user administrative tasks include the definition and management of user accounts and project administrators. Optional system-level administrative tasks include multiple site management, division management, and LDAP directory server integration and administration.

Multiple Site Administration: Multiple site administration includes the definition and management of a site family and management of user licenses for all member sites.

License Management: System-level license administrative tasks include license management and allocation

Web Administration: Web administration includes the definition and configuration of DevTrack Web Server settings for every DevTrack project in the site.

E-mail Integration:E-mail integration and administration is the foundation of DevTrack, DevPlan, and DevSpec e-mail notification and escalation.

Branch Development: Release management administration defines the product structures that are shared across DevTrack, DevSpec, and DevPlan project providing a common framework for organizing and managing the development of products, versions, and builds.

Time And Cost Tracking Administration: Time and cost tracking administration defines the project management tools that enable a business to measure and track project resources.

System-level user administrative tasks include license management and allocation, user account creation and definition, and the management of project administrators. Optional system-level administrative tasks include multiple site management, division management, and LDAP directory server integration and administration.

2.2.2 Understanding Project Administration

DevSuite project administration is the process of defining project-level settings and business objects, customizing project GUIs, configuring workflow rules, managing project team representation and access controls, and integrating add-on modules.

Project administrators define what is tracked in the project, how it is tracked, and who may track the data. All project-level configurations are project-specific and apply to a single KnowledgeWise, DevSpec, DevTrack, or DevTime project.

Project administration tasks may be broken into five inter-related areas:

- Project Definition and Configuration
- Team Representation Administration
- Work Item Definition and GUI Customization
- Workflow Administration
- Add-on Module Integration

2.3 Understanding DevSuite Sample Projects and Templates

DevSuite builds on TechExcel's commitment to the rapid deployment of its ALM by providing development organizations with the tools they need to quickly configure and implement projects that enforce good development practices and drive the quality of releases.

To facilitate the configuration of a fully integrated site and projects that support your development processes, DevSuite provides developers with tools for evaluating and testing DevSuite features.

Sample Projects:A sample project is a project that provides development organizations with a sandbox for configuring and evaluating DevSuite features and integrations in a risk free environment.

Project Templates:A project template is a blueprint for creating a project of a s pecific type. Solution Templates A solution template is a preconfigured project that facilitates the definition of DevSpec projects that implement five proven development methodologies: CMMI, Scrum, SpecDD, TDD, and Iterative Development.

2.3.1 Understanding Sample Projects

In DevSuite, a sample project is a project that provides the development organization with a sandbox for configuring and evaluating features and integrations in a risk free environment.

DevSuite features ten predefined projects that provide development organizations with a sandbox to experiment with KnowledgeWise, DevSpec, and DevTrack tools and features. All sample projects include sample data, sample project team members, and fully configured workflow rules. Project administrators may freely configure sample projects to represent their development processes.

- Project configurations and settings tried and tested in sample projects may be imported into newly created “live projects”.
- Sample projects provide development organizations with a tool for training new users or to introduce existing users to new features or changes in businesses processes.

Using controls in the DevSuite Admin client, administrators may create additional sample KnowledgeWise, DevSpec, and DevTrack projects.

2.3.2 Understanding Project Templates

In DevSuite, a project template is a blueprint for creating a project of a specific type— KnowledgeWise knowledge management projects, DevSpec requirements management projects, or DevTrack development projects.

Each project template defines a complete set of project-level settings including the definition of all project business objects, workflow rules, team representation, and project integrations.

Project templates may be used to quickly configure new projects whenever they are created in a DevSuite site.

2.3.3 Understanding Solution Templates

TechExcel DevSpec, an integrated requirements management solution, features a complete library of preconfigured solution templates that facilitate the implementation of today's most popular and proven development methodologies.

Each solution template defines DevSuite project settings based on software development best practices so that development organizations may get up and running quickly. Once these business rules are in place, development teams may subsequently customize the project settings to suit their individual needs using TechExcel's award-winning project customization tools.

Using DevSuite, development methodologies longer merely serve as guidelines, but define and direct the daily activities of every team member.

TechExcel provides complete documentation for implementing each methodology that describes how DevSuite workflow and automation may be used to implement and enforce best practices.

Chapter 3 Project Administration

3 Chapter 2- Project Administration

In this chapter:

- Understanding DevSpec Project Administration
- Understanding the DevSuite Admin Workspace
- Accessing DevSuite Projects and Views
- Administering DevSpec Projects

3.1 Understanding DevSpec Project Administration

In DevSuite, administrative tasks are managed at two levels: the system-level and the project-level.

This book describes project-level administrative tasks that apply to TechExcel DevSpec work projects.

In DevSuite, the configuration and customization of work projects is the responsibility of a project administrator. Project administrative tasks include the creation and definition of projects, the definition of project settings and workflow, and team management structures.

3.1.1 Understanding Projects

In DevSuite, a project is a tool for storing, organizing, and managing incident, event, employee, asset, knowledge, and project team data.

Every DevSuite implementation—called a*DevSuite site*—may consist of multiple types of work projects including development projects, timesheet projects, KnowledgeWise projects, and DevTest projects.

A work project is a tool for storing and managing work item and event data. The work items managed and tracked in a project are project-specific: development issues in DevTrack, test templates and test tasks in DevTest, and specifications, requirements, and change requests in DevSpec. Every work project represents a distinct area of work and is defined by unique team structures (account types, team groups, and group folders) and workflow rules.

In DevSuite, the work projects in a DevSuite site are organized into a hierarchical structure that determines the relationships between KnowledgeWise projects, DevSpec projects, and DevTrack projects.

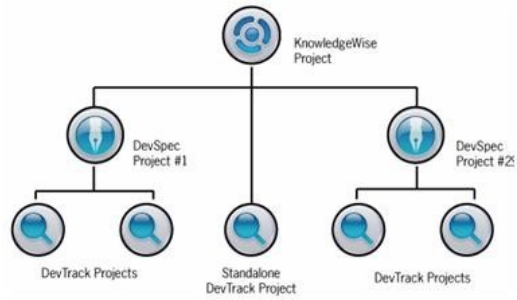


Figure 2-1: DevSuite Project Hierarchy

3.2 Understanding the DevSuite Admin Workspace

In DevSuite, all system- and project-level administrative tasks are performed in the DevSuite Admin client.

When the System Settings project is opened, the DevSuite Admin client displays tools for configuring and administering an entire DevSuite site.

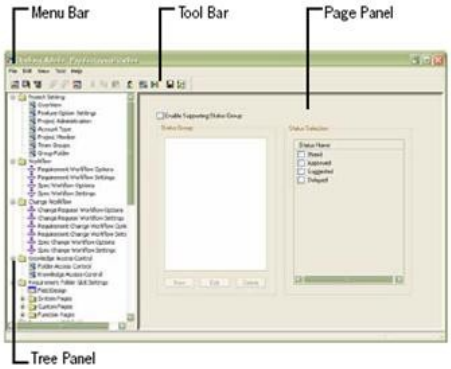


Figure 2-2: DevSuite Admin Client

The DevSuite Admin client organizes the work area into two bars and two panels. Each page is designed to enable an administrator to configure and administer system-level and project-level features.

The DevSuite Admin client is organized into four sections:

Menu Bar: The menu bar displays control buttons that enable the administrator to perform system-level and project-level administrative tasks. Menus include the File, Edit, View, Tool, and Help menus.

Tool Bar: The tool bar displays controls that enable administrators to perform system-level and project-level administrative tasks.

Tree Panel: The Tree Panel organizes administration tools into a hierarchical structure consisting of folders and folders.

Page Panel: The Page Panel displays form-based administrative tools that enable the administrator to configure and maintain system-level settings.

3.2.1 Understanding Menu Bar Commands

The menu bar displays control buttons that enable the administrator to perform system-level and project-level administrative tasks. Menus include the File, Edit, View, Tool, and Help menus.

The majority of the command buttons displayed in the DevSuite Admin tool bar are project-specific commands. Of particular importance to system administrators are the commands displayed in the Tool menu.

File: The File menu displays commands that enable the administrator to create, open, back up, restore work projects.

Edit: The Edit menu displays commands that enable the administrator to manage text and GUI customization tasks.

View: The View menu displays commands that enable the administrator to display or hide the tool bar, status bar, and alignment bar in the Admin client.

Tool: The Tool menu displays commands that enable the administrator to perform a range of system-level administrative tasks including user, login, and license administration.

Help: The Help menu displays commands that enable the administrator to access online help systems and information about the current build.

3.2.2 Understanding the Tool Bar Commands

The tool bar displays controls that enable administrators to perform system-level and project-level administrative tasks.

The majority of the command buttons displayed in the DevSuite Admin tool bar are project-specific commands. Of particular importance to system administrators are the User Manager button, License Manager button, and the Reload Web Settings buttons.

New Project button: The New Project button enables the administrator to create a new work project.

Open Project button: The Open Project button enables the administrator to open an existing work project.

Close Project button: The Close Project button enables the administrator to close a work project.

Back Up Project button: The Back Up button enables the administrator to back up an existing work project.

Restore Project button: The Restore Project button enables the administrator to restore a closed work project.

Delete Project button: The Delete Project button enables the administrator to delete a work project.

User Manager button: The User Manager button enables the administrator to open the User Manager.

Login Manager button: The Login Manager button enables the administrator to open the Login Manager.

Reload Web Settings: The Reload Web Settings button enables the button administrator to stop and restart the DevTrack Web Server ensuring that updated configurations are available to web-based clients.

Save Project button: The Save Project button enables the administrator to save changes to project-level configurations.

Upload to Web Server: The Upload to Web Server button enables the administrator to upload system-level and project-level configurations and settings to the DevSuite Web Server. The DevSuite Web Server must be stopped and restarted for administrative changes to be visible to web clients.

3.2.3 Logging into DevSuite Admin

To log into DevSuite Admin:

1. Select the DevSuite Admin icon in the Windows Start menu.

   By default, the DevSuite Admin icon is added to the DevSuite Admin subfolder within the TechExcel DevSuite folder.

   The DevSuite Admin Login dialog box appears.

2. Enter a user name and password.

Note: DevSuite Admin is delivered with a default system administrator defined (User Name: Admin, Password: (blank)). The password is case-sensitive, however the user name is not case-sensitive. For security reasons TechExcel recommends changing the default login setting as soon as possible.

3. To change web services, select an option from the Web Service dropdown list.

   The DevSuite Admin client connects to the DevSuite Application Server through the DevSuite Web Service. Changing the web service enables the client to connect to a different DevSuite Application Server and manage a different DevSuite site.

4. Click the OK button.

   The DevSuite Admin client opens. By default, the client displays the Project Setup page. To administer a project, open or create a project using tools in the File menu.

3.2.4 Opening DevSpec Projects

The Open Project dialog box displays the System Settings project as well as all live and sample DevTrack, DevSpec, DevPlan, DevTime and KnowledgeWise projects that are available in the DevSuite site.

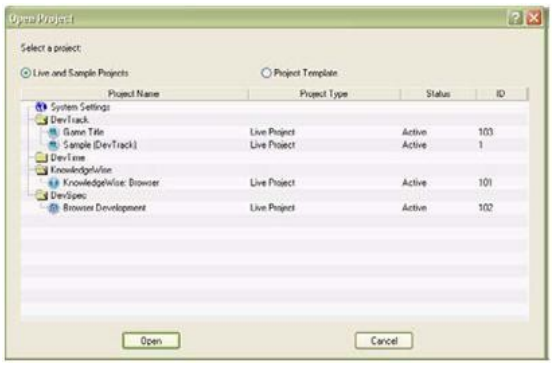


Figure 2-3:Open Project Dialog Box

The Open Project dialog box displays high-level information about every project in the DevSuite site including the project type, project status, and project ID number. Each project is identified by a product icon in the project list.

System Settings

KnowledgeWise

DevSpec

DevTrack

For information on opening and administering KnowledgeWise, DevSpec, and DevTrack projects see the appropriate project administration guide.

To open the System Settings project:

1 Select the Open Project command.

The DevSuite Admin client provides two options for opening projects:

- Select the Open project command in the File menu.
- Press CTRL + O. The Open Project dialog box appears.

2 Select the System Settings Project in the project list.

3 Click the OK button.

The System Settings project is displayed in the workspace.

3.2.5 Closing DevTrack Projects

Administrators may use the Close Project command to close DevTrack projects. The Close Project command may be invoked by two methods:

- Select File > Close Project in the menu bar.
- Press CTRL + S.

3.2.6 Understanding Keyboard Shortcuts

Keyboard shortcuts are combinations of keystrokes that perform predefined functions. Keyboard shortcuts consist of a modifier key and a hot key:

CTRL + X	Cut
CTRL + C	Copy
CTRL + V	Paste
CTRL + N	New Project
CTRL + O	Open Project
CTRL + S	Close Project
CTRL + B	Backup Project
CTRL + R	Restore Project

CTRL + D Delete Project

3.3 Administering DevSpec Projects

In DevSuite, a project is a tool for storing, organizing, and managing incident, event, employee, asset, knowledge, and project team data.

Every DevSuite implementation—called a*DevSuite site*—may consist of multiple types of work projects including development projects,timesheetprojects, KnowledgeWise projects, and DevTest projects.

A work project is a tool for storing and managing work item and event data. The work items managed and tracked in a project are project-specific: development issues in DevTrack, test templates and test tasks in DevTest, and specifications, requirements, and change requests in DevSpec. Every work project represents a distinct area of work and is defined by unique team structures (account types, team groups, and group folders) and workflow rules.

Generally, the configuration and customization of work projects is the responsibility of a project administrator. However, many project administrative tasks are the responsibility of the system administrator. System administrators may create, delete, open, close, back up, and restore work projects and work projects in the General System Settings folder of the DevSuite Admin client.

3.3.1 Creating DevSpec Projects

The New Project dialog box enables administrators to define the project title, the project type, and to import project settings from previously-defined projects. Using controls in the New Project Manager, system administrators may create new projects, sample projects, and project templates.

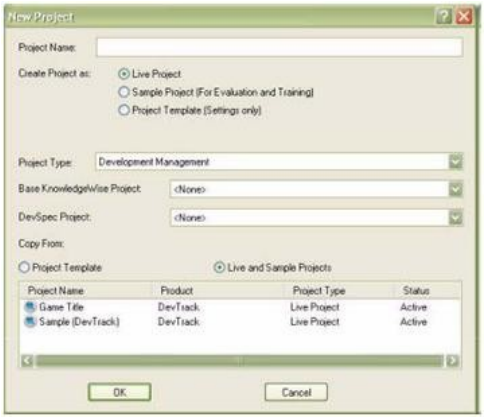


Figure 2-4: Open Project Manager

When creating DevSpec projects administrators may create projects from scratch or they may import project properties from previously defined DevSpec projects. For information on importing project properties from other DevSpec projects see “Copying DevTrack Project Settings” on page 18.

To create a new DevSuite project:

1 Select the New Project command.

The New Project command is accessible by three methods:

- Click the New Project button in the tool bar.
- Press CTRL + N.
- Select the New Project command in the File menu.

The New Project manager appears.

2 Enter a project title in the Project Name text box. Do not use the following characters in the name of the project: / \ : ? “<>|.

3 Select a project type.

DevSuite Admin enables system administrators to create live projects, sample projects, and project templates.

- To create a new sample project, select the Sample Project option button.
- To create a new live project, select the Live Project option button.
- To create a new project template, select the Project Template option button.

4 Select a project type in the Project Type dropdown list.

The Project Type dropdown list displays four options:

- Development Management
- Timesheet Management
- KnowledgeWise Project
- DevSpec Project

5 Optional: Select a project knowledge base in the Base KnowledgeWise Project dropdown list.

Every DevSpec development project and DevSpec project is the child of a single KnowledgeWise project.

6 Optional: Select a DevSpec project in the DevSpec Project dropdown list.

Every DevSpec development project is the child of a single DevSpec project.

7 Optional: Copy project settings from an existing project or project template.

- To copy project settings from a project template, select the Project Template option button and select the project template in the project list.
- To copy project settings from an existing live or sample project of the same project type, select the Live or Sample Project option button and select the project in the project list.

8 Click the OK button.

3.3.2 Backing Up DevSpec Projects

Administrators may use the Backup Project command and the Back Up Project dialog box to back up individual DevSpec projects or the entire DevSpec database.

DevSpec stores backups of DevSpec projects in a proprietary DTK file format. Once the administrator define the project information, the administrator may back up either the entire database or only selected projects. DevSpec Admin creates the backup file with the selected project data.

DevSpec Admin stores backed up projects in a proprietary DTK file format.

Note: DevSpec backup feature is a good way to save project properties, but cannot be considered a substitute for regular database server backups.

To backup DevSpec projects:

1 Invoke the Back Up Project command.

- Select File > Back Up Project in the menu bar.
- Press CTRL + B. The Back Up Project dialog box appears.

2 Click the Ellipsis button to navigate to the location of the backup file.

The Backup dialog box opens.

3 Navigate to the location of the backup files.

4 Enter a name for the DTK file in the File Name field.

5 Select the DevSpec Backup File (DTK) option in the Save as Type dropdown list.

6 Click the Save button. The Backup dialog box closes.

7 Select a back up option:

- Click the Whole Database radio button to backup all projects in a DevSpec implementation.
- Click the Selected Projects radio button to backup some, but not all projects in a DevSpec implementation. Highlight the projects to be backed up.

8 Click the OK button.

The project is backed up.

3.3.3 Restoring DevSpec Projects

Administrators may use the Restore Project command to restore backed up projects.

DevSpec stores backups of DevSpec projects in a proprietary DTK file format. Administrators may choose to back up selected projects or the entire database.

When restoring backed up DevSpec projects administrators may overwrite existing projects or create new projects based on the backed up projects.

To restore DevSpec projects:

1 Invoke the Restore command.

- Select File > Restore in the menu bar.
- Press CTRL + R. The Restore Project dialog box appears.

2 Locate the backed up project. DevSpec stores backups of DevSpec projects in a proprietary DTK file format.

3 Click the Open button. The DevSpec Restore II dialog box appears. The dialog box displays the Project ID number of the project to be restored.

4 Select the Overwrite radio button in the Options area.

5 Click the OK button. The project is restored. If the DTK file selected contains multiple backed up, the DevSpec Restore II dialog box appears for each backed up project.

3.3.4 Creating Projects Based from Backed Up Projects

Administrators may use the Restore command to create new DevSpec projects from backed up copies of existing projects.

Whenever an administrator backs up a DevSpec project, DevSpec saves a copy of that project in a proprietary DTK file format. Administrators may create a new DevSpec project from DTK files.

To create a new project from a backed up project:

1. Invoke the Restore command.
   - Select File > Restore in the menu bar.
   - Press CTRL + R. The Restore Project dialog box appears.
2. Locate the backed up project. DevSpec stores backups of DevSpec projects in a proprietary DTK file format.
3. Click the Open button. The DevSpec Restore II dialog box appears.
4. Select the Create New project radio button in the Options area.
5. Enter a title in the New Project Name field.
6. Click the OK button. The new project is created and the Open Project dialog box appears.

3.3.5 Copying DevTrack Project Settings

System administrators may use the New Project dialog box to create new DevTrack projects based on previously-defined DevTrack projects.

The Copy Settings from an Existing Project command enables administrators to import project properties into the new project. Copying DevTrack project settings cuts down the time required to configure a DevTrack project if the different development projects share similar information.

Project administrators may copy any combination of the following settings to the new project:

- Project members, groups and group folders
- Workflow settings
- Time Tracking settings
- Search and E-mail queries
- Incident routing (autorouting)
- Incident escalation (autoescalation)
- Knowledge view settings
- Event workflow settings
- Interproject settings
- Perforce integration settings
- Issue Template settings

To create DevTrack projects based on existing projects:

1. Select File > New Project.
2. The New Project dialog box appears.
3. Enter a project title in the Project Name field.

   Avoid using the following characters when naming your project:/ \ : ? “<>|.
4. Select a project type from the project dropdown list.

   The only project type available in DevTrack Standard Edition is the Development Management project type. Development Management projects are designed to manage all aspects of defect and feature tracking for any development project.
5. Select the Copy Settings from an Existing Project command. A list of all existing projects is displayed in the window, and you may select the project with the settings you want to import.
6. Select a project in the Project list.
7. Click the OK button.

3.3.6 Understanding Active and Inactive Projects

Every DevTrack project has either an active or inactive status. The status of a project determines the tasks that an administrator may perform on that project.

Administrators define the status of a project by selecting the Active Project check box in the Overview page.

3.3.6.1 Active Projects

Administrators may perform the following tasks using active DevTrack projects:

- Administrators may open active projects in DevTrack Admin and configure project settings.
- Administrators may use the User Manager to add user accounts as project members to active projects.
- Administrators may import project properties from Active projects.

3.3.6.2 Inactive Projects

Administrators may perform the following tasks using inactive DevTrack projects:

- Inactive DevTrack projects may be deleted using the Delete Project command. Active projects cannot be deleted.

3.3.7 Deleting DevTrack Projects

System administrators may use the Delete command to delete inactive DevTrack projects.
Administrators may only delete projects that have an inactive status. For instructions on making a project inactive see “Defining Projects as Active and Inactive” on page 24.

Note: Deleting a project from the system is permanent. Administrators must perform all appropriate back up before deleting a project.

To delete a project:

1. Select File > Delete.
2. The Delete Project dialog box appears. The Delete Project dialog box displays all of the projects that may be deleted. Only those projects that have an inactive status may be deleted.
3. Select a project.
4. Click the Delete button.

3.3.8 Logging into DevSuite Admin

To log into DevSuite Admin:

1. Select the DevSuite Admin icon in the Windows Start menu. By default, the DevSuite Admin icon is added to the DevSuite Admin subfolder within the TechExcel DevSuite folder. The DevSuite Admin Login dialog box appears.
2. Enter a user name and password.

Note: DevSuite Admin is delivered with a default system administrator defined (User Name: Admin, Password: (blank)). The password is case-sensitive, however the user name is not case-sensitive. For security reasons TechExcel suggests changing the default login setting as soon as possible.

3. To change web service, select an option from the Web Service dropdown list.
4. Click the OK button.

   The DevSuite Admin Login dialog box appears.

3.3.9 Creating DevSpec Projects

Chapter 4 Team Administration

4 Chapter 3- Team Administration

In this chapter:

•Understanding Team Administration
•Administering User Accounts
•Administering User Logins
•Managing System Messages
•Administering Account Types, Privileges, and Access Controls
•The user account is removed from the access level.
•Administering Project Members
•Administering Team Groups
•Administering Group Folders

4.1 Understanding Team Administration

The DevSpec client provides project team members with controlled access to all change management artifacts and activities, facilitates communication and workflow, and improves the productivity and effectiveness of the organization.

All DevSpec change management activities are controlled by account typed-based privileges, workflow rules, and access controls. The ability of a DevSpec user account to access projects, views, folders, and pages depends on the “role” (account type)
assigned to that user in the project.

The account type concept provides administrators with the flexibility to customize project workflow and security to support their unique business requirements. DevSpec team representation structures enable development organizations to define sophisticated and flexible workflow rules for managing project data and security.

Project-level team administration is the process of defining team management structures (account types, team groups, and group folders), adding user accounts to the project, and assigning appropriate role and responsibilities to each project member.

Account Type: An account type is a set of access rights, privileges, and responsibilities that define the role that a user account plays in a project. Every user account is assigned one account type in each project.

Team Group:A team group is a collection of project team members that do not necessarily share the same account type but who share common responsibilities in the project.

Group Folder:A group folder is a mechanism for managing the distribution of work items in a project.

4.2 Administering Account Types, Privileges, and Access Controls

In DevSpec, privileges and access rights are not granted to user accounts directly, but assigned to administrator-defined account types. Users are assigned an account type when they join a project and assume the rights and privileges granted to that account type—when a user's responsibility changes, he or she may be assigned a different account type.

In each project, the project administrator is responsible for creating all account types, granting privileges and access rights to those account types, and assigning these account types to each project members based on the role and responsibilities that user performs in the project.

For example, a DevSpec project might provide controlled access to eight different account types: the Program Manager account type, the Project Manager account type,the Designer account type, the Lead Designer account type, the Developer account type, the Sales account type, and the Marketing account type.

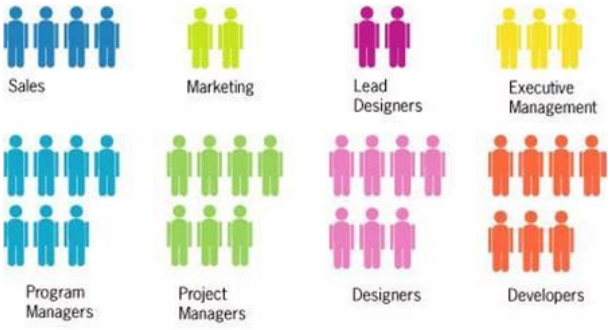


Figure 3-1: Typical DevSpec Project Account Types

Each account type would confer a different set of access rights, privileges, and responsibilities as befit the role played in the project by its team members and stakeholders.

DevSpec uses access rights and privilege controls to coordinate project members in workflow and implement security in projects.

Access Right: An access right defines whether a project member may view or edit a page or data-entry control in the client. DevSpec supports both account type-based access controls and workflow-based access controls.

Privilege:A privilege is a right to perform a task that is granted to project members based on their account type. Every project member is assigned an account type when they join a project. Project member roles, responsibilities, access rights, and privileges are defined by their account type.

4.2.1 Defining Account Types

An account type is a mechanism for assigning and managing access, security, and workflow in DevSpec projects. Every user is assigned an account type when they join a project and with that account type certain privileges, responsibilities and levels of access within the project.

DevSpec team representation provides development organizations with maximum flexibility to manage work items in workflow, enforce security, and ensure accountability. Every user account is assigned one account type in each project. Users may be assigned different account types in different projects.

To create an account type:

1. Select Project Settings > Account Type in the tree panel.
2. Click the New button.

   The New Account Type dialog box appears.
3. Input the name of the account type in the Account Type Name text box.
4. Optional: To copy privilege settings from a previously defined account type, select the Copy Account Type Setting From check box and select an account type in the dropdown list.
5. Click the OK button.

   The New Account Type dialog box closes and the name of the account type is displayed in the Account Types list.

4.2.2 Granting Privileges to an Account Type

In DevSpec, privileges enable project member to perform administrative and management tasks in projects. Project members may only perform tasks that are explicitly granted to them through their account type.

Privileges are granted to account types on a view-by-view basis. In DevSpec, a view is an interface that displays and organizes data in the workspace of the client. Each view is designed to manage different types of data and displays a unique set of tools that enable the user to manage that data.

DevSpec features four views: the specification view, the requirement view, the change request view, and the knowledge view.

Specification View: Specification view privileges enable project members to manage specifications, specification folders, specification access levels, and other features in the specification view of the DevSpec client.

Requirement View: Requirement view privileges enable project members to manage requirements, requirement folders, requirement access levels, and other features in the requirement view of the DevSpec client.

Change Request View: Change request view privileges enable project members to manage change requests, change request folders, change request access levels, and other features in the specification view of the DevSpec client.

Knowledge View: Knowledge view privileges enable project members to manage public and private knowledge reports, web query reports, and other features in the knowledge view of the DevSpec client.

To grant privileges to an account type:

1. Select Project Settings > Account Type in the tree panel.
2. Select an account type in the Existing Accounts list.

   A unique set of privileges may be defined for each account type.
3. Grant or revoke privileges in each DevSpec view.

Each DevSpec view is represented by a tab in the Account Type page: the Specification View tab, the Requirement view tab, the Change Request View tab, and the Report View tab.

- To grant a privilege, select the check box next to the name of a privilege.
- To revoke a privilege, deselect a check box next to the name of a privilege.

4.2.3 Defining Page-Level and Field-Level Access Rights

An access right defines whether a project member may view or edit pages or field in the client. In DevSpec, access rights may be based on both account types and workflow states.

Page-level and field-level access rights define which pages and data-entry controls an account type may view or edit in the DevSpec client.

Specification and requirement pages and data-entry controls may be defined as invisible, read-only, or mandatory for each account type.

Invisible	Invisible pages and controls are not displayed in the DevSpec client.
Read-only	Read-only pages and controls are displayed in the DevSpec client, but cannot be edited.

To define page-level and field-level access rights:

1. Select an account type in the Existing Accounts List of the Account Type page of the Project Settings folder.
2. Optional: To define a page or data-entry control as invisible, select the check box next to the page or control in the Invisible tab.
3. Optional: To define a page or data-entry control as read-only, select the check box next to the page or control in the Read-only tab.
4. Optional: To define a page or data-entry control as mandatory, select the check box next to the page or control in the Mandatory tab.

4.2.4 Renaming Account Types

To rename an account type:

1. Select an account type in the Existing Accounts List of the Account Type page of the Project Settings folder.
2. Click the Edit button.

   The Edit Account Type dialog box appears.
3. Input the name of the account type in the Account Type Name text box.
4. Click the OK button.

   The Edit Account Type dialog box closes.

4.2.5 Deleting Account Types

4.2.5 Deleting Account Types

To delete an account type:

1 Select an account type in the Existing Accounts List of the Account Type page of the Project Settings folder.

2 Click the Delete button.

Before an account type may be deleted, all project members assigned that account type must be deleted or assigned another account type.

A confirmation dialog box appears.

3 Click the Yes button.

A confirmation dialog box appears.

4.2.6 Understanding Specification View Privileges and Access Controls

In DevSpec, every account type is granted privileges to perform certain specification, requirement, change request, and report management tasks. Project members assigned to an account type inherits all of the privileges assigned to that account type.

- Specification privileges enable project members to manage specifications, specification folders, specification access levels, and other features in the specification view of the DevSpec client.
- Specification access controls enable project members to manage specification folders and specifications and managed in public or protected specification folders.

4.2.6.1 Privileges

In DevSpec, a privilege is a right to perform a task that is granted to project members based on their account type.

Specification view privileges include:

Manage All Specification Folder : The Manage All Specification Folder privilege enables an account type to create, edit, and delete specification folders in the specification tree panel of the specification view.

Set Up Access Control : The Assign Folder Access Level privilege enables an account type to define specification folder access controls. Specification folder access controls determine which project members may access and manage specifications in each specification folder.

Unlock Documents Checked Out by Other Users : The privilege enables the user to unlock documents that have been locked or checked out by another user from the knowledge base.

Perform Group Change : The Perform Group Change privilege enables an account type to update or forward groups of specifications using a group action.

Define Public Query : The Define Public Query privilege enables an account type to define and save a query. Public queries are available to all project members in a DevSpec project.

Manually Create Specification Version : The Manually Create Specification Version privilege enables an account type create a new version of a specification.

Delete Specification Version : The Delete Specification Version privilege enables an account type to delete a version of a specification.

Perform Rollback : The Perform Rollback privilege enables an account type to rollback to a previous version of a specification.

Create Default Value Specification : The privilege enables the user to define requirement template— predefined requirements that serves as a template for defining requirements of a particular type quickly and easily.

Import Specification (Including Specification Folder) : The privilege enables the user to import requirements and requirement folders.

Access User Preference : The privilege enables the user to define personal preferences in the DevSpec client.

4.2.6.2 Access Control

DevSpec supports three folder types: public folders, protected folders, and secured folders. Public folders and protected folder are defined by account type-based access controls.

An access control defines the access rights of a project team member to a specific view, page, or folder based on their account type. In DevSpec, each account type may be granted one of five different access rights that enable users assigned that account type to access and manage work items and subfolders in public folders and protected folders: the No Access access right; Read Only access right; Edit; Create and Edit access right; and Create, Edit, and Delete access right.

Distinct work item and folder access rights may be granted to account types for public folders and protected folders:

Public Folder Access: The access control defines the ability of the user to access and manage public folders and subfolders.

Public Specification Access: The access control defines the ability of the user to access and manage the specifications contained in public folders.

Protected Folder Access: The access control defines the ability of the user to access and manage protected folders and subfolders.

Protected Specification Access: The access control defines the ability of the user to access and manage the specifications contained in protected folders.

Secured folders are defined by administrator-defined access levels. For more information on work item access levels and folder access levels see “Administering Secured

Folder Access Levels” and “The user account is removed from the access level.”.

4.2.7 Understanding Requirement View Privileges and Access Controls

In DevSpec, every account type is granted privileges to perform certain specification, requirement, change request, and report management tasks. Project members assigned to an account type inherits all of the privileges assigned to that account type.

- Requirement privileges enable project members to manage requirements, requirement folders, requirement access levels, and other features in the requirement view of the DevSpec client.
- Requirement access controls enable project members to manage requirement folders and requirements and managed in public or protected requirement folders.

4.2.7.1 Privileges

In DevSpec, a privilege is a right to perform a task that is granted to project members based on their account type.

Requirement view privileges include:

Manage All Requirement Folder: The Manage All Requirement Folder privilege enables an account type to create, edit, and delete requirement folders in the requirement tree panel of the requirement view.

Set Up Access Controls: The Assign Folder Access Level privilege enables an account type to define requirement folder access controls. Requirement folder access controls determine which project members may access and manage requirements in each requirement folder.

Unlock Documents Checked Out by Other Users: The privilege enables the user to unlock documents that have been locked or checked out by another user from the knowledge base. Perform Group Change The Perform Group Change privilege enables an account type to update or forward groups of requirements using a group action.

Define Public Query: The Define Public Query privilege enables an account type to define and save a query. Public queries are available to all project members in a DevSpec project.

Manually Create Requirement Version: The Manually Create Requirement Version privilege enables an account type create a new version of a requirement. Delete Requirement Version The Delete Requirement Version privilege enables an account type to delete a version of a requirement.

Perform Rollback: The Perform Rollback privilege enables an account type to rollback to a previous version of a requirement. Create Default Value Requirement The privilege enables the user to define requirement template— predefined requirements that serves as a template for defining requirements of a particular type quickly and easily.

Import Requirement (Including Requirement Folder): The privilege enables the user to import requirements and requirement folders.

Access User Preference: The privilege enables the user to define personal preferences in the DevSpec client.

4.2.7.2 Access Control

DevSpec supports three folder types: public folders, protected folders, and secured folders. Public folders and protected folder are defined by account type-based access controls.

An access control defines the access rights of a project team member to a specificview, page, or folder based on their account type. In DevSpec, each account type may be granted one of five different access rights that enable users assigned that account type to access and manage work items and subfolders in public folders and protected folders: the No Access access right; Read Only access right; Edit; Create and Edit access right; and Create, Edit, and Delete access right.

Distinct work item and folder access rights may be granted to account types for public folders and protected folders:

Public Folder Access: The access control defines the ability of the user to access and manage public folders and subfolders.

Public Requirement Access: The access control defines the ability of the user to access and manage the specifications contained in public folders.

Protected Folder Access: The access control defines the ability of the user to access and manage protected folders and subfolders.

Protected Requirement Access: The access control defines the ability of the user to access and manage the specifications contained in protected folders.

Secured folders are defined by administrator-defined access levels. For more information on work item access levels and folder access levels see “Administering Secured Folder Access Levels” and “The user account is removed from the access level.”.

4.2.8 Understanding Change Request View Privileges

In DevSpec, every account type is granted privileges to perform certain specification, requirement, change request, and report management tasks. Project members assigned to an account type inherits all of the privileges assigned to that account type.

Change request view privileges enable project members to manage change requests, change request folders, change request access levels, and other features in the requirement view of the DevSpec client.

Change request view privileges include:

Can Access Change Request View: The privilege enables the user to access the change request view in the DevSpec client.

Manage Change Request Folder: The privilege enables the user to manage change request folders in the change request tree panel.

Create and Edit Change Request: The Create and Edit Change Request privilege enables an account type to create and update change requests.

Commit Change Request: The Commit Change Request privilege enables an account type to commit change requests.

Delete Change Request: The Delete Change Request privilege enables an account type to delete change requests.

Define Public Query: The Define Public Query privilege enables an account type to define and save a query. Public queries are available to all project members in a DevSpec project.

Access User Preference: The privilege enables the user to define personal preferences in the DevSpec client.

4.2.9 Understanding Report View Privileges

In DevSpec, every account type is granted privileges to perform certain specification, requirement, change request, and report management tasks. Project members assigned to an account type inherits all of the privileges assigned to that account type.

Knowledge view privileges enable project members to manage public and private knowledge reports, web query reports, and other features in the report view of the DevSpec client.

Public Report:A public report is a customized report that is available to all project members.

Private Report:A private report is a customized report that is only available to the report author.

Web Query Report:A web query report is a DevTrack web report that has been saved to the project database and may be accessed through the Internet. Report parameters, the access code, and the report URL are all saved in the database. DevSpec project members, stakeholders, and others may then access web query reports over the Internet using Microsoft Excel.

Report view privileges include:

Can Access Report View: The Can Access Knowledge Report View privilege enables project members to view reports in the report view of the DevSpec client.

Can Create Public Report: The Can Create Private Knowledge Report privilege enables project members to create private knowledge reports in the report view of the DevSpec client.

Can Create Private Report: The Can Create Public Knowledge Report privilege enables project members to create public knowledge reports in the report view of the DevSpec client.

Can Delete Public Report: The Can Delete Public Knowledge Report privilege enables project members to delete public knowledge reports in the report view of the DevSpec client.

4.2.10 Understanding Invisible Pages and Fields

In DevSpec, an invisible page or invisible field is a GUI element is not displayed in the DevSpec client.

- All specification system pages may be defined as invisible pages.
- All data-entry controls displayed in the Specification Description page may be defined as invisible fields.

Pages and fields may be defined as invisible on an account type-by-account type basis. Pages or fields that are invisible for some project members may be visible and editable for others belonging to a different account type.

4.2.11 Understanding Read-Only Pages and Fields

In DevSpec, a read-only page or read-only field is a GUI element is displayed in the DevSpec client but cannot be updated.

- All specification and requirement system pages may be defined as read-only pages.
- All data-entry controls displayed in the Specification Description and Requirement Description page may be defined as read-only fields.

Pages and fields may be defined as read-only on an account type-by-account type basis. Pages or fields that are editable for some project members may be defined as read-only for others belonging to a different account type.

4.3 Administering Secured Folder Access Levels

DevSpec supports three folder types: public folders, protected folders, and secured folders. The ability to create, edit, and manage secured folders and the work items—specifications and requirements— stored in secured folders is defined by administrator-defined secured folder access levels.

A secured folder access level is a collection of access rights —the ability to access, view, edit, create, and delete work items —that are granted to project members based on their account type, team group, or user account. Every secured folder is defined by distinct folder access levels and work item access levels which are assigned to that folder by a privileges project team member.

An access level defines the access rights granted to project team members based on their account type or group. An access level may be defined by one of five different access rights for each account type and team group:

No Access: The No Access access type cannot view or manage knowledge items that are linked to the items stored in a folder.

Read-only: The Read-only access type may view but not edit knowledge items that are linked to the items stored in a folder.

Can Edit: The Can Edit access type may view and edit knowledge items that are linked to the items stored in a folder.

Can Create and Edit: The Can Create and Edit access type may create and edit knowledge items that are linked to the items stored in a folder.

Can Delete, Create, and Edit: The Can Delete, Create, and Edit access type may create, edit, and delete knowledge items that are linked to the items stored in a folder.

Work item access levels apply to secured folders only. Public and protected folders are defined by account type-based access controls. For more information see “Understanding Specification View Privileges and Access Controls” and “Understanding Specification View Privileges and Access Controls” .

4.3.1 Understanding Secured Folder Access Level Conflict Resolution

In the DevSpec client, multiple access levels may be enabled for each specification or requirement knowledge. Each access level defines the access rights that are granted to project members based on their account type, team group, or user account.

Whenever a conflict arises—a user is assigned different access types in the enabled access levels—the access type that grants the user the greatest access to the knowledge is implemented unless an access type is specifically assigned to that user based on their user account.

In the event that a project member is granted multiple access types—one each for their account type, team group, and user account— access rights are granted to the user based on two criteria:*access type prioritization*and*user account prioritization*.

4.3.1.1 Access type prioritization

Access types are prioritized based on the level of access that is granted to a user based on that users account type and team group.

If a project member is assigned one access type based on their account type and another access level based on their team group, the access type that grants the project the greatest access to the knowledge is implemented. No Access The No Access access type cannot view or manage knowledge items that are linked to the items stored in a folder. Read-only The Read-only access type may view but not edit knowledge items that are linked to the items stored in a folder. Can Edit The Can Edit access type may view and edit knowledge items that are linked to the items stored in a folder. Can Create and Edit The Can Create and Edit access type may create and edit knowledge items that are linked to the items stored in a folder. Can Delete, Create, and Edit The Can Delete, Create, and Edit access type may create, edit, and delete knowledge items that are linked to the items stored in a folder

For example, if Terry Johnson were granted the Can Edit access type based on his account type, the Read-only access type based on his team group, he would be granted the ability to edit the knowledge because the Can Edit access type has a higher priority than the Read-only access type.

4.3.1.2 User account prioritization

However, the access type that is assigned to the user account of a project member overrides all other settings and is*always*implemented regardless of the level of access.

If Terry Johnson were granted the Can Edit access type based on his account type, the Read-only access type based on his team group, and the No Access access type based on his account type, he would not be able to access the knowledge because the access type assigned to his user account takes precedence over all other considerations.

4.3.2 Defining Secured Work Item Access Levels

In DevSpec, secured work item access levels define the access rights that are granted to project team members based on their account type, team group, or user account. Distinct access rights may be granted for work items contained within secured folders and for the secured folders themselves.

The ability to create, edit, and manage the*work items*—specifications and requirements— stored in secured folders is defined by administrator-defined secured work item access levels. An access level defines the distinct access rights that are granted to project team members based on their account type, team group, or user account.

To add a secured work item access level:

1. Select Access Controls > Item Access Control in the tree panel. The Work Item Access Control page appears.
2. Click the New button. The New Work Item Access Level dialog box appears.
3. Input the name of the access level in the Name text box.
4. Click the OK button. The New Work Item Access Level dialog box closes.

To define secured work item access level privileges:

1. Select Access Controls > Item Access Control in the tree panel. The Work Item Access Control page appears.
2. Select an access level in the access level list.
3. Optional:To assign the same access to all users, select the All Users Have check box and select an option from the Access Types dropdown list.
4. Select an account type, team group, or user account in the list control.

   The list control displays all administrator-defined account types and team groups. User accounts are only displayed if they are explicitly added to the access level. For more information see See “Adding User Accounts to Access Levels”.
5. Click the Change Selected button. The Access Type dialog box appears.
6. Select an access type in the dropdown list.

   The Access Type dropdown list displays the five access rights: No Access; Read-only; Can Edit; Can Create and Edit; and Can Delete, Create, and Edit.

7 Click the OK button.

To delete a secured folder access level:

1 Select Access Controls > Item Access Control in the tree panel. The Work Item Access Control page appears.

2 Select an access level in the access level list.

3 Click the Delete button.

A confirmation dialog box appears.

4 Click the Yes button.

4.3.3 Defining Secured Folder Access Levels

In DevSpec, secured folder access levels define the access rights that are granted to project team members based on their account type, team group, or user account. Distinct access rights may be granted for work items contained within secured folders and for the secured folders themselves.

The ability to create, edit, and manage secured folders is defined by administrator-defined secured folder access levels.An access level defines the distinct access rights that are granted to project team members based on their account type, team group, or user account.

To add a secured folder access level:

1 Select Access Controls > Folder Access Control in the tree panel. The Folder Access Control page appears.

2 Click the New button.

The New Work Item Access Level dialog box appears.

3 Input the name of the access level in the Name text box.

4 Click the OK button.

The New Work Item Access Level dialog box closes.

To define folder access level privileges:

1 Select Access Controls > Folder Access Control in the tree panel.

The Folder Access Control page appears.

2 Select an access level in the access level list.

3 Optional:To assign the same access to all users, select the All Users Have check box and select an option from the Access Types dropdown list.

4 Select an account type, team group, or user account in the list control.

The list control displays all administrator-defined account types and team groups. User accounts are only displayed if they are explicitly added to the access level. For more information see See “Adding User Accounts to Access Levels” on page 49.

5 Click the Change Selected button.

The Access Type dialog box appears.

6 Select an access type in the dropdown list.

The Access Type dropdown list displays the five access rights: No Access; Read-only; Can Edit; Can Create and Edit; and Can Delete, Create, and Edit.

7 Click the OK button.

To delete a secured folder access level:

1 Select Access Controls > Folder Access Control in the tree panel.

The Folder Access Control page appears.

2 Select an access level in the access level list.

3 Click the Delete button.

A confirmation dialog box appears.

4 Click the Yes button.

4.3.4 Updating Secured Folder Access Levels for All Users

To update the secured folder access level of all users:

1 Select Access Controls > Folder Access Control in the tree panel. The Folder Access Control page appears.

2 Select an access level in the access level list.

3 Click the Change All button.

The

Chapter 5 Specification Administration

5 Chapter 4- Specification Administration

In this chapter:

- Understanding DevSpec Specifications
- Administering Specification Folders
- Administering Specifications
- Administering Specification System Pages
- Administering Specification Workflow
- Administering Specification Types
- Administering Specification-Level Knowledge Management
- Administering Specification State Groups

5.1 Understanding DevSpec Specifications

In DevSpec, a specification represents a formal commitment to implement a feature or enhancement a particular product release. Every specification represents a set of requirements and provides the business with a framework for organizing and tracking the data and knowledge needed to implement those requirements.

Specifications may correspond to a new feature, an enhancement of an existing feature, or even the detailed analysis of a defect that needs to be fixed.

DevSpec specifications are fully customizable and may represent any number of properties: data such as time required, estimated cost, embedded images, attached detailed technical designs, behavior maps, diagrams with use cases, and any other data element that would help a user viewing the specification to understand it completely.

As specifications progress through this lifecycle, development managers work with the product managers to define features, validate requirements, approve designs, and schedule their implementation.

Specifications and Requirements

In DevSpec, a requirement cannot be implemented until it is linked to a specification. Requirements “roll up” into specifications which, in turn, are managed according to custom review and approval processes.

Specifications enable development organizations to manage and track groups of requirements together and to schedule those requirements for implementation in an integrated development project.

Project members may link multiple requirements and knowledge items to specifications. DevTrack development issues and QA test issues may be linked to specifications. ServiceWise support incidents may be linked to specifications.

Specification Tree

Like requirements, DevSpec specifications are organized and managed in a hierarchical tree structure—the *specification tree*—that defines the framework for development, organizes product specifications in the workspace, implements security, enforces good development practices and accountability, and enables collaboration between distributed teams.

Agile Development

In projects using DevSuite blueprint agile development, the specification tree represents a backlog of features—called the *specification backlog*—that organizes the specifications that are yet to be implemented in a development project.

Specifications are implemented when they are assigned to a *iterative subproject*. An iteration subproject is a distinct area of work (as defined by a single specification) that needs to be completed within a set period of time. Every iterative subproject may be defined by a single specification. The specification defines the work required to deliver a feature within that iteration.

DevSuite iterative development is an optional feature that must be enabled in each DevSpec project. For more information on configuring a DevSpec project to support iterative development see “Implementation Integration Administration” on page 171.

5.1.1 Understanding Specification Administration

In DevSpec, specification administration is the process of defining structures for recording, managing, and tracking specifications in a project. All specifications are managed in the specification view of the DevSpec client. Project administrators are responsible for defining the specification object and customizing the client GUI to manage those objects.

5.1.2 Enabling Specifications

In DevSpec, specification management is an optional feature that must be enabled in each DevSpec project. In the DevSpec client, specifications are managed in a hierarchical tree structure—the specification tree—that defines the framework for development, organizes product specifications in the workspace, implements security, enforces good development practices and accountability, and enables collaboration between distributed teams.

5.1.3 Defining Specification ID Prefixes

A specification ID prefix is an administrator-defined text string that is automatically concatenated to the front of each specification ID number. All specification ID prefixes are defined on a project-by-project basis.

If the text string SMB- is defined as the specification ID prefix in a DevSpec project, every specification created in that project is identified by an ID number that begins with that text string. For example, SMB-100, SMB-101, SMB-102 and so on.

Specification ID prefixes enable project teams to quickly identify and sort specifications based on the project that was used to create them.

5.1.4 Defining Specification Starting Numbers

In DevSpec, every specification is assigned a unique identifier—its specification ID— when it is submitted to a project that identifies the specification throughout its life cycle. The title and owner of a specification may change over time, but the specification ID cannot change.

The specification starting number identifies the ID number that is assigned to the first specification created in the DevSpec project and defines the format of all subsequent specifications. Each subsequent specification is automatically assigned an ID number that is one greater than the previous specification.

By default, the initial starting number for both specifications and requirements in a DevSpec project is the number 1. Development organizations may begin the numerical sequence with any integer. For example, 100, 1000, 10000.

5.2 Administering Specification Folders

In the DevSpec client, specifications are managed in a hierarchical tree structure—the specification tree—that defines the framework for development, organizes product specifications in the workspace, implements security, enforces good development practices and accountability, and enables collaboration between distributed teams.

Every area of development within the specification tree framework is represented by a specification folder. A specification folder is a discreet area of work that corresponds to a specific product or version and which implements security based on account type-based access controls.

Every specification folder is defined by its folder type—protected, public, or secured— and a set of folder access and knowledge access controls.

In projects using DevSuite iterative development, specification tree represents the specification backlog.

5.2.1 Customizing the Add Specification Folder Function Page

In the DevSpec client, project members may create new specification folders and define specification folder properties using the Add Specification Folder window.

Figure 4-1: Add Specification Folder Window in the DevSpec Client

The Add Specification Folder window is a manifestation of the Add Specification Folder function page that has been customized by the project administrator."

Using controls in the Specification GUI folder, project administrators may define which pages—system pages and custom pages—are displayed in the client. Project administrators may add up to three system pages and any number of custom pages to the Add Specification Folder function page.

Folder Description: The Folder Description system page enables project members to define and track core specification folder properties.

Applicable Owner: The Applicable Owner system page enables project members to identify the user accounts, team groups, group folders, account types, and user variables that may manage the specifications in the folder.

Access Control: The Access Control system page enables project members to define the folder type, specification access level, and folder access level of each folder.

To customize the Add Specification Folder page:

1Select Spec Folder GUI > Function Pages > Submission Pages in the tree panel.

2 Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.

3Add or remove system pages and custom to the function page.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.2.2 Administering the Edit Specification Folder Function Page

In the DevSpec client, project members may update specification folder properties using the Edit Specification Folder window.

Figure 4-2: Edit Specification Folder Window in the DevSpec Client

The Edit Specification Folder window is a manifestation of the Edit Specification Folder function page that has been customized by the project administrator."

Using controls in the Specification GUI folder, project administrators may define which pages—system pages and custom pages—are displayed in the client. Project administrators may add up to five system pages and any number of custom pages to the Edit Specification Folder function page:

Folder Description: The Folder Description system page enables project members to define and track core specification folder properties.

Access Control: The Access Control system page enables project members to define folder types—public, protected, or secured, specification access levels, and folder access levels of to each folder.

Applicable Owner: The Applicable Owner system page enables project members to identify the user accounts, account types, team groups, group folders, and user variables that may own the specifications in the folder.

Change Log: The Change Log system page provides project members with a read-only log of changes made to the specifications in a folder.

Folder Order: The Folder Order system page enables project members to manage the order of the child subfolder displayed within a parent folder in the specification tree panel.

To customize the Edit Specification Folder page:

1Select Spec Folder GUI > Function Pages >Edit Pages in the tree panel.

2 Optional: To define the title of the page displayed in the DevSpec client, input text in the Page Display Name text box.

3Add or remove system pages and custom to the function page.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.2.3 Administering Specification Folder System Pages

In DevSpec, a system page is a system-defined form that enables DevSpec users to perform a task in the client. A system page generally displays one or more system controls that enable project members to manage and track a special set of data.

Unlike custom pages which are fully customizable, system pages are designed to perform a specific task. System page customizations are generally limited.

DevSpec features five predefined specification folder system pages: the Change Log, Applicable Product, Access Control, Applicable Owner, Folder Order, and Time Report specification folder system pages.

Specification Folder system pages may be displayed in the Add Specification Folder window and the Edit Specification Folder window of the DevSpec client:

Specification Folder System Pages	Specification Folder Function Pages	
	Add Specification Folder	Edit Specification Folder
Folder Description	X	X
Change Log		X
Access Control Page	X	X
Applicable Owner	X	X
Folder Order Page		X

Table 4-1: Specification Folder Function Page

To define the title of a system page:

1Select a system page in the System Pages subfolder in the Specification Folder GUI folder.

DevSpec features five predefined specification folder system pages: Applicable Product system page, Access Control system page, Change Log system page, Folder Order system page, and the Time Report system page.

2Input a title in the Page Display Name text box.

5.2.4 Customizing the Spec Folder Applicable Owner System Page

In the DevSpec client, the Applicable Owner system page enables project members to define which project members may own specifications in each specification folder.

Applicable owners may be identified by user account, account type, team group, group folder, or user variable.

user account:A user account represents an individual DevSpec user that has been assigned a license and account type that enables that user to work in the project.

account type: An account type represents a set of privileges and responsibilities that may be granted to a user account in a DevSpec project. Every user account is defined by its account type.

team group: A team group is an administrator-defined collection of project members—who do not necessarily share the same account type—that share common responsibilities in the project. Team groups organize support teams into smaller, more useful entities that facilitate the assignment and control of items and events.

group folder: A group folder represents a kind of buffer between the team group and the items assigned to that group. Each group folder is owned by a particular group and privilege protected by three access controls: View, Put, and Get. Project members that have been assigned the appropriate account type may view items assigned to a group folder, add items to a group folder, or forward those items to another project member or group.

user variable: A user variable represents a work item stakeholder—a user account— that has a relationship with a work item. System-defined user variables include the {Submitter} user variable, {Current Owner} user variable, and {Previous Owner} user variable.

The Applicable Owner system page may be displayed in the Add Specifications Folder page and the Edit Specification Folder page in the DevSpec client.

5.2.5 Customizing the Change Log System Page

In the DevSpec client, the Change Log page displays a read-only log of the change made to the specifications managed in a folder.

The Change Log page show shows the changes made, the time of the change, and the user account that made the change.

The Change Log page may be displayed in the Edit Specification Folder page in the DevSpec client.

5.2.6 Customizing the Folder Order System Page

In the DevSpec client, the Folder Order page enables project members to manage the order of the subfolders displayed within a folder in the specification tree panel.

Specification folders are displayed in a hierarchical tree structure of folders and subfolders called the specification tree. Every folder within the root specification folder may contain multiple child subfolders. Folder-level access controls and other folder properties may cascade down the hierarchy.

The Folder Order page may be displayed in the Edit Specification Folder page in the DevSpec client.

5.2.7 Customizing the Access Control System Page

In the DevSpec client, the Access Control system page enables project members to define folder-level security for folders and the work items contained within those folders.

The Access Control page enables project team members with the appropriate privileges to define the folder type and define folder-level access controls for folders and the work items contained in folders.

Every requirement folder and specification folder is defined by its folder type—public folder, protected folder, or secured folder. The folder type determines how folder-level access controls are defined for folders and work items.

Public Folder: Public folder security is defined by account type-based access controls.

Protected Folder: Protected folder security is defined by account type-based access controls.

Secured Folder: Secured folder security is defined by an administrator-defined access levels. Within each access level distinct access rights may be granted to project team members based on their account type, team group, or user account.

The Access Control system page may be displayed in the Add Specification Folder page and Edit Specification Folder page in the DevSpec client. Alternately, the folder access level and specification access level controls that are usually displayed in the Access Control system page may be displayed in specification folder custom pages. For more information see See “Displaying Access Controls in Custom Pages” on page 91.

To customize the Access Control page:

1. Select Spec Folder GUI > System Pages > Access Control in the tree panel.
2. Optional: To customize the page title, input a title in the Page Display Name text box.
3. Select a access control display preference:
   - To display the Access Control page as a distinct tabbed page, select the Separate Page option button.
   - To enable Access Control page data-entry controls to be added to specification folder custom pages, select the Separate Fields option button.

Note: If the Separate Fields option is selected, the Access Control page may not be added to the Add Specification Folder or Edit Specification Folder function page. The Folder Access Level system control and the Specification Access Level system control may be added to any specification folder custom page.

5.2.8 Displaying Access Controls in Custom Pages

DevSpec specification folder-level access controls may be defined using the folder access level and specification access level controls. By default, these controls are displayed in the Access Control system page that is in the Add Specification Folder and Edit Specification Folder windows.

The Folder Access Level and Specification Access Level controls may be added to specification folder custom pages as system controls.If this option is selected, the Access Control page is not displayed in the client.

To display access controls in custom pages:

1. Select the Access Control page in the System Pages subfolder in the Specification Folder GUI folder.
2. **Optional**: To enable Access Control page data-entry controls to be added to specification folder custom pages, select the Separate Fields option button.

   If this option is selected, the Access Control page may not be added to the Add Specification Folder or Edit Specification Folder function page. The Folder Access Level system control and the Specification Access Level system control may be added to any specification folder custom page.
3. Select a custom page in the Custom Pages folder.
4. Right-click in the page and select the Add Control command in the shortcut menu.

   The Add New Field window appears.
5. Select the Folder Access Control or Specification Access Control in the System Controls list.

   The Add New Field window closes.
6. Click the OK button.

5.3 Administering Specifications

In DevSpec, a specification represents a formal commitment to implement a feature or enhancement a particular product release. Every specification represents a set of requirements and provides the business with a framework for organizing and tracking the data and knowledge needed to implement those requirements.

Specifications may correspond to a new feature, an enhancement of an existing feature, or even the detailed analysis of a defect that needs to be fixed.

DevSpec project members may define and manage three types of specifications: features, enhancements, and bug fixes. Project members may link multiple requirements and knowledge items to specifications. DevTrack development issues and QA test issues may be linked to specifications. ServiceWise support incidents may be linked to specifications.

5.3.1 Administering the Add Specification Function Page

In the DevSpec client, project members may create new specifications and define specification properties using the Add Specification window—a multiple-page tabbed form.

The Add Specification window displayed in the DevSpec client represents administrator-defined Add Specification function page.

The Add Specification function page may display the Specification Detail system page and one or more custom pages.

To customize the Add Specification page:

1. Select Spec GUI > Function Pages > Submission Pages.
2. Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.
3. Add or remove system pages and custom to the function page.
   - To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
   - To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.3.2 Administering the Edit Specification Function Page

In the DevSpec client, project members may define or update specification properties in the Edit Specification window—a multiple-page form that may display the All Links, Linked Requirements, and Events pages as well as any number of custom pages.

The Edit Specification window displayed in the DevSpec client represents administrator-defined Edit Specification function page.

The Edit Specification function page may display four system pages and any number of custom pages.:

Specification Description: The Specification Description system page enables project members to define and track core specification properties.

All Links: The Specification All Links system page enables project members to track, link, and manage the specifications, requirements, and change requests that are associated with a specification.

Linked Requirement: The Linked Requirement system page enables project members to track, link, and manage the requirements, and knowledge items that are associated with a specification.

Events: The Specification Events page enables project members to create, edit, delete, and track the events that are associated with a specification.

Version: The Version page enables project team members to manage and track multiple versions of specifications.

To customize the Edit Specification page:

1. Select Spec GUI > Function Pages > Editing Pages.
2. Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.
3. Add or remove system pages and custom to the function page.
   - To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
   - To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.3.3 Administering the Specification Detail Function Page

In the DevSpec client, project members may manage and track specification properties in the specification detail panel of the specification view. The specification detail panel shows detailed information about a single specification in multiple tabbed pages including the All Links, Linked Requirements, History, and Events pages as well as any number of custom pages.

Using controls in the Specification GUI folder, project administrators may add or remove system pages and custom pages to the Specification Detail function page:

Specification Description: The Specification Description system page enables project members to define and track core specification properties.

All Links: The Specification All Links system page enables project members to track, link, and manage the specifications, requirements, and change requests that are associated with a specification.

Linked Requirement: The Linked Requirement system page enables project members to track, link, and manage the requirements, and knowledge items that are associated with a specification.

Events: The Specification Events page enables project members to create, edit, delete, and track the events that are associated with a specification

History: The History page provides project members with a read-only audit log of a specification in four standard reports—the Change Log report, the Owner State Change History report, the Summary report, the Tracking History report—and any number of custom reports.

Version: The Version page enables project team members to manage and track multiple versions of specifications.

Quick Report: The Quick Report page enables project members to author and manage specification management reports within the specification view.

To customize the Detail Specification page:

1. Select Spec GUI > Function Pages > Detail Pages.
2. Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.
3. Add or remove system pages and custom to the function page.
   - To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
   - To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.3.4 Administering the Forward Specification Function Page

In the DevSpec client, project members may reassign specifications to other project members in workflow using the Forward Specification window—a multiple-page form that may any number of custom pages.

The Forward Specification window displayed in the DevSpec client represents administrator-defined Forward Specification function page. No system pages are displayed in the Forward Specification page.

The Forward Specification function page may display the Specification Detail system page and one or more custom pages.

To customize the Forward Specification page:

1. Select Spec GUI > Function Pages > Forward Pages.
2. Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.
3. Add or remove system pages and custom to the function page.
   - To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
   - To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.3.5 Administering the Version Function Page

In DevSpec, a specification version is a copy of a specification that represents a distinct stage in the development of that specification. Specification versioning enables organizations to keep track of incrementally different variations of a specification and “rollback” to a previous version.

Using controls in the Version window, project team members may define and submit a version of a specification in the DevSpec client. The Version function page may display the Specification Detail system page and any number of custom pages.

To customize the Version page:

1. Select Spec GUI > Function Pages > Version Pages.
2. Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.
3. Add or remove system pages and custom to the function page.
   - To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
   - To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.3.6 Administering the Specification Template Function Page

In the DevSpec client, project members may define specification templates using the Forward Specification window—a multiple-page form that may any number of custom pages.

The Specification Change window displayed in the DevSpec client represents administrator-defined Specification Template function page.

The Specification Template function page may display the Specification Detail system page and one or more custom pages.

To customize the Specification Template page:

1. Select Spec GUI > Function Pages > Spec Template Pages.
2. Optional:To define the title displayed in the DevSpec client, input text in the Page Display Name text box.
3. Add or remove system pages and custom to the function page.
    - To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
    - To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.3.7 Administering the Group Change Specification Function Page

In the DevSpec client, project members may update multiple specifications using the Group Change Specification window—a multiple-page form that may any number of custom pages.

The Group Change Specification window displayed in the DevSpec client represents administrator-defined Group Change function page. The Group Change Specification function page may display the Specification Detail system page and one or more custom pages.

To customize the Group Change Specification page:

1. Select Spec GUI > Function Pages > Group Change.
2. Optional:To define the title displayed in the DevSpec client, input text in the Page Display Name text box.
3. Add or remove system pages and custom to the function page.
    - To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
    - To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.3.8 Administering the Group Forward Specification Function Page

In the DevSpec client, project members may forward and reassign multiple specifications in workflow using the Group Forward Specification window—a multiple-page form that may any number of custom pages.

The Group Forward Specification window displayed in the DevSpec client represents administrator-defined Group Forward function page.

The Group Forward Specification function page may display the Specification Detail system page and one or more custom pages.

To customize the Group Forward Specification page:

1. Select Spec GUI > Function Pages > Group Forward Pages.
2. **Optional:**To define the title displayed in the DevSpec client, input text in the Page Display Name text box.
3. Add or remove system pages and custom to the function page.
    - To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
    - To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.3.9 Administering the Specification Change Function Page

In the DevSpec client, project members may create changes to specifications using the Change Specification window—a multiple-page form that may display the All Links and Linked Requirement pages as well as any number of custom pages

The Specification Change window displayed in the DevSpec client represents administrator-defined Specification Change function page.

The Specification Detail function page may display three system pages and any number of custom pages:

Specification Description: The Specification Description system page enables project members to define and track core specification properties.

All Links: The Specification All Links system page enables project members to track, link, and manage the specifications, requirements, and change requests that are associated with a specification.

Linked Requirement: The Linked Requirement system page enables project members to track, link, and manage the requirements, and knowledge items that are associated with a specification.

The Specification All Links page may be displayed in the specification detail panel of the DevSpec client.

To customize the Specification Change page:

1. Select Spec GUI > Function Pages > Spec Change.
2. Optional:To define the title displayed in the DevSpec client, input text in the Page Display Name text box.
3. Add or remove system pages and custom to the function page.
   - To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
   - To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

5.3.10 Administering the DevSpec User Function Page

The DevSpec User Specification window displayed in the DevSpec client represents administrator-defined Specification Change function page.

The DevSpec User Function function page may display two system pages and any number of custom pages.

Specification Description: The Specification Description system page enables project members to define and track core specification properties.

All Links: The Specification All Links system page enables project members to track, link, and manage the specifications, specifications, and change requests that are associated with a specification

Change Log Report: The Change Log report displays the complete record of every update made to a requirement in chronological order.

Owner State Change History Report: The Owner and State Change report displays the history of ownership and state changes throughout the entire life cycle of a requirement. Summary Report The Summary report displays a summary of key requirement properties including his requirement ID, title, description, the requirement owner, submitter, and current state.

Tracking History Report: The Tracking History report displays detailed information about requirement updates in a tabular format.

To customize the DevSpec User page:

1. Select Spec GUI > Function Pages > Submission Pages.
2. Optional:To define the title displayed in the DevSpec client, input text in the Page Display Name text box.

Chapter 6 Change Requests Administration

6 Chapter 5- Change Requests Administration

In this chapter:

- Understanding DevSpec Change Request Administration
- Administering Change Request Folders
- Administering Change Requests
- Administering Change Request Workflow

6.1 Understanding DevSpec Change Request Administration

In DevSpec, A*change request*is a package of changes to requirements and specifications that may be managed and tracked together in the project.

A change request represents a formal request to make adjustments to a system and enables change managers to review the effects of changes to specifications or requirements and track the status of the change request.

Figure 5-1: Specification

A change request may represent multiple specification changes and requirement changes and controls the progress of the corresponding specification or requirements.

A change request is defined by its owner, workflow state, and other dynamic properties." Change requests are designed to manage and track the estimated cost and income of a set of changes.

Change requests administration is the process of defining the structures that enable development teams to record, manage, and track change requests in the DevSuite client. Change request management structures include change request folders, change requests, and change request workflow.

6.2 Administering Change Request Folders

In the DevSpec client, change requests are managed in a hierarchical tree structure— the change request tree—that organizes change requests in the workspace.

The change request tree is a hierarchical structure composed of multiple folders and subfolders that organize requirements into distinct areas of work. Project members with the appropriate privileges may create any number of nested change request folders to any level of depth.

Every area of development within the change request tree framework is represented by a change request folder. A change request folder is a discreet area of work that may represent a stakeholder, functional area, or any other category that is useful to a business.

6.2.1 Administering Change Request Folder Description Page

In the DevSpec client, project members may add or edit change request folders and subfolders using the Add Requirement Folder and Edit Requirement Folder windows.

Both the Add Requirement Folder window and the Edit Requirement Folder window displayed in the DevSpec client is a manifestation of a customized Change Folder system page.

Using controls in the DevSuite Admin client, project administrators may add two system fields and any number of custom fields to the Change Request Folder system page.

6.2.1.1 System Fields

System fields are system-defined folder property fields that have a designated use in the DevSpec project— the data integrated is other TechExcel DevSpec features for change request management or reporting. Two system fields may be displayed in the Change Request Folder Description page:

1001 Folder Name Long (<=255) text (edit box)

1002 Folder Description Multiple-line Edit Box

Project administrators may make only limited customizations to system controls.

- The title of the corresponding data-entry control may be customized.
- The Folder Description field may be customized to support spell checking, time stamping, and HTML tags.

To customize the Change Request Folder page:

1 Select Change GUI > Change Request Folder > System Pages > Description Page.

2 Optional: To customize the title displayed in the DevSpec client, input text in the Page Display Name text box.

6.3 Administering Change Requests

In DevSpec, a change request is a package of changes to requirements and specifications that may be managed and tracked together in the project. The change request controls the progress of linked requirements and specifications and enables change managers to ensure that adjustments to a system are implemented throughout that system.

A change request is defined by its owner, workflow state, and other dynamic properties. Administrator-defined workflow rules implement account type-based access controls ensuring that changes are properly processed.

Change requests are designed to manage and track the estimated cost and income of a set of changes.

Change requests are managed and tracked in the change request view of the DevSpec client. Project members may review the templates and work items that are affected by requirement changes.

- Changes to requirements flag the features that are linked to test templates and work items.
- Customers may create, edit, and track change requests in TechExcel DevSpec Express. Each time a requirement is changed a new version is created.

Changing a requirement may not be allowed or will require approval because the product and the version have reached certain release stage. Requirement change may need workflow control because linked development works are already finished.

6.3.1 Administering Change Request System Pages

In DevSpec, a system page is a system-defined form that enables DevSpec users to collect, manage, and track information in the client. System pages feature the built-in functionality of system controls. Unlike custom controls which are fully customizable and may be used to track any type of data, each system control is specifically designed to track a particular set of data that is linked to other controls and functionalities in the DevSpec project.

DevSpec features three predefined change request system pages: the Description, Applicable Product and History change request system pages.

Description: The Change Request Description system page enables project members to define and track core change request properties—title, owner, description.

Applicable Product: The Applicable Product system page enables project members to identify the products and versions that are applicable to the change request.

History: The History page provides project members with a read-only audit log of a change request in four standard reports—the Change Log report, the Owner State Change History report, the Summary report, the Tracking History report—and any number of custom reports.

6.3.2 Administering the Edit Change Request Function Page

In the DevSpec client, project members may edit change requests and define change request properties using the Edit Change Request window.

The Edit Change Request window displayed in the DevSpec client represents administrator-defined Edit Change Request function page. The Edit Change Request Function page may display any number of administrator-defined custom pages.

To customize the Edit Change Request page:

1 Select the Submission Pages page in the Function Pages subfolder in the Change Request GUI folder.

2 Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.

3 Add or remove custom to the Edit Change Request function page.

The Available Pages list displays all administrator-defined custom pages.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

6.3.3 Administering the Change Request Detail Function Page

In the DevSpec client, project members may track and update change request properties in the change request detail panel of the change request view. The change request detail panel shows detailed information about a single change request in multiple tabbed pages.

DevSpec features two predefined change request system pages: the Applicable Product and History change request system pages.

Description: The Change Request Description system page enables project members to define and track core change request properties—title, owner, description.

Applicable Product: The Applicable Product system page enables project members to identify the products and versions that are applicable to the change request.

History: The History page provides project members with a read-only audit log of a change request in four standard reports—the Change Log report, the Owner State Change History report, the Summary report, the Tracking History report—and any number of custom reports.

To customize the Add Change Request page:

1Select the Submission Pages page in the Function Pages subfolder in the Change Request GUI folder.

2 Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.

3Add or remove system pages and custom to the function page.

The Available Pages list displays the Applicable Product system page, the History system page, and all administrator-defined custom pages.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

6.3.4 Administering the Forward Change Request Function Page

In the DevSpec client, project members may forward change requests to other project members using the Forward Change Request window.

The Forward Change Request window displayed in the DevSpec client represents administrator-defined Forward Change Request function page.

The Forward Change Request function page may display all administrator-defined change request custom pages.

To customize the Forward Change Request page:

1Select the Submission Pages page in the Function Pages subfolder in the Change Request GUI folder.

2 Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.

3Add or remove system pages and custom to the function page.

The Available Pages list displays all administrator-defined custom pages.

- To add a page to the function page, select a page in the Available Pages list and click the Right Arrow button.
- To remove a page from the function page, select a page in the Working Pages list and click the Left Arrow button.

6.3.5 Administering the Applicable Product System Page

In DevSpec, a change request may define a function in a particular product or version only if that product or version has been defined as applicable to the change request.

The Applicable Product system page may be displayed in the change request detail panel of the DevSpec client.

Using controls in the Applicable Product system page, project members may identify the products and versions that are applicable to change requests.

To define the title displayed in the DevSpec client:

1Select the Applicable Product system page in the System Pages subfolder in the Change Request Folder GUI folder.

2 Optional: To customize the title of the page displayed in the DevSpec client, input a title in the Page Display Name text box.

6.3.6 Administering the Change Request History System Page

In the DevSpec client, the History page provides project members with a read-only audit log of the changes made to a change request in four standard reports: the Change Log report, the Owner State Change History report, the Summary report, the Tracking History report.

Figure 5-2: History Page in the DevSpec Client

Using controls in the Change Request GUI folder, project administrators may add the History system page to the Change Request Detail function page, Change Request Edit

function page, and the Change Request Change function page."

Each report in the History page corresponds to a change request system page or custom page.

Project administrators may define which reports are displayed in the History page by adding or removing reports to the History page. Four system-defined reports may be displayed in the History page. Custom reports based on custom pages may also be displayed in the page.

Change Log report: The Change Log report displays the complete record of every update made to a change request in chronological order.

Owner State Change History report:The Owner and State Change report displays the history of ownership and state changes throughout the entire life cycle of a change request. Summary report The Summary report displays a summary of key change request properties including his change request ID, title, description, the change request owner, submitter, and current state.

Tracking History report: The Tracking History report displays detailed information about change request updates in a tabular format.

Custom reports: Custom reports display data that is managed and tracked in an administrator-defined custom page. Every administrator-defined page may be represented by a report in the History page.

The History page displays the complete history of the change request across all versions of the change request.

The Change Request All Links page may be displayed in the change request detail panel of the DevSpec client.

To add or remove reports to the History page:

1Select the History page in the System Pages subfolder in the Change Request GUI folder.

2 Optional: To define the title displayed in the DevSpec client, input text in the Page Display Name text box.

3Add or remove reports to the function page.

- To add a report to the function page, select a page in the Available Sections list and click the Right Arrow button.
- To remove a page from the function page, select a page in the History Info list and click the Left Arrow button.

6.4 Administering Change Request Workflow

TechExcel DevSuite solutions enable development organizations to define custom workflow rules for managing every aspect of the ALM life cycle. DevSpec change request workflow defines how change requests are created, managed, and tracked in a DevSpec project.

Administrator-defined workflow rules determine the sequence of workflow states, how and when a change request may pass from one workflow state to the next, and who may own and edit a change request in each stage of its life cycle.

A typical change request workflow might consists of four workflow states: New, Pending, Approved, and Dropped. Each workflow state represents a stage in the life cycle of a change request.

Figure 5-3: Specification Workflow Transitions

The change request life cycle consists of at least three workflow states: an open workflow state and two closed workflow states—a workflow state with a closed: committed status and a workflow state with a closed: abandoned status." By default, all new change requests start in the{New} workflow state, an open state.

Every workflow state is defined as either an open, closed: committed, or closed: abandoned workflow state. based on its status. The status assigned to a workflow state determines how the change request (and the specification changes and requirements changes linked to that change request) are managed in that state.

Whenever a change request enters a closed workflow state—a workflow state that is defined by a Closed: Abandoned status or Closed: Committed status, the project team member may chose to commit or abandon the changes—requirement changes and specification changes—represented by the change request.

6.4.1 Enabling Change Request Workflow Options

Change Request workflow may be defined by six different workflow rules. All workflow rules are optional and must be enabled and defined in each project:

State-to-State Transitions: State-to-State Transition rules define which project members may transition a change request from one state to another through each transition.

Applicable Owners: Applicable Owners rules define which project members may own a change request in each workflow state based on their account type, team group, group folder, system variable, or user account.

Read-Only Fields: Read-only Fields rules define which project members may update specific change request properties in each workflow state based on their account type or team group.

Invisible Fields: Invisible Fields rules define which project members may view specific change request properties in each workflow state based on their account type or team group.

Mandatory Fields: Mandatory Fields rules define which project members must update specific change request properties in each workflow state based on their account type or team group.

Who Can Change: Transition Access Control rules define which project members may update a change request in each workflow state based on their account type or team group.

To enable change request workflow options:

1. Select the Change Request Workflow Options page in the Change Request folder of the Workflow folder.
2. **Optional:**To enable a change request workflow option, select the check box next to the option.

   Six different change request workflow options may be enabled in the Change Request Workflow page: State To State Transition, Applicable Owners, Read-only Fields, Invisible Fields, Mandatory Fields, and Who Can Change State.

6.4.2 Creating Change Request Workflow States

In DevSpec, the lifecycle of a change request is defined and managed in three or more workflow states.

A*workflow state*is a set of workflow and access controls that represents a particular stage in the lifecycle of the change request and which define how a change request is managed at that state in its development.

Every. workflow state is defined by its name, status, and optional access controls.

The status assigned to a workflow state determines how the change request (and the specification changes and requirements changes linked to that change request) are managed in that state.

Open: The Open workflow status defines a workflow state as an open state.

Closed: Committed: The Closed: Committed status defines a workflow state as a closed state and requires that user commit linked specification changes and requirement changes.

Closed: Abandoned: The Closed: Abandoned status defines a workflow state as a closed state and requires that user abandon linked specification changes and requirement changes

To add a change request workflow state:

1. Select the Change Request Workflow Options page in the Change Request folder of the Workflow folder.
2. Click the New button.

   The Add New State dialog box appears.
3. Input the title of the change request state in the State Name text box.
4. Define the status of the change request state.

   Every change request workflow state is defined by its status: open, closed: committed, or closed: abandoned.

   - To define the change request state as an open state, select the Open option button.
   - To define the change request state as an closed: committed state, select the Closed: Committed option button.
   - To define the change request state as an closed: abandoned state, select the Closed: Abandoned state option button.
5. Optional: To define the change request as read-only in the change request state, select the Read-Only check box.

   Change Request properties may be viewed, but not updated when a change request is in a workflow state that has been defined as read-only.
6. Click the OK button.

   The Add New State dialog box closes.

6.4.3 Editing Change Request States

To edit a change request workflow state:

1. Select a change request workflow state in the Change Request States list in the Change Request Workflow Options page in the Change Request folder of the Workflow folder.
2. Click the Edit button.

   The Edit State dialog box appears.
3. **Optional:**To update the title of the change request state, edit the name in the State Name text box.
4. Define the status of the change request state.

   Every change request workflow state is defined by its status: open, closed: committed, or closed: abandoned.

   - To define the change request state as an open state, select the Open option button.
   - To define the change request state as an closed: committed state, select the Closed: Committed option button.
   - To define the change request state as an closed: abandoned state, select the Closed: Abandoned state option button.
5. **Optional:**To define the change request as read-only in the change request state, select the Read-Only check box.

   Change Request properties may be viewed, but not updated when a change request is in a workflow state that has been defined as read-only.

6 Click the OK button.

The Edit New State dialog box closes.

6.4.4 Deleting Change Request States

To delete a change request workflow state:

1 Select a change request workflow state in the Change Request States list in the Change Request Workflow Options page in the Change Request folder of the Workflow folder.

2 Click the Edit button.

A confirmation dialog box appears.

3 Click the Yes button.

6.4.5 Defining Change Request State Transitions

Change request state transitions define the sequence by which a change request may pass through each stage in the change request life cycle.

State transition rules enable organizations to control the change request life cycle by defining possible workflow paths—called*transitions*. A change request may pass from one workflow state to next only if a transition exists between those to workflow states.

If no transition exists, the change request must be assigned to another workflow state. For example,.

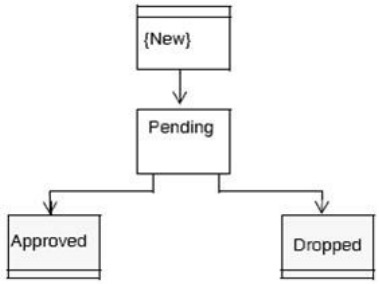


Figure 5-4: Specification Workflow Transitions

Each node represents a change request workflow state. The arrows between each node represent the transition between those workflow states. The inactive states are darkened.

If Who Can Change transition rules are enabled, the project member may restrict the ability of project members transition change requests between particular states based on their account type, team group, user account, or one of four system variables.

If state-to state transition rules are not enabled and defined, a change request may freely pass between any two workflow states without restriction.

To define change request state transitions:

1 Enable state-to-state transition workflow rules.

Change Request state-to-state transition rules are an optional feature that may be enabled and defined in each DevSpec project. For step-by-step instructions see See “Enabling Change Request Workflow Options” .

2 Select the Change Request Workflow Settings page in the Change Request folder of the Workflow folder.

3 Select a change request workflow state in the Change Request States list.

The selected change request workflow state is the beginning state. All transitions enable a change request to pass from the selected state to a target state, but not the other way around.

4 Select the State Transition tab.

The State Transition tab is only displayed if state-by-state transitions are enabled.

5 Add or remove transition between the selected change request state and a target change request state.

- To add a transition to the target state, select the check box next to the name of the targeted change request state.
- To remove a transition to the target state, deselect the check box next to the name of the targeted change request state.

A check mark indicates that a transition exists between the two change request states. A change request may pass from the selected workflow state to the target workflow state.

6.4.6 Defining Change Request State Transition Rules

Who Can Change transition rules define which project members may update the change request workflow state based on their account type, team group, system variable, or user account.

State transitions may be restricted based on the relationship between the user and the change request.

{Submitter}: The {Submitter} system variable represents the project member that initially submitted the change request to the project.

{Current Owner}: The {Current Owner} system variable represents the project member that currently owns the change request.

To restrict change request state transitions:

1. Enable Who Can Change workflow rules.

   Who Can Change rules are an optional feature that may be enabled and defined in each DevSpec project. For step-by-step instructions see See “Enabling Change Request Workflow Options” .

2. Select the Change Request Workflow Settings page in the Change Request folder of the Workflow folder.

3. Select a change request workflow state in the Change Request States list.

4. Select the Who Can Change tab.

   The Who Can Change tab is only displayed change request state applicable owner rule support is enabled.

5. Select a change request workflow state in the Next State list.

   The Next State displays all possible next workflow states.

6. Define which project members may perform the transition by selecting or deselecting account types, team groups, system variables, or user accounts in the Applicable Owner list.

   - To enable project members to perform the transition, select the check box next to an account type, team group, system variable, or user account in the Applicable Owner list.
   - To prevent project members to perform the transition, deselect the check box next to an account type, team group, system variable, or user account in the Applicable Owner list.

6.4.7 Defining Change Request State Applicable Owners

In the DevSpec client, only those project members that have been designated as applicable owners may be assigned ownership of a change request in a each change request workflow state.

State-based applicable owner rules define which project members may own a change request in each workflow state based on their account type, team group, group folder, or user account.

DevSpec project members may be defined as applicable owners based on system variables.

{Unassigned}: The {Unassigned} system variable does not represent a project member, rather it is a placeholder.

{Submitter}: The {Submitter} system variable represents the project member that initially submitted the change request to the project.

{Current Owner}: The {Current Owner} system variable represents the project member that currently owns the change request.

{Previous Owner}: The {Previous Owner} system variable represents the project member that owned the change request prior to the current owner.

To define change request state transitions:

1. Enable state-based applicable owner workflow rules.

   State-based change request state applicable owner rules are an optional feature that may be enabled and defined in each DevSpec project. For step-by-step instructions see See “Enabling Change Request Workflow Options” .

2. Select a change request workflow state in the Change Request States list in the Change Request Workflow Settings page of the Change Request folder in the Workflow folder.

   The Change Request State list displays all administrator-defined change request states.

3. Select the Applicable Owners tab.

   The Applicable Owners tab is only displayed if state-based applicable owner rules are enabled.

4. Add or remove applicable owners by selecting or deselecting account types, team groups, group folders, system variables, or user accounts in the Applicable Owner list.

   - To add an applicable owner, select the check box next to an account type, team group, group folder, system variable, or user account in the Applicable Owner list.
   - To remove an applicable owner, deselect the check box next to an account type, team group, group folder, system variable, or user account in the Applicable Owner list.

6.4.8 Defining Default Owners by Change Request State

To define default owners by change request state:

1. Enable state-based applicable owner workflow rules.

   State-based change request state applicable owner rules are an optional feature that may be enabled and defined in each DevSpec project. For step-by-step instructions see See “Enabling Change Request Workflow Options” .

2. Select a change request workflow state in the Change Request States list in the Change Request Workflow Settings page of the Change Request folder in the Workflow folder.

   The Change Request State list displays all administrator-defined change request states.

3. Select the Applicable Owners tab.

The Applicable Owners tab is only displayed if state-based applicable owner rules are enabled.

4 Define one or more account types, team groups, group folders, user accounts, and system variables

5 Select an applicable owner from the Default Owner dropdown list.

The Default Owner dropdown list displays every account type, team group, group folder, user account, and system variable that has been defined as applicable to the current change request state.

6.4.9 Defining State-Based Read-Only Change Request Fields

In DevSpec, a read-only field is a data-entry control that is displayed in the DevSpec client, but cannot be edited.

State-based read-only field rules define which change request properties project members may view and update in each change request workflow state. Change Request controls defined as read-only may not be updated by any project member regardless of that user's account type or team group as long as the change request is in that change request state.

All system controls and custom controls may be defined as read-only in each workflow state.

To define state-based read-only fields:

1 Enable read-only rules.

Change Request state applicable owner rules are an optional feature that may be enabled and defined in each DevSpec project. For step-by-step instructions see See "Enabling Change Request Workflow Options" .

2 Select the Change Request Workflow Settings page in the Change Request folder of the Workflow folder.

3 Select a change request workflow state in the Change Request States list.

4 Select the Read-Only tab.

The Applicable Owners tab is only displayed if the state-based read-only field rules are enabled.

5 **Optional:** To define a change request state as read-only in the selected change request state, select the check box next to the name of the change request property in the Read-Only Field list.

6.4.10 Defining State-Based Mandatory Change Request Fields

In DevSpec, a mandatory field is a data-entry control that must be defined before a change request may progress to the next stage in the change request life cycle.

State-based mandatory fields rules identify the change request properties that must be defined in a change request workflow state before that change request may progress to another workflow state in the change request life cycle.

All system controls and custom controls may be defined as mandatory in each workflow state.

To define state-based mandatory fields:

1 Enable mandatory rules.

Change Request state applicable owner rules are an optional feature that may be enabled and defined in each DevSpec project. For step-by-step instructions see See "Enabling Change Request Workflow Options" on page 125.

2 Select the Change Request Workflow Settings page in the Change Request folder of the Workflow folder.

3 Select a change request workflow state in the Change Request States list.

4 Select the Mandatory tab.

The Applicable Owners tab is only displayed if the state-based mandatory field rules are enabled.

5 **Optional:** To define a change request state as mandatory in the selected change request state, select the check box next to the name of the change request property in the Mandatory Field list.

6.4.11 Defining State-Based Invisible Change Request Fields

In DevSpec, an invisible field is a data-entry control that is hidden from certain project members based on their account type or team group.

State-based invisible field rules define which change request properties are displayed in each change request workflow state. Change Request controls defined as invisible may not be viewed or updated any project member regardless of that user's account type or team group as long as the change request is in that change request state.

All system controls and custom controls may be defined as invisible in each workflow state.

To define state-based invisible fields:

1 Enable invisible rules.

Change Request state applicable owner rules are an optional feature that may be enabled and defined in each DevSpec project.

2 Select the Change Request Workflow Settings page in the Change Request folder of the Workflow folder.

3 Select a change request workflow state in the Change Request States list.

4 Select the Invisible tab.

The Applicable Owners tab is only displayed if the state-based invisible field rules are enabled.

5 **Optional:** To define a change request state as invisible in the selected change request state, select the check box next to the name of the change request property in the

Invisible Field list.

Chapter 7 DevSpec GUI Gustomization

7 Chapter 6- DevSpec GUI Gustomization

In this chapter:

- Administering System Pages
- Administering Function Pages
- Administering Custom Pages
- Administering System Fields and Controls
- Administering Custom Fields and Controls
- Customizing Selection and Combination Box Controls
- Administering Calculation Controls
- Customizing Voting Grid Controls
- Administering Standard Lookup Fields

7.1 Understanding DevSpec GUI Customization

DevSpec project administration is largely the process of defining structures for recording, managing, and tracking work items—requirements, specifications, change requests—in a project.

Customized project management structures are manifested in the DevSpec client as views, panels, folders, and pages.

A view is an interface that displays and organizes data in the client workspace. DevSpec supports three customizable views: the requirements view, the specifications view, and the change request view.

Views enable project members to manage and track different types of data using structures that are specifically designed to manage that data. The building blocks of each view—folders, pages, and controls—may be customized to support custom business processes.

The building blocks of client interface include: controls, folder pages, and GUI pages.

Control:A control is a GUI element displayed in a folder page or custom page that enables the user to define, update, and track data. DevSpec supports system controls and custom controls. Controls are displayed in system pages and custom pages.

System Page:A system page is a predefined form that enables project members to collect, manage, and track project data in a project." A system page displays a predefined set of system controls.

Custom Page:A custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data." A custom page may display system controls and custom controls.

Function Page:A function page is a compound page composed of multiple custom pages and system pages." In the DevSpec client, function pages are displayed as multiple-page forms that enable the user to perform a particular function such as adding or updating a folder or a requirement.

7.2 Administering System Pages

In DevSpec, a system page is a predefined page that enables project members to perform a specific task—such as collecting, managing, and tracking project data.

Each system pages features one or more system controls. A system control is a systemdefined control that represents a predefined system field and features control-specific functionality. System fields represent core object properties—such as workflow states and object statuses—and the data tracked in these fields is frequently forms the foundation of TechExcel businesses rules.

Like custom pages, system pages may be added to function pages to build the multiple-page forms that enable project members to manage requirements, specifications, and change requests in each view.

7.3 Administering Function Pages

A function page is a compound page composed of multiple custom pages and system pages. In the DevSpec client, function pages are displayed as multiple-page forms that enable the user to perform a particular function such as adding or updating a folder or work item.

The function page acts as a frame for displaying the custom and system pages that enable project team members to manage work items.Project customization consists of creating custom pages and controls and adding those pages to function page so that they are displayed in the client.

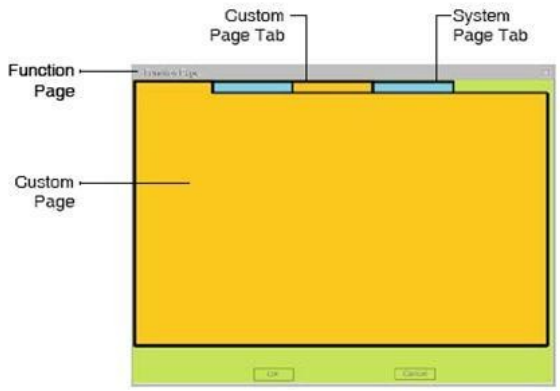


Figure 6-1: Function Page Diagram

7.4 Administering Custom Pages

In DevSpec, a custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data.

Each custom page may display system controls and custom controls. Using custom pages, system controls, and custom controls, a business may create forms that are uniquely designed to manage and track data that is specific to their environment and business.

Figure 6-2: Custom Page

Like system pages, custom pages may be added to function pages to build the multiple-page forms that enable project members to manage requirements, specifications, and change requests in project views."

Project administrators may add and remove system, custom, and multiple-selection controls to custom pages, customize the layout of controls in the page, and define their the tab order.

7.4.1 Adding Custom Pages

A custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data.

Project administrators may create custom pages for use in the requirements view, specification view, and change request view.

To create a custom page:

1. Right-click a Custom Pages folder and select the Add New Page command in the shortcut menu.

   Custom pages may be created for the requirements folder, requirements GUI, specification folder, specification GUI, and change request folder, and change request GUI

   The custom page is displayed in the workspace.

2. Input the title of the custom page in the Page Display Name text box.

   The display name defines the text that identifies the tabbed page in the client. By default, the custom page is given the title New Page.

3. Add custom controls or system controls.

4. Define the tab order of the controls displayed in the custom page.

7.4.2 Adding Controls to Custom Pages

To add a control to a custom page:

1. Right-click in the custom page and select the Add Control command in the shortcut menu. The Add New Field manager appears.

The Add New Field manager displays every custom controls and system controls that may be added to the custom page.

2 **Optional**: To filter controls by control type, select an option from the Field Type dropdown list.

The Field Type dropdown list displays multiple control types including the dropdown list control type, short text box control type, and the field label control type. DevSpec enables administrators to define controls based on 17 different control types.

3 **Optional**: To define how the control is presented in the page, select an option from the Label Alignment dropdown list.

The Label Alignment dropdown list displays three options:

- Right Aligned
- Left Aligned
- Center Aligned

4 **Optional**: To define control as read only in the page, select the Add It as Read Only check box.

A read-only control is a control that is displayed, but cannot be edited.

5 Click the OK button.

7.4.3 Removing Controls From Custom Pages

To remove a control to a custom page:

1 Select a control in the custom page.

2 Right-click and select the Delete Control command in the shortcut menu.

Note: The control is not deleted, but simply removed from the page. The control may subsequently be added to the page or another page.

A confirmation dialog box appears.

3 Click the Yes button.

7.4.4 Ordering Controls in Custom Pages

The control with the lowest tab order number is the control that is immediately active when the page is accessed in the client.

To define the tab order of controls in a custom page:

1 Select a custom page in a Custom Pages folder.

2 Click the Define Field Tab Order button in the custom page.

3 The Field Tab Order dialog box appears.

The Field Tab Order dialog box displays a list of the custom controls and system controls that have been added to the custom page.

4 Define the tab order of each control in the page relative to the other controls displayed in the page.

- To move the control up the list, select the control and click the Up arrow. The lower the tab order number, the earlier the control may be accessed.
- To move the control down the list, select the control and click the Down arrow. The greater the tab order number, the later the control may be accessed.

5 Click the OK button.

7.5 Administering System Fields and Controls

A control is a graphic element displayed in a window, panel, or dialog box that enables the user to define, update, and track data. Each control represents a property—system field or custom field—of a business object. DevSpec controls enable project members to manage folders, requirements, specifications, change requests, and events.

A system control is a system-defined control that represents a predefined system field and features control-specific functionality. System fields represent core object properties—such as workflow states and object statuses—and the data tracked in these fields is frequently forms the foundation of TechExcel businesses rules.

Consequently, system controls are designed for a specific form of interaction and customizations to these controls is limited.

Every system field is defined by four properties: the Field ID, Field Name, Control Title, and Control Type.

Field ID: The Field ID is an identification number uniquely identifies the system field in the project.

Field Name: The Field Name is a unique title that identifies the system field in the project.

Control Title: The Control Title is a customizable field label that identifies the system control in the client GUI

Control Type: The Control Type defines the functionality of a control—its appearance and operation.

The field ID and field name of system fields are fixed in DevSpec and cannot be changed. These properties enable administrators to identify fields across projects and facilitate integration and interproject configurations.

7.5.1 Understanding Specification Folder System Fields

System fields are system-defined folder property fields that have a designated use in the DevSpec project— the data integrated is other TechExcel DevSpec features for specification management or reporting.

Each specification folder system field is represented by a data-entry control in the DevSpec client that enables project members to manage and track data. Project administrators may make only limited customizations to system controls.

- The title of the corresponding data-entry control may be customized.
- The Folder Description field may be customized to support spell checking, time stamping, and HTML tags.

The values displayed as options in these system controls are drawn from the DevSpec database and cannot be customized by the administrator

1001	Folder Name	Long (<=255) Text (Edit Box)
1002	Folder Description	Multiple-line Edit Box
1003	Folder Status	Drop Down List
1004	Is Read-only	Check Box
1005	Default Value	Drop Down List
1006	Folder Identifier	Short (<=50) Text (Edit Box)
1007	Folder Access Control	Access Control
1008	Feature Access Control	Access Control
1009	Time Estimation	Time Estimation
1010	Income	Time Estimation

7.5.2 Understanding Specification System Fields

System fields are system-defined data-entry controls that have a designated use in the DevSpec project and are integrated with other TechExcel DevSpec features for specification management or reporting.

Project administrators may add, remove, or customize sixteen system fields to system or custom pages.

Project administrators may make only limited customizations to system controls. Every system control displayed in the Description page belongs to the dropdown list control type.

The values displayed as options in these system controls are drawn from the DevSpec database and cannot be customized by the administrator.

1101	Title	Long (<=255) Text (Edit Box)
1102	Description	Multiple-line Edit Box
1103	Spec Folder Name	Long (<=255) Text (Edit Box)
1104	Spec Folder Path	Long (<=255) Text (Edit Box)
1105	File Attachment	File Attachment
1106	Assigned by	Short (<=50) Text (Edit Box)
1107	Date Assigned	Date-Time
1108	Last Modified by	Short (<=50) Text (Edit Box)
1109	Date Last Modified	Date-Time
1110	Created by	Short (<=50) Text (Edit Box)
1111	Date Created	Date-Time
1112	Spec ID	Short (<=50) Text (Edit Box)
1113	Spec Owner	Drop Down List
1114	Spec State	Drop Down List
1115	Spec Substate	Drop Down List
1117	Cost Estimate	Time Estimation
1119	Product/Version	Product/Version
1120	Option	Option
1121	Spec Type	Drop Down List
1122	File Attachment	File Attachment
1123	External ID	Long (<=255) Text (Edit Box)
1124	Implementation Module	Implementation Module

7.5.3 Understanding Requirement Folder System Fields

System fields are system-defined property fields that have a designated use in the DevSpec project— the data is integrated with other TechExcel DevSpec features for requirement management or reporting.

Each requirement folder system field is represented by a data-entry control in the DevSpec client that enables project members to manage and track data. Project administrators may make only limited customizations to system controls.

- The title of the corresponding data-entry control may be customized.
- The Folder Description field may be customized to support spell checking, time stamping, and HTML tags.

The values displayed as options in these system controls are drawn from the DevSpec database and cannot be customized by the administrator.

1001	Folder Name	Long (<=255) Text (Edit Box)
1002	Folder Description	Multiple-line Edit Box
1003	Folder Status	Drop Down List
1004	Is Read-only	Check Box
1005	Default Value	Drop Down List
1006	Folder Identifier	Short (<=50) Text (Edit Box)
1007	Folder Access Control	Access Control
1008	Feature Access Control	Access Control
1009	Time Estimation	Time Estimation
1010	Income	Time Estimation
1011	Document	Document

7.5.4 Understanding Requirement System Fields

System fields are system-defined data-entry controls that have a designated use in the DevSpec project and are integrated with other TechExcel DevSpec features for requirement management or reporting.

Project administrators may add, remove, or customize twenty system fields to system or custom pages.

Project administrators may make only limited customizations to system controls. The values displayed as options in these system controls are drawn from the DevSpec database and cannot be customized by the administrator.

1101	Title	Long (<=255) Text (Edit Box)
1102	Description	Multiple-line Edit Box
1103	Requirement Folder Name	Long (<=255) Text (Edit Box)
1104	Requirement Folder Path	Long (<=255) Text (Edit Box)
1105	Document	Document
1106	Assigned by	Short (<=50) Text (Edit Box)
1107	Date Assigned	Date-Time
1108	Last Modified by	Short (<=50) Text (Edit Box)
1109	Date Last Modified	Date-Time
1110	Created by	Short (<=50) Text (Edit Box)
1111	Date Created	Date-Time
1112	Requirement ID	Short (<=50) Text (Edit Box)
1113	Requirement Owner	Drop Down List
1114	Requirement State	Drop Down List
1115	Requirement Sub State	Drop Down List
1117	Time Estimation	Time Estimation
1118	Income	Time Estimation
1119	Product/Version	Product/Version
1120	Option	Option
1121	Requirement Type	Drop Down List
1122	File Attachment	File Attachment
1123	External ID	Long (<=255) Text (Edit Box)

7.5.5 Understanding Change Request Folder System Fields

System fields are system-defined property fields that have a designated use in the DevSpec project— the data is integrated with other TechExcel DevSpec features for requirement management or reporting.

Each requirement folder system field is represented by a data-entry control in the DevSpec client that enables project members to manage and track data. Project administrators may make only limited customizations to system controls.

- The title of the corresponding data-entry control may be customized.
- The Folder Description field may be customized to support spell checking, time stamping, and HTML tags.

The values displayed as options in these system controls are drawn from the DevSpec database and cannot be customized by the administrator.

1001	Folder Name	Long (<=255) Text (Edit Box)
1002	Folder Description	Multiple-line Edit Box

7.5.6 Understanding Change Request System Fields

System fields are system-defined data-entry controls that have a designated use in the DevSpec project and are integrated with other TechExcel DevSpec features for change request management or reporting.

Project administrators may make only limited customizations to system controls. Every system control displayed in the Description page belongs to the dropdown list control type.

The values displayed as options in these system controls are drawn from the DevSpec database and cannot be customized by the administrators

1101	Title	Long (<=255) Text (Edit Box)
1102	Description	Multiple-line Edit Box
1106	Assigned by	Short (<=50) Text (Edit Box)
1107	Date Assigned	Date-Time
1108	Last Modified by	Short (<=50) Text (Edit Box)
1109	Date Last Modified	Date-Time
1110	Created by	Short (<=50) Text (Edit Box)
1111	Date Created	Date-Time
1112	Change Request ID	Short (<=50) Text (Edit Box)
1113	Change Request Owner	Drop Down List
1114	Change Request State	Drop Down List
1115	Cost Impact	Time Estimation
1116	Income Impact	Time Estimation

7.5.7 Understanding Document and File Attachment System Controls

In DevSpec, work item-specific files and URLs may be attached to work items (requirements or specifications) using the Document and File Attachment system fields.

7.5.7.1 Document System Control (1105)

The Document system control enables the user to attach a single file, URL, or document based on a requirement template or specification template.

A template is a predefined requirement or specification that serves as a blueprint for defining requirements or specifications of a particular type quickly and easily. Templates enable project members may save the time and effort that is needed to define many requirement property values from scratch.

7.5.7.2 File Attachment System Control (1121)

The File Attachment control enables the user to attach a multiple files or URLs to a work item.

The File Attachment system control shows the title of the file name, the server version number, the client version number, the date the file was last modified, and the user name of the project team member who made the changes.

7.6 Administering Custom Fields and Controls

A control is a graphic element displayed in a window, panel, or dialog box that enables the user to define, update, and track data. Each control represents a property—system

field or custom field—of a business object. In DevSpec, controls enable project member to manage requirements, specifications, change requests, and events.

Custom controls are administrator-defined data-entry controls that may be added to system pages and custom pages. Project administrators may define appropriate field labels, control types, and predefined values for each control. Data collected in custom fields may be imported or exported, searched, and used to run reports.

7.6.1 Understanding Control Types

In DevSpec, every control is defined by its control type and represents the property— system field or custom field—of a business object.

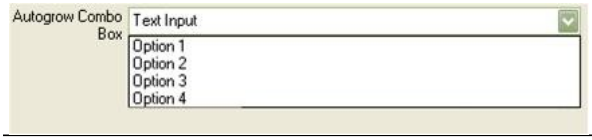
A control type defines the appearance and operation of a control and defines how the graphic element may be used to manage project data. Common control types include text boxes, dropdown lists, and check boxes.

All custom controls are fully customizable. The project administrator may define the title of the control. The number of customizations that may be made to a control depend on its control type.

DevSpec supports 16 different control types: combo boxes, date-time controls, dropdown lists, edit boxes, multiple selection controls, and multiple-line edit boxes.

Autogrow Combo Box: An autogrow combo box combines a dropdown list with a short text box to enable users to choose between inputting a value directly into the control or selecting a value from an existing list of options.

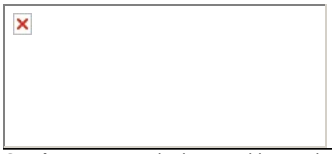
Every value entered into the text box is added to the list of options that are available to subsequent users. Compare with Combo Box.



Calculation Field:A calculation control is a read-only control that displays a*calculation value*—the sum, difference, product, or quotient of two or more factors based on an administrator-defined formula.

The calculation value displayed in the calculation control is defined by the calculation factors, the calculation formula, and the calculation format

Check Box:A check box enables the user to choose between two mutually exclusive options: selected (true) or clear (false).



Combo Box:A combo box combines a dropdown list with a text box to enable users to choose between inputting a value directly into the control or selecting a value from an existing list of options."

Values entered into the text box are*not*added to the list of options that are available to subsequent users. Compare with Autogrow Combo Box.

Date-Time Control:A date-time control enables the user to select a specific date in a popup calendar.

Decimal Edit Box

Definable Combo Box

Dropdown List:A dropdown list enables the user to select a value from a list of predefined options.

HTML Text Message

Integer Edit Box

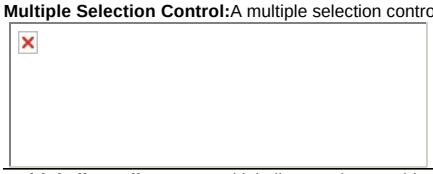
Long (<=255) Text Box

Static Label

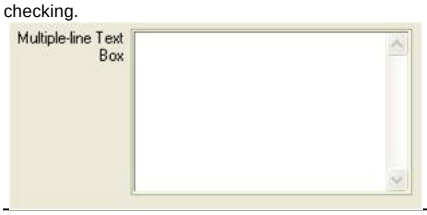
Short (<=50) Text Box

Short Text Box:A short text box enables the user to input a single line of text.

Multiple Selection Control:A multiple selection control enables the user to select multiple values in a predefined list of options.



Multiple-line Edit Box:A multiple-line text box enables the user to input multiple-lines of text." In DevSpec, the control may support HTML tags, time stamping, and spell checking.



Short (<=50) Edit Box:A short text box enables the user to input a single line of text. Static Label A static label is a read-only control that enables the administrator to provide supplementary instructions or information.

Voting Grid:A voting grid tracks data in a tabular list of rows and columns and enables the user to enter integer, decimal, and currency values.

7.6.2 Customizing Check Box Controls

A check box control enables the user to choose between two mutually exclusive options: selected (true) or clear (false).

The field label of the check box is customizable.

To customize a date-time control:

1. Select a Field Design page in the DevSuite Admin client.
2. Create a custom control or edit an existing control.
   - To create a new custom control, click the New button.
   - To edit an existing custom control, select the control in the Custom Fields list and click the Edit button. The Custom Control dialog box appears.
3. Select the Check Box option from the Control Type dropdown list.
4. Define the title of the control in the Field Label text box.
5. Click the OK button.

   The Custom Control dialog box closes and the custom control is displayed in the Custom Fields list.

7.6.3 Customizing Date-Time Controls

A date-time control enables the user to select a date (month, date, and year) and time in a popup calendar.

Figure 6-3: Date-Time Control in the DevSpec Client

In its inactive state the control is displayed as a text box with a small button (the ellipsis button)." Clicking the ellipsis button displays the popup calendar that enables the user to browse and select the date.

To customize a date-time control:

1. Select a Field Design page in the DevSuite Admin client.
2. Create a custom control or edit an existing control.
   - To create a new custom control, click the New button.
   - To edit an existing custom control, select the control in the Custom Fields list.and click the Edit button.

   The Custom Control dialog box appears.

3 Select the Date-Time option in the Control Type dropdown list.

4 Define the title of the control in the Field Label text box.

5 Define the scope of the control.

Date-Time controls may track dates or dates and times.

- To enable the control to track the date only, select the Date option button.

Chapter 8 Event Administration

8 Chapter 7- Event Administration

In this chapter:

- Understanding DevSpec Events
- Administering DevSpec Event Templates
- Administering Event Workflow
- Administering Event Management Privileges

8.1 Understanding DevSpec Events

Throughout the project planning process, project managers may schedule and manage team meetings as events. An event is a management task that facilitates communication and collaboration between project stakeholders. Typical events include brainstorming sessions, team meetings, design reviews, management reviews, presentations, and product demos.

In DevSpec, events are the vehicle that drive the planning process and ensure the quality of the end product. Regular team meetings scheduled within DevSpec enable project managers ensure that all project team members are working together to meet project objectives.

Every DevSpec event is based on an administrator-defined*event template*. An event template is a blueprint for creating a specific type of event—such as a meeting, presentation, or demo—that defines the business rules that determine how that event is managed in the project.

Event administration, therefore, is the largely a matter of creating event templates, defining the rules that manage events based on that template, and determining when events based on that event template may be created in the project.

8.2 Administering DevSpec Event Templates

An event template defines the business rules that are used to manage the events based on that template including the applicable owners, account type-based access controls, and workflow settings.

Using controls in the Event Template page, project administrators may create and define multiple event templates.

For each event template created in the Event Template page, the project administrator must define appropriate properties, access controls, an event group, and event workflow rules.

8.2.1 Defining Event Templates

Each event template represents a specific type of event—such as a meeting, presentation, or demo—and defines who may own the event, who may attend the event, and how the event is managed in workflow.

To add an event template:

1 Click the New button in the Event Template page of the Event subfolder.

The Add a New Event Template dialog box appears.

2 Define a unique, descriptive name for the event template in the Name text box control.

The title of the event template is inherited by every event based on that template and is the default title for those event instances.

3 Define a brief description of the event in the Description field.

Event template descriptions enable project manager to describe the purpose and functionality of an event template.

4 Click the OK button.

The Add a New Event Template dialog box closes. The event template is displayed in the Events folder of the Event Templates tree control.

To delete event templates:

1 Select the Event Template page in the Admin Tree window.

2 Select an event template in the Event Templates list.

3 Click the Delete button in the Event Templates area.

A confirmation dialog box appears.

4 Click the Yes button.

The event template disappears from the Event Templates tree control.

8.2.2 Defining Default Event Titles and Descriptions

Every event is defined by a title and a description of the event. The title and description of events are tracked in two data-entry controls: the Title control and the Description control. Project administrators may define default event titles and event descriptions for each event template. Every event that is based on that event template inherits the predefined title and description.

To define default event titles and descriptions:

1 Select an event template in the Event Templates tree list control of the Event Templates page.

2 Define the title of the event in the Title control.

Every event based on the event template is automatically given the title defined in the event template.

3 Define the description of the event in the Description control.

Every event based on the event template automatically inherits the description defined in the event template.

4 Click the OK button.

8.2.3 Defining Default Event Workflow States

In the DevSpec client, project members may track the progress of a event in the State dropdown list. Each event state represents a particular stage of the life cycle that defines an event from initial creation to its closure and one or more intermediary steps.

The default workflow state is the event state that automatically defines newly created events. Project administrators may define default event workflow states for each event template

Every event that is based on that event template inherits all administrator-defined event settings including the default event state. Project administrators may customize the title of data-entry control, define a default state, and define the options that are available in the State dropdown list in each event template.

8.2.4 Defining Default Event Owners

In the DevSpec client, project members may track the current owner of events in the Owner dropdown list control. The default event owner is the project member that is automatically assigned ownership of the event as soon as it is created.

Project administrators may define default event owners for each event template. Every event that is based on that event template inherits all administrator-defined event settings including the default event owner from that template.

To define a default event owner:

1 Select an event template in the Event Templates tree list control of the Event Templates page.

2 Double-click the Owner control in the Properties tab.

The Event Templates dialog box appears.

3 Select an option from the Default Owner dropdown list.

The Default Owner dropdown list display all project members and administrator-defined group folders, as well as the {Parent Issue Owner} and {Parent Issue Submitter} system variables.

4 Click the OK button.

The Event Templates dialog box closes.

8.2.5 Defining Event Template Scope

In DevSpec, the scope of an event template may be restricted to a subset of work items based on administrator-defined applicable event rules.

An event template may be used to create an event associated with a work item only if that event template has been designated as applicable to the work item type. Event templates may be defined as applicable to four distinct work items: requirements, specifications, requirement changes, and specification changes.

To define an event template applicable to a work item type:

1 Select Advanced Feature > Event in the tree panel.

2 Select an event template in the Event Templates tree list.

3 Select the Applicable Event tab.

4 Select one or more work item types.

The Applicable Event tab displays four work item types: requirements, specifications, requirement changes, and specification changes.

8.3 Administering Event Workflow

Workflow is a business process used to manage tasks within a project. Workflow rules enable project managers to define and track the flow of work between individuals and teams. DevSpec event workflow enable administrators to define sophisticated workflow rules to manage and track issue subtasks.

Event workflow defines the life cycle of a event. Event workflow states represent the various stages of the event life cycle from its initial creation to its closure and one or more intermediary steps.

Event workflow enables organizations to independently manage the various subtasks that must be completed before a work item can be forwarded from one issue workflow state to the next. Event workflow rules enable project teams to manage and track development subtasks independently of one another, assign those tasks to different owners, and define separate schedules (start and due dates) for each event.

Like work items, events are managed within a set of administrator-defined workflow rules. The life cycle of an event is composed of open and closed workflow states.

Figure 7-1: Event Workflow within Work Item Workflow

The workflow of events is typically much simpler than that of their parent work items."

For example, the life cycle of a event frequently consists of no more than two event workflow states: an open state and a closed state. In this scenario, the event merely tracks whether a specific subtask has been completed.

The event has two workflow states:

- To be done (open)
- Done (closed)

Alternatively, event workflow may consist of an opening state and multiple closed states to indicate the result of the subtask.

The event has three workflow states:

- To be tested (open)
- Test passed (closed)
- Test failed (closed)

The life cycle of an event may consist of multiple open and closed events. But no transition rules may be defined for event workflow. Transitions exists between every event workflow state in event workflow and consequently an event may be forwarded to any event state within event workflow.

The event has four workflow states:

- (open)
- (open)
- (open)
- (closed)

All event workflow states are defined on a template-by-template basis. The event life cycle and event workflow rules used to manage and track a event instance are determined by event template used to create that event.

8.3.1 Defining Event States

Project administrators may define the event life cycle of an event template in the State tab of the Event Template page in the Events subfolder.

Project administrators may enable and define two levels of event workflow rules in DevSpec projects: event workflow and advanced event workflow.

To add an event state

1. Select an event template in the Template Tree of the Event Template page.
2. Click the New button in the State tab.

   The Event State dialog box appears.
3. Define a unique name for the event state.
4. Define the event state status.
5. Every event state has one event state status:
   - To define an event state as an open event, select the Open status option.
   - To define an event state as a successfully closed event, select the Closed Successfully status option.
   - To define an event state as a failed closed event, select the Closed Failed status option.
6. Click the OK button.

To edit an event state:

1. Select an event template in the Template Tree of the Event Template page.
2. Select a event state in the State tab.
3. Click the Edit button.

   The Event State dialog box appears.
4. Update the name for the event state.
5. Define the event state status.

   Every event state has one event state status:

   - To define an event state as an open event, select the Open status option.
   - To define an event state as a successfully closed event, select the Closed Successfully status option.
   - To define an event state as a failed closed event, select the Closed Failed status option.
6. Click the OK button.

To delete an event state.:

1. Select an event template in the Template Tree of the Event Template page.
2. Select a event state in the State tab.
3. Click the Delete button. A confirmation dialog box appears.
4. Click the Yes button.

8.4 Administering Event Management Privileges

Event access controls define which project members may view, submit, edit, and delete events in a DevSpec project. Access controls enable project administrators to permit or deny a project member to perform a certain task based on their role in the project.

The ability of project members to perform tasks in DevSpec projects is usually determined by their*account type*, but the privilege to specific perform tasks may also be defined by team group membership or relationship between the project member and the event or issue in question.Each event template is defined by five event access controls. Project members may view, submit, edit, or delete events based on the privileges that are granted to their account type in the event template that was used to create the event.

Can Submit: The Can Submit privilege enables project members to submit events based on an event template.

Can Edit: The Can Edit Open Event privilege enables project members to edit closed events based on an event template.

Can Delete: The Can Delete Event privilege enables project members to delete events based on an event template.

All event access controls are defined on a template-by-template basis. Every event instance inherits the event access control settings from the event template that was used to create it.

8.4.1 Defining Event Management Privileges

Event access controls determine which project members may manage events based two factors: the event template that was used to create the event and the role of that project member in the project.

All event access controls are defined on a template-by-template basis. Project administrators may define event access controls for every event template in the project. Event instances inherit access control settings from the template that was used to create them.

The foundation of all DevSpec access controls are account types. The account type defines the role (responsibilities and privileges) that are granted to a project member in a project. Project administrators may grant event access controls to project members based on their account type, team group membership, or one of four system variables:

{Event Submitter}: The{Event Submitter} variable represents the project member that submitted the event.

{Event Owner}: The {Event Owner} variable represents the project member that currently owns the event.

Project administrators may define event access controls for each event template in the User Access tab of the Event Template page in the Events subfolder.

To define event access controls:

1. Select an event template in the Event Templates list.
2. Select the User Access tab.
3. Select an access control option in the Applicable Account Types list of the User Access tab.
4. Select an option in the in the Applicable Account Types list.

   - The Select the All Account Types radio button enables administrators to select all of the account types displayed in the list.
   - The Select the Selected Account Types option radio button enables administrators to select only some of the account types displayed in the list.

Chapter 9 Implementation Integration Administrati

9 Chapter 8- Implementation Integration Administration

In this chapter:

- Understanding DevSpec Iterative Development Management Administration
- Administering DevSuite Development Integration
- Administering Implementation Linking and Time Management
- Administering Specification Backlogs

9.1 Understanding DevSpec Iterative Development Management Administration

- Subprojects
- Applicable Products and Versions
- Baselines
- Specification Backlog
- Implementation Modules
- Implementation Linking
- Specification State Automation

9.2 Administering DevSuite Development Integration

Optional DevSuite integration features include baselines, applicable product and versions, and specification backlog management.

9.2.1 Enabling Baseline Support

In DevSpec, a baseline is a snapshot of a group of work items (requirements and specifications) at a particular point in time enabling the development team to compare the baseline copies of work items with their current copies.

A baseline may represent a grouping of requirements, specifications, or requirements and specifications based on the user-defined criteria. Each baseline captures key reference points for multiple work items including the title, status, owner, and description.

Using controls in the baseline detail view, project team members may compare the a current requirement or specification with a baseline version of that work item

To enable baseline management:

1. Select Project Setting > Baseline in the tree panel.
2. Select the Enable Baselines check box.

9.2.2 Defining Applicable Products and Versions

In DevSuite, the product tree identifies the products, product versions, and builds under development site and provides a common framework for organizing and managing the development of products, versions, and builds.

Every requirements folder may be defined by one or more applicable products and versions. For more information see “Customizing the Applicable Product and Version System Page” on page 54.

To use the product, version, and build structures, the project administrator must identify which products, versions, and builds in the product tree are applicable to their project. Only those release folders that have been defined as “applicable” may be utilized in the project.

To select applicable products, versions, and builds:

1. Select Product Settings > Applicable Product/Version in the tree panel.
2. Click the Select button.

   The Select Product dialog box appears. The Product Tree displays the product categories, products, versions, and builds that have been defined in site.
3. Select all applicable products, versions, and builds.

   The products, versions, and builds selected in the product tree may define DevTrack subproject structures.
4. Click the OK button.

9.3 Administering Implementation Linking and Time Management

In DevSuite, an implementation link is a set of rules that define a relationship— pending, assigned, or closed—between a specification and a development subproject.

Implementation links may be based on specifications alone or pairings of specifications and implementation modules. Implementation link rules define how specifications are tied to development subprojects and tools that enable development teams to better estimate the work that may be completed in an iteration.

DevSpec implementation link administration is the process of defining applicable implementation modules, enabling and defining implementation time management, enabling implementation linking, and adding a custom page that displays the implementation module system control to the specification view.

9.3.1 Enabling Implementation Link Time Management Tools

In DevSuite, implementation link time management tools enable project managers to estimate and measure the time required for a development team to implement the designs defined in a specific specification.

Implementation link time tracking may be based on time estimates or on specification points.

Time Estimates:A time estimate is an estimation of the work hours required to implement the specification designs in a particular module.

Specification Points:A specification point is a tool for rating the relative complexity of a feature—as defined by the specification and the module in which it is to be implemented—so that the team may better estimate the work that may be completed in an iteration.

The value of a development task may be affected by the module in which it is implemented. An implementation module is a tool for organizing a distinct area of development—such as a platform (Windows, Linux), an application (thick client, thin client, smart client), plug-in, tool, widget, or any other development item.

Project team members may define the estimated time and specification points, whenever they add or update an implementation link.

Once specification points are enabled in a project, team members may begin assigning specification points to features in the specification backlog to indicate the estimated effort required to implement the design. Specification points may be used in conjunction with estimated total time and estimated time remaining to gauge the effort required of development activities.

To enable estimated time total time tracking:

1Select Project Setting > Implementation Setting in the tree panel.

2Select the Enable Estimated Time Total check box.

3 Optional: To customize the term used to refer to specification points in the project, input a word in the Refer To Estimated Time Total As text box.

To enable estimated time remaining tracking:

1Select Project Setting > Implementation Setting in the tree panel.

2Select the Enable Estimated Time Remaining check box.

3 Optional: To customize the term used to refer to specification points in the project, input a word in the Estimated Time Remaining As text box.

To enable specification points:

1Select Project Setting > Implementation Setting in the tree panel.

2Select the Enable Specification Points check box.

3 Optional: To customize the term used to refer to specification points in the project, input a word in the Refer To Specification Points As text box.

9.3.2 Defining Implementation Link Options

Implementation linking enables development organizations to schedule the development of the designed product—as defined by a specification—in development subprojects.

Implementation linking may be based solely on specifications or pairings of specifications with an implementation modules

Specification Only:A specification may be assigned to a single development subproject by means of an implementation link.

Specification and Implementation Module:A specification may be linked to multiple development subprojects—one for each applicable implementation module.

If implementation linking is based on specifications and implementations then each specification may define multiple development subprojects— for each applicable implementation module.

To define project-level implementation link preferences:

1Select Project Setting > Implementation Setting in the tree panel.

2Select an implementation linking option:

- To define development subprojects by specifications only, select the Based on Specification Only option button.
- To define development subprojects by specifications and implementation modules, select the Based on Specification and Implementation Modules option button.

To enable pending implementation links to be added to different backlog folders:

1Select Project Setting > Implementation Setting in the tree panel.

2Select the Enable Pending Implementation Links check box.

9.3.3 Defining Applicable Implementation Modules

An implementation module is a tool for organizing a distinct area of development— such as a platform (Windows, Linux), an application (thick client, thin client, smart client), plug-in, tool, widget, or any other development item.

In DevSuite, development projects may be defined by one or more applicable implementation modules. Each development subproject may also be defined by a specification or a combination of a specification and implementation module which defines the conceptual product to be implemented in that subproject.

To define applicable implementation modules:

1Select Project Setting > Implementation Setting in the tree panel.

2Click the Select button.

The Applicable Implementation Module dialog box appears.

3 Select one or more implementation modules in the implementation module list.

4 Click the OK button.

The Applicable Implementation Module dialog box closes.

9.4 Administering Specification Backlogs

In DevSuite, the specification tree may designated as the *specification backlog* for an integrated development project. As the specification backlog, the tree structure defines the that are to be implemented in a project.

The specification backlog defines the requirements of an entire project including customer and team-defined features as a prioritized list of features. The tasks identified in the project backlog are implemented in iterations according to their priority. The specification backlog is primarily a tool for defining how development project team members assign specifications to development subprojects. Specification backlog controls include specification assignment rules and specification state automation rules.

- Specification assignment rules may be used to restrict which specifications may be assigned to a development subproject based on specification states.
- Specification state automation enables development organizations to streamline and control the implementation of specification designs in development subprojects.

9.4.1 Enabling Specification Backlog Management

In DevSuite, the specification backlog identifies all of the specifications—development tasks—that are ready to be scheduled for development. Each specification represents “a unit of work small enough to be completed by a team in one Sprint iteration.”

The specification backlog enables development organizations to control which specifications may be assigned to iterative subprojects—sprints— in development projects using state-based specification assignment rules.

Specification backlog administration is the process of enabling support for backlog management, defining the terms used for backlog management in the project, and defining and managing subproject templates.

To enable specification backlog management:

1 Select Project Setting > Backlog in the tree panel.

2 Select the Enable Specification Backlog check box.

3 **Optional**: To define the term used to refer to the specification backlog in a project, input the term in the Refer to Specification Backlog As text box.

Different terms are used to refer to iterative processes in different development methodologies. In DevSuite, project administrators may customize the terms used to refer to key development terms to suite their individual needs

9.4.2 Defining Spec Management Options for Child Development Projects

In DevSuite development projects, iterative development subprojects may be defined by one or more subprojects—one for each enabled implementation module.

The assignment of specifications to subprojects is performed in the Backlog tab of the DevTrack and DevPlan clients. Depending on the options defined in the parent DevSpec project, development project team members may filter specifications by specification backlog or by specification backlog and defect backlog.

To define how specs are managed in child development projects:

1 Select Project Setting > Backlog in the tree bar.

2 Select option for defining the specification backlog in the project.

- To organize specifications by the subproject tree only, select the Organize by Subproject Tree Only option.
- To organize specifications by the specification backlog and the defect backlog, select the Organize by Subproject Tree Only option.

9.4.3 Defining State-Based Specification Assignment Rules

State-based specification assignment rules enable development organizations to control which specifications are assigned to development subprojects for implementation based on their completeness.

Specification state checking is an optional feature that enables development organizations to control which specifications may linked to development subprojects based on their specification workflow state.

In DevSuite, specifications may be defined as pending or associated subprojects using an implementation link. If specification state checking is enabled, only those specifications in an administrator-defined applicable state may be linked to a pending of assigned development subproject.

To enable specification state checking:

1 Select Workflow > Spec States Automation in the tree bar.

2 Select the Enable Specification Status Check When Planning Implementation check box.

To define applicable specification workflow states:

1 Select Workflow > Spec States Automation in the tree bar. The Available States list displays only specification workflow states.

2 Add or remove applicable specification workflow states.

- To add a specification state, select a spec state in the Available States list and click the Right Arrow button.
- To remove a specification state, select a spec state in the Selected States list and click the Left Arrow button.

9.4.4 Defining State-Based Pending Specification Linking Rules

State-based pending specification linking defines rules for automatically creating pending implementations for all specifications that are in applicable workflow states.

To enable state-based specification planning:

1. Select Workflow > Spec States Automation in the tree bar.
2. Select the Auto Create Pending Modules for Allowed States check box.

9.4.5 Defining Assigned Implementation Link Spec State Automation

Assigned specification state automation defines rules for changing the workflow state of specifications based on assigned implementation links.

To enable and define automated specification state assignment:

1. Select Workflow > Spec States Automation in the tree bar.
2. Select the Auto Set Specification State Check When All Implementation Links are Assigned check box.
3. Select an open specification state in the Select a State dropdown list.

9.4.6 Defining Closed Link Specification State Automation

Closed specification state automation defines rules for changing the workflow state of specifications based on closed implementation links.

To enable and define automated specification state assignment:

1. Select Workflow > Spec States Automation in the tree bar.
2. Select the Auto Set Specification State Check When All Implementation Links are Closed check box.

   Select an open specification state in the Select a State dropdown list.

9.4.7 Administering the Implementation Module Control

In DevSpec, the implementation module system control is a system-defined control type that enables DevSpec project team members to manage and track the assignment of specifications to development subprojects using*implementation links*.

In DevSuite, an implementation link is a set of rules that define a relationship— pending, assigned, or closed—between a specification and a development subproject.

Using controls in an implementation module control, project team members may link specifications to implementation modules and development projects.

Figure 8-1: Implementation Module System Control in Custom Page

Each implementation link may be a pending link, an assigned link, or a closed link.

Pending Link:A pending implementation link ties a specification to a development subproject but does not schedule the implementation of the designs in that specification."

Assigned Link: An assigned implementation link ties a specification to a development subproject.

Closed Link:A closed implementation link indicates that the specification has been implemented in a development project.

To display the implementation module control in the DevSpec client, the system control must be added to a specification view custom page.

Chapter 10 Time and Cost Analysis Administration

10 Chapter 9- Time and Cost Analysis Administration

In this chapter:

- Understanding DevSpec Time, Cost, and Income Estimates
- Administering Project, Time, Cost, and Income Estimation Settings
- Administrating Time, Cost, and Income Estimation Controls

10.1 Understanding DevSpec Time, Cost, and Income Estimates

DevSpec enables development organizations to track the estimated time, cost, and expected income of development efforts.

Using tools in the DevSpec client, project team members may log estimations of the time required, the cost per unit, and the total cost of development efforts.

Figure 9-1: Time and Cost Tracking in the DevSpec Smart Client

All DevSpec time, cost, and income analysis is based on estimates entered in the DevSpec smart client. Entries are tracked in system-defined time estimation controls which may be added system or custom pages.

The time and cost estimates defined and tracked in DevSpec mirror the actual time and costs of development efforts tracked in DevTrack development projects.

The time and cost of development tasks may be calculated on a requirement-by-requirement basis or specification-by-specification basis.

Requirements-Based Time and Cost Estimates: Time and cost estimates that are logged for each requirement in the DevSuite smart client. Time and cost estimates “roll up” into specifications.

Specifications-Based Time and Cost Estimates: Time and cost estimates that are logged for each specification in the DevSuite smart client.

All time and cost analysis tools and structures are defined by a project administrator in the DevSuite Admin client.

Both DevSpec time estimates and DevTrack time and cost entries are based on administrator-defined*time categories*. In DevSpec, time categories enable the development organization to measure the estimated time, cost, and

10.2 Administering Project, Time, Cost, and Income Estimation Settings

Using controls in the Time Settings page, project administrators may enable time, cost, and income estimation, choose time estimation tracking preferences, and identify the time categories that are applicable to the project.

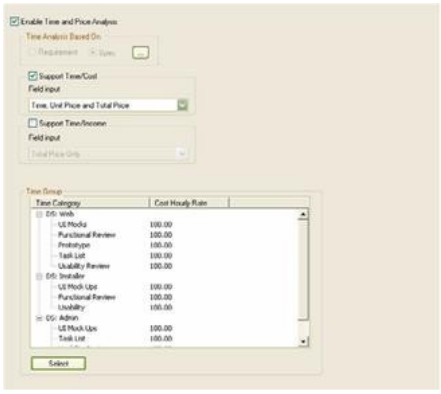


Figure 9-2: Time Settings Page in the Advanced Setting Folder

10.2.1 Enabling Time and Cost Analysis

In DevSpec, time, cost, and income analysis are optional features that must be enabled and configured on a project-by-project basis.

To enable time and cost analysis:

1Select Advanced Features > Time and Price Analysis.

2Select the Support Time/Cost check box.

10.2.2 Defining Time and Cost Analysis Preferences

In each DevSpec project, the time and cost of development tasks may be calculated on a requirement-by-requirement basis or specification-by-specification basis.

Requirements-Based Time and Cost Estimates: Time and cost estimates that are logged for each requirement in the DevSuite smart client. Time and cost estimates “roll up” into specifications.

Specifications-Based Time and Cost Estimates: Time and cost estimates that are logged for each specification in the DevSuite smart client.

DevSpec requirements “roll up” into DevSpec specifications. Time and costs that are tracked on a requirement-by-requirement basis are automatically totalled to provide the estimate time, cost, and income of design tasks.

To enable time analysis preferences:

1Select Advanced Features > Time and Price Analysis.

2Select requirements-based or specifications-based settings.

- To select requirements-based,
- To select specification-based,

10.2.3 Defining Time/Cost Tracking

All DevSpec time, cost, and income analysis is based on estimates entered in the DevSpec smart client by project team members in the course of updating requirements or specifications.

Using tools in the DevSpec client, project team members may log estimations of the time required, the cost per unit, and the total cost of development efforts.

Figure 9-3: Time and Cost Tracking in the DevSpec Smart Client

All time/cost estimates are tracked in system-defined time estimate controls that may be displayed in the DevSpec client." The data that may be entered into each time/cost estimate control is customizable.

The Time/Cost time estimation control may track the time, the total cost, or the time, unit cost, and total cost of a development effort.

Time Only: The Time Only option enables project members to enter time estimates into the Time/Cost estimation control in the DevSpec client.

Total Price Only: The Total Price Only option enables project members to enter cost estimates into the Time/Cost estimation control in the DevSpec client.

Time, Unit Price, and Total Price: The Time, Unit Price, and Total Price option enables project members to enter time, cost per unit, and total cost estimates into the Time/Cost estimation control in the DevSpec client.

All entries are tracked in system-defined time estimation controls which may be added system or custom pages by the project administrator. Depending on project time and cost analysis preferences, the Time/Cost estimation control may be displayed in the DevSpec requirements view or specifications view. For more information see “Administrating Time, Cost, and Income Estimation Controls” .

To define support time/cost tracking:

1Select Advanced Features > Time and Price Analysis.

2Select the Time/Cost check box.

3Select an option in the Time/Cost dropdown list:

The dropdown list displays three options: Time Only, Total Price Only, and Time, Unit Price, and Total Price.

The option selected determines the entries that may be entered into the time/cost estimation control in the DevSpec client.

10.2.4 Defining Time/Income Tracking

All DevSpec time, cost, and income analysis is based on estimates entered in the DevSpec smart client. Entries are tracked in system-defined time estimation controls which may be added system or custom pages.

Using tools in the DevSpec client, project team members may log estimations of the time required, the cost per unit, and the total cost of development efforts.

Figure 9-4: Time and Income Tracking in the DevSpec Smart Client

All time/income estimates are tracked in system-defined time estimate controls that may be displayed in the DevSpec client." The data that may be entered into each time/income estimate control is customizable.

The Time/Income time estimation control may track the time, the total income, or the time, unit income, and total income of a development effort.

Time Only: The Time Only option enables project members to enter time estimates into the Time/Income estimation control in the DevSpec client.

Total Price Only: The Total Price Only option enables project members to enter cost estimates into the Time/ Income estimation control in the DevSpec client.

Time, Unit Price, and Total Price: The Time, Unit Price, and Total Price option enables project members to enter time, cost per unit, and total cost estimates into the Time/

All entries are tracked in system-defined time estimation controls which may be added system or custom pages by the project administrator. Depending on project time and cost analysis preferences, the Time/Income estimation control may be displayed in the DevSpec requirements view or specifications view. For more information see “Administrating Time, Cost, and Income Estimation Controls” on page 184.

To define support time/income tracking:

1. Select Advanced Features > Time and Price Analysis.
2. Select the Enable Time/Income check box.
3. Select an option in the Time/Income dropdown list:

   The dropdown list displays three options: Time Only, Total Price Only, and Time, Unit Price, and Total Price.

   The option selected determines the entries that may be entered into the time/ income estimation control in the DevSpec client.

10.2.5 Selecting Applicable Time Categories

Using controls in the Select Time Category window, project administrators may identify applicable time categories.

Figure 9-5:

To define a time category as applicable:

1. Select Advanced Features > Time and Price Analysis."
2. Click the Select button in the Time Category area.

   The Select Time Category window appears.
3. Select one or more time categories in the tree and click the OK button.

   The Select Time Category window closes.

10.2.6 Updating Time Categories

Using controls in the Select Time Category window, project administrators may identify applicable time categories.

Figure 9-6: Set Project Time Category Dialog Box

To update a time category:

1. Select Advanced Features > Time and Price Analysis."
2. Click the Select button in the Time Category area.

   The Select Time Category window appears.
3. Select a time category.
4. Click the Edit button.
5. The Set Project Time Category dialog box appears.
6. Define the cost rate.

   - To override the default rate, input a value in the Cost Rate text box.
   - To accept the default rate, click the Default button.

7Define the income rate.

- To override the default rate, input a value in the Income Rate text box.
- To accept the default rate, click the Default button.

8Click the OK button.

10.3 Administrating Time, Cost, and Income Estimation Controls

DevSpec time, cost, and income analysis is based on estimates entered in the DevSpec smart client by project team members. Estimated time, cost, and income entries may be made for each specification or requirement tracked in a DevSpec project.

Entries are tracked in system-defined time estimation controls which may be added system or custom pages.

Time, cost, and income estimation administration is the task of adding time estimation controls to appropriate system or custom pages and ensuring that those pages are displayed in the client by adding them to a function page. DevSpec features four system controls of the time estimation control type:

Requirement Time Estimate: The Requirement Time Estimate control may enable project team member to estimate the time, cost per unit, and total cost of development efforts on a requirementbyrequirement basis.

Requirement Income Estimate: The Requirement Income Estimate may enable project team member to estimate the time, income per unit, and total income of development efforts on a requirementbyrequirement basis.

Specification Time Estimate: The Specification Time Estimate may enable project team member to estimate the time, cost per unit, and total cost of development efforts on a specification-byspecification basis.

Specification Income Estimate: The Specification Income Estimate may enable project team member to estimate the time, income per unit, and total income of development efforts on a specificationbyspecification basis.

The time and cost of development tasks may be calculated on a requirement-by-requirement basis or specification-by-specification basis.

10.3.1 Administering Requirement Estimation Controls

DevSpec time, cost, and income analysis is based on estimates entered in the DevSpec smart client by project team members. Estimated time, cost, and income entries may be made for each specification or requirement tracked in a DevSpec project.

In DevSpec project in which time, cost, and income estimations are defined on a requirement-by-requirement basis, estimations are tracked in two time estimate controls.

1117	Requirement Time Estimate	The Requirement Time Estimate control may enable project team member to estimate the time, cost per unit, and total cost of development efforts on a requirementbyrequirement basis.
1118	Requirement Income Estimate	The Requirement Income Estimate may enable project team member to estimate the time, income per unit, and total income of development efforts on a requirementbyrequirement basis.

Time, cost, and income estimation administration is the task of adding time estimation controls to appropriate system or custom pages and ensuring that those pages are displayed in the client by adding them to a function page.

To administer requirement estimation controls:

1To enable time estimation support in the project, select the Support Time/Cost check box in the Time and Price Analysis page of the Advanced Features folder.

2To enable requirements-based time analysis, select the ellipsis button and select the Requirement option button in the dialog box. For more information see “Defining Time and Cost Analysis Preferences” on page 180.

3To enable time/cost time analysis support, select the Support Time/Cost option button and select a tracking option in the dropdown list.

The Time/Cost dropdown list displays three options: Time Only, Total Price Only, and Time, Unit Price, and Total Price.

The option selected determines the entries that may be entered into the time/cost estimation control in the DevSpec client. For more information see “Defining Time/Cost Tracking” on page 181.

4 Options: To enable time/income time analysis support, select the Support Time/ Income option button and select a tracking option in the dropdown list.

The Time/Income dropdown list displays three options: Time Only, Total Price Only, and Time, Unit Price, and Total Price.

The option selected determines the entries that may be entered into the time/ income estimation control in the DevSpec client. For more information see “Defining Time/Income Tracking” on page 182.

5Add time estimate controls to a system page or custom page and resize to display in the DevSpec smart client.

- The Time/Cost Estimate control (1117) requires that time/cost estimation is enabled and configured in the project.
- The Time/Income Estimate control (1118) requires that time/cost estimation is enabled and configured in the project.

10.3.2 Administering Specification Estimation Controls

DevSpec time, cost, and income analysis is based on estimates entered in the DevSpec smart client by project team members. Estimated time, cost, and income entries may be made for each specification or requirement tracked in a DevSpec project.

In DevSpec project in which time, cost, and income estimations are defined on a requirement-by-requirement basis, estimations are tracked in two time estimate controls.

1117	Specification Time Estimate	The Specification Time Estimate may enable project team member to estimate the time, cost per unit, and total cost of development efforts on a specification-byspecification basis.
1118	Specification Income Estimate	The Specification Income Estimate may enable project team member to estimate the time, income per unit, and total income of development efforts on a specificationbyspecification basis.

Time, cost, and income estimation administration is the task of adding time estimation controls to appropriate system or custom pages and ensuring that those pages are displayed in the client by adding them to a function page.

To administer specification estimation controls:

1 To enable time estimation support in the project, select the Support Time/Cost check box in the Time and Price Analysis page of the Advanced Features folder.

2 To enable specification-based time analysis, select the ellipsis button and select the Specification option button in the dialog box. For more information see “Defining Time and Cost Analysis Preferences” on page 180.

3 To enable time/cost time analysis support, select the Support Time/Cost option button and select a tracking option in the dropdown list.

The Time/Cost dropdown list displays three options: Time Only, Total Price Only, and Time, Unit Price, and Total Price.

The option selected determines the entries that may be entered into the time/cost estimation control in the DevSpec client. For more information see “Defining Time/Cost Tracking” on page 181.

4 **Options**: To enable time/income time analysis support, select the Support Time/ Income option button and select a tracking option in the dropdown list.

The Time/Income dropdown list displays three options: Time Only, Total Price Only, and Time, Unit Price, and Total Price.

The option selected determines the entries that may be entered into the time/ income estimation control in the DevSpec client. For more information see “Defining Time/Income Tracking” on page 182.

5 Add time estimate controls to a system page or custom page and resize to display in the DevSpec smart client.

- The Time/Cost Estimate control (1117) requires that time/cost estimation is enabled and configured in the project.
- The Time/Income Estimate control (1118) requires that time/cost estimation is enabled and configured in the project.

access level: An access level defines which project members may access, view, edit, create, and delete work items stored in a secured folder. Secured folder access rights may be customized on an account-by-account basis. Using controls in the Access Control tab, the may add and define the access rights granted to individual project team members. Each project member may be assigned one of five different levels of access in a secured folder.

access level conflict resolution rule: In DevSpec, project members are granted access to knowledge by user accounts, account types, and group membership. In the event that a project member is granted multiple access types—one each for their user account, account type, and group memberships—knowledge access conflict resolution rules determine which access rights are granted to the user based on two criteria: **access type prioritization** and **user account prioritization**.

access rights: DevSpec uses access rights and privilege controls to manage workflow and implement security in projects. An access right determines whether a project member may view or edit a page or data-entry control in the client. Access rights may be account type-based and workflow-based. Compare with **privileges**.

access type: An access type is a bundle of access rights that is assigned to project members based on their account type, team group, or user accounts which enable those users to access and manage specification or requirement folders and the knowledge that is linked to items within those folders. See also **folder access level** and **knowledge access level**.

access type prioritization rule: DevSpec folder access conflict resolution rules and knowledge access conflict resolution rules are based on two criteria: access type prioritization and **user account prioritization**. Access type prioritization ensures that the access type that permits the greatest level of access is assigned to a project member. User account prioritization always overrides access type prioritization.

account type: In DevSpec, an account type defines the role of a **user account**—the privileges and responsibilities granted to that user—in a DevSpec project. TechExcel DevSpec uses a account types to coordinate project members in project workflow and to implement security in DevSpec projects.

account type prioritization rule: Account type prioritization rules apply to both folder access level and knowledge access level conflict resolution rules.

applicable owner: An applicable owner is a project member, team group, or group folder that may own a work item—specification, requirement, change request, event or knowledge item— based on administrator-defined workflow rules.

applicable owner rules: Applicable owner rules determine which project members, team groups, or group folders may own items in each item state. Project administrators may define a different set of applicable owners for each item state in the item life cycle.

Baseline: In DevSpec, a baseline is a snapshot of a group of work items (requirements and specifications) at a particular point in time enabling the development team to compare the baseline copies of work items with their current copies. A baseline may represent a grouping of requirements, specifications, or requirements and specifications based on the user-defined criteria.

Build: In DevSpec, a build is a copy of a software program that is in development. Every build is the child of a parent **version** which, in turn, is the child of a parent **product**. Builds, versions, products, and product categories are defined and managed in the **product tree**.

build state: In DevSpec, the life cycle of a build may be defined by administrator-defined build states. Every build state is either an open state or a closed state. New **development issues** may only be added to builds that are in an open build state.

change flagging event: In DevSpec, a change flagging event is development task that links a requirement change or specification change to one or more DevTrack development issues. The links between the change flagging event and the development issues are managed and tracked as change event flagging subtasks.

change requests: In DevSpec, a change request is a package of changes to requirements and specifications that are managed and tracked together in the project. The change request controls the progress of linked requirements and specifications and enables change managers to ensure that a group of **requirement changes** and **specification changes** are committed or abandoned together.

change request states

change request detail panel: In the DevSpec client, the change request detail panel displays detailed about a single change request in two or more tabbed pages including the Description and Change Request History page.

change request list panel: In the DevSpec client, the change request list panel displays high-level information about multiple change request items in a tabular format of rows and columns. Controls in the tool bar enable users to filter and sort the change requests displayed in the list panel by ownership, status, or query.

change request tree panel: In the DevSpec client, the change request tree panel organizes **change requests** into folders. The change request tree shows the relationship between the change requests managed in a DevSpec project.

change request view: In the DevSpec client, the change request view is the primary view for creating, managing, and tracking change requests, requirement changes, and specification

closed release state: An closed release state indicates that the product, version, or build is currently not under development. The product, version, or build may be completed or may still be in the future. Changes may not be made to releases with a closed release state.

closed: committed status: The Closed: Committed status defines a change request workflow state as a closed state and requires that user commit linked specification changes and requirement changes.

closed: abandoned status: The Closed: Abandoned status defines a change request workflow state as a closed state and requires that user abandon linked specification changes and requirement changes.

Control:A control is a graphic element displayed in a window, panel, or dialog box that enables the user to define, update, and track data. Each control represents a property—system field or custom field—of a business object. Every control is defined by its**control type**.

control type:A control type defines the appearance and operation of a control. DevSpec supports seven control types: combo boxes, date-time controls, dropdown lists, edit boxes, multiple selection controls, and multiple-line edit boxes.

custom field:A custom field represents a customized object property. Custom fields enable businesses to manage and track data that is specific to their environment, industry, and organization. Custom fields are represented in the client by custom-defined controls.

custom page:A custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data. Each custom page may display system controls and custom controls. Using custom pages, system controls, and custom controls, a business may create forms that are uniquely designed to manage and track data that is specific to their environment, industry, and organization.

default owner: The default owner of a work item is the user account or group folder that that is assigned*by default*as the owner of that item—specification, requirement, change request, or event—in workflow. In DevSpec, default ownership rules are a time saving device that enable project administrators to define rules for automatically assigning work items to user accounts or group folders in each workflow state. Project teams members may choose to reassign a work item to another applicable owner or do nothing and accept the default owner as the work item owner.

DevSuite Admin client: The DevSuite Admin client provides system and project administrators with a common interface for configuring and customizing TechExcel DevSpec, DevTrack, and DevPlan projects.

document requirement: In DevSpec, a*document requirement*is a special type of requirement that has been generated from an Adobe PDF or Microsoft Word file using an XML-based template. Document requirements provide development organizations a quick method of importing existing requirement documents into DevSpec so that they may be managed and tracked in workflow.

document root directory: The document root directory is a directory where all project documents are stored. The document root directory is defined during the installation of the DevSpec Document Server.

e-mail integration, advanced: Advanced e-mail integration enables project administrators to define multiple incoming e-mail servers and multiple team e-mail accounts. Unique e-mail received event templates and e-mail sent event templates may be defined for each team e-mail account.

e-mail integration, standard: Standard e-mail integration enables project administrators to define*one*incoming mail server and*one*team e-mail account. All incoming e-mail messages generate events based on the same e-mail received event template or e-mail sent event template.

Events: In DevSpec, an event is a management task that facilitates communication and collaboration between project stakeholders so that they may achieve a greater goal. Every event is unique and defined by definite beginning and ending date or time. Every event created in a DevSpec project is an instance of an**event template**and inherits administrator-defined settings from that event template and system-defined business rules from its**event type**.

event template: An event template is a blueprint for creating a specific type of event—such as a meeting, presentation, or demo—that defines the business rules that determine how that event is managed in the project. Every event is based on an administrator-defined*event template*.

event types: An event type represents a class of tasks that project members must perform in the course of managing items. Making phone calls, sending and receiving e-mail, and adding documents to the knowledge base all represent very different processes and are managed by distinct business rules in DevSpec projects.

Filter: In DevSpec, a filter is a simple query that returns development issues matching a defined set of criteria. Records matching the criteria are displayed in the report—all other work items are “filtered out” and not shown.

folder access level:A folder access level defines which project members may access, create, manage, and delete specification and requirement subfolders. Every specification folder and requirement folder in a DevSpec project is assigned a folder access level, which defines the access rights of each user.

folder access conflict resolution rule: In DevSpec, project members are granted access to folders by user accounts, account types, and group membership. In the event that a project member is granted multiple access types—one each for their user account, account type, and group memberships—folder access conflict resolution rules determine which access rights are granted to the user based on two criteria:**access type prioritization**and**user account prioritization**.

function page:A function page is a compound page composed of multiple custom pages and system pages. In the DevSpec client, function pages are displayed as multiple-page forms that enable the user to perform a particular function such as adding or updating a folder or a work item.

group folder:A group folders represents a kind of buffer between the team group and the items assigned to that group. Each group folder is owned by a particular group and privilege protected by three access controls: View, Put, and Get. Project members that have been assigned the appropriate account type may view items assigned to a group folder, add items to a group folder, or forward those items to another project member or group.

implementation link: In DevSuite, an implementation link is a set of rules that define a relationship— pending, assigned, or closed—between a specification and a development subproject. Implementation links may be based on specifications alone or pairings of specifications and implementation modules.

implementation module: An implementation module is a tool for organizing a distinct area of development— such as a platform (Windows, Linux), an application (thick client, thin client, smart client), plug-in, tool, widget, or any other development item.

indicator icons:A indicator icon is an administrator-defined graphic that indicates the current status of the specification based on a single specification property—its owner or workflow state. The significance and appearance of specification indicator icons is customizable and may vary from project to project

invisible field: In DevSpec, an invisible field is a data-entry control is not displayed in the DevSpec client. Data-entry controls may be defined as invisible fields based on account type-based access controls or state-based workflow rules.

invisible page: In DevSpec, an invisible page is a GUI element is not displayed in the DevSpec client. All specification and requirement**system pages**may be defined as invisible pages based on account type-based**access controls**.

item escalation: In DevSpec, items may be “escalated” based on administrator-defined items escalation rules. item escalations help support organizations to ensure that items do not “fall through the cracks” by automatically notifying item stakeholders by e-mail, pager, or mobile phone when no action is taken on an item. If still no action is taken, events may be automatically reassigned to another project member, group, or group folder.

item escalation rule: An item escalation rule is defined by an escalation trigger, optional escalation conditions, an escalation schedule, one or more escalation actions, and escalation subscription rules.

item notification: DevSpec event notification enables development teams to ensure that all appropriate stakeholders are notified (by e-mail, pager, or mobile phone) whenever events are created, updated, or closed in a DevSpec project. DevSpec event notification is based on administrator-defined event notification rules. Event notification rules consist of triggers, conditions, actions, and subscribers.

item notification rule: An item notification rule is defined by a notification trigger, optional notification conditions, notification actions, and notification subscription rules.

item owner: The item owner is the project member that is deemed responsible for processing an item at any given time. Every item is owned b y one-and-only project member. item ownership may be restricted based administrator-defined workflow rules.

item state:item states are administrator-defined workflow states that represent the various stages that an item may pass through in the item life cycle. At any one time during the item life cycle and item belongs to one-and-only one workflow state. Every workflow state represents either an open item status or a closed item status.

item status: The status of an item indicates whether that item is open or closed based on its current*item state*—a stage in the item life cycle or All item states are defined as either open or closed states. Project members may filter the items displayed in the client by item status.

item submitter: The item submitter is the project member that initially submitted an item to the project. item submitters may be defined as potential recipients of notification and escalation actions.

knowledge base:A knowledge base is a KnowledgeWise knowledge management project that is shared by multiple DevSuite projects.

knowledge portal:A knowledge portal provides project members (and employees) with the appropriate privileges to browse and search for project knowledge items (documents, knowledge topics, and HTML links), FAQs, or an integrated third-party knowledge management solution.

Login Manager: The Login Manager displays high-level information the current status and login history of DevSuite project members including login names, login and logout times, machine names, and license types.

License Manager: In DevSuite, the License Manager is a system-level administrative tool that enables administrators to manage user licenses. Using the Login Manager, administrators may connect directly to the TechExcel License Server, register, and load their licenses.

mandatory field: In DevSpec, a mandatory field is a data-entry control that must be defined before an item (such as a specification or requirement) may progress to the next stage in its life cycle. Mandatory fields may be defined on a state-by-state basis.

menu bar: The menu bar organizes DevSpec commands into six different menus: the File menu, Edit menu, View menu, Tool menu, System menu, and Help menu.

notification action: Notification actions are system-generated actions (e-mail messages, pager or mobile phone notifications, or personal folder notifications) that are delivered to project members whenever a notification rule is triggered.

notification condition:A notification condition limits the scope of an item or event notification rule based on item, event, or employee property values. All notification conditions are optional.

notification trigger:A notification trigger identifies the issue management task that triggers an administrator-defined action or set of actions to be performed. Project administrators may choose between six system-defined triggers (New Issue, Forward Issue, Close Issue, Reopen.

open release state: An open release state indicates that the product, version, or build is currently under development. Changes may be made to releases with an open release state.

PDF requirement:A PDF requirement is a requirement that has been generated from an Adobe PDF file using an XML-based PDF template and stored in a PDF requirement folder in the requirements tree window.

potential recipients:A potential recipient of a notification or escalation action is generally a DevSpec project member that is a stakeholder in the item or event that is the subject of the notification or escalation rule. Potential recipients may be identified based system variables, account types, or team group membership. Potential recipients receive notification actions only if they are subscribed to the corresponding notification or escalation rule. See**subscribers**.

PowerPad text editor: The PowerPad text editor is a customized word processing utility that project members may use to create and edit TXT and RTF files in the DevSpec client.

private report:A private report is a customized report that is only available to the report author.

Privilege: DevSpec uses access controls and privilege controls to manage workflow and implement security in projects. Most administrative and management tasks are protected by privileges—**project members**may only perform tasks that are explicitly granted to them through their**account type**. Compare with**access control**.

Product: In DevSpec, a product is an application that is in development. Every**product**is the child of a**product category**(such as a*software suite*) and the parent of one or more**versions**, which represent copies of that product. Products, versions, builds, and product categories are defined and managed in the**product tree**. In the DevSpec client, products and version may be defined as applicable to**specification folders**and**requirement folders**.

product category: In DevSpec, a product category represents a group of products that a business may wish to manage as a group—such as a suite of applications. Every product category consists of one or more**products**. Product categories, products, versions, and builds are defined and managed in the**product tree**.

product state: In DevSpec, the life cycle of a**product**may be defined by administrator-defined product states. Every product state is either an open state or a closed state. New**versions**,**builds**, and**development issues**may only be added to products that are in an open product state.

product tree: The product tree is a hierarchical tree structure that represents the product categories, products, versions, and builds under development in a DevSpec site.

Profile: In DevSuite, a user account profile identifies the**account types**and state-based access that has been granted to a user account across multiple projects.

Project: In DevSuite, a project is a tool that enables project members to organize, manage, and track a distinct area of work. The tools that are available in a project depend on the**project type**and on administrator-defined customizations. A project is defined by its team structure, business rules, and workflow rules.

project administrator: In DevSuite, a project administrator is a project member that has been assigned a project administrator account type.

project member:A project member is a**user account**that has been assigned an account type and user license to work in a particular DevSuite project—a DevSpec project, DevTrack project, DevTime project, or KnowledgeWise project.

project tip: In DevSpec, the project tip window is an administrator-defined message that is displayed in the client workspace when a user logs into a DevSpec project. The window is designed to enable project administrators to provide project team members with information or links to information that will enable them to begin working in a project.

project type: Every DevSuite project is defined by its project type. DevSuite supports five different project types: DevSpec projects, DevTime projects, DevTrack projects, and KnowledgeWise projects.

protected folder:A protected folder represents the more restrictive level of access. A requirement folder type is an administrator-defined set of account type-based access rights that define which project members may create, edit, or delete requirement subfolders.

public folder:A public folder represents the least restrictive level of access. A requirement folder type is an administrator-defined set of account type-based access rights that define which project members may create, edit, or delete requirement subfolders.**public report**A public report is a customized report that is available to all project members.

read-only field: In DevSpec, a read-only field is a data-entry control is displayed but not editable in the DevSpec client. Data-entry controls may be defined as read-only fields based on account type-based access controls or state-based workflow rules.

read-only page: In DevSpec, a read-only page is a GUI element that is displayed but cannot be edited in the DevSpec client. All specification and requirement**system pages**may be defined as read-only pages based on account type-based**access controls**.

release states: Every product, version, and build in the global product tree is identified by its release state. An object may have an open or a closed release state. Objects with a closed release state cannot be updated by DevSpec administrators or DevSpec project members. Open statuses are definable, and system administrators may enable project administrator or project members to edit objects in their projects.

Requirement: In DevSpec, a requirement is a formal request that identifies an attribute, capability, characteristic, or quality that is required of a system that is managed and tracked in the DevSpec client as a requirement work item.

Requirement All Links system page: The Requirement All Links system page enables project members to track, link, and manage the requirements, specifications, and change requests that are associated with a requirement.

requirement attachment:A requirement attachment is a file, Word document, or HTML link that provides the details of a requirement in text, images, or tables. Each requirement item may be linked to a single attachment.

requirement change:A requirement change is an adjustment to a requirement that is tracked as part of a change request. The requirement change is the child of a parent requirement and may be “linked” to one or more change requests.

requirement detail panel: In the DevSpec client, the requirement detail panel displays detailed about a single requirement in six tabbed pages: the Description, History, Versions, Linked Specs, All Links, and Events pages.

requirement folders: In the DevSpec client, products and version may be defined as applicable to**specification folders**and**requirement folders**.

Requirement Events system page: The Requirement Events page enables project members to create, edit, delete, and track the events that are associated with a requirement

requirement ID prefix:A requirement ID prefix is an administrator-defined text string that is automatically concatenated to the front of each requirement ID number. All requirement ID prefixes are defined on a project-by-project basis.

requirement list panel: In the DevSpec client, the requirement list panel displays high-level information about multiple**requirement**items in a tabular format of rows and columns. Controls in the tool bar enable users to filter and sort the requirements displayed in the list panel by ownership, status, or query.**requirement tree panel**In the DevSpec client, the requirement tree panel organizes**requirements**into**requirement folders**. The requirement tree shows the relationship between the requirements managed in a DevSpec project.

requirement view: In the DevSpec client, the requirement view is the primary view for creating, managing, and tracking requirements and requirement attachments.

runtime key: Runtime keys do not replace Web login processes; runtime keys provide an additional level of access control. After passing the appropriate runtime key, user authentication is still required to access a project. Runtime key configuration is an optional features.

secured folder: Secured folder security is defined by an administrator-defined access levels. Within each access level distinct access rights may be granted to project team members based on their account type, team group, or user account

specifications: In DevSpec, a specification represents a formal commitment to implement a feature or enhancement a particular product release. Every specification represents a set of**requirements**and provides the business with a framework for organizing and tracking the data and knowledge needed to implement those requirements.

specification access level:A specification access level defines which project members may access, view, edit, create, and delete the specifications in a specification folder.

specification change:A specification change is an adjustment to a specification that is tracked as part of a change request. The specification change is the child of a parent specification and may be “linked” to one or more change requests.

specification detail panel: In the DevSpec client, the specification detail panel displays detailed about a single specification in six tabbed pages: the Description, History, Versions, Linked Specs, All Links, and Events pages.

specification folders: In the DevSpec client, products and version may be defined as applicable to**specification folders**and**requirement folders**.

specification ID prefix:A specification ID prefix is an administrator-defined text string that is automatically concatenated to the front of each specification ID number. All specification ID prefixes are defined on a project-by-project basis.

specification list panel: The specification list panel displays high-level information about multiple specification items in a tabular format of rows and columns. Controls in the tool bar enable users to filter and sort the specifications displayed in the list panel by ownership, status, or query.

specification points: In DevSuite, a specification point is a tool for rating the relative complexity of a feature—as defined by the specification and the module in which it is to be implemented—so that the team may better estimate the work that may be completed in an iteration.

specification state group: In DevSpec, a specification state group is a set of specification workflow states that have been grouped together for reporting purposes.

specification tree: In the DevSpec client, specifications are managed in a hierarchical tree structure—the specification tree—that defines the framework for development, organizes product specifications in the workspace, implements security, enforces good development practices and accountability, and enables collaboration between distributed teams. In projects using DevSuite blueprint agile the specification tree represents the**specification backlog**.

specification types: In DevSpec, project members may manage three types of specifications: features, enhancements, and bug fixes.

specification view: In the DevSpec client, the specification view is the primary view for creating, managing, and tracking specifications and specification attachments.

standard lookup field: In DevSuite, a standard lookup field is an administrator-defined custom field that may be used to define and track business object data in projects throughout a DevTest site. Each standard lookup field defines a set of field values, which may potentially define that field in the project. In the DevTest clients, the field is represented by a multiple selection control (such as a dropdown list) and the field values are displayed as options within that control.

State: In DevSpec, a state represents a stage in the life cycle of a business object (such as a**specification**,**requirement**, or**change request**). Every workflow state is

defined by its **status**. State-based rules enable businesses to manage items in workflow.

state group: In DevSpec, a state group is a set of workflow **states** that are grouped together for tracking and reporting purposes. State groups may be created that are based on **specification** states or **requirement** states.

Status: In DevSpec, every workflow **state** is defined by its status. The status assigned to a workflow state determines how requirements, specifications, or change requests are managed in that state. Requirement workflow states and specification workflow states may be defined one of two statuses: the open status or the closed status. Change request workflow states may be defined by three statuses: the open status, **closed: committed status**, **closed: abandoned status**.

status Bar: The status bar displays information about the number of items displayed in the list panel.

Subproject: In DevSuite, a subproject is a tool for organizing development issues into smaller, more manageable groupings of issues. Development teams may create subproject folders with subproject folders to organize their work and define subproject statuses, prioritize subprojects, and set due dates for all subprojects. Each subproject has its own description, priority, status, and start and end dates.

system administrator: System administrators are responsible for creating user accounts, defining user account properties, and assigning licenses to DevSpec users.

system administrator account type: A system administrator account type enables a licensed user account to perform system-level administrative tasks in a DevSpec site.

system fields: In DevSpec, a system field is a system-defined field that represents a core object property and which has a designated use in the project— the data tracked in these fields is frequently integrated with other TechExcel DevSpec features for requirement management or reporting, System field data is the foundation of TechExcel business and workflow rules and enables project members to manage and track the objects defined by those properties. Common system fields include the title, ID number, workflow state, and owner fields.

system pages: In DevSpec, a system page is a system-defined form that enables DevSpec users to collect, manage, and track information in the client. System pages feature the built-in functionality of system controls. Unlike custom controls which are fully customizable and may be used to track any type of data, each system control is specifically designed to track a particular set of data that is linked to other controls and functionalities in the DevSpec project.

system runtime key: The system runtime key enables employees to access all TechExcel DevSpec projects

system team group: A system team group is an administrator-defined grouping of users that enables the administrator to represent internal structure of the company—for example, product divisions or geographic divisions—and to define and manage company addresses.

tool bar: The tool bar displays buttons and controls that enable change managers to perform common tasks such as switching between views or filtering the items displayed in the list panel.

user variable: In DevSpec, user variables enable project administrators to define rules for assigning work items—requirements, specifications, change requests, and events—to project members based on the relationships that currently exist between work item stakeholders and those work items. System user variables include the {Submitter System} user variable, {Current Owner} user variable, and {Previous Owner} user variable.

team group: A team group is an administrator-defined collection of **project members**—who do not necessarily share the same **account type**—that share common responsibilities in a DevSpec **project**. Team groups organize support teams into smaller, more useful entities that facilitate the assignment and control of **items** and **events.**

team hierarchy: DevTime

time off type: In DevTime, a time off type defines a general category of employee absences—such as vacation or illness—that may be used to track employee work schedules.

time stamping: A time stamp is a signature that is concatenated to text entered in the data-entry field. DevSpec time stamps log the exact date and time of the entry, the name of the project member entering the data, and the action that accompanied the work

DevTrack Projects Administration

Chapter 1 What's new

Chapter 2 DevTrack Concepts

2 DevTrack Concepts

This chapter provides a broad overview of DevSuite features and modules describing how these tools may be used to enhance application life cycle management processes.

In this chapter:

•Understanding TechExcel DevTrack and DevSuite

2.1 Understanding TechExcel DevTrack and DevSuite

TechExcel DevSuite is a family of integrated application lifecycle management (ALM)tools that place knowledge management —from ideas, to formal specifications, to competitive information to issue resolution and customer insight—at the core of anyproduct development initiative.

The DevSuite knowledge-centric strategy enables improve communication, keep uptodate on changes, and reduce the development cycles so that the business may deliver the right products for the right markets in the shortest possible time.

Figure 1-1: DevSuite Knowledge-Centric Development

DevSuite places knowledge management at the core of all development processes. TechExcel KnowledgeWise provides for the easy and efficient collection and organization of informal ideas, gathered from a wide variety of sources, that area shared across multiple DevSpec, DevTrack, and DevTest projects.

KnowledgeWise projects provide controlled access to documents, improve communication and coordination between distributed development teams, and facilitate the management and sharing of information between development teams and project stakeholders.

TechExcel DevSuite leverages intellectual assets with KnowledgeWise, communicating a clear product vision and tactical execution strategy by linking ideas and customer feedback, specifications, requirements, designs, prototypes, and other documents to specific areas of work.

2.1.1 Understanding DevSuite ALM Solutions

TechExcel DevSuite features five ALM solutions that operate in an n-tier architecture:
TechExcel KnowledgeWise, TechExcel DevTrack, TechExcel DevPlan, TechExcel DevSpec, and TechExcel DevTest.

All DevSuite products share the same core architecture—the DevSuite Database Server, DevSuite Application Server, DevSuite Document Server, and DevSuite Web Services—and are fully integrated enabling every branch of the development organization to communicate and work together.

KnowledgeWise	TechExcel DevSuite leverages intellectual assets with KnowledgeWise, communicating a clear product vision and tactical execution strategy by linking ideas and customer feedback,specifications, requirements, designs, prototypes, and other documents to specific areas of work during a development project.Documents are shared with all resources involved in the executionof the project allowing for an uncompromised vision to direct the path of any development project.
DevPlan	TechExcel DevPlan manages the transformation of concepts into formal strategic plans. DevPlan offers an intuitive planninghierarchy to formalize scope an optimize resource usage,team-based planning and calendaring capabilities. These features enable complete control over all product development projects from design planning to implementation and enables increased team efficiency and collaboration.
DevSpec	TechExcel DevSpec is an integrated requirements management solution that is specifically designed to provide visibility,traceability and validation of your product or project requirements.DevSpec provides a frame work to create new requirements, specifications and features that can be linked to development and testing implementation projects.
DevTrack	TechExcel DevTrack enables development teams to manage every aspect of the development process including issue management, team management, and communications management.
DevTest	TechExcel DevTest is a test management solution that enables test organizations to manage every stage of testing life cycle—from test case design, to test execution, to test analysis. DevTest provides testing groups with the tools they need work more effectively and efficiently, hold down costs, and to deliver higher quality products.

While each element is valuable even when functioning independently, the DevSuite architecture is founded upon the assumption of future scaling and is capable of advanced inter-module integration. Whether used as separate components or as an incorporated set of applications, the DevSuite solutions are ideal for companies at any stage in their product development.

2.1.2 Understanding TechExcel Project Hierarchy

At the core of every DevSuite implementation—a*DevSuite site*— is one or more KnowledgeWise knowledge management projects that organize, manage, and track the knowledge collected in multiple DevSpec, DevTrack, and DevTest projects.

All projects in a DevSuite site are organized into a hierarchical structure that defines the relationships between the projects in a DevSuite site. Every DevSpec or DevTrack project is the child of a parent KnowledgeWise knowledge management project and relies on that project to manage project intelligence.

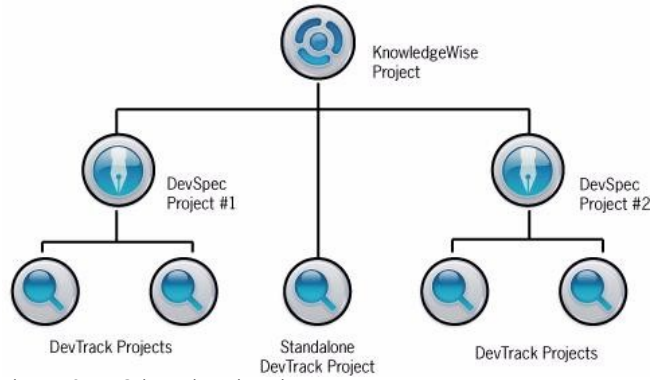


Figure 1-2: DevSuite Project Hierarchy

KnowledgeWise enables development organizations to build a knowledge base of shared knowledge in many different media. All knowledge items may be linked with different areas of work so that internal teams and external customers can search the knowledge base for self-service content based on account type-based privileges and access controls.

KnowledgeWise projects may be accessed by three methods:
(1) the KnowledgeWise Web client,
(2) the knowledge view of a child DevSpec project;
(3) the knowledge view of a child DevTrack project.

2.2 Understanding DevSuite Administration

In DevSuite, administrative tasks are managed at two levels: system administration and project administration.

Site Administration: DevSuite system administration is the processof configuring, administering, and maintaining a DevSuite site. System-level administration activities include maintaining the implementation, creating and defining new projects, managing user accounts, managing licenses, and configuring the DevSuite Server.

Project Administration: DevSuite project administration is the process of defining project-level settings and business objects, customizing project GUIs, configuring workflow rules, managing project team representation and access controls, and integrating add-on modules.

All DevSuite site and project administrative tasks are performed in the DevSuite Admin client. The DevSuite Admin client provides DevTrack, DevPlan, DevSpec, and KnowledgeWise system and project administrators with a single interface for defining, configuring, and administering TechExcel sites and projects.

All DevSuite site and system-level administrative tasks are performed in a System Settings project. Every DevSuite site consists of a single System Settings project. Settings

defined in a system settings project apply to every work project in that site.

All project-level administrative tasks are defined on a project-by-project basis and are only applicable to a single work project. In DevSuite 1, project administrators may define four types of projects: KnowledgeWise projects, DevSpec projects, DevTrack projects, and DevTime projects.

> **Note:** DevTest projects may be defined, configured, and administered in the DevTest Admin client. All other DevSuite projects are managed in the DevSuite Admin client.

2.2.1 Understanding DevSuite Site (System-Level) Administration

DevSuite system administration is the process of configuring, administering, and maintaining a DevSuite site. A site consists of a system settings project and one or more work projects. Every work project in a DevSuite site shares a common database, knowledge base, and other system-level configurations that are defined in the DevSuite Admin client.

System-level configurations define the DevSuite site and the integration of every project within that site.

In DevSuite, system administration is the process of defining system, user, and project settings that define work in a DevSuite site. A DevSuite site is an implementation of a DevSuite server and may include multiple DevTrack, DevSpec, DevPlan, and DevSpec projects. System-level settings are settings that apply to every project in the DevSuite site.

DevSuite system administration tasks fall into seven general areas:

System Administration: System-level settings administrative tasks include the configuration of the DevSuite Server.

User Administration: System-level user administrative tasks include the definition and management of user accounts and project administrators. Optional system-level administrative tasks include multiple site management, division management, and LDAP directory server integration and administration.

Multiple Site Administration: Multiple site administration includes the definition and management of a site family and management of user licenses for all member sites.

License Management: System-level license administrative tasks includelicense management and allocation.

Web Administration: Web administration includes the definition and configuration of DevTrack Web Server settings for every DevTrack project in the site.

E-mail Integration: E-mail integration and administration is the foundation of DevTrack, DevPlan, and DevSpec e-mail notification and escalation.

Branch Development: Release management administration defines the product structures that are shared across DevTrack, DevSpec, and DevPlan project providing a common framework for organizing and managing the development of products, versions, and builds.

Time And Cost Tracking: Time and cost tracking administration defines the project

Administration: management tools that enable a business to measure and track project resources.

System-level user administrative tasks include license management and allocation, user account creation and definition, and the management of project administrators. Optional system-level administrative tasks include multiple site management, division management, and LDAP directory server integration and administration.

2.2.2 Understanding Project Administration

DevSuite project administration is the process of defining project-level settings and business objects, customizing project GUIs, configuring workflow rules, managing project team representation and access controls, and integrating add-on modules.

Project administrators define what is tracked in the project, how it is tracked, and who may track the data. All project-level configurations are project-specific and apply to a single KnowledgeWise, DevSpec, DevTrack, or DevTime project.

Project administration tasks may be broken into five inter-related areas:

- Project Definition and Configuration
- Team Representation Administration
- Work Item Definition and GUI Customization
- Workflow Administration
- Add-on Module Integration

2.3 Understanding DevSuite Sample Projects and Templates

DevSuite builds on TechExcel's commitment to the rapid deployment of its ALM by providing development organizations with the tools they need to quickly configure and implement projects that enforce good development practices and drive the quality of releases.

To facilitate the configuration of a fully integrated site and projects that support your development processes, DevSuite provides developers with tools for evaluating and testing DevSuite features.

Sample Projects: A sample project is a project that provides development organizations with a sandbox for configuring and evaluating DevSuite features and integrations in a risk free environment.Project Templates.A project template is a blueprint for creating a project of a specific type.

Solution Templates: A solution template is a preconfigured project that facilitates the definition of DevSpec projects that implement five proven development methodologies: CMMI, Scrum, SpecDD, TDD, and Iterative Development.

TechExcel strongly recommends that development organizations deploy and test DevSuite in a test environment prior to your DevSuite implementation. A test environment enables development organizations to test project configurations and to train users in new processes prior to the “go live” date in a production environment.

2.3.1 Understanding Sample Projects

In DevSuite, a sample project is a project that provides the development organization with a sandbox for configuring and evaluating features and integrations in a risk free

environment.

DevSuite features ten predefined projects that provide development organizations with a sandbox to experiment with KnowledgeWise, DevSpec, and DevTrack tools and features. All sample projects include sample data, sample project team members, and fully configured workflow rules. Project administrators may freely configure sample projects to represent their development processes.

- Project configurations and settings tried and tested in sample projects may be imported into newly created “live projects”.
- Sample projects provide development organizations with a tool for training new users or to introduce existing users to new features or changes in businesses processes.

Using controls in the DevSuite Admin client, administrators may create additional sample KnowledgeWise, DevSpec, and DevTrack projects.

2.3.2 Understanding Project Templates

In DevSuite, a project template is a blueprint for creating a project of a specific type—KnowledgeWise knowledge management projects, DevSpec requirements management projects, or DevTrack development projects.

Each project template defines a complete set of project-level settings including the definition of all project business objects, workflow rules, team representation, and project integrations.

Project templates may be used to quickly configure new projects whenever they are created in a DevSuite site.

2.3.3 Understanding Solution Templates

TechExcel DevSpec, an integrated requirements management solution, features a complete library of preconfigured solution templates that facilitate the implementation of today's most popular and proven development methodologies.

Each solution template defines DevSuite project settings based on software development best practices so that development organizations may get up and running quickly. Once these business rules are in place, development teams may subsequently customize the project settings to suit their individual needs using TechExcel's award-winning project customization tools.

Using DevSuite, development methodologies longer merely serve as guidelines, but define and direct the daily activities of every team member.

TechExcel provides complete documentation for implementing each methodology that describes how DevSuite workflow and automation may be used to implement and enforce best practices.

Chapter 3 Project Administration

3 Chapter 2- Project Administration

This chapter introduces the DevSuite Admin client workspace and describes basic project administration tasks including the creation of DevTrack development projects and the definition of core project settings.

In this chapter:

- Understanding DevTrack Project Administration
- Understanding the DevSuite Admin Workspace
- Administering DevTrack Projects
- Administering Project Settings

3.1 Understanding DevTrack Project Administration

In DevSuite, administrative tasks are managed at two levels: the system-level and the project-level.

This book describes project-level administrative tasks that apply to TechExcel DevTrack work projects.

In DevSuite, the configuration and customization of work projects is the responsibility of a project administrator. Project administrative tasks include the creation and definition of projects, the definition of project settings and workflow, and team management structures.

3.1.1 Understanding Projects

In DevSuite, a project is a tool for storing, organizing, and managing incident, event, employee, asset, knowledge, and project team data.

Every DevSuite implementation—called a DevSuite site—may consist of multiple types of work projects including DevTrack development projects, DevSpec requirements management projects, KnowledgeWise knowledge management projects, and DevTest QA management projects.

A work project is a tool for storing and managing work item and event data. The work items managed and tracked in a project are project-specific: development issues in DevTrack, test templates and test tasks in DevTest, and specifications, requirements, and change requests in DevTrack. Every work project represents a distinct area of work and is defined by unique team structures (account types, team groups, and group folders) and workflow rules.

In DevSuite, the work projects in a DevSuite site are organized into a hierarchical structure that determines the relationships between KnowledgeWise projects, DevSpec projects, and DevTrack projects.

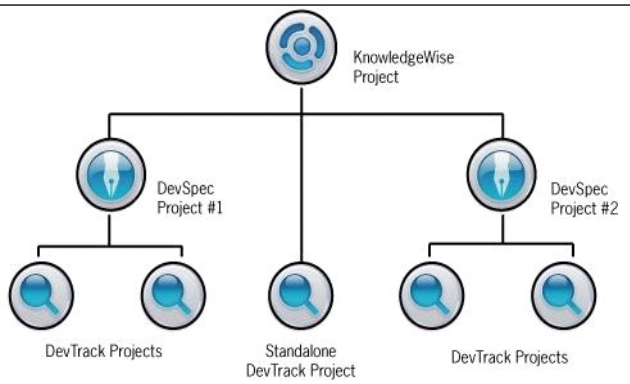


Figure 2-1: DevSuite Project Hierarchy

3.2 Understanding the DevSuite Admin Workspace

In DevSuite, all system- and project-level administrative tasks are performed in the DevSuite Admin client.

When the System Settings project is opened, the DevSuite Admin client displays tools for configuring and administering an entire DevSuite site.

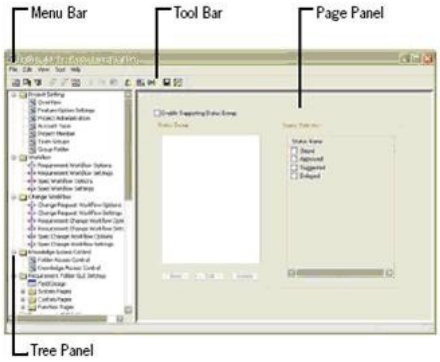


Figure 2-2: DevSuite Admin Client

The DevSuite Admin client organizes the work area into two bars and two panels. Each page is designed to enable an administrator to configure and administer system-level and project-level features.

The DevSuite Admin client is organized into four sections:

Menu Bar: The menu bar displays control buttons that enable the administrator to perform system-level and project-level administrative tasks. Menus include the File, Edit, View, Tool, and Help menus.

Tool Bar:The tool bar displays controls that enable administrators to.perform system-level and project-level administrative tasks.

Tree Panel: The Tree Panel organizes administration tools into a hierarchical structure consisting of folders and folders.

Page Panel: The Page Panel displays form-based administrative tools that enable the administrator to configure and maintain system-level settings.

3.2.1 Understanding Menu Bar Commands

The menu bar displays control buttons that enable the administrator to perform system-level and project-level administrative tasks. Menus include the File, Edit, View, Tool, and Help menus.

The majority of the command buttons displayed in the DevSuite Admin tool bar are project-specific commands. Of particular importance to system administrators are the commands displayed in the Tool menu.

File: The File menu displays commands that enable the administrator to create, open, back up, restore work projects.

Edit: The Edit menu displays commands that enable the administrator to manage text and GUI customization tasks.

View: The View menu displays commands that enable the administrator to display or hide the tool bar, status bar, and alignment bar in the Admin client.

Tool: The Tool menu displays commands that enable the administrator to perform a range of system-level administrative tasks including user, login, and license administration.

Help: The Help menu displays commands that enable the administrator to access online help systems and information about the current build

3.2.2 Understanding the Tool Bar Commands

The tool bar displays controls that enable administrators to.perform system-level and project-level administrative tasks.

The majority of the command buttons displayed in the DevSuite Admin tool bar are project-specific commands. Of particular importance to system administrators are the User Manager button, License Manager button, and the Reload Web Settings buttons.

New Project button	The New Project button enables the administrator to create a new work project.
Open Project button	The Open Project button enables the administrator to open an existing work project.
Close Project button	The Close Project button enables the administrator to close a work project.
Back Up Project button	The Back Up button enables the administrator to back up an existing work project.
Restore Project button	The Restore Project button enables the administrator to restore a closed work project.
Delete Project button	The Delete Project button enables the administrator to delete a work project.
User Manager button	The User Manager button enables the administrator to open the User Manager.
Login Manager button	The Login Manager button enables the administrator to open the Login Manager.
Reload Web Settings button:	The Reload Web Settings button enables the administrator to stop and restart the DevTrack Web Server ensuring that updated configurations are available to web-based clients.
Save Project button	The Save Project button enables the administrator to save changes to project-level configurations.
Upload to Web Server:	The Upload to Web Server button enables the administrator to upload system-level and project-level configurations and settings to the DevSuite Web Server. The DevSuite Web Server must stopped and restarted for administrative changes to be visible to web clients.

3.2.3 Logging into DevSuite Admin

To log into DevSuite Admin:

1Select the DevSuite Admin icon in the Windows Start menu.

By default, the DevSuite Admin icon is added to the DevSuite Admin subfolder within the TechExcel DevSuite folder. The DevSuite Admin Login dialog box appears

2Enter a user name and password.

> **Note:**DevSuite Admin is delivered with a default system administrator defined (User Name: Admin, Password: (blank)). The password is case-sensitive, however the user name is not case-sensitive. For security reasons TechExcel recommends changing the default login setting as soon as possible

3To change web services, select an option from the Web Service dropdown list.

The DevSuite Admin client connects to the DevSuite Application Server through the DevSuite Web Service. Changing the web service enables the client to connect to a different DevSuite Application Server and manage a different DevSuite site.

4Click the OK button.

The DevSuite Admin client opens. By default, the client displays the Project Setup page. To administer a project, open or create a project using tools in the File menu.

3.2.4 Opening DevTrack Projects

The Open Project dialog box displays the System Settings project as well as all live and sample DevTrack, DevSpec, DevPlan, DevTime and KnowledgeWise projects that are available in the DevSuite site.

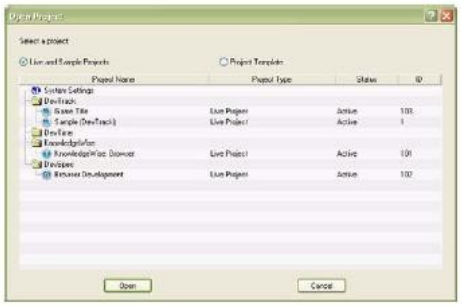


Figure 2-3: Open Project Dialog Box

The Open Project dialog box displays high-level information about every project in the DevSuite site includingthe project type, project status, and projectID number. Each project is identified by a product icon in the project list.

System Settings

KnowledgeWise

DevSpec

DevTrack

For information on opening and administering KnowledgeWise, DevSpec, and DevTrack projects see the appropriate project administration guide.

To open a development project:

1. Select the Open Project command.

   The DevSuite Admin client provides two options foropening projects:

   - Select the Open project command in the File menu.
   - Press CTRL + O.

   The Open Project dialog box appears.

2. Select the DevTrack Project in the project list.
3. Click the OK button.

   The System Settings project is displayed in the workspace.

3.2.5 Closing DevTrack Projects

Administrators may use the Close Project command to close DevTrack projects. The Close Project command may be invoked by two methods:

- Select File > Close Project in the menu bar.
- Press CTRL + S.

3.2.6 Understanding Keyboard Shortcuts

Keyboard shortcuts are combinations of keystrokes that perform predefined functions. Keyboard shortcuts consist of a modifier key and a hot key:

CTRL + X	Cut
CTRL + C	Copy
CTRL + V	Paste
CTRL + N	New Project
CTRL + O	Open Project
CTRL + S	Close Project
CTRL + B	Backup Project
CTRL + R	Restore Project

CTRL + D	Delete Project

3.3 Administering DevTrack Projects

In DevSuite, a project is a tool for storing, organizing, and managing incident, event, employee, asset, knowledge, and project team data.

Every DevSuite implementation—called a*DevSuite site*—may consist of multiple types of work projects including development projects, timesheet projects, KnowledgeWise projects, and DevTest projects.

A work project is a tool for storing and managing work item and event data. The work items managed and tracked in a project are project-specific: development issues in DevTrack, test templates and test tasks in DevTest, and specifications, requirements, and change requests in DevSpec. Every work project represents a distinct area of work and is defined by unique team structures (account types, team groups, and group folders) and workflow rules.

Generally, the configuration and customization of work projects is the responsibility of a project administrator. However, many project administrative tasks are the responsibility of the system administrator. System administrators may create, delete, open, close, back up, and restore work projects and work projects in the General System Settings folder of the DevSuite Admin client.

3.3.1 Creating DevTrack Projects

The New Project dialog box enables administrators to define the project title, the project type, and to import project settings from previously-defined projects. Using controls in the New Project Manager, system administrators may create new projects, sample projects, and project templates.

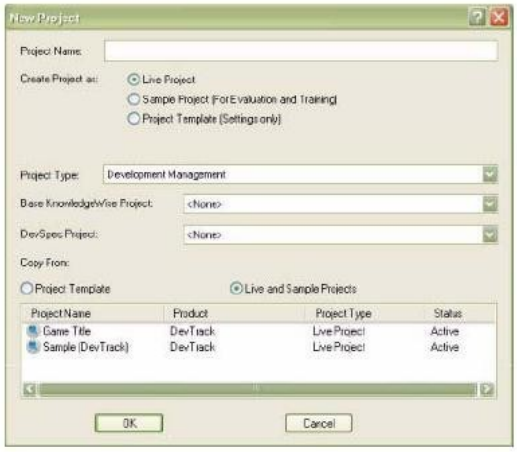


Figure 2-4: Open Project Manager

When creating DevTrack projects administrators may create projects from scratch or they may import project properties from previously defined DevTrack projects. For information on importing project properties from other DevTrack projects see “Copying DevTrack Project Settings” on page 16.

To create a new DevSuite project:

1Select the New Project command.

The New Project command is accessible by three methods:

- Click the New Project button in the tool bar.
- Press CTRL + N.
- Select the New Project command in the File menu.

The New Project manager appears.

2Enter a project title in the Project Name text box. Do not use the following characters in the name of the project:/ \ : ? “<>

3Select a project type.

DevSuite Admin enables system administrators to create live projects, sample projects, and project templates.

- To create a new sample project, select the Sample Project option button.
- To create a new live project, select the Live Project option button.
- To create a new project template, select the Project Template option button.

4Select a project type in the Project Type dropdown list.

The Project Type dropdown list displays four options:

- Development Management
- Timesheet Management
- KnowledgeWise Project
- DevSpec Project

5 Optional: Select a project knowledge base in the Base KnowledgeWise Project dropdown list.

6 Optional: Select a DevTrack project in the DevTrack Project dropdown list.

7 Optional: Copy project settings from an existing project or project template.

- To copy project settings from a project template, select the Project Template option button and select the project template in the project list.
- To copy project settings from an existing live or sample project of the same project type, select the Live or Sample Project option button and select the project in the project list.

8 Click the OK button.

3.3.2 Backing Up DevTrack Projects

Administrators may use the Backup Project command and the Back Up Project dialog box to back up individual DevTrack projects or the entire DevTrack database.

DevTrack stores backups of DevTrack projects in a proprietary DTK file format.Once the administrator define the project information, the administrator may back up either the entire database or only selected projects. DevSuite Admin creates the backup file with the selected project data.

DevSuite Admin stores backed up projects in a proprietary DTK file format.

Note: DevTrack backup feature is a good way to save project properties, but cannot be considered a substitute for regular database server backups

To backup DevTrack projects:

1 Invoke the Back Up Project command.

- Select File > Back Up Project in the menu bar.
- Press CTRL + B. The Back Up Project dialog box appears.

2 Click the Ellipsis button to navigate to the location of the backup file. The Backup dialog box opens.

3 Navigate to the location of the backup files.

4 Enter a name for the DTK file in the File Name field.

5 Select the DevTrack Backup File (DTK) option in the Save as Type dropdown list.

6 Click the Save button.

The Backup dialog box closes.

7 Select a back up option:

- Click the Whole Database radio button to backup all projects in a DevTrack implementation.
- Click the Selected Projects radio button to backup some, but not all projects in a DevTrack implementation. Highlight the projects to be backed up.

8 Click the OK button.

The project is backed up.

3.3.3 Restoring DevTrack Projects

Administrators may use the Restore Project command to restore backed up projects.

DevTrack stores backups of DevTrack projects in a proprietary DTK file format. Administrators may choose to back up selected projects or the entire database.

When restoring backed up DevTrack projects administrators may overwrite existing projects or create new projects based on the backed up projects.

To restore DevTrack projects:

1 Invoke the Restore command.

- Select File > Restore in the menu bar.
- Press CTRL + R.

The Restore Project dialog box appears.

2 Locate the backed up project.

DevTrack stores backups of DevTrack projects in a proprietary DTK file format.

3 Click the Open button.

The DevTrack Restore II dialog box appears. The dialog box displays the Project ID number of the project to be restored.

4 Select the Overwrite radio button in the Options area.

5 Click the OK button.

The project is restored. If the DTK file selected contains multiple backed up, the DevTrack Restore II dialog box appears for each backed up project.

3.3.4 Creating Projects Based from Backed Up Projects

Administrators may use the Restore command to create new DevTrack projects from backed up copies of existing projects.

Whenever an administrator backs up a DevTrack project, DevTrack saves a copy of that project in a proprietary DTK file format.

Administrators may create a new DevTrack project from DTK files.

To create a new project from a backed up project:

1. Invoke the Restore command.
   - Select File > Restore in the menu bar.
   - Press CTRL + R. The Restore Project dialog box appears.
2. Locate the backed up project.

   DevTrack stores backups of DevTrack projects in a proprietary DTK file format.
3. Click the Open button.

   The DevTrack Restore II dialog box appears.
4. Select the Create New project radio button in the Options area.
5. Enter a title in the New Project Name field.
6. Click the OK button.

   The new project is created and the Open Project dialog box appears.

3.3.5 Copying DevTrack Project Settings

System administrators may use the New Project dialog box to create new DevTrack projects based on previously-defined DevTrack projects.

The Copy Settings from an Existing Project command enables administrators to import project properties into the new project. Copying DevTrack project settings cuts down the time required to configure a DevTrack project if the different development projects share similar information.

Project administrators may copy any combination of the following settings to the new project:

- Project members, groups and group folders
- Workflow settings
- Time Tracking settings
- Search and E-mail queries
- Incident routing (autorouting)
- Incident escalation (autoescalation)
- Knowledge view settings
- Event workflow settings
- Interproject settings
- Perforce integration settings
- Issue Template settings

To create DevTrack projects based on existing projects:

1. Select File > New Project.
2. The New Project dialog box appears.
3. Enter a project title in the Project Name field.

   Avoid using the following characters when naming your project/ \ : ? “<>|.
4. Select a project type from the project dropdown list.

   The only project type available in DevTrack Standard Edition is the Development anagement project type. Development Management projects are designed to manage all aspects of defect and feature tracking for any development project.
5. Select the Copy Settings from an Existing Project command.

A list of all existing projects is displayed in the window, and you may select the project with the settings you want to i

Chapter 4 Team Administration

4 Chapter 3- Team Administration

DevTrack team administration is the process of defining team management structures (account types, team groups, and group folders), adding user accounts to the project, and assigning appropriate role and responsibilities to each project member.

In this chapter:

- Understanding Team Administration
- Administering Account Types, Access Controls, and Privileges
- Administering Project Members
- Administering Team Groups
- Administering Group Folders

4.1 Understanding Team Administration

The DevTrack clients provide project team members with*controlled*access to all development work items and activities, facilitate communication and workflow, and improve the productivity and effectiveness of the organization.

All DevTrack development activities are controlled by account typed-based privileges, workflow rules, and access controls. The ability of a DevTrack user account to access projects, views, folders, and pages depends on the “role” (account type) assigned to that user in the project.

The account type concept provides administrators with the flexibility to customize project workflow and security to support their unique business requirements. DevTrack team representation structures enable development organizations to define sophisticated and flexible workflow rules for managing project data and security.

Project-level team administration is the process of defining team management structures (account types, team groups, and group folders), adding user accounts to the project, and assigning appropriate role and responsibilities to each project member.

- An account type is a set of access rights, privileges, and responsibilities that define the role that a user account plays in a project. Every user account is assigned one account type in each project.
- A team group is a collection of project team members that do not necessarily share the same account type but who share common responsibilities in the project.
- A group folder is a mechanism for managing the distribution of development issues in a project.

4.2 Administering Account Types, Access Controls, and Privileges

In DevTrack, privileges and access rights are not granted to user accounts directly, but assigned to administrator-defined*account types*.

An account type defines the role assigned to a user account in a project and determines the projects, views, data, and reports that are available to a project team team member through the DevTrack clients.

Users are assigned an account type when they join a project and assume the rights and privileges granted to that account type—when a user’s responsibility changes, he or she may be assigned a different account type.

In each project, the project administrator is responsible for creating all account types, granting privileges and access rights to those account types, and assigning these account types to each project members based on the role and responsibilities that user performs in the project.

4.2.1 Defining Account Types

DevTrack team representation provides development organizations with maximum flexibility to manage work items in workflow, enforce security, and ensure accountability.

All account types are project-specific and defined by the project administrator. In DevTrack, organizations are free to define the account types that best represent their business processes.

For example, a typical development project might feature five different account types: the Manager account type, the System Architect account type, the Programmer account type, the Tech Support account type, and the QA account type.

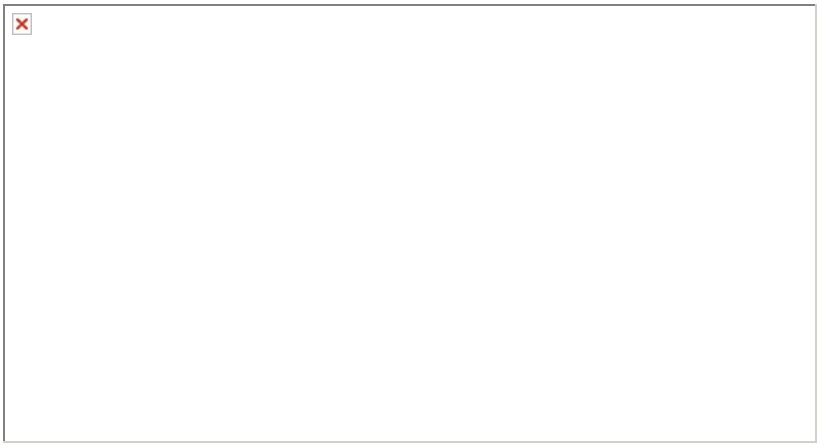


Figure 3-1: Typical DevTrack Project Account Types

Every licensed DevTrack user may be a member of multiple projects and may be assigned a different account type in each project."

To create an account type:

1Select Project Settings > Account Type.

2Click the New button in the Account Type page. The New Account Type dialog box appears.

3Enter a name in the Account Type Name text box.

4 Optional: To copy standard access privilege settings from another account type:

- Select the Copy Account Type Setting From check box.
- Select an account type from the dropdown list.

5Click the OK button.

To rename an account type:

1. Select Project Settings > Account Type.
2. Click the Rename button. The Rename Account Type dialog box appears.
3. Enter a name in the Name field.
4. Click the OK button.

To delete an account type:

1. Select Project Settings > Account Type.
2. Select an account type in the Account Type list.
3. Click the Delete button. The Delete Account Type warning appears.

4.2.2 Enabling Advanced Access Control Support

In DevTrack, access controls determine whether a project team member may view or edit a page or data-entry control in the client. DevTrack pages and fields (data-entry controls) may be defined as read-only or invisible to project members belonging to certain account types.

DevTrack supports two types of account type-based access controls: standard access controls and advanced access controls:

Standard Access Controls: Standard access controls enable the project administrator to define account type-based access rights for pages and fields.

Advanced Access Controls: Advanced access controls enable the project administrator to grant distinct access rights to project team members based on the status of a development issue. Account types may be granted different access rights depending the whether they submit an issue or update an existing issue.

Advanced Access Control management is a DevTrack Enterprise Edition feature. DevTrack Enterprise Edition is an optional upgrade that includes all of the features in the Standard Edition of DevTrack and adds a number of enterprise features including event management and the Beta Customer Portal.

To enable advanced access control support:

1. Select Project Settings > Account Type page in the tree panel.
2. Click the Setup button.

   The Page and Field Access Control dialog box appears.
3. Select the Advanced Access Control option button.
4. Click the OK button.

   The Page and Field Access Control dialog box closes. The On Submission and On Existing Issue tabs are displayed in the Account Type page.

4.2.3 Enabling the New Issue Deletion Option

DevTrack privileges or workflow rules may restrict the ability of project team members to delete development issues based on the user's account type or the issue's workflow state.

Project administrators may use the Allow Users to Delete Issues if They Have Not Been Updated controls to enable project members to delete development issues that they have submitted to a project, if those issues have not been updated subsequent to their initial submission.

To enable new issue deletion:

1. Select Project Settings > Account Type page in the tree panel.
2. Select the Allow Users to Delete Issues if They HaveNot Been Updated check box.

4.2.4 Administering Issue Management Privileges

In DevTrack, a privilege is a right to perform a task that is granted to project members based on their account type. Issue management privileges determine which project team members may submit and manage development issues in a project.

The scope of certain issue management tasks may be limited based on team group membership.

Using controls in the Privileges tab of the Account Type page, project members may grant standard issue management privileges to project members of each account type in a DevTrack project

- Can submit issues
- Can perform forward actions on own issues within own group
- Can perform forward actions on other users' issues within own group
- Can perform forward actions on own issues to all project members
- Can perform forward actions on other users' issues to all project members
- Can perform close actions on issues

Can perform reopen actions on issues

Can delete issues

Can import issues

Can duplicate issues

Can perform Inter-Project submit/copy

Can perform operations on multiple issues

Can edit linked issue

Can define public query/report

Can define public incident template

Can change issue status without workflow (state transition) restriction

Can view other team member's contact information

Can view issues within own group

Can view issues of all groups

Can edit other team member's shortcut

Can add, update, and remove project members

Can publish dictionary changes to server

Can change parent specification

Can set parent specification on submission

Can associate specification with subproject

Can create auto-generated reports

4.2.5 Administering Invisible Pages and Fields (Standard Access Controls)

Invisible page and field access controls enable project administrators to choose the data is displayed to project team members based on their account type.

Pages or data-entry controls that track confidential data may be displayed to project members of one account type—for example, management—and hidden from other account types.

Using controls in the Account Type page, project administrators may define selected fields (data-entry controls) or entire pages as invisible.

Figure 3-2: Invisible Tab in Account Type Page (Detail)

The Invisible tab is only displayed in the Account Type page if standard access controls are implemented in the project." In DevTrack, project administrators may choose to use standard access controls or advanced access controls. For more information see “Enabling Advanced Access Control Support” .

To define invisible pages and fields:

1 Select Project Settings > Account Type page in the tree panel.

2 Select an account type in the Account Types list.

Invisible pages and fields may be uniquely defined for each account type.

3 Select the Invisible tab.

The Invisible tab is only displayed in the Account Type page if Standard Access Controls are enabled in the project.

4 Select a page or field in the Invisible tab.

The Invisible tab displays all pages and fields in a hierarchical tree structure.

- An invisible field is a data-entry control that cannot be viewed or edited in the client.Data-entry controls may be defined as invisible fields based on account type-based access controls or state-based workflow rules.
- An invisible page is a system or custom page that cannot be viewed or edited in the client. Any system or custom page except the Description page may be defined as invisible to project account types. Administrators may, however, make all the individual fields in the Description page invisible.

4.2.6 Administering Read-Only Pages and Fields

Read-only page and field access controls enable project administrators to control which project team members may update project data based on their account type.

Pages or data-entry controls that track confidential data may be editable to project members of one account type—for example, management—and read-only to project members belonging to other account types.

Using controls in the Account Type page, project administrators may define selected fields (data-entry controls) or entire pages as read-only.

Figure 3-3: Read-Only Tab in Account Type Page (Detail)

The Read-Only tab is only displayed in the Account Type page if standard access controls are implemented in the project." In DevTrack, project administrators may choose to use standard access controls or advanced access controls. For more information see “Enabling Advanced Access Control Support” .

To define read-only pages and fields:

1 Select Project Settings > Account Type page in the tree panel.

2 Select an account type in the Account Types list.

Read-only pages and fields may be uniquely defined for each account type.

3 Select the Read-Only tab.

The Read-Only tab is only displayed in the Account Type page if Standard Access Controls are enabled in the project.

4 Select a page or field in the Read-only tab.

The Read-Only tab displays all pages and fields in a hierarchical tree structure.

- A read-only field is a data-entry control that may be viewed, but not edited in the client. Data-entry controls may be defined as read-only fields based on account type-based access controls or state-based workflow rules.
- A read-only page is a system or custom page that cannot be viewed or edited in the client.

4.2.7 Administering On Submission Access Controls

To define On Submission access controls:

1 Select Project Settings > Account Type page in the tree panel.

2 Select an account type in the Account Types list.

3 Select the On Submission tab.

The On Submission tab is only displayed in the Account Type page if Advanced Access Controls are enabled in the project.

4 **Optional**: To define page-level On Submission access controls, select a page in the Page Level Access Control list and click the Edit button. The On Submission Page dialog box appears. Select an option in the Access Control dropdown list and click the OK button.

5 **Optional**: To define field-level On Submission access controls, select a field in the Field Level Access Control list and click the Edit button. The On Submission Field dialog box appears. Select an option in the Access Control dropdown list and click the OK button.

4.2.8 Administering On Existing Issues Access Controls

To define On Existing Issue access controls:

1. Select Project Settings > Account Type page in the tree panel.
2. Select an account type in the Account Types list.
3. Select the On Existing Issue tab.

   The On Existing Issue tab is only displayed in the Account Type page if Advanced Access Controls are enabled in the project.

4. **Optional**: To define page-level On Existing Issue access controls, select a page in the Page Level Access Control list and click the Edit button. The On Existing Issue Page dialog box appears. Select an option in the Access Control dropdown list and click the OK button.
5. **Optional**: To define field-level On Existing Issue access controls, select a field in the Field Level Access Control list and click the Edit button. The On Existing Issue Field dialog box appears. Select an option in the Access Control dropdown list and click the OK button.

4.2.9 Administering DevTrack Knowledge Management Privileges

DevTrack knowledge management privileges enable project team members to access, manage, publish, and update project knowledge in the knowledge view of the DevTrack Windows client

Note: DevTrack knowledge management privileges are only used in projects that are not integrated with a TechExcel KnowledgeWise project. In an integrated DevSuite site all knowledge management privileges are defined in the KnowledgeWise knowledge management project and the permissions defined in the DevTrack project are ignored.

Using controls in the Knowledge Access tab of the Account Type page, project administrators may grant knowledge access privileges to project team members based on their account type.:

Can Publish and Modify Knowledge Items: The Can Publish and Modify Knowledge Items privilege enables account types to create and edit knowledge items.

Can Modify Knowledge Structure privilege: The Can Modify Knowledge Structure privilege enables account types to modify the directory in the Knowledge Tree window.

Can Get Read-only Copy of a File: The Can Get a Read-only Copy of a File privilege enables account types to obtain a read-only version of any knowledge document.

Can Get and Modify a File: The Can Get and Modify a File privilege enables account types to check out documents, modify those documents, and check them back into the knowledgebase.

Can Lock a File: The Can Lock a File privilege enables account types to lock knowledge items. Other project members may still open read-only copies of locked files.

Can Unlock a File: The Can Unlock a File privilege enables account types to unlock files locked by other project members.

Delete a File: The Can Delete a File privilege enables account types to delete files

Close a File: The Can Close a File privilege enables account types to close any file. Closed files may only be accessed by account types that have been granted the Can Reopen a File privilege.

Reopen a File: The Can Reopen a File privilege enables account types to reopen closed files.

Can Get Read-only Copy of a Closed File: The Can Get Read-only Copy of a Closed File privilege enables account types to obtain a read-only versions of closed files.

4.2.10 Administering Report Access Controls

DevTrack features a built-in, robust reporting tool for end users. DevTrack features over 150 predefined reports and graphs, including customizable text reports for issue lists, graphic reports for defect distribution and trends, tabular reports, lifetime analysis reports, Web Query reports, and time tracking reports at the Issue level and project level .

Using controls in the Report Access tab of the Account Types page, project administrators may define account type-based access rights to reports in the DevTrack Windows client and the DevTrack Web client.

Figure 3-4: Report Access Tab

Distinct access rights may be defined for the reports in the DevTrack Windows client and reports in the DevTrack Web client."

4.2.10.1 Windows Client Reporting

In the DevTrack Windows client, report authors may define and run ten types of reports.

- Issue List
- Event List
- Distribution
- Trend
- Project Time Track
- Issue Time Track
- User Defined
- Custom Crystal Report
- Web Report
- Issue Flow Trend

4.2.10.2 Web Client Reporting

In the DevTrack Web client, report authors may define and run more than 20 different types of reports.

- Issue List
- Event List
- Distribution
- Submit / Close Trend
- Perforce
- Tabular
- Life Time Listing
- Life Time Total
- Life Time Average
- Web Query
- Issue Time Track
- Issue Flow: Summary
- Issue Flow: Details
- Issue Flow: Status Group
- Issue Flow: History

Custom Report

Define Custom Report

Flow Trend: Snapshot By Status

Flow Trend: Status Change

AccuRev

Total Productivity

Flow Trend: Summary

Flow Trend: Delta By Status

Administrators may make reports visible to an account type by selecting the check box next to the report type in the Client Access and Web Access lists. Reports may be viewed by an account type only if a black check mark appears in the check box next to the report type.

4.2.11 Administering Subproject Management Privileges

In DevTrack, a privilege is a right to perform a task that is granted to project members based on their account type. Issue management privileges determine which project team members may submit and manage development issues in a project.

Subprojects enable development teams to divide a DevTrack project into smaller, more manageable groupings of issues. Project members may schedule, prioritize, and track those issues using subproject properties and, optionally, subproject workflow.

Can Create Subproject: The Can Create Subproject privilege enables the user to create subprojects in the DevTrack project.

Can Edit Subproject Properties: The Can Edit Subproject Properties privilege enables the user to edit subproject settings.

Can Delete Subproject: The Can Delete Subproject privilege enables the user to delete subprojects.

Can Associate Specifications With Subproject: The Can Associate Specifications With Subproject privilege enables the user to associate DevSpec specification with a subproject.

Can Define Subproject Access Control: The Can Define Subproject Access Control privilege enables the user to define folder access controls in subprojects.

Can Access Subproject Marked As "Protected": The Can Access Subproject Marked As "Protected" privilege enables the user to access subproject folders designated as protected folders.

Can Change Subproject: The Can Change Subproject privilege enables the user to change a subproject.

4.3 Administering Project Members

In DevTrack, all development and project management tasks are managed and tracked by a project team consisting of multiple*project team members.*A project team member is a licensed DevSuite user that has been assigned an account type.

A DevSuite user account cannot access a DevTrack project unless they have been assigned a DevTrack (or DevSuite) license by the system administrator and a project account type by the DevTrack project administrator.

Using controls in the Project Member page, project members may add user accounts to DevTrack projects, assign account types to user accounts, and view project member information.

4.3.1 Adding User Accounts to Projects

To add user accounts to a project:

1Select Project Setting > Project Member in the tree panel.

2Select a user from the Available Members list.

Administrators may use the SHIFT and CTRL keys to select multiple users accounts.

3Click the Add button.

The Account Type dialog box appears.

4Select an account type option from the Account Type dropdown list.

5Click the OK button.

The user account appears in the Members in Project list. Inactive user accounts are displayed with the word Inactive to show they are inactive.

4.3.2 Editing Project Member Account Types

To edit a project member account types:

1Select Project Setting > Project Member in the tree panel.

2 Select a user from the Members in Project list.

3 Select an account type option from the Account Type dropdown list.

4.3.3 Removing User Accounts from Projects

To remove a user account from a project:

1 Select Project Setting > Project Member in the tree panel.

2 Select a user from the Members in Project list.

3 Click the Remove button.

The user account disappears from the Members in Project list and appears in the Available Members list.

4.3.4 Viewing Project Member Information

View project member information

Chapter 5 Development Issue Administration

5 Chapter 4- Development Issue Administration

This chapter describes issue management processes and describes how project administrators may define development issues and customize a graphical user interface that enables the development organization to effectively manage those issues including pages and controls.

In this chapter:

- Understanding DevTrack Client GUI Customization
- Administering the Issue List Panel
- Administering the Issue Detail Panel
- Administering Function Pages
- Administering System Pages
- Administering Custom Pages
- Administering System Fields and Controls
- Administering Custom Fields and Controls
- Customizing Selection and Combination Box Controls
- Understanding DevTrack Field Mapping

5.1 Understanding DevTrack Client GUI Customization

DevTrack project administration is largely the process of defining structures for recording, managing, and tracking work items—requirements, specifications, change requests—in a project.

Customized project management structures are manifested in the DevTrack client as views, panels, folders, and pages.

A view is an interface that displays and organizes data in the client workspace. DevTrack supports three customizable views: the requirements view, the specifications view, and the change request view.

Views enable project members to manage and track different types of data using structures that are specifically designed to manage that data. The building blocks of each view—folders, pages, and controls—may be customized to support custom business processes.

The building blocks of client interface include: controls, folder pages, and GUI pages.

	Control	A control is a GUI element displayed in a folder page or custom page that enables the user to define, update, and track data. DevTrack supports system controls and custom controls. Controls are displayed in system pages and custom pages.
	System Page	A system page is a predefined form that enables project members to collect, manage, and track project data in a project. A system page displays a predefined set of system controls.
	Custom Page	A custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data. A custom page may display system controls and custom controls.
	Function Page	A function page is a compound page composed of multiple custom pages and system pages. In the DevTrack client, function pages are displayed as multiple-page forms that enable the user to perform a particular function such as adding or updating a folder or a requirement.

The function page acts as a frame for displaying the custom and system pages that enable project team members to manage work items.Project customization consists of creating custom pages and controls and adding those pages to function page so that they are displayed in the client.

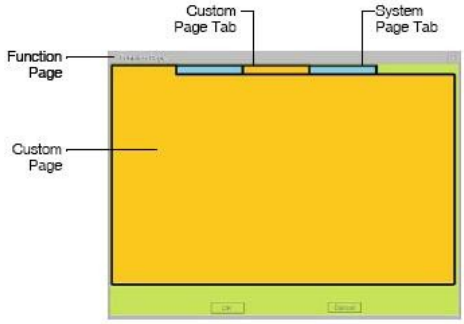


Figure 4-1: Function Page Diagram

5.2 Administering the Issue List Panel

Project administrators may use the Issue List manager to determine which issue properties are displayed by default in the issue list panel of the DevTrack clients, edit the column headers for each data field, and set the default order for the columns in the issue list panel.

The issue list panel provides project members with high-level information about many issues simultaneously. Project members may use controls in the DevTrack clients to filter and sort the issues displayed in the issue list panel by issue properties.

5.2.1 Adding or Removing List Columns

Project administrators may use the Issue List manager to define which list columns are displayed in the issue list panel of the DevTrack clients.

Project administrators may only define the*default*settings for each project. Individual project members may customize the issue list panel in their clients to display or hide issue properties.

To add or remove list columns:

1Select the Issue List manager in the Main View GUI folder of the Admin Tree window.

2Add or remove columns by moving them between the Available Data Fields list and the Select List Columns list.

- To add columns to the Issue list highlight options in the Data Fields list and click the Right arrow to move them to the Issue List View Columns list.
- To add columns to the Issue list highlight options in the Issue List View Columns list and click the Left arrow to move them to the Data Fields list.

3Click the OK button.

4Click the Reset button to set the Issue list view back to the default view.

5.2.2 Editing List Column Headers

Project administrators may use the Issue List page to edit the title of list column headers displayed in the issue list panel of the DevTrack client.

Project administrators may change the column headers for the Issue ID, Progress Status, Current Owner and other columns in the issue list panel. Customizations made to the field are manifested in other pages, including the Current Status page, the New page, and the Search manager.

To edit list column headers:

1Select Main View GUI Design > Issue List in the Tree window.

2Move the field from the Data Fields list to the Issue List View Column list.

3Highlight the field.

4Edit the header label in the Edit Column Header textbox.

5.2.3 Changing Column Order

Project administrators may use the Issue List page to change the order of columns displayed in the issue list panel of the DevTrack client.

To change issue list panel column order:

1Select Main View GUI Design > Issue List in the Tree window.

2Select a field in the Issue List View Columns List.

3Change the placement of the field in the issue list panel.

- Press the Up button to move the field to the left in the issue list panel.
- Press the Down button to move the field to the right in the issue list panel.

5.2.4 Defining List Indicator Icons and Text Colors

Project administrators may use the List View Graphic Indicator Setting manager to customize the appearance of issues in the issue list panel based on issue properties.

Project administrators may customize the following features in the issue list panel:

- Customize the font weight of text of issues in the Issue list based on issue property values.
- Customize the color of text of the issue in the Issue list based on issue property values..
- Customize the icons that identify issue state based on issue property values.

The List View Graphic Indicator Setting manager enables administrators to define the appearance of the issue displayed in the issue list panel.

Figure 4-2: Graphical Indicator Manager

Project administrators may choose to display a list indicator icon in the issue list panel to indicate the current status of issues based on a single issue property." In the List View Graphic Indicator manager administrators may associate a icon with the property values of 11 system fields:

Type	Priority	Component
Version	Platform	Reason
Current Owner	Progress Status	Target Release
Submitted By	Assigned By	

If the administrator associates an icon for each Priority field value (for example, Urgent, High, Medium, Low), issues in the issue list panel are identified by the appropriate icon.

Project administrators may also use colored and boldface text to identify specific issue property values.

Note: Indicator icons may be defined for one and only one issue property at a time. Indicator icons (and indicator text) identifying the value assigned to an issue based on a single issue property. Defining indicator icons for other issue properties immediately deletes all settings defined for the previous issue property.

To define Issue List issue icons:

1. Select the Issue List in the Main View GUI folder of the Admin Tree window.

   The Issue List manage appears.

2. Select the List View Graphic Display Settings button.

   The List View Graphic Indicator Setting manager appears.

3. Select an option from the Indicator Field Selection dropdown list.

   The Field Selection dropdown list displays multiple issue properties.

   The options displayed in the Available Choices list depend on the issue property option selected in the Field Selection dropdown list.

Note: List indicators may be defined for one and only one issue property at a time. Each indicator color is associated with the values of a single issue property.

4. Select a field option in the Available Choices list.

   The Available Choices list displays the options available for the selected issue property as well as multiple system-defined objects.

   Project administrators may not customize the icon displayed for system-defined objects.

5. Click the Select Icon button.

   The Select an Icon dialog box appears.

6. Choose an icon option from the Icon Field area.

7. Click the OK button.

The Select an Icon dialog box closes.

The icon appears next to the option selected in the Available Options list.

8 Click the OK button.

To change the color of issue records:

1 Select the Issue List in the Main View GUI folder of the Admin Tree window.

The Issue List manage appears.

2 Select the List View Graphic Display Settings button.

The List View Graphic Indicator Setting manager appears.

3 Select an option from the Indicator Field Selection dropdown list.

The Field Selection dropdown list displays multiple issue properties. The options displayed in the Available Choices list depend on the issue property option selected in the Field Selection dropdown list.

Note: List indicators may be defined for one and only one issue property at a time. Each indicator color is associated with the values of a single issue property.

4 Select a field option in the Available Choices list.

The Available Choices list displays the options available for the selected issue property as well as multiple system-defined objects. Project administrators may not customize the icon displayed for system-defined objects.

5 To set the issue in boldface type select the Text Bolded check box.

6 To define the color of issue text select an option from the Text Color dropdown list.

7 Click the OK button.

5.2.5 Displaying Note and Event Attachment Icons

Project administrator may enable project members to indicate in the issue list panel whether there is an attachment to a DevTrack Issue.

The General Settings page of the Main View GUI Design contains two controls:

- The Show Event File Attachment control enables project members to see if there is an attachment to a child event.
- The Show Note File Attachment control enables project members to see if there is an attachment to an issue note.

Project members must add the Attachment property to the columns displayed in the issue list panel. Once the Attachment property is added, the Attachment icon is displayed in the issue list panel for all applicable issues.

5.3 Administering the Issue Detail Panel

The issue detail panel displays detailed information about a single DevTrack issue. In the DevTrack Windows client, the issue detail panel is displayed in the bottom third of the Main view. In the DevTrack Web client, the issue detail panel is displayed in the Detail page.

Project administrators may define the order in which pages are displayed in the issue detail panel and whether the pages are editable or read-only in the DevTrack Windows client.

Project administrators may also define how pages are displayed in the Issue Detail page of the DevTrack Web client. The the issue detail panel in the Detail page may display system pages and custom pages as multiple tabs or as sections in a single page.

5.3.1 Defining Issue Detail Panel Page Order

Project administrators may use the Page Order control in the Page Setup page to define the order that pages are displayed in the DevTrack clients.

In the DevTrack Windows client, pages appear as tabs in the issue detail panel of the Main view. The higher the page order the further to the left the page is displayed.

In the DevTrack Web client, pages appear either as tabbed pages or as sections in a single Detail page.

- If the multiple page option is selected, the higher the page order the further to the left the page is displayed.
- If the single page option is selected, the higher the page order the closer to the top of the Detail page the section is displayed.

Project administrators may choose how pages are displayed in the Issue Detail page of the Web client. For more information see See “Customizing the Issue Detail Page of the Web Client” 1.

To define detail page order:

1 Select a page in the Page Order list of the Page Setup page. Only those pages enabled in the Issue Submission Page controls are displayed in the Page Order list.

2 Move pages up or down in order.

- Click the Up button to move a page up in the page order.
- Click the Down button to move a page down in the page order.

5.3.2 Customizing the Issue Detail Panel in the Windows Client

Project administrators may define the issue detail panel of the DevTrack Windows client as editable or read-only using controls in the User Interface Settings page of the Main View GUI Design folder.

Issue detail panel settings for the DevTrack Windows client determine how project members may update issues in DevTrack projects.

- The Multi Tab and Always Editable option defines the pages displayed in the issue detail panel as editable in the DevTrack Windows client.
- The Read Only Perform Update Operation to Edit option defines the Issue Detail pages (the Description page, Current Status page, and all custom pages) as read only in the issue detail panel. Project members must update or forward an issue to edit issue properties.

If the Read Only Perform Update Operation to Edit option is selected, project members “lock” an issue whenever they open that issue for editing. The issue remains locked, and no other project member may edit that issue.for as long as the project member has the Update Issue manager or Forward Issue manager open.

Project administrators may grant privileges to project members that enable them to*override*locked issues.

- The Can Grab Locks on Own Issues from Other Users privilege enables a project member to override locks placed on issues that they own.
- The Can Grab Locks on Other User’s Issues privilege enables a project member to override locks placed on any issue regardless of the issue owner.

All project privileges are granted to project members based on their account type in the Privileges tab of the Account Types page in the Project Settings folder. For more information see See “Administering Issue Management Privileges” on page 28.

5.3.3 Customizing the Issue Detail Page of the Web Client

Project administrators may use controls in the User Interface Settings page to determine whether the pages displayed in the DevTrack Web client are displayed as single page or as a multiple tabbed pages.

- The Multi Tab option means that pages are displayed in the Web client as multiple tabbed pages. Project members may click tabs to switch between the pages displayed in the Web client.
- The One page option means that pages are displayed as a single page in the DevTrack Web client.

5.3.3.1 Multiple page display

The Multi Tab option means that pages are displayed in the Detail page of the DevTrack Web client as multiple tabbed pages. Project members may click tabs to switch between the pages displayed in the DevTrack Web client.

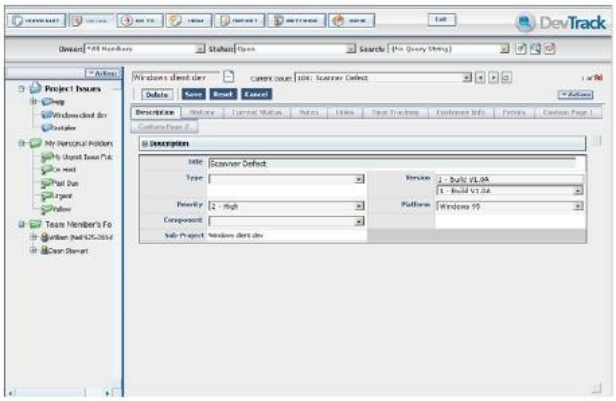


Figure 4-3: Multiple Page Display in the DevTrack Web Client

5.3.3.2 Single page display

The Single page option means that pages are displayed as sections of a single page in the Issue Detail page.

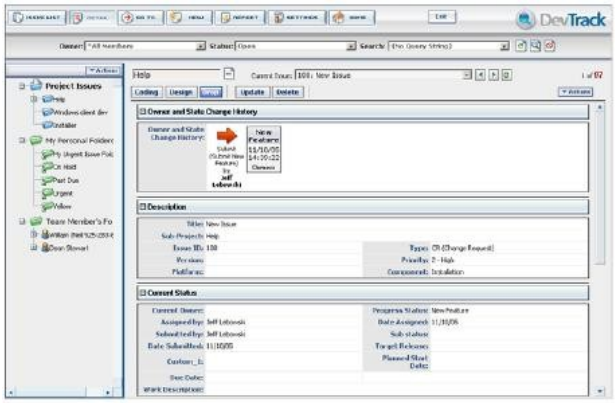


Figure 4-4: Single Page Display in the DevTrack Web Client

5.4 Administering Function Pages

In DevTrack, a function page is a compound page composed of multiple custom pages and system pages. In the DevTrack client, function pages are displayed as multiple-page forms that enable the user to perform a particular function such as adding or updating a folder or a requirement.

DevTrack function pages include the issue detail panel, the New Issue window, the Forward Issue window, and the Close Issue window.

Issue Detail Panel: The issue detail panel displays detailed information about a single DevTrack issue. In the DevTrack Windows client, the issue detail panel is displayed in the bottom third of the Main view. In the DevTrack Web client, the issue detail panel is displayed in the Detail page.

New Issue Window:

Forward Issue Window:

Close Issue Window:

5.4.1 Administering the New Issue Window

The New Issue window is displayed in the DevTrack client whenever a project member submits a new issue.

The controls displayed in the New page enable project members to define issue properties. Most of the fields displayed in the New page are shared with the Description page and the Forward page.

Using controls in the Page Setup page, project administrators may choose which system pages and custom pages are displayed in the New Issue window.

The Description system page is always displayed in the New Issue manager. The Description page contains the Issue Title, Issue Type, Version, and other issue properties that do not generally change during the life cycle of the issue.

In addition to the Description page, project administrators may choose to display up to six different additional pages in the DevTrack client.

Current Status page

Custom page

Custom 2 page

Custom 3 page

Custom 4 page

Custom 5 page

Pages are only displayed in the New Issue manager if a black check mark appears next to the page name.

The custom pages may only be added to the New Issue manager if they have been enabled by the project administrator. The check boxes next to the custom pages are grayed out and cannot be selected until the pages are made visible.The issue properties tracked in these pages depend on administrator defined customizations.

5.4.1.1 system controls

Project administrators may use the New page to customize the title of the New page as displayed in the DevTrack clients, the Progress Status control, and the Assigned To control, and the Substatus control.

Progress Status

Assigned To

Substatus

The Progress Status control is displayed elsewhere in the client: the issue list panel, the Current Status page, and the Search manager. The Progress Status field may be the child of any other field during issue submission.

5.4.2 Defining Manager Options in the Windows Client

Project administrators may use controls in the User Interface Settings page to determine whether the Forward Issue manager, the Update Issue manager, and the Close Issue manager are displayed as multi-page tabbed forms (Detailed Action) or as single pages (Simplified Actions).

- The Detailed Action option enables project members to update many different issue properties whenever they forward, update, or close an issue.
- The Simplified Action option restricts the issue properties that project members may update whenever they forward, update, or close an issue.

5.4.2.1 Detailed action display

The Detailed Action option enables project members to update many different issue properties whenever they forward, update, or close an issue.

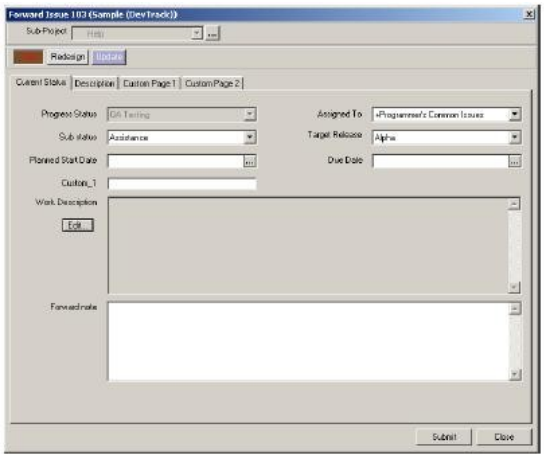


Figure 4-6: Update Issue Manager (Detailed Action)

5.4.2.2 Simplified action display

The Simplified Action option restricts the issue properties that project members may update whenever they forward, update, or close an issue.

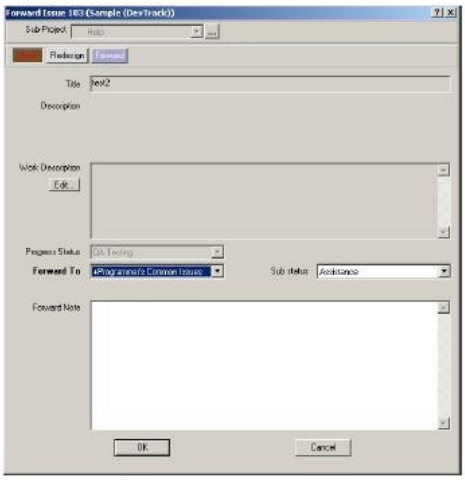


Figure 4-7: Update Issue Manager (Simplified Action)

5.4.3 Administering the Forward Issue Window

Project administrators may use the*Forward page*to customize the graphical user interface of the Forward page in the DevTrack client.

The Forward page is displayed in the DevTrack client whenever a project member forwards an issue to another project member.

Project administrators may rename the page title and edit the Forward To control, the Reason control, and the Forward Note control. All other data-entry controls may be customized using the related field in the Description page

The Forward To control

The Reason control

The Forward Note control

Project administrators may also add up to six predefined controls to the Forward page: Custom Field 1, Custom Field 2, Custom Field 3, Custom Field 4, Custom Field 5, and

Custom Field 6.

The Custom Field controls displayed in the Forward page are placeholders for controls in the Description page, Current Status page, New page, or any of the six custom pages. Project administrators may customize Custom Field controls so that they represent those other controls.

5.4.4 Administering the Close Issue Window

5.5 Administering System Pages

In DevTrack, a system page is a predefined page that enables project members to perform a specific task—such as collecting, managing, and tracking project data.

Each system pages features one or more system controls. A system control is a system-defined control that represents a predefined system field and features control-specific functionality. System fields represent core object properties—such as workflow states and object statuses—and the data tracked in these fields is frequently forms the foundation of TechExcel businesses rules.

Like custom pages, system pages may be added to function pages to build the multiple-page forms that enable project members to manage requirements, specifications, and change requests in project view.

5.5.1 Administering the Issue Description System Page

Using controls in the Description system page, project administrators may customize the field labels of the data-entry controls displayed in the Description page, update their control types, define the options displayed in selection controls, make fields mandatory, invisible, read-only, and change the title of the Description page itself.

Project administrators cannot, however, hide the Description page. The Description page is mandatory in all DevTrack projects and must be displayed in the New Issue window.

Five system controls are mandatory and cannot be hidden in DevTrack projects:

Description Platform Priority

Title Type

5.5.2 Administering the Issue Detail System Page

The Issue Detail page is a complete audit log tracking all changes to an issue and, if the project employs transition-based workflow, provides tools that enables project members to perform actions on DevTrack issues.

Using controls in the Issue Detail page, project administrators may add or remove issue detail reports to the system page, define the order of the reports displayed in the client, and update the title of the page.

Figure 4-8: Issue Detail Page

Using tools provided in DevSuite Admin project administrators can add or remove sections to the report displayed in the Issue Detail pages. Each Issue Detail page presents a report composed of multiple sections.

Change Log

Current Status

Custom pages

Description

DevPlan

Links

Notes

Owner and State Change History

Tracking History

To customize the Issue Detail system page:

1Select Main View GUI Design > Issue Detail in the tree panel.

2Add or remove sections to the Issue Detail page:

- To add a section to the Issue Detail page, select a section in the Available Issue Detail sections list and click the right-arrow button.
- To remove a section from the Issue Detail page, select a section from the Issue Detail Sections Displayed list and click the left-arrow button.

3 Optional: To change the order the issue detail reports are displayed, select a report in the Issue Detail Sections Displayed list.

- Press the Up button to move the section forward.
- Press the Down button to move the section backward.

4 Optional: To define the sort order of issue detail reports, select the Ascending or Descending option button.

5.5.3 Administering the Current Status System Page

In the DevTrack clients, the Current Status system page may be displayed in the issue detail panel, Forward Issue window, and Close Issue window.

Using controls in the Current Status page, project team members may record and track the “status” of development issues including its current workflow state, work description, and other dynamic properties.

Data-entry controls displayed in the Current Status page include:

Progress Status Current Owner Work Description

5.5.4 Administering the Notes System Page

Issue notes and note attachments enable project teams to communicate issue-specific information with one another.

The issue notes provide project teams with a repository for storing issue-specific rather than project-related information. Project members may add any number of notes to each issue. Project members may create and to add notes and file attachments to DevTrack issues in the Notes control.

The Notes page of the main view is displayed in the issue detail panel and is divided into three areas: the Note list, Note detail area, and the File Attachment area.Issue notes and note attachments enable project teams to manage a repository of issuespecific information.

The Notes system page features three unique features:

Timestamping of issue notes

Definition of note types

Note access controls

5.5.5 Administering the Link System Page

Project administrators may use the tools in the*Links page*to enable the Links page in the DevTrack Windows client and to define the link types managed by this page in the DevTrack client application.

The Link page enables project members to link related issues and information to an issue.

- Project administrators may enable the Link page by selecting the Show link page option at the top of the page.
- Project administrators may use the Define Link Type button to create and edit link types. For step-by-step instructions see See “Administering Link Types” on page 119.

5.6 Administering Custom Pages

In DevTrack, a custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data.

Each custom page may display system controls and custom controls. Using custom pages, system controls, and custom controls, a business may create forms that are uniquely designed to manage and track data that is specific to their environment and business.

Figure 4-9: Custom Page

Like system pages, custom pages may be added to function pages to build the multiple-page forms that enable project members to manage requirements, specifications, and change requests in project views."

Project administrators may add and remove system, custom, and multiple-selection controls to custom pages, customize the layout of controls in the page, and define their the tab order.

5.6.1 Creating Custom Pages

A custom page is a form conceived, designed, and built by a project administrator that enables project members to collect, manage, and track project data.

Project administrators may create custom pages for use in the requirements view, specification view, and change request view.

To create a custom page:

1. Right-click a Custom Pages folder and select the Add New Page command in the shortcut menu.

   Custom pages may be created for the requirements folder, requirements GUI, specification folder, specification GUI, and change request folder, and change request GUI
   The custom page is displayed in the workspace.

2. Input the title of the custom page in the Page Display Name text box.

   The display name defines the text that identifies the tabbed page in the client. By default, the custom page is given the title New Page.

3. Add custom controls or system controls.

4. Define the tab order of the controls displayed in the custom page.

5.7 Administering Page Design

5.7.1 Adding Controls to Custom Pages

To add a control to a custom page:

1. Right-click in the custom page and select the Add Control command in the shortcut menu. The Add New Field manager appears.

   The Add New Field manager displays every custom controls and system controls that may be added to the custom page.

2. **Optional**: To filter controls by control type, select an option from the Field Type dropdown list.

   The Field Type dropdown list displays multiple control types including the dropdown list control type, short text box control type, and the field label control type. DevTrack enables administrators to define controls based on 17 different control types.

3. **Optional**: To define how the control is presented in the page, select an option from the Label Alignment dropdown list.
   The Label Alignment dropdown list displays three options:

   - Right Aligned
   - Left Aligned
   - Center Aligned

4. **Optional**: To define control as read only in the page, select the Add It as Read Only check box. A read-only control is a control that is displayed, but cannot be edited.

5. Click the OK button.

5.7.2 Ordering Controls in Custom Pages

The control with the lowest tab order number is the control that is immediately active when the page is accessed in the client.

To define the tab order of controls in a custom page:

1. Select a custom page in a Custom Pages folder.

2. Click the Define Field Tab Order button in the custom page.

3. The Field Tab Order dialog box appears.

   The Field Tab Order dialog box displays a list of the custom controls and system controls that have been added to the custom page.

4. Define the tab order of each control in the page relative to the other controls displayed in the page.

   - To move the control up the list, select the control and click the Up arrow. The lower the tab order number, the earlier the control may be accessed.

   - To move the control down the list, select the control and click the Down arrow. The greater the tab order number, the later the control may be accessed.

5. Click the OK button.

5.7.3 Removing Controls From Custom Pages

To remove a control to a custom page:

1. Select a control in the custom page.

2.

Chapter 6 Development Issue Workflow Administrati

6 Chapter 5- Development Issue Workflow Administration

In DevTrack an administrator may define project workflow based on one of two workflow models: the state-based workflow model and the transition-based workflow model.

This chapter describes transition-based workflow, provides an overview for implementing transition-based workflow in a project, and step-by-step instructions for defining workflow rules in project that uses transition-based workflow.

In this chapter:

- Administering Development Issue Workflow
- Configuring Project Workflow
- Administering Issue Workflow States
- Administering Transitions
- Administering Conditional Transitions
- Administering Interproject Action Automations
- Administering Issue Creation Triggers
- Administering Flowchart Graphics
- Administering Development Issue Workflow Reports

6.1 Administering Development Issue Workflow

A workflow is a business process in which all activities are organized into repeatable patterns of work.

DevTrack issue workflow is conceived as a series of issue*workflow states*that are connected by*transitions*. Each workflow state represents a distinct stage in the issue life cycle. A transition represents a passage between two workflow states.

A workflow is a business process in which all activities are organized into repeatable patterns of work.

These patterns—called*workflow rules*—facilitate the efficient processing of work items by identifying which tasks are applicable to each work item at every point in the development life cycle. Each issue must go through a series of stages (workflow states) from its initial submission to its closure

DevTrack issue workflow is conceived as a series of issue*workflow states*that are connected by*transitions*. Each workflow state represents a distinct stage in the issue life cycle. A transition represents a passage between two workflow states.In DevTrack, workflow is conceived as a series of issue workflow states that are connected by transitions. Each workflow state represents a distinct stage in the issue life cycle. A transition represents a passage between two workflow states.

States: Issue workflow states represent the various the stages that an issue may go through during the development life cycle. Each issue workflow state has an open status or a closed status.

Transitions: Transitions define the path that issues may take in project workflow. Issues may be forwarded from one issue workflow state to another issue workflow state only if a transition exists between those two workflow states.

Incident workflow rules define the tasks that are required to manage issues in a DevTrack project, identify the project members that perform those tasks, and determine which tools are displayed in the DevTrack client.

In DevTrack, project workflow may be based on one of two different workflow models: state-based workflow and transition-based workflow. The tasks required to update and forward issues differ in each model:

- In the state-based workflow model, issues progress from workflow state to workflow state based on explicit changes to the issue workflow state property of development issues by project members. All workflow rules are state-dependent.
- In the transition-based workflow model, issues progress from workflow state to workflow state based on actions performed by project members. Workflow rules may be state-dependent or transition-dependent.

In DevTrack, workflow rules may be*state-dependent*or*transition-dependent*.

- State-dependent workflow rules are defined on a state-by-state basis and control the changes that may be made to an issue in each workflow state.
- Transition-dependent workflow rules are defined on a transition-by-transition basis and control the changes that may be made to an issue in each transition.

Transition-based workflow supports*both*transition-dependent and state-dependent workflow rules.

Transition-based workflow provides project administrators much greater flexibility and control when defining workflow rules in a project.

- Transition-based workflow rules may be both state-dependent and transition-dependent.
- Transition-based workflow supports multiple workflow (subworkflows) by which DevTrack workflow is integrated with subprojects.

DevTrack transition-based workflow administration is the process of enabling support for state-dependent and transition-dependent workflow, defining issue workflow states and transitions and the rules that govern each state and transition.

6.2 Configuring Project Workflow

Using controls in the Workflow Options page, project administrators may enable transition-based workflow and support for state-dependent and transition-dependent workflow rules.

Figure 5-1: Transition-Based Workflow Options Page

Transition-based workflow provides administrators much greater control and flexibility than state-based workflow and each workflow option enabled both adds to the flexibility of the workflow model and complexity of defining workflow rules in that project."

- In transition-based workflow, project administrators may enable multiple project workflow, state-dependent workflow rules, transition-dependent workflow rules, transition-driven parent-child relationships between selection controls, and either state-dependent or transition-dependent issue creation triggers.
- In state-based workflow, project administrators may enable state-dependent workflow rules and state-dependent issue creation triggers.

In DevTrack, transition-based workflow supports both*state-dependent*and*transition-dependent*workflow rules:.

State-Dependent Rules: State-dependent workflow rules are based on the current workflow state of issues. Administrators may define rules that define applicable owners, mandatory fields, read-only fields, and invisible fields based on the workflow state of DevTrack issues. For more information see “Administering Issue Workflow States” .

Transition-Dependent Rules: Transition-dependent workflow rules are based on the transitions between workflow states. Administrators may define rules that define mandatory fields, read-only fields, invisible fields, and Who Can Perform rules based on the transitions between two workflow states. For more information see “Administering Transitions”.

6.2.1 Enabling Transition-Based Workflow and Transition-Dependent Rules

Transition-based workflow provides administrators much greater control and flexibility than state-based workflow and each workflow option enabled both adds to the flexibility of the workflow model and complexity of defining workflow rules in that project.

In transition-based workflow, project members may submit, edit, forward, and close development issues by performing “actions” in the client. An action is a task—such as creating, editing, or forwarding an issue—that represents a transition between two workflow states or between a workflow state and itself.

In transition-based workflow, multiple transitions may exist between each state, a transition may exist between a workflow state an itself, and each transition is explicitly named and represented by an action button in the client.

Figure 5-2: Transition-Based Workflow

The Workflow Type control displays two options: State-based Workflow option and the Transition-based Workflow option."

If the Transition-based option is selected, the Workflow Options page displays the transition-based workflow options that the administrator may enable and define in the project.

Transition-based workflow supports the both state-dependent workflow rules and transition-dependent workflow rules:

Multiple Workflow: Multiple project workflow (subworkflow) enables administrators to define subworkflows within the larger master workflow. Each subworkflow may be managed by a set of autonomous workflow rules for the workflow states and transitions in the subworkflow.

State-Dependent Access Controls: State-dependent access controls define which project team members to own, view, or update development issues in each workflow state. DevTrack supports four state-dependent access controls: applicable owner rules, mandatory fields, read-only fields, and invisible fields.

Transition-Dependent Access Controls: Transition-dependent access controls define which project team members may perform “actions” and update issue properties in each transition. DevTrack supports four transition-dependent access controls: Who Can Change rules, mandatory fields, read-only fields, and invisible fields.

Conditional Transitions:A conditional transition rule is defined by its name, a condition, and a target state. Every transition may be defined by multiple conditional transition rules.

Issue Creation Triggers:In DevTrack, an issue creation trigger is a state-workflow rule that prompts the creation of new development issues based administrator-defined criteria. DevTrack supports both transition-dependent and statedependent issue creation triggers.

To enable transition-based workflow:

1. Select Workflow > Workflow Options.
2. Click the Workflow Type ellipsis button.

   The Workflow Type dialog box appears.
3. Select the Transition-based option in the Workflow Type dropdown list.
4. Click the OK button.

   The Workflow Type dialog box closes.

To enable workflow rules in transition-based projects:

1. Select Workflow > Workflow Options.
2. Enable the transition-based workflow model.
3. **Optional**: To enable support for state-dependent access controls, select the check box next to the rule:
   In the transition-based workflow model, five state-dependent access controls are supported.

   Applicable Owners: Applicable owner access controls define which project team members may own development issues in each workflow state based on their account type or team group.

   Mandatory Fields: Mandatory field access controls define specific fields as mandatory in particular workflow states.

   Read-Only Fields: Read-only field access controls define specific fields as read-only in particular workflow states.

   Invisible Field: Invisible field access controls define specific fields as invisible in particular workflow states.
4. **Optional**: To enable support for transition-dependent access controls, select the check box next to the rule:

   In the transition-based workflow model, five state-dependent access controls are supported:

   Mandatory Fields: Mandatory field access controls define specific fields as mandatory in particular transitions.

   Read-Only Fields: Read-only field access controls define specific fields as read-only in particular transitions.

   Invisible Fields: Invisible field access controls define specific fields as invisible in particular transitions.

   Who Can Perform:Who Can Perform access controls define which project team members may perform each transition based on their account type or team group.
5. **Optional**: To enable support for transition-dependent field choices, select the Support Transition-driven Field Choice Selection and select a rule option:
6. **Optional**: To enable transition-dependent new issue triggering, select a triggering option button.
7. **Optional**: To enable transition-dependent automation triggering, select the an automation option button.
8. Click the OK button.

6.2.2 Enabling State-Based Workflow and Workflow Rules

The state-based workflow model enables development organizations to define simple workflow defined by rudimentary workflow rules, access controls, and triggers.

In the state-based workflow model, issues progress from workflow state to workflow state based on explicit changes to the issue workflow state property of development issues by project members.

State-based workflow may be graphically represented as a series of boxes and arrows.In state-based workflow, a single,*unnamed*transition may connect any two

Figure 5-3: State-Based Workflow

states in workflow." Every workflow state may be defined by one or more applicable next states—workflow states that development issues may pass to from that state.

In project using the state-based workflow model, all workflow rules, access controls, and automation triggers are state-dependent. State-dependent workflow rules define the ability of project team members to own, view, or update development issues on a state-by-state basis.

The relative lack of flexibility and power of the state-based workflow model means that workflow states and workflow rules may be quickly defined. The state-based workflow model does not support transition-dependent workflow rules, multiple workflows,

State-based workflow supports the following state-dependent workflow rules:

To enable state-based workflow:

1. Select Workflow > Workflow Options.

2 Click the Workflow Type ellipsis button.

The Workflow Type dialog box appears.

3 Select the State-based option in the Workflow Type dropdown list.

4 Click the OK button.

The Workflow Type dialog box closes.

To enable state-dependent workflow rules in state-based projects:

1 Select Workflow > Workflow Options.

2 Enable the state-based workflow model.

3 **Optional**: To enable support for applicable workflow state rules or graphical workflow, click the Change button in the Workflow Options area.

- To enable applicable workflow state rules, click the Applicable State Changes (Next State) check box.
- To define state-based workflow using graphical workflow tools, select the Define Graphically option button.
- To define state-based workflow using workflow settings controls, select the define through Workflow Settings option button.

4 **Optional**: To enable state-dependent support for access controls, select the check box next to the rule:

In the state-based workflow model, five state-dependent access controls are supported.

Applicable Owners: Applicable owner access controls define which project team members may own development issues in each workflow state based on their account type or team group.

Mandatory Fields: Mandatory field access controls define specific fields as mandatory in particular workflow states.

Read-Only Fields: Read-only field access controls define specific fields as read-only in particular workflow states.

Invisible Field: Invisible field access controls define specific fields as invisible in particular workflow states.

5 **Optional**: To enable state-dependent new issue triggering, select the Enable New Issue Triggering Creation check box.

6 **Optional**: To enable interproject action automation triggering, select the Based on Target State option button.

7 Click the OK button.

6.2.3 Enabling Transition Dependent Access Controls

A transition-dependent access control is a workflow rule that is enforced on a transition-by-transition basis.

Using transition-dependent workflow rules, development organizations may restrict the ability of project team members to perform certain transitions or restricts their ability to make changes within a transition.

Transition-dependent access controls may be defined in projects using transitionbased workflow only. DevTrack supports four transition-dependent access controls:

Mandatory Fields: Mandatory field access controls define specific fields as mandatory in particular transitions.

Read-Only Fields: Read-only field access controls define specific fields as read-only in particular transitions.

Invisible Fields: Invisible field access controls define specific fields as invisible in particular transitions.

Who Can Perform Access Controls: Who Can Perform access controls define which project team members may perform each transition based on their account type or team group.

6.3 Administering Issue Workflow States

In DevTrack, the lifecycle of a development issue is defined by *workflow states* and *transitions*. An issue workflow state represents a distinct stage in the lifecycle of a development issue. and determines how that issue is managed in workflow.

An issue workflow state may be defined by its name and description, status (*open*, *closed: success*, or *closed: failed*), graphical attributes, and state-dependent workflow rules.

Figure 5-4: States and Transitions in State-Based Workflow

Issue workflow state administration is the process of enabling support for state-dependent workflow rules, defining issue workflow states, and defining state-dependent workflow rules for each workflow state."

Workflow state statuses

In DevTrack, every issue workflow state is defined by its*state status*. The status of a workflow state indicates the relative success of development issue, the next steps (transitions) in the development lifecycle, and rules for managing issues.

DevTrack supports three workflow state statuses: Open, Closed: Success, or Closed: Failed.

- An open workflow state—a workflow state with an*Open status*—represents a stage in the issue lifecycle. Open development issues are actively tracked in a project.While an issue is in an open state it may be updated and forwarded to other issue workflow states and project members.
- A closed workflow state—a workflow state with either the*Closed: Success status*or*Closed: Failure status*—represents the last stage of the issue lifecycle. Closed issues cannot be updated or forwarded unless they are reopened by a project member and routed to an open state. A project may have multiple closed issue workflow states.

The ability to reopen closed issues may be controlled using account type-based access controls.

State-dependent workflow rules

State-dependent workflow rules are an optional DevTrack feature that may be enabled in projects using either state-based workflow or transition-based workflow.

Using state-dependent workflow rules, development organizations may restrict the ability of project team members to own, view, or update development issues when those issues are in specific workflow states.

State-dependent workflow rules define how development issues are managed in each issue workflow state.DevTrack supports four optional state-dependent workflow rules: applicable owner, mandatory field, invisible field, and read-only field access controls.

Applicable Owner: An*applicable owner*is a project team member or group folder that may be assigned ownership of a development issue when that issue is in a particular workflow state.

Mandatory Fields:A mandatory field is a data-entry control that must be defined in an issue workflow state before the issue may progress to the next stage in its lifecycle.

Read-Only Fields:A read-only field is a data-entry control that may be viewed, but not edited in an issue workflow state before the issue may progress to the next stage in its lifecycle.

Invisible Fields page: An invisible field is a data-entry control that is not displayed in an issue workflow state. Data tracked in the invisible control may not be viewed or edited while the issue is in that workflow state.

6.3.1 Adding Workflow States

An issue workflow state is defined by its name and description, status (open, closed: success, or closed: failed), graphical attributes, and state-dependent workflow rules.

The Add a State manager may display up to eight tabbed form pages including the General Settings tab, the Applicable Owners tab, the Read-Only Fields tab, the Mandatory Fields tab, the Invisible Fields tab, the Triggering tab. Each tab enables project administrators to define*state-dependent workflow rules*for the current workflow state. For more information see “Defining State-Dependent Workflow Rules”.

To add an issue workflow state (in graphical workflow):

1Select Workflow > Workflow Design.

2Right-click in the workflow design page and select the Add a State command in the shortcut menu.

3Define the title and description of the workflow state in the Properties tab.

4Define the workflow status of the workflow state.

5To define the status of the issue workflow state, select the Open, Closed - Success, or Closed - Failed option button.

In DevTrack, every issue workflow state is defined by a workflow status: Open, Closed: Success, or Closed: Failed.

Open: An open workflow state—a workflow state with an Open status—represents a development stage in which development issue are considered active.

Closed: Success:A closed workflow state—a workflow state with a*Closed: Success status*—represents the final stage in the issue lifecycle of work items that have been successfully completed.

Closed: Failed:A closed workflow state—a workflow state with a*Closed: Failure status*—represents the final stage in the issue lifecycle of work items that have not been successfully completed.

6.3.2 Deleting Issue Workflow States

Deleting a workflow state automatically deletes all of the transitions linked to that state.

6.3.3 Defining State-Dependent Workflow Rules

Using controls in the Add State manager, project administrators may create and define issue workflow states.

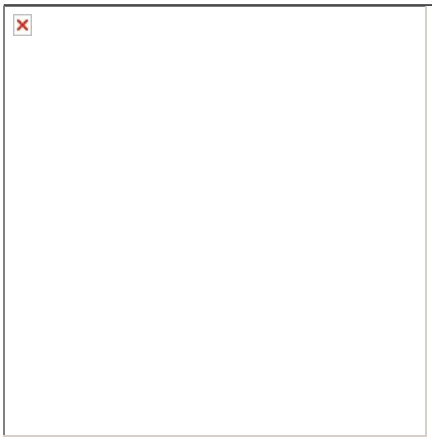


Figure 5-5: Add a State Manager

To define state-dependent workflow rules:

1Select Workflow > Workflow Design."

2Add or edit an issue workflow state.

Properties tab

3Define the appearance of the state using controls in the State Node Settings area of the Properties page. Project administrators may define the color of nodes, node borders, and node text.

Defining a unique appearance for nodes with similar settings communicates the function of the mode to project members.

Applicable Owners tab

4To define applicable owners in the workflow state, select one or more user accounts, account types, group folders, team groups, or user variables.

Invisible Fields tab

5To define data-entry controls as invisible in the workflow state, select one or more issue property fields.

The Invisible Fields tab is only displayed if support for state-dependent applicable owner access controls is enabled. For more information see “Enabling State-Based Workflow and Workflow Rules” .

Mandatory Fields tab

6To define data-entry controls as mandatory in the workflow state, select one or more issue property fields.

The Invisible Fields tab is only displayed if support for state-dependent applicable owner access controls is enabled. For more information see “Enabling State-Based Workflow and Workflow Rules” .

Read-Only tab

7To define data-entry controls as read-only in the workflow state, select one or more issue property fields.

The Invisible Fields tab is only displayed if support for state-dependent applicable owner access controls is enabled. For more information see “Enabling State-Based Workflow and Workflow Rules”.

6.3.4 Defining State-Dependent Applicable Owner Access Controls

State-dependent applicable owner rules enable development organizations to control the flow of development issues in each stage of the development lifecycle. An*applicable owner*is a project team member or group folder that may be assigned ownership of a development issue when that issue is in a particular workflow state.

In DevTrack, a development issue has one-and-only-one owner at any one time. Development issues may be owned by project team members, group folders, or{Unassigned} user variables.

Applicable owners may be identified by account type, user account, team group, or user variable. If a group folder is selected as an applicable owner, the group folder itself is listed as an option in the Assign To dropdown list when an issue is forwarded to that workflow state.

User variables

Five user variables as applicable owners in each workflow state:{Auto-Assignment}, {Previous Owner}, {Current Owner}, {Submitter}, and {Unassigned}.

{Auto Assignment}: The {Auto Assignment} user variable enables issue routing rules to automatically assign development issues to project team members based on administrator-defined rules. For more information see “Administering DevTrack Development Issue Routing” .

{Previous Owner}: The {Previous Owner} user variable identifies the project team member that owned a development issue prior to the current owner as an applicable owner of the issue in the workflow state.

{Current Owner}: The {Current Owner} user variable identifies the current owner of an issue as an applicable owner of that issue in the workflow state.

{Submitter}: The {Submitter} user variable identifies the project team member that initially submitted the development issue as an applicable owner of that issue in the workflow state.

{Unassigned}: The {Unassigned} user variable indicates that the issue may be designated as unassigned in the workflow state.

The Applicable Owners page is displayed in the Workflow Settings page only if the administrator has enabled the Applicable Owners workflow option in the Workflow Options page.

6.3.5 Defining State-Dependent Default Issue Owners

State-dependent default owner rules enable development organization to designate certain project team members or group folders as the default owners of development issues in each workflow state.

Three “default owners” may be defined for each workflow state:

Default Owner of Unassigned Issues: The Default Owner of Unassigned Issues identifies the project team member or group folder that by default owns unassigned issues—issues currently assigned to the {Unassigned} user variable.

Default Owner of Assigned Issues (Current Owner Applicable): The Default Owner of Assigned Issues (Current Owner Applicable) identifies the project team member or group folder that by default owns issues when the previous owner is an applicable owner in the new workflow state.

Default Owner of Assigned Issues (Current Owner Unapplicable): The Default Owner of Assigned Issues (Current Owner Unapplicable) identifies the project team member or group folder that by default owns issues when the previous owner is *not* an applicable owner in the new workflow state.

The list of potential default owners is constrained by state-dependent applicable owners rules. To be designated as the default owner of an issue in a workflow state, a project team member or group folder must first be designated an applicable owner. For more information see “Defining State-Dependent Applicable Owner Access Controls”.

6.3.6 Defining Applicable Workflow States

In projects using state-based workflow, applicable workflow state rules may be defined on a state-by-state basis. An applicable workflow state identifies all possible next stages in the lifecycle of development issues.

Applicable workflow state rules need only be defined in projects using the state-based workflow model. In transition-based workflow, applicable workflow states are defined by the transitions between workflow states. If applicable workflow states are not supported, development issues may pass freely between workflow states without restriction.

The steps required to define applicable workflow states are different if graphical workflow is enabled in the project.

To define applicable workflow states (in graphical workflow):

1. Enable applicable next state workflow rules.
2. Select Workflow > Workflow Design.
3. Select a box in the workflow flowchart and drag the mouse to another state box in one continuous motion.

   As the mouse is dragged a transition line is drawn in the workflow flowchart.
4. Lift the mouse to connect the two state boxes with the transition line.

To define applicable workflow states (in non-graphical workflow):

1. Enable applicable next state workflow rules.

   The Next State tab is only displayed in the Workflow Settings page if applicable next workflow state rules are enabled in the project.
2. Select Workflow > Workflow Settings.
3. Select a workflow state in the Progress States list.
4. Select one or more issue workflow states in the Next State tab.

6.3.7 Defining State-Dependent Mandatory Fields

A mandatory field is a data-entry control that must be defined in an issue workflow state before the issue may progress to the next stage in its lifecycle.

The Mandatory Fields page is displayed in the Workflow Settings page only if the administrator has enabled the Mandatory Fields workflow option in the Workflow Options page.

To define state-dependent mandatory fields:

1. Enable state-dependent mandatory field rules.

   The Mandatory Fields tab is only displayed in the Workflow Settings page if support for mandatory field access controls is enabled in the project.
2. Select Workflow > Workflow Settings.
3. Select a workflow state in the Progress States list.
4. Select one or more development issue fields in the Mandatory Fields tab.

6.3.8 Defining State-Dependent Read-Only Fields

A read-only field is a data-entry control that may be viewed, but not edited in an issue workflow state before the issue may progress to the next stage in its lifecycle.

The Read-Only Fields page is displayed in the Workflow Settings page only if the administrator has enabled the Read-Only Field workflow option in the Workflow Options page.

To define state-dependent read-only fields:

1. Enable state-dependent read-only field rules.

The Read-Only Fields tab is only displayed in the Workflow Settings page if support for read-only field access controls is enabled in the project.

2 Select Workflow > Workflow Settings.

3 Select a workflow state in the Progress States list.

4 Select one or more development issue fields in the Read-Only Fields tab.

6.3.9 Defining State-Dependent Invisible Fields

An invisible field is a data-entry control that is not displayed in an issue workflow state. Data tracked in the invisible control may not be viewed or edited while the issue is in that workflow state.

The Invisible Fields page is displayed in the Workflow Settings page only if the administrator has enabled the Invisible Fields workflow option in the Workflow Options page.

To define state-dependent invisible fields:

1 Enable state-dependent invisible field rules.

The Invisible Fields tab is only displayed in the Workflow Settings page if support for invisible field access controls is enabled in the project.

2 Select Workflow > Workflow Settings.

3 Select a workflow state in the Progress States list.

4 Select one or more development issue fields in the Invisible Fields tab.

6.4 Administering Transitions

In DevTrack, the lifecycle of a development issue is defined by *workflow states* and *transitions*. An issue workflow state defines a distinct stage in the lifecycle of a development issue and transitions mark the connections between each stage.

In transition-based projects workflow rules are both state-dependent and transition-dependent:

- State-dependent rules are based on the current workflow state of the issue and determine which changes may be made to an issue in a particular workflow state and which project members may own the issue and make those changes.
- Transition-dependent rules are based on a transition between two workflow states and determine which changes may be made to during that transition and which project members may perform that action.

6.4.1 Adding Transitions

The tabs displayed in the Add a Transition manager depend on the transition-based workflow options that the administrator enables in the Workflow Options page. For more information see “Configuring Project Workflow” .

To add a transition:

1 Select Workflow > Workflow Design.

2 Select a box in the workflow flowchart and drag the mouse to another state box in one continuous motion.

As the mouse is dragged a transition line is drawn in the workflow flowchart.

3 Lift the mouse to connect the two state boxes with the transition line.

The Add a Transition manager appears.

4 Enter the name of the transition in the Transition Name text box.

5 **Optional**: To define the transition as a quick action, select the If Quick Action check box.

A quick action transition enables project members to update the owner of the issue in the transition. All other issue properties are read-only in the transition.

6.4.2 Editing Transitions

Each page in the Edit a Transition manager enables administrators to define transition-based workflow properties for that transition. The tabs displayed in the Edit a Transition manager depend on the transition-based workflow options enabled in the Workflow Options page. For more information see “Configuring Project Workflow” on page 83.

6.4.3 Deleting Transitions

6.4.4 Defining Transition-Dependent Workflow Rules

To define transition-dependent workflow

Chapter 7 Administering Advanced Features

7 Chapter 6- Administering Advanced Features

This chapter describes advanced DevTrack features including automatic issue routing, personal folders, issue templates, link types, status groups, and issue priority settings.

In this chapter:

- Administering DevTrack Development Issue Routing
- Administering Personal Folders
- Administering Issue Templates
- Administering Link Types
- Administering Custom Crystal Reports
- Managing Issue Priority
- Administering Status Groups

7.1 Administering DevTrack Development Issue Routing

In DevTrack, issue “autorouting” is an optional feature by which development issues may be automatically assigned to applicable issue owners based on the workload or skill-level of those project team members.

The recipients of routed issues are determined by administrator-defined issue routing rules. Routing rules define the criteria for selecting an appropriate issue owner (workload, skill, or a combination of workload and skill), the criteria for routing development issues (fields and field values), and the default owner of issues that do not meet the routing criteria.

To be subject to issue routing rules a development issue must be “forwarded” to the{Auto-Assignment} rule in the course of project workflow. Issues that are owned by the {Auto-Assignment} rule are automatically routed to the appropriate project team member or group folder based on administrator-defined routing rules.

Issue routing may be very valuable to development organizations that need to automatically distribute development issues to project team members based on their workload or the skill-level of its project team members. Issue routing may be used to meet the following needs:

- A programmer fixes an issue, but does not have the privilege to assign the issue to another owner.
- A team member submits a new issue, but does not know the appropriate owner for the issue.
- A manager needs to submit a large number of issues before an approaching deadline. The Routing agent can assign the issues to owners with the proper skill level and reasonable workload.

Using controls in the Auto-Routing page, administrators may enable issue routing, define the multiple rules for routing issues based on issue properties, and determine the criteria that is used to assign those issues to DevTrack project members.

7.1.1 Enabling Issue Routing

In DevTrack, issue “autorouting” is an optional feature that must be enabled in each DevTrack development project. All issue routing rules are project-specific.

To be subject to issue routing rules a development issue must be “forwarded” to the{Auto-Assignment} rule in the course of project workflow. Issues that are owned by the {Auto-Assignment} rule are automatically routed to the appropriate project teammember or group folder based on administrator-defined routing rules. For more information see “Defining State-Dependent Applicable Owner Access Controls”

To enable development issue routing:

1. Select Advanced Features > Auto-Routing in the tree panel.
2. Select the Enable Auto-Routing check box.

7.1.2 Defining Issue Routing Rule Criteria

In DevTrack, administrator-defined routing rules are based on one of three different criteria:*workload*,*skills*, or*workload & skills*.

The issue routing rule criteria selected determines the method that is used in that project for identifying the appropriate owners of routed issues. Issue routing rules define a formula for identifying the most appropriate owner of development issues based on a combination of issue property values and the skills or workload of potential issue owners.

Skill-Based Assignment: In projects employing skill-based routing rules, development issues are automatically assigned to project team members based on their expertise (skill-level)—a numerical value assigned to each project team member by the project administrator.

Workload-Based Assignment: In projects employing workload-based routing rules, development issues are automatically assigned to project team members based on their workload—the project member that owns the fewest issues is assigned the issue.

Formula-Based Assignment: Formula-based routing rules route issues to target owners based an administrator defined formula that weighs both the skill level of the potential target owners and the workload of those potential owners.

To define project routing criteria:

1. Select Advanced Features > Auto-Routing in the tree panel.
2. Select a routing criteria in the Routing Property list.
   - To use workload-based criteria for routing issues, select the Based on Workload option button.
   - To user skill-based criteria for routing issues, select the Based on Skill option button.
   - To user formula-based criteria for routing issues, select the Based on Formula option button and define the relative weights of workload and skill in evaluating routing options.

7.1.3 Defining the Default Recipient of Routed Issues

The default target of routed issues is the project team member or group folder that is automatically assigned routed development issues that do not meet any of the criteria identified by at least one routing rule.

Project administrators may control issue routing using*issue routing rules*which define the criteria (issue field values) that must be met for a development issue to be automatically routed to the appropriate project team member. Issues that do not meet the criteria defined by at least one issue routing rule are automatically assigned to the default owner of routed issues.

To define the default recipient of routed issues:

1. Select Advanced Features > Auto-Routing in the tree panel.
2. Select a project team member in the Default Target If No Criteria Are Met dropdown list.

   The Default Target If No Criteria Are Met dropdown list displays the names of all project team members, group folders, and the{Unassigned} variable.

7.1.4 Defining Issue Routing Rules (Criteria)

In DevTrack, an issue routing rule is a set of criteria that identify which development issues are subject is administrator-defined routing rules based on issue field values.

Every issue routing rule consists of a set of one or more issue fields and corresponding issue field values. Issue routing rule criteria may be based on the following fields:

Component	Fix Target	Platform
Priority	Progress Status	Type
Version		

If a development issue is routed and the issue does not meet the criteria identified by any of the routing rules, the DevTrack workflow engine automatically assigns the issue to the default target. For more information see “Defining the Default Recipient of Routed Issues” .

To define an issue routing rule:

1. Select Advanced Features > Auto-Routing in the tree panel.
2. Click the New Criteria button.

   The New Routing Rule dialog box appears.
3. Select a issue property in the Field Name list.

   The Field Name list displays seven issue property fields.
4. Select an issue property value in the Selection list.

   The Selection list displays field values that define the selected issue property field.
5. Click the OK button.

7.1.5 Defining Issue Routing Targets (Potential Owners)

To define the default recipient of routed issues:

1. Select Advanced Features > Auto-Routing in the tree panel.
2. Select the Add Owner button. The Add Owner dialog box appears.
3. Select a project team member.

   Routing targets (owners) must be created to complete an auto-assignment rule. An issue that meets the defined criteria defined will be routed to one of these targets. Each criterion has its own list of target owners.
4. Enter a numerical value in the Skill Level text box.
5. Click the OK button.

   The Add Owner dialog box closes.

7.2 Administering Personal Folders

In DevTrack, personal folders are an optional feature that enable project team members organize and manage their development issues and, optionally, those of their teams using “personal folders” the issue tree panel of the DevTrack client.

Personal folders may be used to organize DevTrack issues for many different purposes. For example, personal folders may represent the relative priority of issues, organize issues by specific products or components, or by any other category that is useful to the business.

DevTrack issue may be added to personal folder by three methods:

- Project members may drag and drop issues and events into their personal folders in the DevTrack client.
- Issues may be automatically sent to DevTrack project members based on issue notification rules or issue escalation rules.

- Shortcuts to development issues may be added or deleted from personal folders based on event notification rules or event escalation rules. Project administrators may define unique personal notification actions for each personal folder.

All personal folders are administrator-defined. Project administrators may enable personal folder support, create, edit, and manage personal folders in the Personal Folder page of the Enterprise Editions folder.

DevTrack project members may view two different types of personal folders in their clients:

- The My Personal Folder contains the personal folders that may contain issues assigned to the project member. Project members may drag and drop issues into these folders.
- The Team Member's Folder contains personal folders belonging to other project members belonging to the same group.

7.2.1 Enabling Personal Folder Support

To enable personal folder support:

1. Select Advanced Features > Personal Folders in the tree panel.
2. Select the Enable Personal Folder Support check box.

7.2.2 Defining Personal Folders

To create a personal folder:

1. Select Advanced Features > Personal Folders in the tree panel.
2. Click the New button.

   The Add New E-mail Personal Folder dialog box appears.
3. Enter the name of the personal folder in the Folder Name field.
4. Enter a brief description of the personal folder in the Description field.

   The Add New E-mail Personal Folder dialog box closes.

To edit a personal folder:

1. Select Advanced Features > Personal Folders in the tree panel.
2. Select a personal folder in the Personal Folders list.
3. Click the Edit button.

   The Edit Personal Folder dialog box appears.
4. Update the name of the personal folder in the Folder Name field.
5. Update a brief description of the personal folder in the Description field.

   The Edit E-mail Personal Folder dialog box closes.

To delete a personal folder:

1. Select Advanced Features > Personal Folders in the tree panel.
2. Select a personal folder in the Personal Folders list.
3. Click the Delete button.

   A warning dialog box appears.
4. Click the Yes button.

7.3 Administering Issue Templates

An issue template is a user-defined template representing the issue submission page that project members must complete to submit an issue in a DevTrack project. The data-entry controls in the default value template are pre-populated with common values— values that project members might need to enter repeatedly in each new issue.

DevTrack project members may use issue templates to save time and effort when submitting issues to a DevTrack project.

Default value templates may store default values for every control in every page of the issue submission page, including custom controls.

7.3.1 Defining Issue Template Options

Administrators may use controls in the Default Value page to define the project-level issue template and to manage issue template options.

The project-level issue template is the default template for all project members throughout a DevTrack project. Administrators may define the project-level issue template by selecting it from the Project Level Default Issue Template dropdown list in the Default Value page.

Administrators may also determine whether use of the project-level default value template is mandatory throughout the DevTrack project or whether individual project members may create and use their own default value templates.

The Default value option dropdown list displays two options:

- The Every User Uses the Project Level Default Value Issue Template option mandates that the selected template always used in the project.
- The Every User Can Select Own Default Value Issue Template option enables individual project members to define their own default value settings in the User Preferences manager in their client application.

To define issue template options:

1. Select the Default Value page in the Advanced Features folder of the Admin Tree window.
2. Select an option from the Project-level Default Issue Template dropdown list.

   The Project-level Default Issue Template dropdown list displays every template created in the current project. All default value templates are created in the DevTrack clients.

   If the No Default Value option is selected, no template is identified as the project-level default value template.
3. Select an option from the Default value option dropdown list.

7.4 Administering Link Types

In DevTrack, a link type is a set of business rules that defines the relationship between two linked development issues. Issue linking enables developers to more easily manage groups of related issues.

A link type is a kind of template for creating links and defining relationships of a certain type. Every link type is defined by its title and link type category (reference link or parent-child link).

Using controls in the Define Link Type manager, project administrators may create the link types that project team members may use to the create links between development issues in DevTrack projects.

Figure 6-1: Define Link Type Manager

Whenever a project member creates a link between two issues in the DevTrack client, the project member must select an administrator-defined link type to define and identify each issue link."

DevTrack supports two classes of link types: referential link types and parent-child link types.

Referential: Referential link types may be used to link together two issues within a DevTrack project.

Parent-Child: Parent-Child link types may be used to create a parent-child relationship between two issues within a DevTrack project.

Once a link type is created in the Link Type manager of DevSuite Admin that link type is available for selection in the DevTrack client whenever a project member creates a link between two issues.

The Link Type manager is accessible through both the Parent-Child Issue Link page and the Define Link Type button in the Interproject Action page.

7.4.1 Understanding Interproject Linking

Administrator-defined link types may be used to create links between development issues within a single DevTrack project, between development projects in different projects, or between development issues and incidents managed in Tech Excel CustomerWise or ServiceWise projects.

Project members may link issues in different project using either a referential link or a parent-child link. For more information see “Administering Interproject Actions and Linking” .

All links between issues or between issues and incidents are*bi-directional.*When a link is created between issues is separate projects, that link is displayed in both projects. Members of both projects may view and track details about the linked issue in the Issue page of the issue detail panel.

7.4.2 Defining Referential Link Types

In DevTrack, a link type is a set of business rules that defines the relationship between two linked development issues. Issue linking enables developers to more easily manage groups of related issues.

DevTrack supports two classes of link types: referential links and parent-child links.

A referential link type defines a bidirectional relationship between two issues by which information about a linked issue may be tracked in the DevTrack client along with every other development issue, specification, requirement, or incident that is related to the same issue.

A referential link type is defined by its title and link type category, and a description. All referential links are simple relationships and do not enforce workflow rules.

To define a referential link type:

1. Access the Link Type manager.

   The Link Type manager is accessible through two different pages in DevSuite Admin.

   - Click the Define Link Type button in the Parent-Child Issue Link page.
   - Click the Define Link Type button in the Interproject Action page. The Define Link Types manager opens.

2. Click the New button.

   The Define a Link Type Define dialog box opens.

3. Enter a name in the Link Type Name field.

4. Select the Reference Links option from the Link Type Category dropdown list.

   The Link Type Category dropdown list displays the referential link type option and the parent-child link type option.

5. Enter a brief description of the link type in the Link Type Description page.

6. Click the OK button.

   The Link Type Define dialog box closes.

7. Click the Close button in the Link Types manager.

   The Link Types manager closes. Once a link type is created in the Link Type manager that link type is available for selection in the DevTrack client whenever a project member creates a link between two issues.

7.4.3 Defining Parent-Child Link Types

In DevTrack, a link type is a set of business rules that defines the relationship between two linked development issues. Issue linking enables development organizations to more easily manage and tracks groups of related issues.

Parent-child link types define a relationship between two issues by which the status of a one issue is controlled by the status of another linked issue. These relationships—called*close time enforcement rules*— define which issue in the parent-child relationship is to be closed before the other.

A parent-child link type may be defined by one of five different close time enforcement rules:

No enforcement: freely close parent or child issue: If the No Enforcement rule defines the link type, either the parent or the child issue may be closed without affecting the status of the other.

Warning: do not close a parent issue until all the child issues get closed: If the Warning: Do Not Close Parent rule defines the link type, project team members are warned if they attempt to close a parent issue before the child issue is closed.

Enforce: cannot close a parent issue until all the child issues get closed: If the Enforce: Cannot Close Parent rule defines the link type, the child development issue cannot be closed until the parent development issue is closed.

Warning: do not close child issue until the parent issue gets closed: If the Warning: Do Not Close Child rule defines the link type, project team members are warned if they attempt to close a child issue before the parent issue is closed.

Enforce: cannot close child issues until the parent issue get closed: If the Enforce: Cannot Close Child rule defines the link type, the child development issue cannot be closed until the parent development issue is closed.

Using controls in the Define Link Type manager, project administrators may create the link types that project team members may use to the create links between development issues in DevTrack projects.

Once a link type is created in the Link Type manager that link type is available for selection in the DevTrack client whenever a project member creates a link between two issues.

To define a parent-child link type:

1. Access the Link Type manager.

   The Link Type manager is accessible through two different pages in DevSuite Admin.

   - Click the Define Link Type button in the Parent-Child Issue Link page.
   - Click the Define Link Type button in the Interproject Action page.

   The Define Link Types manager opens.

2. Click the New button.

   The Define a Link Type Define dialog box opens.

3. Enter a name in the Link Type Name field.

4. Select the Parent-Child Links option from the Link Type Category dropdown list. The Link Type Category dropdown list displays the referential link type option and the parent-child link type option.

5. Select a link enforcement rule option in the Close Time Enforcement area.

   The Close Time Enforcement area displays five options:

   - No enforcement: freely close parent or child issue
   - Warning: do not close a parent issue until all the child issues get closed
   - Enforce: cannot close a parent issue until all the child issues get closed
   - Warning: do not close child issue until the parent issue gets closed
   - Enforce: cannot close child issues until the parent issue get closed

6 Enter a brief description of the link type in the Link Type Description page.

7 Click the OK button.

The Link Type Define dialog box closes.

8 Click the Close button in the Link Types manager.

The Link Types manager closes.

7.4.4 Deleting Link Types

To delete a link type:

1 Access the Link Type manager.

The Link Type manager is accessible through two different pages in DevSuite Admin.

- Click the Define Link Type button in the Parent-Child Issue Link page.
- Click the Define Link Type button in the Interproject Action page.

The Link Types manager opens.

2 Select a link type option in the Link Type list.

3 Click the Delete button.

A warning dialog box opens.

4 Click the Yes button.

7.5 Administering Custom Crystal Reports

Crystal Reports enables development organizations to create flexible, presentationquality reports that are accessible through the Report view of the DevTrack Windows client.

Using controls in the Report Setting page, project administrators may choose between Crystal Reports and HTML-based reporting and identify the Crystal Report templates (RPT files) that may be used in a DevTrack development project.

7.5.1 Choosing DevTrack Reporting Platforms

DevTrack supports two platforms for generating and presenting reports in the DevTrack Windows client: Crystal Reports and HTML-based web reports.

- Crystal Reports is the leading reporting engine that enables businesses to generate and customize “presentation-quality” reports with sophisticated charting and graphing.
- Web reports are HTML-based reports that are displayed as web pages in the client. If the HTML-based reports option is selected the reports run in the DevTrack Windows client are the same as the reports run in the DevTrack Web client.

Using controls in the Report Setting page, project administrators may choose between Crystal Reports and HTML-based reporting.

To choose a reporting platform:

1 Select Advanced Features > Report Setting in the tree panel.

2 Select a reporting platform.

- To use Crystal Reports as the reporting platform in a DevTrack project, select the Use the Crystal Report option button.
- To use HTML-based reports as the reporting platform in a DevTrack project, select the Use the HTML Report option button.

7.5.2 Defining Custom Crystal Report Template Folders

The DevTrack system may use the RPT files stored in the report template folder to create user -defined Crystal Reports in the DevTrack Windows client.

To define custom Crystal Report template folders:

1 Select Advanced Features > Report Setting in the tree panel.

2 Select the Use Crystal Report option button.

3 Click the Browse button.

The Browse for Folder dialog box appears.

4 Locate the folder that contains the Crystal Reports template files.

5 Click the OK button.

7.5.3 Adding User-Defined Custom Reports

Self-defined Crystal Reports are reports that are based on user-defined Crystal Report template files (RPT files) that are stored in an administrator-defined Crystal Reports Template folder.

Using controls in the Report Setting page administrators may define the titles of self-defined Crystal Reports and identify an appropriate Crystal Report template files for each report.

The title of each self-defined Crystal Report is displayed in the Self-defined Crystal Report folder in the Report Tree window of the DevTrack Windows client.

To add user-defined Crystal Reports:

1. Add a custom Crystal Report template file (a RPT file) to the Custom Report Template File folder.

   Administrators may define the Custom Report Template folder for each DevTrack project in the Custom Crystal Report Template File Folder field.

2. Select Advanced Features > Report Setting in the tree panel.
3. Click the Add button.

   The Add a Custom Report dialog box appears.

4. Enter a title for the report in Report Name field.

   The report name is displayed in the Self-defined Crystal Report folder in the Report Tree window of the DevTrack Windows client.

5. Identify the report template file.
   - Administrators may manually enter the file name in the File Name field. For exampleList_Report.rpt.
   - Administrators may click the Browse button and search for the appropriate RPT file in the Custom Report Template File folder.

To edit self-defined Crystal Reports:

1. Select Advanced Features > Report Setting in the tree panel.
2. Select a custom Crystal Report in the Custom Crystal Report Template Files list.
3. Click the Modify button.

   The Modify Custom Report dialog box appears.

4. Update the title for the report in Report Name field.

   The report name is displayed in the Self-defined Crystal Report folder in the Report Tree window of the DevTrack Windows client.

5. Identify the report template file.
   - Administrators may manually enter the file name in the File Name field. For exampleList_Report.rpt.
   - Administrators may click the Browse button and search for a RPT file in the template folder.

To delete self-defined Crystal Reports:

1. Select the Report Setting page in the Advanced Features folder of the Admin Tree window.
2. Select a custom Crystal Report in the Custom Crystal Report Template Files list.
3. Click the Delete button.

   A warning dialog box appears.

4. Click the Yes button.

7.6 Managing Issue Priority

Administrators may use control in the Issue Priority page of the Advanced Features folder in DevSuite Admin to enable support and define settings for priority order management and priority value management in a DevTrack project.

DevTrack project teams many prioritize issues using two different methods:

- Priority order management enables project members to assign a number (a rank) to each issue. A project member that has ten issues may rank each issue with a unique priority order number from one to ten.
- Priority value management enables administrators to define a formula for assigning a priority value to each issue based on its property definitions.

Issue priority management tools enable project members to prioritize the issues assigned to them. Every DevTrack issue may have both a priority value and priority order number associated with it, giving issue owners tools to determine the most critical issues in their issue list.

Note: The Priority control displayed in the Description page and the priority value and priority order number are unrelated. The Priority field may be customized and is a field associated with an issue. These priority numbers are simply values assigned to an issue based on the priority formula.

7.6.1 Defining Priority Order Management Settings

Administrators may use controls in the Issue Priority page of the Advanced Features folder in DevSuite Admin to enable and define priority order management in DevTrack projects.

Priority order management enables project members to prioritize DevTrack issues based on a user-defined priority order number.

Project members that have been granted appropriate privileges may rank DevTrack issues in the current project using priority order numbers. Each issue may be assigned a unique priority number that represents that issues relative importance. A project member that has ten issues may rank each issue with a unique priority order number from one to ten.

Administrators may grant two priority order management privileges to each account type.

- The Can Change Priority Order of Own Issues privilege
- The Can Change Priority Order of All Issues privilege

Priority order management privileges are granted to project account types. Project members inherit all privileges assigned to their account type.

To define priority order management settings:

1 Select Advanced Features > Issue Priority in the tree panel.

2 Select the General Setting tab

3 Select the Support Priority Order Management tab

4 Select the Privilege tab.

5 Select an account type in the Account Type list.

6 Select privilege options for the selected account type.

7.6.2 Defining Priority Value Management

Administrators may use controls in the Issue Priority page of the Advanced Features folder in DevSuite Admin to enable and define priority value management in DevTrack projects.

When a project team uses priority value management the priority value of an issue is automatically calculated based on an administrator-defined formula. The formula that calculates the priority value of each issue is based on two primary factors: Issue property definitions and time factors.

Issue property definitions

Administrators may define select which data-entry controls are applicable for determining issue priority values and assign weights to the options displayed in each applicable field.

Issue time factors

Time factor priority values are based on the number of days that an issue remains open and the weight that the administrator assigns to the factor. Administrators may enable up to three time factors and assign a weight for each time factor. Time factors include the number of days that the issue has been open, the number of days since the issue was last modified, or the number of days the issue is past due. For each time factor the administrator-defined weight is multiplied by the number of days the issue has remained open to arrive at a value.

To define priority value management settings:

1 Select Advanced Features > Issue Priority in the tree panel.

2 Select the General Setting tab

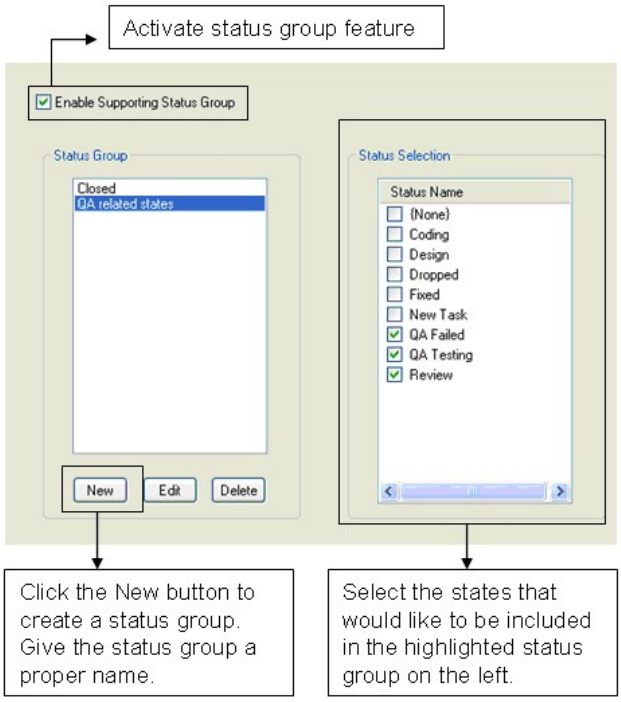
3 Select the Support Priority Value (Severity Number) tab.

4 Select options in the Applicable Fields for Priority Value field.

7.7 Administering Status Groups

Astatus group is a set of development issue workflow states thathave been grouped together for issue filtering and reporting purposes. Statusgrouping enable project teams to quickly identify and report on groups ofstatuses instead of a single status.

For example, a user wants to run a quick search on allissues with a status of QA Failed, QA Passed, or QA Testing. If a status groupwas defined containing these three statuses, the user could quickly get thedesired results.



7.7.1 Enabling Status Group Support

To enable status group support:

1Select the Status Grouppage in the Advanced Features folder of the Admin Tree window.

2Select the EnableSupporting Status Group check box.

7.7.2 Defining Status Groups

To create a status group:

1Select Advanced Features> Status Group in the tree panel.

2Select the EnableSupporting Status Group check box.

3Click the New button. TheAdd a New Item dialog box appears.

4Enter a name for theStatus Group in the Name field.

5Click the OK button. TheAdd a New Item dialog box closes.

6Select issue statusoptions in the Status Selection list.

To edit a status group:

1Select Advanced Features> Status Group in the tree panel.

2Select the EnableSupporting Status Group check box.

3Select a status group inthe Status Group list.

4Click the Edit button. TheEdit the Item dialog box appears.

5Update the name for theStatus Group in the Name field.

6Click the OK button. TheEdit the Item dialog box closes.

7 Select issue statusoptions in the Status Selection list.

To delete a status group:

1 Select Advanced Features> Status Group in the tree panel.

2 Select a status group inthe Status Group list.

3 Click the Delete button. Awarning dialog box appears.

4 Click the Yes button.

Chapter 8 Issue Notification & Escalation Adminis

8 Chapter 7- Issue Notification & Escalation Administration

DevTrack issue notifications and issue escalations provide development organizations the means to ensure that issue stakeholders are automatically notified at key junctures in the issue lifecycle, enforce accountability, and drive the quality of project deliverables.

In this chapter:

- Administering Issue Notifications and Escalations
- Administering Issue Notifications
- Administering Issue Escalations
- Administering Issue Notification and Escalation Conditions
- Administering Issue Notification and Escalation Actions
- Administering Potential Recipients and Subscriptions
- Administering Notification Messages

8.1 Administering Issue Notifications and Escalations

DevTrack issue notifications and issue escalations provide development organizations the means to ensure that issue stakeholders are automatically notified at key junctures in the issue lifecycle, enforce accountability, and drive the quality of their products.

All issue notifications and issue escalations are based on administrator-defined rules which identify the triggers, conditions, notification methods and messages, and recipients of every notification or escalation action.

Issue notifications and escalations enable to ensure that issue stakeholders are kept abreast of the status of their issues. The primary difference between DevTrack issue notification and issue escalation is how these rules are triggered:

Issue Notification: Issue notifications alert issue stakeholders whenever key changes are made to their development issues— subscribers may be notified whenever issues are created, updated, or closed in a development project.

Issue Escalation: Issue escalations alert issue stakeholders at key points in the lifecycle of development issues or when insufficient action is taken on an issue in a defined time period and provide for the reassignment of escalated issues to other project members and workflow states.

Both DevTrack issue notification and issue escalation support:

- Customized notification triggers. The trigger defines the specific *action* (issue notification) or *lack of action* (issue escalation) that prompts the delivery of the notification message.
- Customized conditions that limit the scope of the trigger to a subset of issues based on an administrator-defined query.
- Customized email notification messages and notification methods (email, pager, mobile phone, and personal folder)
- Customized subscription rules including mandatory and optional notification subscriptions.

Notification and escalation triggers

The primary difference between DevTrack issue notifications and issue escalations is how these rules are triggered:

Push and pull notification subscriptions

TechExcel DevTrack supports both *push notification subscriptions* and *pull notification subscriptions*.

All issue notifications and escalations are defined by administrator-defined *subscription rules*. Subscription rules determine whether a project team member must, can, or cannot receive a notification.

Subscriptions rules may mandate that certain project members receive issue notifications for certain issues while others may *opt in* or *opt out* of individual subscriptions. Project administrators may also mandate that certain project member *never* receive issue notifications based on issue notification rules.

- Push notifications are “instantly and actively transferred (pushed)” to subscribers. A project member is subscribed to all issues and is always notified when changes are made to issues based on that rule.
- Pull notifications are optionally sent. A project member may choose to subscribe to a notifications on an issue-by-issue basis.

8.2 Administering the DevTrack Mail Service

The DevTrack Mail Service enables development organizations to integrate DevTrack workflow, issue notification, event scheduling, and issue escalation with the convenience of e-mail.

All email, pager, and mobile phone notifications are sent to subscribers through the the DevTrack Mail Service. DevTrack issue notification and escalation and event notification and escalation require the installation and configuration of the DevTrack Mail Service.

The DevTrack Email Server communicates with the DevSuite Database Server and email server to find and send email messages generated within a DevTrack project. The DevTrack Mail Server retrieves the user name and password needed to access to the DevSuite Database Server from the DevSuite Application Server.

8.2.1 Managing Outgoing Email Server Settings

To define outgoing mail server properties:

1. Select Advanced Features > Email Settings.
2. Select the Outgoing Mail Server tab.
3. Define SMTP server address.

   Specify the name or the TCP/IP address of the outgoing mail server.

4. **Optional**: To define authentication, select the mail server requires authentication option and specify required user account information.

   The account information can be either a domain user account or the server computer account with privileges of using the outgoing mail server service.

To define outgoing email character set:

1. Select Advanced Features > Email Settings.
2. Select the Outgoing Mail Server tab.
3. Select the character set from the dropdown list.

To define outgoing email text wrapping:

1. Select Advanced Features > Email Settings.
2. Select the Outgoing Mail Server tab.
3. Select the character set from the dropdown list.
4. Define text wrapping for outgoing email.

8.2.2 Defining Issue Notification Sending Addresses

All email, pager, and mobile phone issue notifications are sent from a single, administrator-defined email address. Project administrators may define the sending address and the name of the sender for all project issue notifications in the Issue Notifications page of the Issue Notification subfolder in the Advanced Settings folder.

To define the issue notification sending address:

1. Select the Issue Notifications page of the Issue Notification subfolder.
2. Define the sending email address for project issue notifications in the Email Address control.
3. Define the name of the sender of issue notifications in the Name control.

8.2.3 Reloading Settings to DevTrack Mail Service

The DevTrack Email Notification Service must be stopped and restarted whenever a change is made to issue notification settings.

To reload email notification settings, click the Reload Email Service Setting button in the in the Issue Notifications page of the Issue Notification subfolder in the Advanced Settings folder.

8.2.4 Defining Issue Notification Activity Dates

The issue notification activity date identifies the beginning date for all issue notifications in a DevTrack project.

- Notification rules triggered subsequent to the activity date generate issue notifications.
- Notification rules triggered *prior* to the activity date *do not* generate issue notifications.

To define project issue notification activity dates, select the Start Date for Applying Notification rules control in the Issue Notifications page of the Issue Notification subfolder in the Advanced Settings folder.

8.3 Administering Issue Notifications

In DevTrack, issue notification is the process by which stakeholders are notified (by email, pager, mobile phone, or personal folder) whenever a specific change—such as the creation or deletion—is made to a development issue.

All issue notifications are based on administrator-defined issue notification rules. An issue notification rule is defined by a *notification trigger*, one or more *notification actions*, and

one or more*subscribers*. The scope of an issue notification rule may be defined by a*notification condition*.

Using controls in the Add Notification Rule manager and the Properties tab of the Issue Notifications page, project administrators may define and update many notification rule properties.

Notification Trigger:A notification trigger identifies the issue management task that triggers a notification action.

Notification Condition:A notification condition limits the limit the scope of the trigger to a subset of issues based on an administrator-defined query.

Notification Action:A notification action defines the notification message and the notification method (email, personal folder, pager, mobile phone).

Subscribers: Subscribers represent project members that are both potential recipients of an issue notification and who are subscribed to the issue notification rule. Project administrators may assign one of four subscriptions to project members, account types, and groups: the Must Subscribe subscription, the Optional Yes subscription, the Optional No subscriptions, and the Never Subscribe subscription.

8.3.1 Enabling Issue Notification in a DevTrack Project

Issue notification is an optional DevTrack feature that must be enabled on a project-by-project basis.

To enable issue notification select the Enable Auto Email Notification check box in the Issue Notifications page of the Issue Notification subfolder in the Advanced Settings folder.

8.3.2 Defining Issue Notification Rules

An issue notification rule identifies which project team members are notified whenever a specific change is made to a development issue.

An issue notification rule is defined by a*notification trigger*, one or more notification actions, and*notification subscription rules*. Optional issue notification conditions limit the scope of the rule to a subset of development issues

Any number of notification rules may be defined in a project.

To define an issue notification rule:

1Select Advanced Features > Notifications/Escalations > Issue Notifications.

2Click the New button.

The Add Notification Rule dialog box appears.

3Provide a name and description for the issue notification rule.

A unique name and note that describe the logic of the issue notification rule enables project administrators to save time when they review or update issue notifications.

Notification trigger

4Select a trigger in the Trigger dropdown list.

The Trigger dropdown list displays six system triggers and all administrator-defined status change and field change triggers. Every issue notification rule is defined by a single notification trigger.

Custom notification triggers may be defined in the Define Triggers window.For step-by-step instructions see “Administering Issue Notification Triggers”.

Notification conditions

5 Optional: To limit the scope of a notification rule, select an issue notification condition in the Condition dropdown list.

The Condition dropdown list displays all administrator-defined conditions.

Custom notification conditions may be defined in the Notification & Escalation Conditions window. For step-by-step instructions see “Administering Issue Notification and Escalation Conditions” .

6Click the OK button.

The Add Notification dialog box closes and the issue notification rule is displayed in the Notification Rules list.

Notification actions

7Define one or more notification actions in the Actions tab.

A notification action is defined by a*notification method*(email, personal folder, pager, or mobile phone) and a*notification message*.

- To define an email notification action, select the Notify By Sending Email option and select appropriate internal and external messages.
- To define a pager notification action, select the Notify By Pager option and select an appropriate message.
- To define a mobile phone notification action, select the Notify By Mobile Phone option and select an appropriate message.
- To define a personal folder notification action, select the Notify By Personal Folder option. Custom notification messages may be defined in the Define Email window. For step-by-step instructions see “Administering Notification Messages” .

Potential recipients

8Define potential recipients in the Potential Recipient tab.

A potential recipient is a project team member who*may subscribe*to an issue notification rule. A project team member may be identified as a potential recipient based on their account type, team group, or user variable.

Potential recipient subscription rules

9Define subscription rules in the Subscription Option tab.

Potential recipient subscription rules define whether a potential recipient can, cannot, or must subscribe to an issue notification. Subscription rules may be defined for each account type, team group, or user account and affect only those project team members that have been designated as potential recipients of issue notifications.

Subscribers

10Define subscriptions in the Potential Recipient tab.

Issue notification subscriptions define which project team members are subscribed and which project team members are not subscribed to each issue

notification rule.Issue notification rules are specifically defined for each user account and override all potential recipient subscription rules.

To rename an issue notification rule:

To rename an issue notification rule, select the rule in the Notification Rules list and click the Rename button. The name of the rule may be changed in the dialog box.

To delete an issue notification rule:

To delete an issue notification rule, select the rule in the Notification Rules list and click the Delete button

8.3.3 Administering Issue Notification Triggers

In DevTrack, an issue notification trigger is an*action*—such as the creation, edit, or deletion of an issue—that prompts the DevTrack Mail Service to deliver a notification message to a list of issue stakeholders.

An issue notification is sent every time that the triggering action is performed. If the action is not performed, the issue notification rule is not triggered and the notification message is not sent.

DevTrack supports both*system triggers*and*custom triggers*.

System triggers

DevTrack system triggers prompt the DevTrack Mail Service to send notification messages whenever an issue management task is performed on issues meeting the conditions of the issue notification rule.

DevTrack supports six system triggers:

New Issue: The New Issue trigger generates an issue notification whenever a new development issue is created in a project.

Forward Issue: The Forward Issue trigger generates an issue notification whenever a development issue is forwarded to another project member or workflow state.

Closed Issue: The Closed Issue trigger generates an issue notification whenever a development issue is closed in a project.

Reopen Issue: The Reopen Issue trigger generates an issue notification whenever a development issue is reopened in a project.

New Note: The New Note trigger generates an issue notification whenever a note is added to a development issue.

Edit Note: The Edit Note trigger generates an issue notification whenever a note attached to a development issue is edited.

Custom triggers

DevTrack custom triggers prompt the DevTrack Mail Service to send notification messages whenever a specified change is made to an issue property. DevTrack supports two types of custom triggers:*status change triggers*and*field change triggers.*

Status Change Trigger: Status change triggers generate issue notifications based on changes to the workflow state of development issues. Status change triggers may be state-dependent or transition-dependent

Field Change Trigger:Field change triggers generate issue notifications based on changes to issue property values. Every field change trigger is defined by an issue field, a beginning field value, and an ending field value.

8.3.4 Defining Custom Issue Notification Status Change Triggers

In DevTrack, a custom notification trigger defines an*action*—such as a change to the ownership or workflow state of an issue—that prompts the DevTrack Mail Service to deliver a notification message to a list of subscribers.

Status change triggers generate issue notifications based on changes to the workflow state of development issues. Status change triggers may be*state-dependent*or*transition-dependent*.

- State-dependent triggers generate issue notifications whenever an issue if forwarded from one specified workflow state to another specified workflow state.
- Transition-dependent triggers generate issue notifications whenever a specified transition action is performed. A trigger may represent multiple transitions.

Using controls in the Triggers window, project administrators may create, edit, and delete custom status change triggers and field change triggers.

Figure 7-1:Add New Custom Trigger DialogBox

To define a state-dependent notification trigger:

1. Click the Setup button in the Properties tab of the Incident Notification page." Custom notification triggers may be defined whenever a project administrator creates or edits an issue notification rule.

   The Triggers window appears.

2. Click the New button.

   The Add New Custom Trigger dialog box appears.

3. Select the Status Change option button.

   The Add New Custom Trigger dialog box displays controls that enable the user to define a state-dependent or transition-dependent custom status change trigger.

4. Select the Change by State option button.

5. Select an issue workflow state in the Changed From dropdown list.

   The issue workflow state selected in the Changed To control the starting state.

6. Select an issue workflow state in the Changed To dropdown list.

   The issue workflow state selected in the Changed To control the ending state.

7. Click the OK button.

   The custom status change trigger is saved and may be used to define an issue notification rule.

To define a transition-dependent notification trigger:

1. Click the Setup button in the Properties tab of the Incident Notification page.

   Custom notification triggers may be defined whenever a project administrator creates or edits an issue notification rule.

   The Triggers window appears.

2. Click the New button.

   The Add New Custom Trigger dialog box appears.

3. Select the Status Change option button.

   The Add New Custom Trigger dialog box displays controls that enable the user to define a state-dependent or transition-dependent custom status change trigger.

4. Select the Change by Transition option button.

5. Select one or more transitions in the Transition tree menu.

   A transition-dependent custom status change trigger may be defined by multiple transitions. Development issues are subject to the issue notification rule any of the selected transition actions are performed.

6. Click the OK button.

   The custom status change trigger is saved and may be used to define an issue notification rule.

8.3.5 Defining Custom Field Change Triggers

In DevTrack, a custom notification trigger defines an*action*—such as a change to the ownership or state of an issue—that prompts the DevTrack Mail Service to deliver a notification message to a list of subscribers.

Field change triggers generate issue notifications based on changes to issue property values. Every field change trigger is defined by an issue field, a beginning field value, and an ending field value.

Using controls in the Triggers window, project administrators may create, edit, and delete custom status change triggers and field change triggers.

Figure 7-2: Add New Custom Trigger Dialog Box

To define a field change notification trigger:

1. Click the Setup button in the Properties tab of the Incident Notification page."

   Custom notification triggers may be defined whenever a project administrator creates or edits an issue notification rule. The Triggers window appears.

2. Click the New button.

   The Add New Custom Trigger dialog box appears.

3. Select the Field Change option button.

   The Add New Custom Trigger dialog box displays controls that enable the user to define a custom field change trigger.

4. Select a field (issue property) in the Field dropdown list.

   The Field dropdown list displays fields that defined development issue properties. The field selected determines the field values displayed in the From and To dropdown list.

5. Select a field value in the From dropdown list.

6. Select a field value in the To dropdown list.

7. Click the OK button.

   The custom field change trigger is saved and may be used to define an issue notification rule.

8.4 Administering Issue Escalations

Issue escalations alert issue stakeholders at key points in the lifecycle of development issues or when insufficient action is taken on an issue in a defined time period and provide for the reassignment of escalated issues to other project members and workflow states.

Issue escalation enables project teams to ensure that DevTrack issues do not “fall through the cracks”. Project teams may be notified by email, pager, mobile phone, or personal folder based on an administrator-defined escalation schedule. If still no action is taken, issues may be automatically reassigned to another project member, group, or group folder.

An issue escalation rule is defined by an*escalation trigger*, one or more optional*escalation condition*s, one or more*escalation actions*, and its*subscribers*—the project team members that may be notified when an issue is escalated based on the rule.

Escalation triggers

Unlike issue notifications (which are based on the changes that project members make to DevTrack issues), issue escalation rules are based on changes that are not made to issues.

Every issue escalation rule is defined by one of nine, system-defined*escalation triggers*: No Acknowledge, Open Too Long, No Progress, No Status Change Only, No Owner Change Only, Before Start Date, After Start Date, Before Finish Date, and After Finish Date.

Escalation conditions

The scope of an escalation rule may be limited to development issues that meet a set of criteria—the*issue escalation condition*.

An escalation condition is essentially a query that filters issues by their issue property values. For example, a condition may limit the scope of an escalation rule to issues that have a High or Urgent status.

Escalation actions

All escalation triggers are “time triggers”. DevTrack executes an escalation action when the “status” defined by the escalation trigger remains unchanged for the escalation wait period.

Issue escalation actions define the time period that must pass before notifications are sent, the method used to notify subscribers (email, personal folder, pager, or mobile phone), the message delivered by the notification, and the number of times that notification are sent before an escalated issue is reassigned to another project team member.

Subscribers

DevTrack enables administrators to manage both push email and pull email for each issue escalation rule using subscription rules.

Project administrators may mandate that certain project members receive issue escalation notifications for certain issues while others may opt in or opt out of individual

subscriptions. Project administrators may also mandate that certain project member never receive issue escalation notifications based on issue escalation rules.

8.4.1 Administering Issue Escalation Rules

Issue escalations alert issue stakeholders at key points in the lifecycle of development issues or when insufficient action is taken on an issue in a defined time period and provide for the reassignment of escalated issues to other project members and workflow states. An issue escalation rule is defined by an*escalation trigger*, one or more notification actions, and*escalation subscription rules*. Optional issue*escalation conditions*limit the scope of the rule to a subset of development issues. Any number of notification rules may be defined in a project.

To create issue escalation rules:

1Select Advanced Features > Notifications/Escalations > Issue Escalations.

2Click the New button. The Add Escalation Rule dialog box appears.

3Provide a name and description for the issue escalation rule.

A unique name and note that describe the logic of the issue notification rule enables project administrators to save time when they review or update issue notifications.

Escalation triggers

4Select an option from the Trigger dropdown list.

The Trigger dropdown list displays nine system-defined triggers: No Acknowledge, Open Too Long, No Progress, No Status Change Only, No Owner Change Only, Before Start Date, After Start Date, Before Finish Date, and After Finish Date. For more information see “Understanding Issue Escalation Triggers”.

Escalation conditions

5 Optional: To limit the scope of an escalation rule, select an issue escalation condition in the Condition dropdown list.

The Condition dropdown list displays all administrator-defined conditions.

Custom escalation conditions may be defined in the Notification & Escalation Conditions window. For step-by-step instructions see “Administering Issue Notification and Escalation Conditions”.

6Click the OK button.

The Add Escalation dialog box closes and the issue escalation rule is displayed in the Escalation Rules list.

Escalation actions

An escalation action is defined by an*escalation schedule*, an*escalation method*(email, personal folder, pager, or mobile phone) and an*escalation message*.

7Define the escalation schedule.

An*escalation schedule*is defined by an*escalation wait period*, a*repeat schedule*, and a*repeat number*.

- To define the escalation wait period, enter the number of hours and minutes in the Start Escalation controls.
- To define the repeat schedule, enter the number of hours and minutes in the Repeat Escalation controls.
- To define the repeat number, enter a digit in the Repeat Number text box.

8Select an appropriate notification message for each notification method:

- To define an email notification action, select the Notify By Sending Email option and select appropriate internal and external messages.
- To define a pager notification action, select the Notify By Pager option and select an appropriate message.
- To define a mobile phone notification action, select the Notify By Mobile Phone option and select an appropriate message.
- To define a personal folder notification action, select the Notify By Personal Folder option.

Custom notification messages may be defined in the Define Email window. For step-by-step instructions see “Administering Notification Messages”.

Issue reassignment actions

9 Optional: To enable development issues to be automatically forwarded to a different workflow state or owner if no action is taken on that issue, define issue reassignment rules.

Issue reassignment actions may be defined in the Escalation Action window. For step-by-step instructions see “Administering Issue Reassignment Actions”.

Potential recipients

10Define potential recipients in the Potential Recipient tab.

A potential recipient is a project team member who*may subscribe*to an issue notification rule. A project team member may be identified as a potential recipient based on their account type, team group, or user variable.

Potential recipient subscription rules

11Define subscription rules in the Subscription Option tab.

Potential recipient subscription rules define whether a potential recipient can, cannot, or must subscribe to an issue notification. Subscription rules may be defined for each account type, team group, or user account and affect only those project team members that have been designated as potential recipients of issue notifications.

Subscribers

12Define subscriptions in the Potential Recipient tab.
Issue notification subscriptions define which project team members are subscribed and which project team members are not subscribed to each issue notification rule.Issue notification rules are specifically defined for each user account and override all potential recipient subscription rules.

Project administrators may use controls in the Email Escalations page to rename, issue escalation rules.

8.4.2 Understanding Issue Escalation Triggers

In DevTrack, all issue escalation triggers are “time triggers” which prompt the DevTrack Mail Service to send notification messages to issue stakeholders at key points in the issue lifecycle—for example, its start and finish dates—or when insufficient progress is made on a development issue.

An*escalation trigger*identifies the*unperformed*task that triggers an administrator-defined action (issue notification or issue re-assignment). Unlike issue notification triggers which represents specific acts performed on an issue, issue escalation triggers represents acts that*are not*performed on an issue.

DevTrack supports four issue escalation triggers:.

Open Too Long:The Open Too Long escalation trigger prompts escalation actions when an issue remains open beyond the period defined in the escalation schedule.

No Progress:The No Progress escalation trigger prompts escalation actions when there is no change in ownership or status beyond the period defined in the escalation schedule.

On Start Date:The On Start Date trigger prompts escalation actions when work on an issue has not started within the period defined in the escalation schedule.

On Finish Date:The On Finish Date trigger prompts escalation actions when work on an issue has not ended within the period defined in the escalation schedule.

In DevTrack, every escalation rule is defined by a single trigger. The scope of the escalation rule may be limited by one or more escalation conditions which limit the number of development issues that are subject to it.

8.4.3 Administering Issue Escalation Actions

Issue escalation actions define the time period that must pass before notifications are sent, the method used to notify subscribers (email, personal folder, pager, or mobile phone), the message delivered by the notification, and the number of times that notification are sent before an escalated issue is reassigned to another project team member.

An escalation action is defined by an*escalation schedule*, a*notification method*and, optionally, an*issue reassignment action:*

Escalation schedules

All escalation triggers are “time triggers”. DevTrack executes an escalation action when the “state” defined by the escalation trigger remains unchanged for the period defined in the*escalation schedule*.

An escalation schedule defines the time that must pass before notifications are delivered to escalation subscribers, the number of times that notifications are sent, and the time between notifications.

Notification methods

A notification method defines the method by which subscribers are notified that an issue has been escalated.

DevTrack supports four issue escalation notification methods:

Email Notification: Email notification actions notify subscribers by email whenever an issue notification rule is triggered.

Personal Folder Notification: Personal folder notification actions notify subscribers by personal folder whenever an issue notification rule is triggered.

Page Notification: Pager notification actions notify subscribers by pager whenever an issue notification rule is triggered.

Mobile Phone Notification: Mobile phone notification actions notify subscribers by mobile phone whenever an issue notification rule is triggered.

A different email message may be sent for every notification method.

Issue reassignment actions

An issue reassignment action defines rules for routing escalated development issues to specific workflow states or project team members at the end of the issue escalation schedule.

Issue reassignment actions are an optional feature that must be enabled and defined in each issue escalation rule.

8.4.4 Administering Issue Escalation Schedules

An escalation schedule defines the time that must pass before notifications are delivered to escalation subscribers, the number of times that notifications may be sent, and the time between each notification.

All escalation triggers are “time triggers”. DevTrack executes an escalation action when the “state” defined by the escalation trigger remains unchanged for the escalation wait period.

Using controls in the Escalation Action window, project administrators may define an escalation wait period, an escalation schedule, and a repeat number for each issue escalation rule.

Figure 7-3: Schedule Area of the Add Escalation Action Dialog Box

An*escalation schedule*is defined by an*escalation wait period*, a*repeat schedule*, and a*repeat number*."

Chapter 9 Email Integration Administration

9 Chapter 8- Email Integration Administration

In this chapter:

- Administrating DevTrack Email Integration
- Administrating the Email Retrieval Service
- Managing Issue Submission by Email

9.1 Administrating DevTrack Email Integration

This chapter covers a wide range of email-related features and email-related administrative tasks. The tasks described in this chapter are both system-level administrative tasks and project-level administrative tasks.

System administrators are responsible for installing the DevTrack Email Server service, the DevTrack Email Retrieval service, and configuring the DevTrack environment to operate properly. System administrators are also responsible for defining email error handling rules.

Project administrator may enable and define email-based tools that enable project members to work together more effectively. DevTrack email integration is a prerequisite for the following features:

- Issue autosubmission enables project members to submit issues to DevTrack by email. DevTrack automatically creates new issues based on the content of the subject line and body of the email.
- Customer contacts may send emails containing issue information to co-workers through the Beta Customer Portal.
- Issue notification enables project members to be notified automatically whenever an administrator-defined issue notification rule is triggered. Administrators may define all appropriate triggers, conditions, and recipients.
- Issue escalation enables project members to be notified automatically whenever an administrator-defined issue escalation rule is triggered. Administrators may define all appropriate triggers, conditions, and recipients.
- Event notification enables project members to be notified automatically whenever an administrator-defined event notification rule is triggered. Administrators may define all appropriate triggers, conditions, applicable event templates, and recipients.
- Event escalation enables project members to be notified automatically whenever an administrator-defined event escalation rule is triggered. Administrators may define all appropriate triggers, conditions, and recipients.

The DevTrack mail server must be installed before project administrator may enable email notification features.

9.2 Administrating the Email Retrieval Service

Project administrators may use tools in the Email Integration page to define the incoming mail server, the outgoing mail server, and specify which account types may submit issues through email.

The Email Integration page is divided in three tabs: the General Settings tab, Incoming Mail Server tab, the Outgoing Mail Server tab, and the Email Auto Submit tab. Each tab enables administrators to manage a different aspect of their project email integration settings.

- The*General Settings tab*enables administrators to enable email submission, define the email submission address, and define the email protocol for the project.
- The Incoming Mail Server tab enables administrators to define the POP3 incoming mail server properties. Incoming mail settings enable the DevTrack system to retrieve email from the server.
- The Outgoing Mail Server tab enables administrators to define SMTP outgoing mail server properties. Outgoing mail settings enable users to send email from within the DevTrack client, and email notification to be automatically delivered.
- The Email Auto Submit tab enables administrators to define which account types may submit issues through email.

Email retrieval administrative tasks include:

- Defining Incoming Mail Server Settings
- Defining Incoming Mail Server Properties
- Defining Email Autosubmit Settings
- Managing Issue Submission by Email

9.2.1 Defining Incoming Mail Server Settings

Project administrators may use controls in the*General Settings tab*of the in the Email Integration page to enable or disable the email submission, define the email address to which issues are to be sent, and define the email protocol for the project, and select an email protocol.

Administrators must enable email submission and otherwise project members may not submit issues through email.

Administrators must define an email address to which all issues submitted through email are sent. Only emails submitted to this email address by approved users will be automatically converted into new issues. The return email address is specific to the current project, as different development projects may use different accounts for email submission.

To define general email integration settings:

1. Select the Email Settings page in the Admin Tree window.
2. Select the General Settings tab.
3. Select the Support Email Submission check box.

   If this check box is not selected, no user may submit an issue through email regardless of other settings.
4. Specify the dedicated incoming mail account where team members will submit issues in the Email Address for issue submission field.
5. Select the email protocol to use for email integration: SMTP/POP3 or Microsoft MAPI.

   If administrators choose to use Microsoft MAPI, administrators must also specify a profile name.

9.2.2 Defining Incoming Mail Server Properties

Project administrators may use the Incoming Mail Server tab in the Email Integration page to define the POP3 incoming mail server properties that enable DevTrack to retrieve emails from the server.

DevTrack uses information entered in this form to log into the mail server and retrieve email submitted to the specified account.

Note: Entering an actual working email address here will cause all emails sending to this address disappear from the inbox and go directly into DevTrack issue queue. TechExcel *strongly recommends* using a test email account, or a new email account especially for DevTrack email submission purpose.

To define incoming mail server properties:

1. Select the Email Settings page in the Admin Tree window.
2. Select the Incoming Mail Server tab.
3. Specify the name or the TCP/IP address of the incoming mail server.

   Administrators must also specify a predefined email account name and password to access the server.
4. Specify the account name in the Account Name field.
5. Specify the account password in the Password field.

9.2.3 Defining Email Autosubmit Settings

The Email Auto-Submit tab in the Email Integration page enables administrators to define which account types may submit issues through email. all account types defined in the project.

To submit issues through email, an account type must have both the privilege to submit issues through email defined in the Email Auto-Submit tab of the Email Integration page and the Can Submit Issues privilege.

To define outgoing mail servers:

1. Select the Email Integration page in the Tree window.
2. Select the Email Auto Submit tab.
3. Select account type options displayed in the User Account Type list.

Any email these users send to the email address specified in the General Settings page will be submitted as an issue.

9.3 Managing Issue Submission by Email

DevTrack email submission enables project members to submit new issues to DevTrack projects by sending specially formatted email messages to an administrator-defined address.

Every five minutes, the DevTrack Email Retrieval Server automatically checks the mail server for new submission emails. Once received, DevTrack maps the text of the submission email to the corresponding issue properties and generates new issues based on the email subject line, body, and file attachments.

- The subject line of the email is converted to the title of the new issue (provided that the title is not defined in the body of the email itself).
- The body of the email is converted to description of the new email.
- Files attached to the email are converted to new issue file attachments.

Using advanced issue autosubmission, project members may identify and define multiple issue properties in each submission email.
Issues can only be automatically created by email submissions that are received from email addresses that have been defined by a system administrator in the User Manager. Email sent to the issue submission email address from addresses that are not associated with a DevTrack project cannot create new issues.

- DevTrack validates that the submitting email address is associated with a DevTrack user.
- If the email address is valid, DevTrack then checks to see that the associated user is assigned to an account type with the autosubmit privilege.

Only after both criteria are met will the email be converted into an issue.

TechExcel recommends that administrators create*email templates*for users who will be submitting issues through email. A sample email submission template may be sent as a template for project members who submit new issues by email. A user can paste this document in the body of an email, enter the properties for each field, and send the email to the dedicated incoming email account to automatically create more clearly defined issues.

9.3.1 Understanding Advanced Issue Autosubmission

Development teams may use advanced issue submission tags to identify and define multiple issue properties in each submission email.

As with standard submission email, the subject line of advanced submission email may be used to define the title of the new issue and email file attachments may be converted to issue attachments. Issue submitters may define much more than the issue description property in the body of submission email.

Issue submitters may divide the body of the submission email into two main sections introduced by markup tags that identify the information contained in that section of the email.

- The[Issue Properties] tag identifies the beginning of the issue property definitions. The field names and values must match the field names and values in the project. Issue submitters may add any issue fields from the Description page, the Current Status page, or the custom pages. However, the email must follow the same format for all fields.
- The [Description] tag identifies the beginning of the issue description. Text in this section of the email populates the issue Description field.

Both the [Issue Properties] tag and the [Description] tag must included in the body of the email for DevTrack to properly parse the email and populate the new issue fields.

If tags are not in the body of the email, DevTrack populates the issue title with the email subject and the issue description with the email body.

The following is an example of an email using the proper format for submission.

```
[Issue Properties]

Title = Screen refreshing problem

Current Owner = Scott Williams

Progress Status = New Defect

Type = Bug (Discrepancy Report)

Priority = 2 - High

Component = GUI, Reports, Web Server

Version = 3 - Build V2.0A

Platform = Windows 95

[Issue Description]

The executable needs more screen refreshing.
```

In the example above, if the GUI option is not an available value in the Component dropdown list in the DevTrack project, the field will remain undefined when the issue is created.

In addition to the two main areas of submission email (the[Issue Properties] and the [Description] area), issue submitters may also define custom tags that introduce a third area of the submission email. A custom issue property tag enables project members to define multiple lines of text as the issue property value for that issue property field. Project members may define one custom tag per issue submission email. The custom tag and its value definitions must be on separate lines and appear at the end of the email.

9.3.1.1 The [Issue Properties] tag

The[Issue Properties] tag identifies the beginning of the issue property definitions of the issue submission email.

Issue submitters may define issue properties for any data-entry control displayed in the Description page, the Current Status page, or the custom pages. The issue properties and issue property values must match the names of the data-entry controls and issue property values tracked in the project.

The syntax used to define issue properties depends on the control type that is used to manage and track the issue property:

- Single value fields (such as Edit Box controls and Dropdown List controls) require the title of the field, the equal sign(=), and an applicable value. For selection controls, the value defined must be a valid administrator-defined value.
- Multiple value fields (the Multi-Select control) require the title of the field, the equal sign(=), and one or more applicable values separated by commas. For example, Components = GUI, Reports, Web Server

DevTrack may only associate issue property values that already exist for each corresponding data-entry field, even if the control type of the issue property is an auto-grow combo box.

9.3.1.2 The [Description] tag

To adequately describe an issue generally requires a lengthy discussion of symptoms that is broken into multiple lines of text. Issue submitters may describe issues in the body of submission email using the [Description] tag.

The [Description] tag introduces the beginning of issue description property definitions and enables DevTrack to properly parse the submission email.

9.3.1.3 Custom tags for multiple lines of text

The custom tag introduces the beginning of an issue property definition consisting of*multiple lines of text*and enables DevTrack to properly parse that issue property in submission email.

DevTrack generally truncates the text of issue property definitions whenever it encounters a line break. Text following a line break is not parsed or included in the issue property definition. Issue submitters may define issue properties using multiple lines of text for two issue properties in each submission email: the issue description introduced by the [Description] tag and issue definitions introduced by a custom tag.

A custom tag consists of the name of the issue property enclosed in square brackets and must appear at the end of the submission email, after the [Properties] and [Description] tags.Examples of custom tags include [Title], [Work Description], or [Comments] tags.

To define an issue property using multiple lines of text, the following requirements must be met:

- The custom tag and the multiple lines of text must appear at the end of the submission email after the [Properties] and [Description] sections.
- The name of the custom tag must be that of an issue property and enclosed in square brackets: [and].
- The multiple lines of text that define the issue property must be on separate lines of body of the email.

For example, the [Title] custom tag introduces multiple lines of text that define the title property of an issue.

```
Component=GUI
Type=TP
(Third Party)
[Title]
First
Second
Third
```

The multiple lines of text are not separated by line breaks once they have been parsed and applied to the issue property. The example submission email creates an issue with the title “First Second Third”.

Chapter 10 Subproject Administration

10 Chapter 9- Subproject Administration

This chapter describes how development organizations may use subprojects to organize, prioritize, and schedule development issues.

In this chapter:

- Understanding DevSuite Subprojects
- Administering Subproject Settings
- Administering Subproject Statuses
- Administering Subproject Priorities

10.1 Understanding DevSuite Subprojects

In DevSuite, a subproject is a tool for organizing development issues into smaller, more manageable groupings of issues. Development teams may create subproject folders with subproject folders to organize their work and define subproject statuses, prioritize subprojects, and set due dates for all subprojects. Each subproject has its own description, priority, status, and start and end dates.

Every work project represents a distinct area of work and is defined by unique team structures (account types, team groups, and group folders) and workflow rules. Subprojects provide developers the means to organize, schedule, prioritize, and track groups of issues and to implement an additional layer of security within the project. Subprojects are ideal for development organizations that wish to develop multiple products within a single DevTrack project.

Subprojects are also useful to organizations developing products made up of multiple components or modules, or who wish to track many different types of issues in a single DevTrack project.

The subproject hierarchy is displayed in both the DevTrack Windows and Web client and the DevPlan smart client providing development organizations with a common framework for managing, tracking, and scheduling tasks.

In DevTrack, the subproject hierarchy is displayed in the issue tree panel and organizes development issues by product, version, and build.

Figure 9-1: Subproject Enabled Client GUI

In DevPlan, the subproject hierarchy is displayed in the issue tree panel and organizes development issues by product, version, and build.

Figure 9-2: Subproject Enabled Client GUI

Subprojects serve three primary purposes in DevTrack projects:

- Subprojects facilitate the management of multiple development projects within a single DevTrack project."
- Subprojects enable development teams to define rules for managing issues within a subproject.
- Subprojects enable development teams to use multiple workflow to manage project issues

Note: If TechExcel DevPlan integration is enabled for a DevTrack project, all subproject management tasks are performed in the DevPlan smart client. Project team members may create and manage subprojects in the DevTrack client only if DevPlan integration is not enabled.

In DevTrack Admin, project administrators may define rules for defining the status and priority of subprojects, and account type access privileges, which determine which project members may create, edit, and delete subprojects within the DevTrack project.

- The status of a subproject enables project members to indicate the current progress state of a subproject. The status of a subproject may be defined independently or be linked to the progress state of the issues in the subproject.
- The priority of a subproject enables project members to indicate the current priority of a subproject relative to other subprojects.

All subproject properties may be defined in the Subproject manager within the DevTrack client.

Administrators may use tools in the Subprojects subfolder to enable subprojects, define subproject statuses, subproject priorities, and subproject access privileges.

10.1.1 Understanding Subprojects and the Beta Customer Portal

Projects that are using the Beta Customer Portal may wish to route all issues submitted from beta customers through the Beta Customer Portal to a subproject.

Project administrators may define a default subproject for every customer created in a customer base project. All issues submitted by that customer are submitted to a DevTrack project would by default be created in the selected subproject.

10.2 Administering Subproject Settings

In DevSuite, a subproject is a discreet area of work that corresponds to a specific product, feature, version, or build. Using subprojects, development teams may divide a DevTrack project into smaller, more manageable groupings of issues. Project members may schedule, prioritize, and track those issues using subproject properties and, optionally, subproject workflow.

Using controls in the General Settings page, project administrators may enable subproject support in a DevTrack project and to enable team member access controls, applicable issue owners, and applicable issue statuses in subprojects.

Project administrators must enable support for subprojects in each DevTrack project. If subproject support is not enabled no subproject settings can be implemented within the project and the issue tree panel does not appear in the DevTrack Windows client.

10.2.1 Enabling Subproject Support

In TechExcel DevTrack, subprojects are an optional Enterprise Edition feature that must be enabled in each work project. Basic subproject administrative tasks include enabling subproject support, enabling DevPlan integration support, and defining how subprojects are identified and managed in the integrated DevSuite site.

To enable subprojects:

1. Select the General Settings page in the Subprojects subfolder.
2. Select the Enable Subproject Support check box.

3 Enter an alternative title for subprojects the Refer As field.

The Refer As option enables project administrators to change the Subproject label displayed in the DevTrack client. The text entered in the Refer As field is displayed instead of the word *Subproject* throughout the project in both the Windows and Web clients.

The text entered does not change the title of the Project Issues folder in the Issue Tree window of the DevTrack Windows client.

4 Select a default subproject from the Default Subproject Creation dropdown list.

10.2.2 Defining Subproject Terminology

The Refer As option enables project administrators to change the Subproject label displayed in the DevTrack, DevSpec, and DevPlan clients. The text entered in the Refer As field is displayed instead of the words *subproject* throughout the project in both the Windows and Web clients.

To define subproject terminology:

1 Select Advanced Features > Subproject > General Settings in the tree panel.

2 Enter an alternative title for subprojects the Refer As text box. The text entered does not change the title of the Project Issues folder in the issue tree panel of the DevTrack Windows client.

10.2.3 Defining Default Subprojects

To define default subprojects:

1 Select Advanced Features > Subproject > General Settings in the tree panel.

2 Select a default subproject from the Default Subproject Creation dropdown list.

10.2.4 Defining Applicable Issue Types

In DevTrack, an issue type represents a class of development issues share similar properties and which need to be managed in a unique way. De vTrack workflow rules are generally based on issue types.

The issue type of an issue is typically linked to its type definition (for example, bug, new feature, enhancement), but alternatively may be linked to its priority, component, platform, build, version, or any other criteria.

To define applicable issue types:

1 Select Advanced Features > Subproject > Subproject Issue Type in the tree panel.

2 Select the Enable Applicable Issue Type Based on Subproject check box.

3 Select an issue type in the Issue Type list.

4 Click the Edit button. The Associate Issue Type with Subproject dialog box appears.

5 Select one or more issue types.

6 Click the OK button.

The Associate Issue Type with Subproject dialog box closes and the issue type is associated with the subproject.

10.3 Administering Subproject Statuses

In DevSuite, a subproject is a tool for organizing development issues into smaller, more manageable groupings of issues so that development teams may manage and track related de velopment issues as a unit.

Every subproject is defined by its *subproject status* which indicates the current progress of the subproject within the development lifecycle. The lifecycle of subprojects may be uniquely defined in each project or be based on the lifecycle of the development issues managed in that project..

The status of a subproject may be automatically calculated based on administrator-defined rules or manually defined by project members in the DevPlan client.

DevSuite supports four methods of deteriming the status of subprojects in a project: the Independent Of Its Incidents method, Derived from the Highest-ranked Incident Status method, Derived from the Lowest-Ranked Incident Status method, and the Linked to Incident Status Definition method.

Independent of its Incidents Status: The Independent of its Incidents Status method requires that project team members manually define the status of all subprojects using controls in the DevPlan client. Subproject states are fully customizable.

Derived from the Highest-ranked Incident Status: The Derived from the Highest-ranked Incident Status method ties the status of a subproject to the highest ranked issue within the subproject. Subproject statuses are automatically calculated based on administrator-defined issue status rankings.

Derived From The Lowest-ranked Incident Status: The Derived From The Lowest-ranked Incident Status status method ties the status of a subproject to the lowest ranked issue within the subproject. Subproject statuses are automatically calculated based on administrator-defined issue status rankings.

Linked to Incident Status Definition: The Linked to Incident Status Definition status method method requires that project team members manually define the status of all subprojects using controls in the DevPlan client.. Subproject states are identical to development issue states.

Using controls in the Subproject Status page, project administrators may define the method used to define the subproject status of subprojects.

10.3.1 Defining Independent Subproject Statuses

Independent subproject statuses are one of four types of subproject status rules that an administrator may define in the Subproject Status page of DevTrack Admin. Administrators must select the*Independent of its Incidents Status option*in the Status Option dropdown list of the Subproject Status page.

Adminsitrators may add, edit, and delete independent subproject statuses in the Subproject Status page. Although administrators may rank independent subproject statuses the ranks assigned have no effect using this option.

> **Note:**Adminstrator-defined independent subproject status options are overwritten if the administrator selects the Linked to Incident Status option in the Status Option dropdown list.

To add independent subproject statuses:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Status option from the Status Option dropdown list.
3. Click the Add button.

   The Status dialog box appears.
4. Enter the name of the status in the Name field.
5. Click the OK button.

To edit independent subproject statuses:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Status option from the Status option dropdown list.
3. Select a status in the Status list.
4. Click the Edit button. The Status dialog box appears.
5. Enter the name of the status in the Name field.
6. Click the OK button.

To delete independent subproject statuses:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Status option from the Status option dropdown list.
3. Select a status in the Status list.
4. Click the Delete button. A Warning dialog box appears.
5. Click the Yes button.

10.3.2 Defining Subproject Status Based on Highest Ranking

Administrators may use controls in the Subproject Status page to define rules for determining the status of subprojects in DevTrack projects based on issue status rankings.

Administrators may select the Derived from the Highest-ranked Incident Status option in the Status Option dropdown list to tie the status of a subproject to the status of the highest-ranked issue in that subproject. If this option is selected, the status of subprojects is defined automatically based on the statuses of the issues contained in that subproject.

Each subproject represents multiple issues. The progress status of each issue is represented by its issue status. Administrators may use controls in the Subproject Status page to rank the issue status options used in the project. The issue with the highest-ranking issue status determines the subproject status of the entire subproject.

The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1” is ranked higher than a status assigned a “5.”

To rank issue status options:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Derived From

   The Highest-ranked Incident Status Status option from the Status Option dropdown list.
3. Select an option in the Status list.

   The Status list displays all of the status options that may be selected for a DevTrack issue.
4. Click the Edit button.

   The Status dialog box appears.
5. Enter the rank of the status in the Status field.

   The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1” is ranked higher than a status assigned a “5.”
6. Click the OK button.

   The rank of the issue status is displayed in the Status list.

10.3.3 Defining Subproject Status Based on Lowest Ranking

Administrators may use controls in the Subproject Status page to define rules for determining the status of subprojects in DevTrack projects based on issue status rankings.

Administrators may select the Derived from the Lowest-ranked Incident Status option in the Status Option dropdown list to tie the status of a subproject to the status of the lowest-ranked issue in that subproject. If this option is selected, the status of subprojects is defined automatically based on the statuses of the issues contained in that subproject.

Each subproject represents multiple issues. The progress status of each issue is represented by its issue status. Administrators may use controls in the Subproject Status page to rank the issue status options used in the project. The issue with the lowest-ranking issue status determines the subproject status of the entire subproject.

The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1” is ranked higher than a status assigned a “5.”

To rank issue status options:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Derived From

   The Lowest-ranked Incident Status Status option from the Status Option dropdown list.
3. Select an option in the Status list.

   The Status list displays all of the status options that may be selected for a DevTrack issue.
4. Click the Edit button.

   The Status dialog box appears.
5. Enter the rank of the status in the Status field.

   The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1” is ranked higher than a status assigned a “5.”
6. Click the OK button.

   The rank of the issue status is displayed in the Status list.

10.3.4 Defining Subproject Status Based on Issue Status

Administrators may use controls in the Subproject Status page to link subproject status to the options used to define the status of issues in the project.

The Linked to Incident Status Definition Status option enables project teams to use the status options that define DevTrack issues to define DevTrack subprojects as well. Project members may manually define the status of a subproject whenever they create or edit a subproject in the Subproject manager. The options displayed in the subproject Status Option dropdown list are identical to the options displayed in the Status dropdown list that is used to define issue status.

Administrators may edit, but not add or delete status options in the Subproject Status page. Because subproject status options are tied to issue status options, new sub-status options are made available whenever the administrator creates new issue statuses in the Workflow page.

Although administrators may rank subproject statuses in the Subproject Status page, the ranks assigned have no effect using this option.

To link subproject status option to issue status options:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Linked To Incident Status Definition status option from the Status option dropdown list.
3. Select an option in the Status list.

   The status options displayed in the Status list are identical the status options used to define issues in the DevTrack project.
4. Click the Edit button.

   The Status dialog box appears.
5. Click the OK button.

10.4 Administering Subproject Priorities

The subproject priority defines the relative importance of a subproject relative to other areas of development.

The priority of each subproject may be defined manually or defined automatically depending on administrator-defined subproject definition rules.

DevSuite supports four methods of deteriming the priority of subprojects in a project: the Independent Of Its Incidents method, Derived from the Highest-ranked Incident Status method, Derived from the Lowest-Ranked Incident Status method, and the Linked to Incident Status Definition method:

Independent of its Incidents Priority: The Independent of its Incidents Priority option enables administrators to define a unique set of options to indicate the priority of a subproject. Project members may manually define the priority of a subproject by selecting one of these administrator defined options in the Priority dropdown list in the Subproject Manager.

Derived from the Highest-ranked Incident Priority: The Derived from the Highest-ranked Incident Priority option ties the priority of a subproject to the highest ranked issue within that subproject. The rank of issues in a subproject is based on administrator-define rankings of the options in one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field. The priority of each subproject is determined automatically based on the rank the administrator assigns to

the options available in the corresponding issue control. The subproject Priority dropdown list in the Subproject Manager is grayed out and cannot be edited.

Derived from the Lowest-ranked Incident Priority: The Derived from the Lowest-ranked Incident Priority option ties the priority of a subproject to the lowest ranked issue within that subproject. The rank of issues in a subproject is based on administrator-define rankings of the options in one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field. The priority of each subproject is determined automatically based on the rank the administrator assigns to the options available in the corresponding issue control. The Subproject Priority dropdown list in the Subproject Manager is grayed out and cannot be edited.

Linked to Incident Priority Definition: The Linked to Incident Priority Definition option enables the administrator to tie the priority of a subproject to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field. The options available for selection in the subproject Priority field are identical to the options displayed in the selected issue control.

10.4.1 Creating Independent Subproject Priorities

Administrators may use controls in the Subproject Priority page to define a unique set of options that DevTrack project members may use to identify the current priority of a subproject.

In the DevTrack client project members may change the priority of a subproject by selecting an option from the Priority dropdown list of the Subproject manager. Administrators may define these priority options using controls in the Subproject Priority page.

Independent subproject priorities are one of four types of subproject priority rules that an administrator may define in the Subproject Priority page of DevTrack Admin. Administrators must select the*Independent of its Incidents Priority option*in the Priority Option dropdown list of the Subproject Priority page.

Adminsitrators may add, edit, and delete independent subproject priority optons in the Subproject Priority page. Although administrators may rank independent subproject priorities the ranks assigned have no effect using this option.

> **Note:**Adminstrator-defined independent subproject priority options are overwritten if the administrator selects the Linked to Incident Priority option in the Priority Option dropdown list.

To add an independent subproject priority:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Priority option from the Priority Option dropdown list.
3. Click the Add button. The Priority Dialog dialog box appears.
4. Enter a title for the priority in the Name field.
5. Click the OK button.

   The Priority Dialog dialog box closes and the priority is displayed in the Priority list.

To edit an independent subproject priority:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Priority option from the Priority Option dropdown list.
3. Select a priority in the Priority list.
4. Click the Edit button.

   The Priority Dialog dialog box appears.
5. Enter a title for the priority in the Name field.
6. Click the OK button.

   The Priority Dialog dialog box closes and the updated priority is displayed in the Priority list.

To delete independent subproject priority:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Priority option from the Priority Option dropdown list.
3. Select a priority in the Priority list.
4. Click the Delete button.

   A Warning dialog box appears.
5. Click the Yes button.

10.4.2 Defining Highest Subproject Priorities

Project administrators may use controls in the Subproject Priority page to define rules for determining the priority of subprojects in DevTrack projects based on issue priority rankings.

Administrators may select the Derived from the Highest-ranked Incident Priority option in the Priority Option dropdown list to tie the priority of a subproject to the “priority” of the highest-ranked issue in that subproject. If this option is selected, the priority of subprojects is defined automatically based on the priorites of the issues contained in that subproject.

The relative priority of each subproject may be based on a number of different criteria. Administrators may link the subproject Priority control to one of five issue system fields (the

Type field, Priority field, the Component field, the Version field, or the Platform field) and rank each of the options displayed in the selected issue control.

Each subproject represents multiple issues. The issue with the highest-ranking priority determines the subproject priority of the entire subproject.

The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1” is ranked higher than a status assigned a “5.”

To define highest subproject priorities:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Derived From Its Highest-ranked Incident Priority option from the Priority option dropdown list.
3. Select an option in the Link to Priority dropdown list.

   Administrators may link subproject priority to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field.
4. Select an option in the Priorities field.

   The Priorities field displays options belonging to the list control selected in the Link to Priority dropdown list.
5. Click the Edit button. The Priority dialog box appears.
6. To edit the rank of the option enter a value in the Rank field.

   The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1” is ranked higher than a status assigned a “5.”
7. Click the OK button.

10.4.3 Defining Lowest Subproject Priorities

Administrators may use controls in the Subproject Priority page to define rules for determining the priority of subprojects in DevTrack projects based on issue priority rankings.

Administrators may select the Derived from the Lowest-ranked Incident Priority option in the Priority Option dropdown list to tie the priority of a subproject to the “priority” of the lowest-ranked issue in that subproject. If this option is selected, the priority of subprojects is defined automatically based on the priorites of the issues contained in that subproject.

The relative priority of each subproject may be based on a number of different criteria. Administrators may link the subproject Priority control to one of five issue system fields (the Type field, Priority field, the Component field, the Version field, or the Platform field) and rank each of the options displayed in the selected issue control.

Each subproject represents multiple issues. The issue with the highest-ranking priority determines the subproject priority of the entire subproject.

The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1” is ranked higher than a status assigned a “5.”

To define lowest subproject priorities:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Derived From Its Lowest-ranked Incident Priority option from the Priority option dropdown list.
3. Select an option in the Link to Priority dropdown list.

   Administrators may link subproject priority to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field.
4. Select an option in the Priorities field.

   The Priorities field displays options belonging to the list control selected in the Link to Priority dropdown list.
5. Click the Edit button.

   The Priority dialog box appears.
6. To edit the rank of the option, enter a value in the Rank field.

   The lower the number assigned to the issue status, the higher the ranking of that issue status. For example, an issue status that is assigned a “1” is ranked higher than an issue status that is assigned a “5.”
7. Click the OK button.

10.4.4 Defining Linked to Issue Subproject Priorities

Administrators may use controls in the Subproject Priority page to link subproject priorities to options belonging to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field.

The Linked to Incident Status Definition Priority option enables project teams to use options that define DevTrack issue properties to define DevTrack subproject priorities as well. Project members may manually define the priority of a subproject whenever they create or edit a subproject in the Subproject manager. The options displayed in the subproject Priority Option dropdown list are identical to the options displayed in the linked issue list control (the Type field, Priority field, the Component field, the Version field, or the Platform field)

Adminisitrators may edit, but not add or delete priority options in the Subproject Priority page. Because subproject priority options are tied to system field options, new sub-status priority options are made available whenever the administrator defines new issue options for the corresponding issue control.

Although administrators may rank subproject priorities in the Subproject Status page, the ranks assigned have no effect using this option.

To define linked to incident priority subproject priorities:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Linked to Incident Priority Definition option from the Priority Option dropdown list.
3. Select an option in the Link to Priority dropdown list.

   Project administrators may link subproject priority to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field.
4. Select an option in the Priorities field.

   The Priorities field displays options belonging to the list control selected in the Link to Priority dropdown list.
5. Click the Edit button.

   The Priority dialog box appears.
6. Enter the rank of the priority in the Rank field.
7. Click the OK button.

Chapter 11 Time and Cost Tracking Administration

11 Chapter 10- Time and Cost Tracking Administration

This chapter describes DevTrack issue-level time and cost tracking, time group and time category administration, time and cost tracking methods, and the assignment of project-level time and cost tracking privileges.

In this chapter:

- Understanding DevSuite Development Time and Cost Tracking
- Administering Time Groups and Categories
- Administering Time and Cost Tracking Privileges

11.1 Understanding DevSuite Development Time and Cost Tracking

DevTrack enables development organizations to track the time that project members spend working on development issues and calculate the cost of their development efforts.

In DevTrack, the time and cost spent on development tasks Time and cost tracking data may be viewed enabling development organizations to identify and analyze development and user trends.

Time Tracking:Time tracking measures the amount of time that project members spend working on different kinds of development tasks. Time entries enable project teams to track the time that each project member spends on an issue.

Cost Tracking:Cost tracking calculates the cost of the hours each project member spends on development tasks based on an administrator-defined hourly rate that is defined for each work type. The hourly rate for each work type may be defined by account type or user account.

Issue-level time cost tracking are optional DevTrack features that must be enabled in the project by an administrator.

Time tracking may be implemented without cost tracking, but the cost-tracking feature requires that time tracking features be enabled and defined.

11.1.1 Understanding DevSuite Time and Cost Tracking Administration

DevTrack development time and costs are tracked by*time group*and*time category*. DevSuite time tracking and cost tracking are interrelated tools that rely on administrator-defined time groups and time categories to manage and track development tasks.

Development tasks and costs are defined at two levels providing the organization the flexibility to group many different types of tasks for reporting and cost calculation. In DevSuite, time and cost tracking is defined at both the system-level and the project-level.

System Level: Time groups and time categories are defined at the systemlevel and are shared by every project (DevSpec or DevTrack) in a DevSuite site.

Project Level: In DevTrack and DevSpec, time and cost tracking may be enabled on a project-by-project basis. Time groups and time categories must be defined as applicable to a project to be used for time and costing tracking in that project. Time category default costs may be overridden at the project-level.

In DevSuite, time groups and time categories are defined at the system-level and are shared by all DevTrack and DevSpec projects in a DevSuite site.

Project administrators may enable time tracking and cost tracking in DevTrack projects and to define time tracking settings in the Issue Level Time Track page.

11.2 Administering Time Groups and Categories

All DevTrack issue-level time and cost tracking is based on administrator-defined time groups and time categories. DevSuite time groups and time categories provide the organization the flexibility to group many different types of tasks for reporting and cost calculation.

Each administrator-defined time category represents a development task or a set of development tasks that the business wishes to track together.

Time Group:A time group is an administrator-defined grouping of time categories that identifies different types of work that share common characteristics. Each time group belongs to one of four time types: Design Time time type, Development Time time type, Testing Time time type, and Other time type.

Time Category: Time categories enable development organizations and group together many different tasks for reporting and cost calculation.

Time groups and time categories enable development organizations to manage and track development tasks and costs. Time groups and child time categories are managed in the time category tree, a hierarchical structure of folders and subfolders.

In DevSuite, all time groups and time categories are defined at the system-level and are shared by every other DevTrack and DevSuite project in the DevSuite site. Project administrators are responsible for enabling support each time category that is used in their project.

11.2.1 Selecting Applicable Time Categories

In DevTrack, development organizations may use system-defined time categories to measure the time and cost of their development efforts.

All time groups and time categories are defined and maintained in a system settings project and shared by every project in a DevSuite site. To use a time category to measure the time and cost of development efforts in a DevTrack project, the project administrator must enable support for that time category.

Using controls in the Select Time Category window, project administrators may identify applicable time categories.

Figure 10-1: Select Time Category Window

A time category is a system-defined set of business rules that represent a specific set of development tasks." Every time category is defined by its title, default hourly rate, and a parent time group, which determines its time type (Design Time time type, Development Time time type, Testing Time time type, and Other time type).

To define a time category as applicable:

1 Select Advanced Features > Time and Price Analysis.

2 Click the Select button in the Time Category area.

The Select Time Category window appears.

3 Select one or more time categories in the tree and click the OK button.

The Select Time Category window closes.

11.2.2 Enabling Time Tracking

To enable time tracking, select the Enable Time Tracking check box in the Issue Level Time Track page of the Advanced Features folder.

11.2.3 Enabling Cost Tracking

To enable cost tracking, select the Enable Cost Tracking check box in the Issue Level Time Track page of the Advanced Features folder.

Time tracking may be implemented without cost tracking, but the cost-tracking feature requires that time tracking features be enabled and defined.

11.2.4 Defining Project Time Tracking Methods

Time tracking enables development organizations to track the amount of time that project members spend working on development tasks.

All time tracking calculations are based on data entered by project members in the Issue Time Track manager of the DevTrack client. Project members may add time entries whenever they forward, close, or create development issue.

Using controls in the Account Type Hourly Rate tab of the Issue Level Time Track page, project administrators may define the hourly rate of a time category by account type.

Figure 10-2: Select Time Category Window

All time entries are prompted by an administrator-enabled time tracking settings." The time that project members spend on a development task may be tracked by one of three different methods.

Record Time Usage when an Issue is Forwarded:The Record Time Usage when an Issue is Forwarded method requires that project members enter time tracking data whenever they forward an issue.

Record Time Usage when an Issue is Closed:The Record Time Usage when an Issue is Closed method requires that project members enter time tracking data whenever they close an issue.

Record Time Usage when an Issue is Created:The Record Time Usage when an Issue is Created method requires that project members enter time tracking data whenever they create an issue.

Project administrators may mandate that time tracking figures are manually entered whenever an issue is created, forwarded, or closed.

To define time tracking settings:

1. Select the Issue Level Time Track page in the Advanced Features folder of the tree panel.
2. Select the Enable Issue Level Time Tracking check box in the General Setting tab.
3. Identify the time categories that are applicable to the current project.

   A time category is a system-defined set of business rules that represent a specific set of development tasks. Every time category is defined by its title, default hourly rate, and a parent time group, which determines its time type (Design Time type, Development Time type, Testing Time type, and Other time type).
4. Select time tracking options:
   - The Record Time Usage when an Issue is Forwarded option requires that project members enter time tracking data whenever they forward an issue.
   - The Record Time Usage when an Issue is Closed option requires that project members enter time tracking data whenever they close an issue.
   - The Record Time Usage when an Issue is Created option requires that project members enter time tracking data whenever they create an issue.
5. Define time and cost tracking privileges.

   For step-by-step instructions see “Administering Time and Cost Tracking Privileges”.

11.2.5 Defining Hourly Rate by Account Type

Cost tracking enables development organizations to calculate the cost of development efforts performed in projects. The cost of development activities is determined by time category hourly rates.

The hourly rate of a time category may be defined at three levels: the default hourly rate, the account type hourly rate, and the project member hourly rate. The account type member hourly rate defines the rate per hour for project team members belonging to a specific account type and overrides the default hourly rate.

Using controls in the Account Type Hourly Rate tab of the Issue Level Time Track page, project administrators may define the hourly rate of a time category by account type.

Figure 10-3: Account Type Hourly Rate Tab

A time category is a system-defined set of business rules that represents a development task." Every time category is defined by its title, default hourly rate, and a parent time group. The default hourly rate of a time category may be overridden at the project-level by an updated default hourly rate, an hourly rate by account type, or an hourly rate by project team member.

To define hourly rate by account type:

1. Select the Issue Level Time Track page in the Advanced Features folder of the tree panel.
2. Select an account type in the Account Type list of the Account Hourly Rate tab.
3. Select a time category in the Time Category list.
4. Enter a figure in the Default Hourly Rate field.

11.2.6 Defining Hourly Rate by User Account

Cost tracking enables development organizations to calculate the cost of development efforts performed in projects. The cost of development activities is determined by time category hourly rates.

The hourly rate of a time category may be defined at three levels: the default hourly rate, the account type hourly rate, and the project member hourly rate. The project member hourly rate defines the specific rate per hour for a specific project team member and overrides the default and account type-based rates.

Using controls in the Member Hourly Rate tab of the Issue Level Time Track page, project administrators may define the hourly rate of a time category by project team member.

Figure 10-4:Member Hourly Rate Tab

A time category is a system-defined set of business rules that represents a development task." Every time category is defined by its title, default hourly rate, and a parent time group. The default hourly rate of a time category may be overridden at the project-level by an updated default hourly rate, an hourly rate by account type, or an hourly rate by project team member.

The hourly rate of each time categories may be defined at three-levels: the default hourly rate, the account type hourly rate, and the project member hourly rate

Project administrators may enable and define time tracking settings for DevTrack projects in the Issue Level Time Track page.

To define hourly rate by user account:

1. Select the Issue Level Time Track page in the Advanced Features folder of the tree panel.
2. Select a project member in the Member list of the Member Hourly Rate tab.
3. Click the Edit button.

   The Member Hourly Rate dialog box appears.
4. Select a time category in the Time Category list.
5. Define the project member hourly rate for the selected time category
6. Click the OK button.

   The Member Hourly Rate dialog box closes and project member hourly rate is displayed in the Project Member Hourly Rate tab.

11.3 Administering Time and Cost Tracking Privileges

In DevTrack, a privilege is the right to perform a specific task such as creating, editing, or deleting a work items such as a n issue, event, or report.

DevTrack privileges and access rights are not granted to user accounts directly, but assigned to administrator-defined account types. Users are assigned an account type when they join a project and assume the rights and privileges granted to that account type—when a user's responsibility changes, he or she may be assigned a different account type.

Using controls in the Account Privileges tab of the Issue Level Time Track, project administrators may grant time and cost management privileges to project members of each account type:

Can Edit and Delete Other's Time Entry:The Can Edit and Delete Other's Time Entry privilege enables managers to view, edit, and delete time entries made by other project members.

Can Access Cost Information: The Can Access Cost Information privilege enables managers to define the hourly cost of work type activities.

To define time and cost tracking privileges:

1. Select the Issue Level Time Track page in the Advanced Features folder of the tree panel.
2. Select the Account Privilege tab.
3. Select an account type in the Account Type list.
4. Select privilege options for the selected account type.
   - Can Edit and Delete Other's Time Entry
   - Can Access Cost Information.

Chapter 12 Knowledge Management Administration

12 Chapter 11- Knowledge Management Administration

Communication is a key component to the development of quality products. Issue knowledge management tools enable development organizations to ensure that all project stakeholders may access up-to-date and accurate information about development tasks and the project itself.

In this chapter:

- Understanding DevTrack Knowledge Management
- Administering Project-level Knowledge Management
- Administering Issue-level Knowledge Management

12.1 Understanding DevTrack Knowledge Management

- **Project knowledge management** is the control of knowledge items (documents, files, topics, and URLs) that are relevant to multiple issues, subprojects, or the project itself. DevTrack knowledge management tools enable development organizations to manage multiple versions of documents and to link version-controlled documents to relevant development issues.

- **Issue knowledge management** is the control and tracking of issue-specific data. Development issues are tools for tracking development tasks and for sharing key information with all issue stakeholders. The Note page provides development organizations with a central repository for storing and sharing information about that specific development task.

12.2 Administering KnowledgeWise Integration

12.2.1 Enabling KnowledgeWise Integration

To integration a DevTrack development project with a KnowledgeWise knowledge base project, select the Enable KnowledgeWise Integration check box in the KnowledgeWise Integration page.

12.2.2 Enabling DevSpec Integration

To integration a DevTrack development project with a DevSpec project, select the Enable DevSpec Integration check box in the KnowledgeWise Integration page.

12.2.3 Migrating Knowledge View Data

Using the Fields Mapping wizard, project administrators may map knowledge item fields and field values to KnowledgeWise fields and values and map knowledge item statuses.

Figure 11-1: Fields Mapping Wizard

To map knowledge fields and field values:

1 Select Project Settings > KnowledgeWise Integration in the tree panel."

2 Click the Migrate Data from Knowledge View button.

The Fields Mapping wizard appears.

map knowledge fields

3 Select a knowledge item field in the Knowledge View Field list and click the Map button.

The Field Map dialog box appears.

4 Select a KnowledgeWise knowledge field in the KnowledgeWise Field dropdown list.

5 Click the OK button.

The Field Map dialog box closes.

define open and closed knowledge states

6 Select a workflow state in the Workflow State for Open Documents dropdown list.

7 Select a workflow state in the Workflow State for Closed Documents dropdown list.

The Choice Fields Map page appears.

map knowledge field values

8 Map DevTrack knowledge item field values to KnowledgeWise field values.

9 Click the Next button.

A confirmation dialog box appears.

10 Click the Yes button.

The Fields Mapping wizard closes.

12.3 Administering Project-level Knowledge Management

DevTrack provides development organizations with a central knowledge base for all project-related documentation through integration with TechExcel KnowledgeWise.

All project knowledge is managed in the Knowledge view of the DevTrack Windows client.

12.3.1 Understanding KnowledgeWise

All documents that are created as part of the planning processes including customer requirements, functional specifications, technical specifications, database schemas, GUI design documents may be managed and stored in an integrated KnowledgeWise knowledge base.

KnowledgeWise enables development organizations to enforce change management, security, transparency to project deliverables, and facilitates collaboration by distributed development teams.

- KnowledgeWise enables all stakeholders across multiple projects to access and view the most recent version of all documents.
- The DevTrack Knowledge view enables development teams to manage all project-related knowledge items (documents, HTML links, topics and attachments) including internal design documents, system requirements, user specifications, and test cases in a central repository.
- KnowledgeWise organizes documents by functional area. All documents are linked to an activity and all stake holders involved in that activity may quickly view all relevant documentation.
- KnowledgeWise features built-in version control tools that ensure that all project development teams no matter where they are in world always have access to the most up-to-date documents.
- KnowledgeWise provides development organizations with a secure repository for all for all project documentation. Access to all knowledge items is protected by privilege-based access controls. The ability of project members to view, edit, lock, check out, and check in protected files is based on their role in the planning project.Knowledge items include documents, HTML links, knowledge topics, and issue

Project administrators may enable Knowledge view support in DevTrack projects and to customize many aspects of the Knowledge view*graphical user interface*(GUI) in the Knowledge folder of the Admin Tree window.

12.3.2 Understanding Knowledge Items

DevTrack enables project members to manage knowledge items (documents, web pages, topics, and attachments) in the Knowledge view. Knowledge items, especially design documents, are a key link between the project planning processes and the software development processes that are managed and tracked in TechExcel DevTrack.

The Knowledge view contains tools designed to manage four different types of knowledge items. The Knowledge Tree window organizes knowledge items into a hierarchy of folders.

- Documents include all external files that saved in the DevTrack database through the Knowledge view. Documents may either be created internally from document templates or externally using a word processing or spreadsheet program.
- HTML links enable project teams to create a library of project-related URLs. Development issues can be linked with a dynamically changing web site or page-based HTML document for easy reference and access.
- Topics are knowledge items specifically designed to enable project teams to build a knowledge base of problems and resolutions. Each topic knowledge item consists of a description page, a resolution page, and a links page.No file is attached and the information is maintained in the database. Topics can help save time in resolving redundant issues.
- Attachments are documents that have been attached to issues in the Main view Notes page. Although these knowledge items are visible through the Knowledge view, they are managed using tools in the Notes page.

12.3.3 Understanding Knowledge View Access Controls

The ability of project members to perform tasks in the knowledge view is based on the Knowledge view privileges granted to their account type by an administrator in DevSuite Admin.

Project administrators are responsible for assigning account types to individual user accounts, and Knowledge view privileges to each account type. In DevTrack all system and project-level privileges are assigned to account types rather than to individual user accounts. Each user account gains all of the access controls granted to their account type.

Can Publish and Modify Knowledge Item: The Can Publish and Modify Knowledge Item privilege enables an account type to create and edit all knowledge items including documents, topics, HTML links, and attachments.

Can Modify Knowledge Structure: The Can Modify Knowledge Structure privilege enables an account type to create, rename, and edit folders in the Document tree window.

Can Get Read Only Copy of File: The Can Get Read Only Copy of File privilege enables an account type to open and view copies of knowledge items. Account types assigned this privilege cannot make changes to the file, create, or save knowledge items.

Can Get and Modify a File: The Can Get and Modify a File privilege enables an account type to create and edit documents attached to knowledge items.

Can Unlock a File: The Can Unlock a File privilege enables an account type to unlock locked documents.

Can Close: The Can Close privilege enables an account type to close knowledge items.

Can Reopen The Can Reopen privilege enables an account type to reopen closed knowledge items.

Can Get a Read-Only Copy of a Closed File: The Can Get a Read-Only Copy of a Closed File privi*lege*enables an account type to open an read-only copy of closed files.

12.3.4 Enabling the Knowledge View

To enable Knowledge view support, select the Enable Document View Support check box in the Settings page of the Knowledge View subfolder.

Knowledge management must be enabled in each DevTrack project.

12.3.5 Selecting a Revision System

To select a revision system, select an option in the Revision System dropdown list in the Settings page of the Knowledge View subfolder.

In the current implementation of DevTrack the DevTrack revision system is the only available option.

12.3.6 Defining the Maximum Number of Revisions

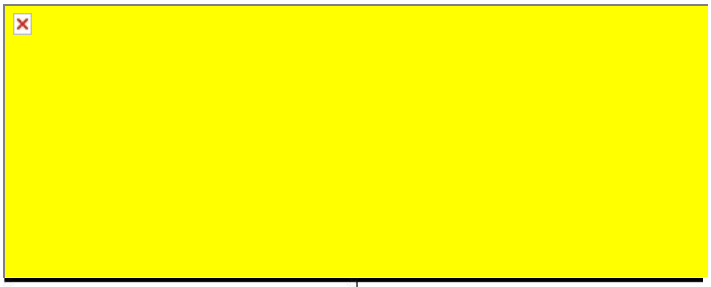
To define the number of revisions of documents that are stored and tracked in the knowledge base, enter a number in the Number of Revisions field.

12.3.7 Customizing the Description Page

In DevTrack, the Description page in the Knowledge view tracks general information about an knowledge items including the title, type, and keywords that are used to identify that knowledge item.

Description page data-entry controls are completely customizable. Each field and its respective values can be changed to better fit each project team's development environment.

By default, the Description page displays six data-entry controls: the Title control, the File, control, the Type control, the Keyword control, the Priority control, the Description control. Project administrators may make different customizations to each data-entry control displayed in the Description page.



Priority	• Customize the field label • Define control as visible or invisible • Define the dropdown list options
Description	• Customize the field label • Define the dropdown list options

Table 11-1: Description Page Control Customizations

Project administrators may customize the Description page in the Knowledge view."

To customize the Title control:

1Double-click the Title control in the Description page.

The Setup UI Field - Description Edit Box dialog box appears.

2Update the title of the control in the Current field.

3Click the OK button.

To customize the Keywords control:

1Double-click the Key Words control in the Description page.

The Setup UI Field - Description Edit Box dialog box appears.

2Update the title of the control in the Current field.

3Click the OK button.

To customize the File control:

1Double-click the File control in the Description page.

The Setup UI Field - Description Edit Box dialog box appears.

2 Update the title of the control in the Current field.

3 Click the OK button.

To customize the Type control:

1 Double-click the Title control in the Description page.

The Setup UI Field - Description Dropdown List dialog box appears.

2 Update the title of the control in the Current field.

3 Define the options displayed in the Type dropdown list control.

Project administrators may use the New button, Edit button, and Delete button to add, edit, or delete the options displayed in the Type dropdown list in the project

Project administrators may use the Import button to import options from structured text.

4 Click the OK button.

To customize the Priority control:

1 Double-click the Priority control in the Description page.

The Setup UI Field - Description Dropdown List dialog box appears.

2 Update the title of the control in the Current field.

3 Define the options displayed in the Priority dropdown list control.

Project administrators may use the New button, Edit button, and Delete button to add, edit, or delete the options displayed in the Priority dropdown list in the project

Project administrators may use the Import button to import options from structured text.

4 Click the OK button.

To customize the Description control:

1 Double-click the Title control in the Description page.

The Setup UI Field - Description Edit Box dialog box appears.

2 Update the title of the control in the Current field.

3 Click the OK button.

12.3.8 Customizing the History Page

The History page tracks the history of knowledge items. History page fields include:

- The Version column label
- The Name column label
- The Date column label
- The Action column label
- The Notes column label

Project administrators may use the tools in the*History system page*to customize the field labels for each control in the Issue Detail page in the Knowledge view of the DevTrack Windows client.

Each field label in the History page, including the page title, may be customized by double-clicking the label in the History page and updating the name in the dialog box.

12.3.9 Customizing the Resolution Page

Project administrators may use the tools in the Resolution page to customize the field labels for each control in the Resolution page in the Knowledge view of the DevTrack Windows client.

Project administrators may customize the labels for the page title, the Symptom field, and the Resolution field.

12.3.10 Backing up the Document Files

To ensure that files saved in the Documents and Attachments folders are not lost, project administrators should back up the DevTrack Document Server on a regular basis.

- All files in the Documents and Attachments folders are stored on the document server.
- HTML links and topics are stored in the DevTrack database.

The default directory for document storage on the document server is:

C:\Program Files\Techexcel\DTDoc\P#

The pound sign (#) represents the project ID.

12.4 Administering Issue-level Knowledge Management

Development organizations may manage and track information that is specific to development issues in many different issue property controls including the Issue Description control, the work Description control, the Issue History page, the Note control, and in issue note attachments.

All issue-specific information is directly embedded in, or attached to, the development issue used to track the issue.

- Issue descriptions define the content and context of development issues and are the primary vehicle for sharing issue-specific information among project team members.
- Work descriptions enable development organizations to share information about the work performed on a development issue.
- Issue histories enable development organizations to view all development issue property definitions and the entire history of the issue in a single page.
- Issue notes are issue-specific comments that are defined and tracked in the Notes page. Project members may add any number of notes to each development issue.
- Issue note attachments are documents, HTML links, and knowledge topics that are attached to issue notes. Project members may attach new documents or HTML links directly to each issue, or they may link the issue note to an existing issue managed in the Knowledge view. Only one attachment may be added to each issue note.
- Screen captures are graphic images that may be attached to issues. DevTrack Smart Screen Capture enables project members to take screen captures and immediately attach the JPEG file to an issue or issue note.

DevTrack project-level administrative tasks include:

- Understanding the Notes Page
- Enabling the Notes Page
- Customizing Note Page List Columns
- Understanding Memo Field Time Stamping
- Managing Note Types
- Managing Note Access Controls

12.4.1 Understanding the Notes Page

Issue notes and note attachments enable project teams to manage a repository of issue-specific information. Project members may add any number of notes to each issue. Project members may create and to add notes and file attachments to DevTrack issues in the Notes control.

Project administrators may enable and customize the Notes page in the Notes system page of the DevSuite Admin client. Project administrator-defined customizations determine how project members may manage notes in the DevTrack project.

Project administrators may define the following features:

- Timestamping of issue notes
- Definition of note types
- Note access controls

12.4.2 Enabling the Notes Page

To enable the Notes page in a DevTrack project, select the Show Notes page check box in the Notes system page of Main View GUI Design folder in the DevSuite Admin client.

12.4.3 Customizing Note Page List Columns

The Notes page displays high-level information about every note that is added to a DevTrack issue in the Notes list. The Notes list consists of multiple rows, which represent individual notes and three columns that represent note properties. By default these columns are titled*Note*,*Name*, and*Date*.

Although Note page controls may be renamed, TechExcel recommends that the data tracked in these columns are not changed.

12.4.4 Understanding Memo Field Time Stamping

A time stamp is a signature that is concatenated to text entered in the data-entry field. DevTrack time stamps log the exact date and time of the entry, the name of the project member entering the data, and the action that accompanied the work description entry.

The Description control, Work Description control, and Note control support the time stamping of all data entered into the field. In addition, project administrators may define time stamped entries as indelible: once entered the entries cannot be edited or deleted. Subsequent entries may be made, but these entries are marked by a unique time stamp.

Time stamping is an optional feature that must be enabled by a project administrator in each DevTrack project.

12.4.5 Managing Note Types

Note types enable development organizations to define distinct categories of issue notes, which enable development organizations to better manage issue-level notes. Typical note types might include:*Comment*,*Resolution*,*Release Note*, or*Design*.

Note types enable development teams to readily identify, sort, and manage different types of notes in their projects. Project members may use the Filter controls in the Notes page to choose which notes are displayed in the DevTrack client.

In addition, all issue-level knowledge access controls are defined by note type. The ability of project members to view, create, edit, and delete notes may be restricted by note type. For more information on note access controls see “Managing Note Access Controls”.

Project administrators may add, edit, and delete note types in the Notes page of the Main View GUI Design folder in the DevSuite Admin client.

To add a note type:

1. Click the Add button in the Notes page of the Main View GUI Design folder.

   The Add a New Note Type dialog box appears.

2 Enter a unique title in the Name control.

3 Click the OK button.

To edit a note type:

1 Select a note type in the Note Types list of the Notes page of the Main View GUI Design folder.

2 Click the Edit button.

The Edit Note Type dialog box appears.

3 Enter a unique title in the Name control.

4 Click the OK button.

To delete a note type:

1 Select a note type in the Note Types list of the Notes page of the Main View GUI Design folder.

2 Click the Delete button.

A confirmation dialog box appears.

3 Click the Yes button.

12.4.6 Managing Note Access Controls

Note access controls define which project members may view, add, edit, and delete different types of notes in DevTrack projects. Access controls enable project administrators to permit or deny a project member to perform a certain task based on their role in the project.

All note access controls are defined on a*note type-by-note type basis*. Depending on their role in a project, a user may be able to view, add, edit and delete on type of note and only view another note type.

For each note type an account type may be granted four different privileges:

Table 11-2: Document View Privileges

Access controls are usually determined by the*account type*that is assigned to a user, but note access controls may be granted to project members based their relationship to a specific issue." Project members may be granted the ability to view, add, edit, or delete issue notes if they initially submitted or currently own the associated issue.

To define note access controls:

1 Select a note type in the Note Types list of the Notes page of the Main View GUI Design folder.

The Note Types list displays all of the note types that may be managed in the project.

2 Grant access controls to each account type or system user variable listed in the Access Privileges tab.

The Account Type column of the Access Privileges tab displays every administrator-defined account type as well as two system user variables: the {Parent Issue Owner} variable and the {Parent Issue Submitter} variables.

Four privileges may be granted for each note type:

- To grant the View privilege to the account type, select the View check box.
- To grant the Edit privilege to the account type, select the Edit check box.
- To grant the Create privilege to the account type, select the Create check box.
- To grant the Delete privilege to the account type, select the Delete check box.

DevTrack (Enterprise)

Chapter 1 DevTrack Concepts

1 Chapter 1- DevTrack Concepts

This chapter provides a broad overview of DevSuite features and modules describing how these tools may be used to enhance application life cycle management processes.

In this chapter:
- Understanding TechExcel DevTrack and DevSuite

1.1 Understanding TechExcel DevTrack and DevSuite

TechExcel DevSuite is a family of integrated application lifecycle management (ALM) tools that place knowledge management —from ideas, to formal specifications, to competitive information to issue resolution and customer insight—at the core of any product development initiative.

The DevSuite knowledge-centric strategy enables improve communication, keep up-to-date on changes, and reduce the development cycles so that the business may deliver the right products for the right markets in the shortest possible time.

Figure 1-1: DevSuite Knowledge-Centric Development

DevSuite places knowledge management at the core of all development processes. TechExcel KnowledgeWise provides for the easy and efficient collection and organization of informal ideas, gathered from a wide variety of sources, that area shared across multiple DevSpec, DevTrack, and DevTest projects.

KnowledgeWise projects provide controlled access to documents, improve communication and coordination between distributed development teams, and facilitate the management and sharing of information between development teams and project stakeholders.

TechExcel DevSuite leverages intellectual assets with KnowledgeWise, communicating a clear product vision and tactical execution strategy by linking ideas and customer feedback, specifications, requirements, designs, prototypes, and other documents to specific areas of work.

1.1.1 Understanding DevSuite ALM Solutions

TechExcel DevSuite features five ALM solutions that operate in an n-tier architecture: TechExcel KnowledgeWise, TechExcel DevTrack, TechExcel DevPlan, TechExcel DevSpec, and TechExcel DevTest.

All DevSuite products share the same core architecture—the DevSuite Database Server, DevSuite Application Server, DevSuite Document Server, and DevSuite Web Services—and are fully integrated enabling every branch of the development organization to communicate and work together.

KnowledgeWise	TechExcel DevSuite leverages intellectual assets with KnowledgeWise, communicating a clear product vision and tactical execution strategy by linking ideas and customer feedback,specifications, requirements, designs, prototypes, and other documents to specific areas of work during a development project.Documents are shared with all resources involved in the executionof the project allowing for an uncompromised vision to direct the path of any development project.
DevPlan	TechExcel DevPlan manages the transformation of concepts into formal strategic plans. DevPlan offers an intuitive planninghierarchy to formalize scope an optimize resource usage,team-based planning and calendaring capabilities. These features enable complete control over all product development projects from design planning to implementation and enables increased team efficiency and collaboration.
DevSpec	TechExcel DevSpec is an integrated requirements management solution that is specifically designed to provide visibility,traceability and validation of your product or project requirements.DevSpec provides a frame work to create new requirements, specifications and features that can be linked to development and testing implementation projects.
DevTrack	TechExcel DevTrack enables development teams to manage every aspect of the development process including issue management, team management, and communications management.
DevTest	TechExcel DevTest is a test management solution that enables test organizations to manage every stage of testing life cycle—from test case design, to test execution, to test analysis. DevTest provides testing groups with the tools they need work more effectively and efficiently, hold down costs, and to deliver higher quality products.

While each element is valuable even when functioning independently, the DevSuite architecture is founded upon the assumption of future scaling and is capable of advanced inter-module integration. Whether used as separate components or as an incorporated set of applications, the DevSuite solutions are ideal for companies at any stage in their product development.

1.1.2 Understanding TechExcel Project Hierarchy

At the core of every DevSuite implementation—a DevSuite site— is one or more KnowledgeWise knowledge management projects that organize, manage, and track the knowledge collected in multiple DevSpec, DevTrack, and DevTest projects.

All projects in a DevSuite site are organized into a hierarchical structure that defines the relationships between the projects in a DevSuite site. Every DevSpec or DevTrack project is the child of a parent KnowledgeWise knowledge management project and relies on that project to manage project intelligence.

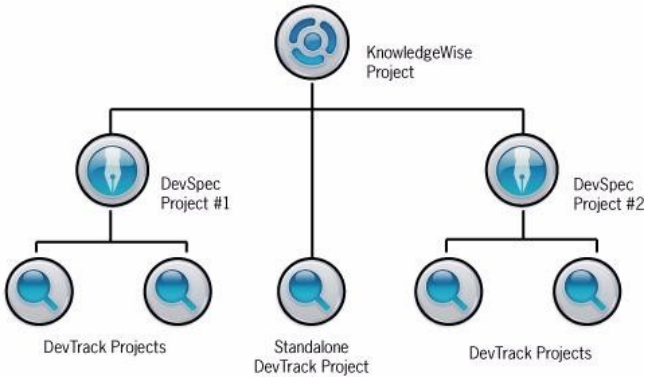


Figure 1-2: DevSuite Project Hierarchy

KnowledgeWise enables development organizations to build a knowledge base of shared knowledge in many different media. All knowledge items may be linked with different areas of work so that internal teams and external customers can search the knowledge base for self-service content based on account type-based privileges and access controls.

KnowledgeWise projects may be accessed by three methods: (1) the KnowledgeWise Web client, (2) the knowledge view of a child DevSpec project; (2) the knowledge view of a child DevTrack project.

1.2 Understanding DevSuite Administration

In DevSuite, administrative tasks are managed at two levels: system administration and project administration.

Site Administration: DevSuite system administration is the process of configuring, administering, and maintaining a DevSuite site. System-level administration activities include maintaining the implementation, creating and defining new projects, managing user accounts, managing licenses, and configuring the DevSuite Server.

Project Administration: DevSuite project administration is the process of defining project-level settings and business objects, customizing project GUIs, configuring workflow rules, managing project team representation and access controls, and integrating add-on modules.

All DevSuite site and project administrative tasks are performed in the DevSuite Admin client. The DevSuite Admin client provides DevTrack, DevPlan, DevSpec, and KnowledgeWise system and project administrators with a single interface for defining, configuring, and administering TechExcel sites and projects.

All DevSuite site and system-level administrative tasks are performed in a System Settings project. Every DevSuite site consists of a single System Settings project. Settings defined in a system settings project apply to every work project in that site.

All project-level administrative tasks are defined on a project-by-project basis and are only applicable to a single work project. In DevSuite 1, project administrators may define four types of projects: KnowledgeWise projects, DevSpec projects, DevTrack projects, and DevTime projects.

Note: DevTest projects may be defined, configured, and administered in the DevTest Admin client. All other DevSuite projects are managed in the DevSuite Admin client.

1.2.1 Understanding DevSuite Site (System-Level) Administration

DevSuite system administration is the process of configuring, administering, and maintaining a DevSuite site. A site consists of a system settings project and one or more work projects. Every work project in a DevSuite site shares a common database, knowledge base, and other system-level configurations that are defined in the DevSuite Admin client.

System-level configurations define the DevSuite site and the integration of every project within that site.

In DevSuite, system administration is the process of defining system, user, and project settings that define work in a DevSuite site. A DevSuite site is an implementation of a DevSuite server and may include multiple DevTrack, DevSpec, DevPlan, and DevSpec projects. System-level settings are settings that apply to every project in the DevSuite site.

DevSuite system administration tasks fall into seven general areas:

System Administration: System-level settings administrative tasks include the configuration of the DevSuite Server.

User Administration: System-level user administrative tasks include the definition and management of user accounts and project administrators. Optional system-level administrative tasks include multiple site management, division, management, and LDAP directory server integration and administration.

Multiple Site Administration: Multiple site administration includes the definition and management of a site family and management of user licenses for all member sites.

License Management: System-level license administrative tasks include license management and allocation.

Web Administration: Web administration includes the definition and configuration of DevTrack Web Server settings for every DevTrack project in the site.

E-mail Integration:E-mail integration and administration is the foundation of DevTrack, DevPlan, and DevSpec e-mail notification and escalation.

Branch Development: Release management administration defines the product structures that are shared across DevTrack, DevSpec, and DevPlan project providing a common framework for organizing and managing the development of products, versions, and builds.

Time And Cost Tracking Administration: Time and cost tracking administration defines the project management tools that enable a business to measure and track project resources.

System-level user administrative tasks include license management and allocation, user account creation and definition, and the management of project administrators. Optional system-level administrative tasks include multiple site management, division management, and LDAP directory server integration and administration.

1.2.2 Understanding Project Administration

DevSuite project administration is the process of defining project-level settings and business objects, customizing project GUIs, configuring workflow rules, managing project team representation and access controls, and integrating add-on modules.

Project administrators define what is tracked in the project, how it is tracked, and who may track the data. All project-level configurations are project-specific and apply to a single KnowledgeWise, DevSpec, DevTrack, or DevTime project.

Project administration tasks may be broken into five inter-related areas:

- Project Definition and Configuration
- Team Representation Administration
- Work Item Definition and GUI Customization
- Workflow Administration
- Add-on Module Integration

1.3 Understanding DevSuite Sample Projects and Templates

DevSuite builds on TechExcel's commitment to the rapid deployment of its ALM by providing development organizations with the tools they need to quickly configure and implement projects that enforce good development practices and drive the quality of releases.

To facilitate the configuration of a fully integrated site and projects that support your development processes, DevSuite provides developers with tools for evaluating and testing DevSuite features.

Sample Projects: A sample project is a project that provides development organizations with a sandbox for configuring and evaluating DevSuite features and integrations in a risk free environment.

Project Templates: A project template is a blueprint for creating a project of a specific type.

Solution Templates: A solution template is a preconfigured project that facilitates the definition of DevSpec projects that implement five proven development methodologies: CMMI, Scrum, SpecDD, TDD, and Iterative Development.

TechExcel strongly recommends that development organizations deploy and test DevSuite in a test environment prior to your DevSuite implementation. A test environment enables development organizations to test project configurations and to train users in new processes prior to the “go live” date in a production environment.

1.3.1 Understanding Sample Projects

In DevSuite, a sample project is a project that provides the development organization with a sandbox for configuring and evaluating features and integrations in a risk free environment.

DevSuite features ten predefined projects that provide development organizations with a sandbox to experiment with KnowledgeWise, DevSpec, and DevTrack tools and features. All sample projects include sample data, sample project team members, and fully configured workflow rules. Project administrators may freely configure sample projects to represent their development processes.

- Project configurations and settings tried and tested in sample projects may be imported into newly created “live projects”.
- Sample projects provide development organizations with a tool for training new users or to introduce existing users to new features or changes in businesses processes.

Using controls in the DevSuite Admin client, administrators may create additional sample KnowledgeWise, DevSpec, and DevTrack projects.

1.3.2 Understanding Project Templates

In DevSuite, a project template is a blueprint for creating a project of a specific type— KnowledgeWise knowledge management projects, DevSpec requirements management projects, or DevTrack development projects.

Each project template defines a complete set of project-level settings including the definition of all project business objects, workflow rules, team representation, and project integrations.

Project templates may be used to quickly configure new projects whenever they are created in a DevSuite site.

1.3.3 Understanding Solution Templates

TechExcel DevSpec, an integrated requirements management solution, features a complete library of preconfigured solution templates that facilitate the implementation of today's most popular and proven development methodologies.

Each solution template defines DevSuite project settings based on software development best practices so that development organizations may get up and running quickly. Once these business rules are in place, development teams may subsequently customize the project settings to suit their individual needs using TechExcel's award-winning project customization tools.

Using DevSuite, development methodologies longer merely serve as guidelines, but define and direct the daily activities of every team member. TechExcel provides complete documentation for implementing each methodology that describes how DevSuite workflow and automation may be used to implement and enforce best practices.

Chapter 2 Iterative Development Administration

2 Chapter 2 Iterative Development Administration

In this chapter:

- Understanding DevSuite Iterative Development
- Administering Project-Level Iterative Development
- Administering Development Backlogs

2.1 Understanding DevSuite Iterative Development

TechExcel DevSuite provides complete documentation for implementing each methodology that describes how DevSuite workflow and automation may be used to implement and enforce best practices.

Each solution template defines DevSuite project settings based on software development best practices so that development organizations may get up and running quickly. Once these business rules are in place, development teams may subsequently customize the project settings to suit their individual needs using TechExcel's award-winning project customization tools.

Using DevSuite, development methodologies longer merely serve as guidelines, but define and direct the daily activities of every team member.

2.2 Administering Project-Level Iterative Development

Enabling Iterative Development

Iterative development is an optional DevSuite feature that must be enabled on a project-by-project basis.

To enable iterative development:

1 Select Project Settings > Iterative Development.

2 Select the Enable Iterative Development check box.

2.2.1 Specifying Iterative Development Terminology

Different terms are used to refer to iterative processes in different development methodologies. In DevSuite, project administrators may customize the terms used to refer to key

development terms to suite their individual needs.

DevSuite supports five development methodologies.

Methodology	Iteration Term	Iteration Group Term
CMMI	Iteration	Iteration Group
Iterative Development		
Scrum Development	Sprint	Milestone
Specification Driven Development		
Test Driven Development		

Table 2-1: Iterative Terms

To specify iterative development terminology:

1 Select Project Settings > Iterative Development.

2 Input text in the Refer Iteration As text box.

3 Input text in the Refer Iteration Group text box.

2.2.2 Defining Iteration Durations

The default duration of an iteration may be predefined to any period between a week and three months:

One Week Two Weeks Three Weeks

Four Weeks Semi Month One Month

Two Months Three Months

Alternately, project administrators may choose to enable project team members to freely define the duration of each iteration by selecting the {Freely Defined} option.

To define iteration duration settings:

1 Select Project Settings > Iterative Development.

2 Select the Enable Predefined Iteration Duration check box.

3 Select an option from the Iteration Duration dropdown list.

2.3 Administering Development Backlogs

In DevSuite, a backlog is a prioritized list of development tasks (work items) that must be completed to meet the goals of a particular development cycle. The tasks that comprise a backlog may include

DevSuite supports backlog management for five distinct development cycles: projects, iterations, specifications, and defects.

Project Backlog:A project backlog defines the requirements of an entire project including customer and team-defined features as a prioritized list of features. The tasks identified in the project backlog are implemented in iterations according to their priority.

Iteration Backlog: An iteration backlog identifies the development task to be implemented within a specific development iteration (usually 30 days) as a prioritized list of features.

Specification Backlog: Specification backlog support is an optional DevSuite feature that must be enabled in each DevTrack or DevSpec project.

Defect Backlog: Defect backlog support is an optional DevSuite feature that must be enabled in each DevTrack or DevSpec project.

Backlog administration is the process of enabling support for backlog management, defining the terms used for backlog management in the project, and defining and managing subproject templates.

2.3.1 Specifying Backlog Terminology

Different terms are used to refer to iterative processes in different development methodologies. In DevSuite, project administrators may customize the terms used to refer to key development terms to suite their individual needs.

Specification Backlog

Defect Backlog

Project Backlog

Iteration Backlog

Chapter 3 Enterprise Edition Features

3 Chapter 3 Enterprise Edition Features

In this chapter:

- Administering the DevTrack Search Engine
- Administering the DevTrack Search Engine
- Administering User Identity Authentication

3.1 Understanding DevTrack Enterprise Edition

*DevTrack Enterprise Edition*is an optional upgrade that includes all of the features in the Standard Edition of DevTrack and adds a number of enterprise features including event management, subproject management, and the Beta Customer Portal.

DevTrack Enterprise Edition features include:

- Multi Sites
- Subproject management
- DevTrack Search Engine
- Event management
- Event Notification
- Event Escalation
- Beta Customer Web Portal
- Multiple workflow

3.2 Administering the DevTrack Search Engine

The DevTrack search engine enables project members to quickly search the notes, memo fields, and descriptions of DevTrack issues and events.

Unlike the standard search engine built into each DevTrack project, the DevTrack Search Engine enables project members to search multiple notes fields (the Description field, issue history, link comments, and so on). Project administrators may also periodically index these fields to enhance project searches.

Project administrators may install multiple DevTrack Search Engines in their DevTrack site. Each DevTrack project may have its own Search Engine or multiple projects may share the same DevTrack Search Engine.

Once the Search Engine is installed and configured, project members may use keywords to search the following DevTrack fields:

- Description
- Close Note
- History
- Issue Notes
- Link Comments
- Event Description
- All custom-defined memo fields

The DevTrack Search Engine is a feature of the DevTrack Enterprise Edition.*DevTrack Enterprise Edition*is an optional upgrade that includes all of the features in the Standard Edition of DevTrack and adds a number of enterprise features including event management, subproject management, and the Beta Customer Portal.

DevTrack Search Engine management tasks include:

- Installing the DevTrack Search Engine
- Defining Search Engine Settings
- Indexing the DevTrack Search Engine

3.2.1 Installing the DevTrack Search Engine

Project administrators may install the DevTrack Search Engine to enhance the searching capabilities of project members in DevTrack projects.

The DevTrack Search Engine runs as a service and may managed using the Services manager in Administrative Tools of a Windows operating system.

The DevTrack search service must be installed on the machine*before*a project may enable DevTrack Search Engine support and index the searchable fields in the project.

To install the DevTrack Search Engine:

1Run the SearchSetup executable.

The DevTrack Search Server Installation wizard installs the DevTrack Search Engine.

2During the installation the administrator must:

- Accept the license agreement.
- Ensure that the DevTrack Search Engine can connect to the DevTrack Application Server.

3 The TechExcel DevTrack Search Engine Configuration dialog box appears.

4 Define the name of the current DevTrack Search Engine in the Referred as Service Name field.

Each DevTrack Search Engine installed in a DevTrack site may have its own name. Project administrators may choose which search engine is used for each DevTrack project. Multiple projects may use the same search engine. Naming search engines enable administrators to identify search engines in the system.

5 Define the location of the index directory.

6 Click the OK button.

7 Click the Finish button.

3.2.2 Defining Search Engine Settings

Project administrators may enable DevTrack Search Engine support, select an appropriate DevTrack Search Engine for a project, and index the memo fields in that project using controls in the DevTrack Search Engine page.

To define DevTrack Search Engine settings:

1 Select the Search Engine page in the Enterprise Edition Features folder of the DevSuite Admin Tree window.

2 Select the Enable Memo Fields Search check box.

3 Click the Select a Search Engine button. The Select a Search Engine dialog box appears.

4 Select a search engine option in the Search Engine list.

Project administrators may install multiple DevTrack Search Engines in their DevTrack site. Each DevTrack project may have its own Search Engine or multiple projects may share the same DevTrack Search Engine.

5 Click the Select button.

The Select a Search Engine dialog box closes.

3.2.3 Indexing the DevTrack Search Engine

Project administrators may enable DevTrack Search Engine support, select an appropriate DevTrack Search Engine for a project, and index the memo fields in that project using controls in the DevTrack Search Engine page.

The search engine performs search indexing on memo fields. The first time an administrator indexes a large database, the process may take hours. Subsequent indexing of project memo fields is only performed on newly added records and the execution time is significantly reduced.

To index project memo fields:

1 Select the Search Engine page in the Enterprise Edition Features folder of the DevSuite Admin Tree window.

2 Click the Index Now button.

The Initialization of Search Index dialog box appears.

3 Click the Start button.

The Initialization of Search Index dialog box closes.

The Current Status control displays the current status of the indexing process. When indexing is complete the Current Status control displays the tex*t Indexing process finished*.

3.3 Administering User Identity Authentication

User authentication enables development teams to ensure that key transition and state changes to issues and events are approved in DevTrack projects.

Development teams may wish to restrict certain issue or event changes to certain project members. DevTrack account types and workflow rules are designed to ensure that only project members with the appropriate privileges may make changes in projects.

User identity authentication enables development teams to add an addition level of security to their projects by requiring that project members verify their identity before they may update issues or events in specific workflow states.

- Identity authentication on issue state or transition changes: Issue authentication rules may require that project members enter their password before an issue can be updated based on transitions or state changes identified by the administrator.
- Identity authentication on event state or transition changes: Event authentications rules may require that project members enter their password before an event can be updated based on event templates and event states identified by the administrator.

All authentication rules are administrator defined. Project administrators may use the User Identity Management page in the DevSuite Admin client to enable user authentication, identify which issue and event state or transition changes require user authentication, and define appropriate messages to alert users that they must verify their identity.

The User Authentication Management page is divided into three main areas:

- The User Identity Description area enables project administrators to define the message used to prompt users to enter their password."
- The Applicable Issue States area enables project administrators to define authentication rules based on changes to issue states or transitions. Issues are not updated unless a valid password is entered.
- The Applicable Event area enables project administrators to define authentication rules based on event states changes for each event template. Event templates and child event states are organized as a tree menu in the Applicable Events area. Events are not updated unless a valid password is entered.

Once enabled and defined, user authentication rules ensure that no changes may be made unless the user confirms their identity by entering their password.

Project members are prompted to enter their password whenever they attempt to update an issue or event covered by an authentication rule.

All identity authentications are recorded and tracked in the History page in the DevTrack issue detail panel." All identity authentications are tracked in the DevTrack projects, enabling administrators to identity unauthorized or fraudulent changes.

User identity authentication is a DevTrack Enterprise Edition feature.

Chapter 4 Issue Types and Multiple Workflow Admin

4 Chapter 4 Issue Types and Multiple Workflow Administration

In this chapter:

- Administering Multiple Workflow and Workflow Assignment
- Administering Issue Types

4.1 Understanding Multiple Workflow and Issue Types

TechExcel DevTrack is a tool for managing and tracking development tasks such as bug fixes, enhancements, or features in workflow. All development tasks are represented and tracked in the project as*development issues*.

A development issue is a collection of data that represents a particular task or set of tasks that must be processed in the course of a development project. Every issue is defined by a unique issue ID, description, workflow state, owner, work description, and other dynamic properties.

Issue typing and multiple workflow enable development organizations to define many different workflows that are specifically designed to suit development issues of a particular type. In DevTrack, different workflows and workflow rules may be defined for different issue types.

Support for multiple workflow and issue typing is a DevTrack Enterprise Edition that must be enabled in each DevTrack project. Administrative tasks include the definition of issue types, the definition of subworkflows, and the assignment of subworkflows to the administrator-defined issue types.

4.2 Administering Multiple Workflow and Workflow Assignment

In DevTrack, a single development project may support multiple workflows (*subworkflows*) to better manage different areas of development within the project.

A workflow is a business process in which all activities are organized into repeatable patterns of work. Using multiple workflow, a development organization may define distinct workflows and access controls to manage different*issue types*(for example, product defects, enhancements, and features).

Multiple project workflow enables administrators to define subworkflows within the larger master workflow. Each subworkflow may be managed by a set of autonomous workflow rules for the workflow states and transitions in the subworkflow.

Multiple workflow administration is the process of enabling support for multiple workflow and issue typing, selecting multiple workflow settings, defining workflow assignment settings, and workflow initialization options.

4.2.1 Enabling Multiple Workflow

DevTrack also features multiple project workflow (subproject workflow) that is only available in projects using transition-based workflow. Multiple project workflow enables administrators to define workflow rules for individual subprojects within the larger master workflow.

Administrators may optionally define a unique set of workflow rules for the workflow states and transitions within each subworkflow. Using multiple workflow options, each subproject may have its own workflow and workflow rules within the larger project workflow (Master workflow).

Subproject workflow enables administrators to define multiple sets of workflow rules for workflow states and transitions, and tie those workflow rules to a subproject. An administrator may define different sets of subproject workflow rules to manage different subsets of issues. For example, both product defect issues and new feature issues may share the same workflow path, but be governed by entirely different workflow rules.

For more information see “Administering Multiple Workflow and Workflow Assignment”.

4.2.2 Defining Multiple Workflow Settings

Multiple workflow and workflow assignment are DevSuite Enterprise Edition features that must be enabled and configured in each DevTrack project. These features are only available in projects using transition-based workflow.

DevTrack features a highly customizable workflow engine that enables the administrator to customize project workflow to support many different development methodologies.

Using controls in the Multiple Workflow Options dialog box, project administrators may enable support for multiple workflow and issue types and choose how subworkflows are defined and implemented in a project.

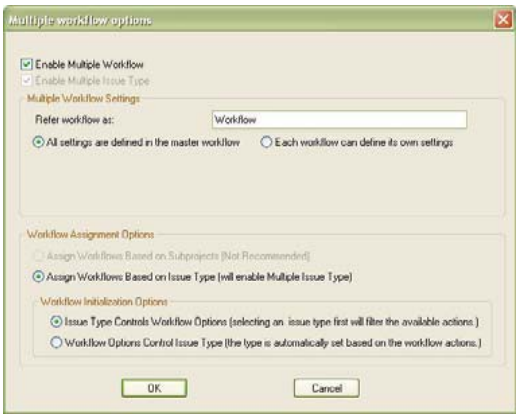


Figure 4-1: Multiple Workflow Options Window

In each project multiple workflow is defined by a multiple workflow model, workflow assignment model, and workflow initialization model.

Multiple Workflow Method: Multiple workflow method determines how subproject workflow is defined and managed in the DevSuite Admin client.

Workflow Assignment Method: Workflow assignment method determines how subworkflows are implemented in a project.

Workflow Initialization Method: Workflow initialization method determines how subworkflows are assigned to issue types in projects using the Assigned Workflows Based on Issue Type workflow assignment option.

To enable multiple workflow and choose subproject option:

1Select Workflow > Multiple Workflow in the tree panel.

2Click the Change button in the Multiple Workflow Options area of the Multiple Workflow page.

The Multiple Workflow Options window appears.

3Select the Enable Multiple Workflow check box.

If the Enable Multiple Workflow check box is selected, the New Workflow control is displayed in the tree panel. Administrators may use this control to create subworkflows in the project and, optionally, define workflow rules for the workflow states and transitions in that subworkflow. For more information see “Administering Multiple Workflow and Workflow Assignment”.

4 Optional: To define the term used to refer to project workflow, enter a term in the Refer to Workflow As text box.

By default, project workflow is referred to as “workflow” throughout the DevSuite Admin client and all client GUIs.

5 Choose a workflow assignment method:

Workflow assignment option determines how subworkflows are implemented in a project.

- To define subworkflows for each subproject, select the Assigned Workflows Based on Subproject option button.
- To define subworkflows for each issue type, select the Assigned Workflows Based on Issue Type option button.

6 Choose a workflow initialization option:

Workflow initialization option determines how subworkflows are assigned to issue types in projects using issue type workflow assignment.

- To automatically assign a subworkflow to a development issue based on its issue type, select the Issue Type Controls Workflow Options option button.
- To assign an issue type to a development issue based on subworkflow assignment, select the Workflow Options Controls Issue Types option button.

7 Click the OK button.

The Multiple Workflow Options window closes.

4.2.3 Defining Subworkflows

In DevTrack a subworkflow is a set of rules for managing a subset of issues (based on their issue type) that defines the issue lifecycle, state-dependent access controls and transition-dependent access controls.

A subworkflow is defined by one or more applicable transitions, a default submit action, and a default close action. Distinct access controls may be defined for every workflow state and transition in the subworkflow.

Using controls in the New Workflow page, project administrators may create, edit, and view subworkflows and define default submit and close actions.

Figure 4-2: Sub Workflow Page

The Applicable Transitions list displays every transition defined in the Master Workflow, as well as the two states linked by the transition. Administrators may choose which transitions and which states from the parent Master Workflow are included in the sub workflow. The source and target states of each transition are automatically added to the subworkflow.

To define a subworkflow:

1 Select Workflow > Master Workflow Design > [New Workflow] in the tree panel.

The Subworkflow window appears.

2 Define the title of the subworkflow in the Workflow Name text box.

3 Select one or more transitions in the Select Workflow Applicable Transitions list.

The Select Workflow Applicable Transitions list displays transitions defined in the master workflow. Selected transitions are designated as applicable to the subworkflow and may be customized.

- To select all transitions, click the Check All button.
- To deselect all transitions, click the Uncheck All button.

4 Select a transition in the Default Submit Action dropdown list.

The default submit action identifies the default initial transition in the subworkflow.

5 Select a transition in the Default Close Action dropdown list.

The default close action identifies the default final transition in the subworkflow.

6 Click the OK button.

The Subworkflow window closes.

4.2.4 Understanding Workflow Assignment Methods

DevTrack multiple workflow is highly configurable and may be customized to support the unique needs of any business. In each development project, multiple workflow is defined by its multiple workflow method, workflow assignment method, and workflow initialization method.

The workflow assignment method determines whether subworkflows are assigned to development issues based on their issue type or subproject.

The Assigned Workflows Based on Subproject method is no longer supported in DevTrack. Existing implementations that assign subworkflows to subprojects may continue to use this method, but new sites and projects cannot implementation this method.

4.2.5 Defining Issue Type Subworkflow Initialization

DevTrack multiple workflow is highly configurable and may be customized to support the unique needs of any business. In each development project, multiple workflow is defined by its multiple workflow method, workflow assignment method, and workflow initialization method.

The workflow initialization method determines how subworkflows are assigned to issue types.

To define the workflow initialization method:

1. Select Workflow > Master Workflow Design > [New Workflow] in the tree panel.

   The Subworkflow window appears.

2. Choose a workflow initialization option:

   Workflow initialization option determines how subworkflows are assigned to issue types in projects using the Assigned Workflows Based on Issue Type workflow assignment option.

   - To automatically assign a subworkflow to a development issue based on its issue type, select the Issue Type Controls Workflow Options option button.
   - To assign an issue type to a development issue based on subworkflow assignment, select the Workflow Options Controls Issue Types option button.

4.2.6 Granting Issue Type Assignment Privileges

To grant issue type assignment privileges:

1. Select Workflow > Workflow Assignment in the tree panel.

   The Workflow Assignment page appears.

2. Select one or more account types in the Who Can Assign Workflow list.

   A black check mark is displayed in the check box if the account type is enabled to change the issue type of a development issue.

4.3 Administering Issue Types

In DevTrack, project administrators may define multiple workflows within a single DevTrack project to enable development organizations to assign different workflow paths (subworkflows) to the issues (bugs, features, enhancements) managed in a project.

Issue typing offers development team maximum flexibility and power to customize project workflow to support and manage their business processes.

Using controls in the Issue Type page, project administrators may add, edit, delete, and define development issue types and QA test issue types.

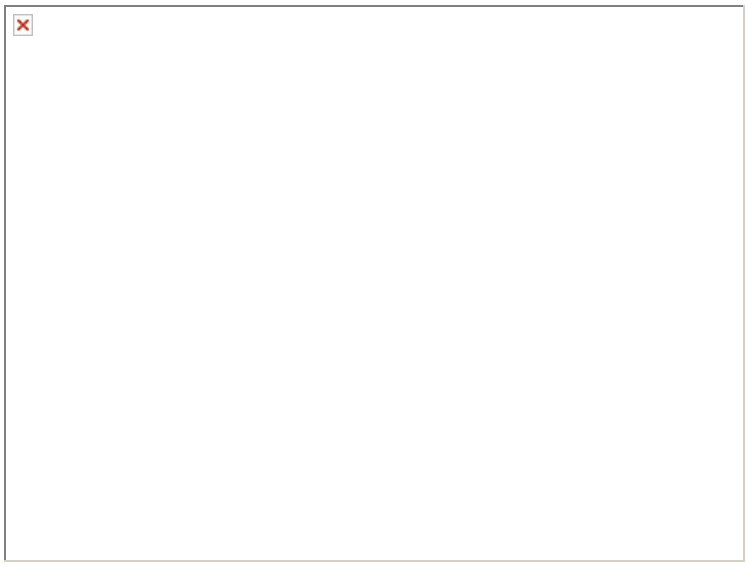


Figure 4-3: Define Issue Type Page

An issue type represents a group of issues that share similar properties and which need to be managed in a unique way." The issue type of an issue is typically linked to its type

definition (for example, bug, new feature, enhancement), but alternatively may be linked to its priority, component, platform, build, version, or any other criteria.

Project administrators may choose to link the issue type to one of many different issue properties including type, component, platform, priority, and version. Note however, that issue type linked properties cannot be the child issue property of another, parent issue property.

DevTrack supports two classes of issue types: development issues and branch issues.

Development Issue:A development issue is a tool for managing and tracking a development task in a project. Every development issue is i defined by a unique issue ID, description, workflow state, owner, work description, and other dynamic operties.

Branch Issue:A branch issue is a copy of development issue that is submitted to a release folder by means of a*branch action*. Every branch issue is linked to a parent development issue by a branch link. Branch issues enable developers to manage and track the implementation of development tasks across multiple builds of a product.

Both development issues and branch issues may also be defined by a unique subworkflow and workflow rules that determine how that issue is managed in the project. Development issues and branch issues inherit their workflow from their issue type.

Issue types are a DevTrack Enterprise Edition feature that enable project administrators to define different type of issues (issue types) and to define subworkflow rules on issue types.

4.3.1 Defining Development Issue Type Terms

To define the term that is used to refer to issue types in a project, enter the term in the Refer to Development Issue As edit box control in the Define Issue Type page of the Enterprise Edition folder.

4.3.2 Linking Issue Type Property to Issue Property Fields

In DevTrack, the issue type of a development issue may be linked to another system or custom issue field so that the issue type of the development issue is determined by the issue property definitions tracked in the DevTrack client.

Eight issue fields may be linked to the issue type:

Type {Description}

Priority {Description}

Component {Description}

Version {Description}

Platform {Description}

Fix Target {Current Status}

Custom Field 1 {}

Custom Field 2 {}

To define issue type linked to fields:

1Click the Change button in the Define Issue Type page of the Enterprise Edition folder.

The Issue Type Link a Field dialog box appears.

2Select an option from the Select a Field dropdown list.

The Select a Field dropdown list displays eight options: the Type, Priority, Component, Version, Platform, Fix Target, Custom Field 1, and Custom Field 2 fields.

3Click the Next button.

The [Field Name]: Link a Field window appears. The [Field Name]: Link a Field window displays the field values for the linked issue property field.

4Select a field value in the list and click the Change Category button.

The Issue Type Category dialog box appears.

5Select a class of issue type in the dropdown list.

6Click the OK button.

The Issue Type Category dialog box closes.

4.3.3 Adding Issue Types

To add an issue type:

1SelectEnterpriseEdition Features > Define Issue Type.

2Click the Add button.

The Add New Issue Type dialog box appears.

3Enter the name of the issue type in the New Issue Type Name control.

4Select the class of issue type to be defined.

- To define the issue type as a development issue, select the Development Issue radio button.
- To define the issue type as a QA test issue, select the QA Test Issue radio button.

5Click the OK button.

The Add New Issue Type dialog box closes and the issue type is displayed in the Issue Type list.

4.3.4 Defining Default Issue Types

To define a default issue type:

1Select Enterprise Edition Features > Define Issue Type.

2Select an issue type in the Issue Type list.

3Select a default issue type in the Default Issue Type list.

4.3.5 Defining Issue Types

To define an issue type:

1SelectEnterpriseEdition Features > Define Issue Type.

2Select an issue type in the Issue Type list.

3Select a workflow in the Assign Workflow dropdown list of the Properties tab.

The Assign Workflow dropdown list displays the {Master Workflow} and all subworkflows.

4 Optional: Define editable and read-only fields.

To define a issue property field as read-only if the issue is the child of another develop-ment issue, select a field in the Edit list and click the Right arrow.

Chapter 5 Development Event Administration

5 Chapter 5 Development Event Administration

Development events represent the “to do” items or “subtasks” that project members must complete to process DevTrack issues. Every development event is based on an administrator-defined event template.

In this chapter:

- Administering DevTrack Development Events
- Administering Regular Development Event Templates
- Administering Co-Owner Event Templates
- Administering Event Groups
- Defining Regular Development Event Attachment Rules
- Administering Event Workflow
- Administering Applicable Owners of Development Events
- Administering Event Management Privileges and Access Controls
- Administering Customer Access to Development Events
- Administering Standard Event Workflow
- Administering Advanced Event Workflow

5.1 Administering DevTrack Development Events

A development event represents a subtask of a development issue. Development events enable organizations to divide a development task into many different subtasks, assign each of those subtasks to a different project member, define separate start and due dates for each subtask, and manage and track each of those tasks independently in its own workflow.

Development events enable project members to manage issue subtasks within a specific issue workflow state. Development events may be manually created by project members or automatically created based on administrator-defined workflow rules.

All development events are based on administrator-defined development event templates. A development event template is a set of rules for managing events in project workflow. The life cycle of a development event and the rules that determine how that event is managed in workflow are defined by the event template that was used to create it.

Development event administration, therefore, is the largely a matter of creating development event templates, defining the rules that manage events based on that template, and determining when events based on that event template may be created in the project.

Development events and development event workflow are optional, DevTrack Enterprise Edition features.

5.2 Administering Event Groups

An event group is a group of event templates that share a common set of workflow rules and that are managed together in a DevTrack project.

DevTrack supports two different types of event groups in each DevTrack project.

- Regular event groups are event groups that may be used to represent and manage event templates belonging to the regular event type. Project administrators may create any number of regular event groups to manage regular development event templates.
- Co-owner event groups are event groups that represent and manage event templates belonging the co-owner event type. The co-owner event group represents both linked and unlinked single-selection co-owner event templates. Project administrators may create any number of co-owner event groups to manage regular development event templates.

In addition to the administrator-defined regular event groups and co-owner event groups, DevTrack also uses a single, system-defined {All Development Event} group to organize and manage multiple-selection co-owner event templates.

5.2.1 Creating Event Groups

Event groups are tools that enable project administrators to better manage development event workflow in DevTrack projects. Project administrators may create, edit, and delete regular event groups and co-owner event groups in the Event Group page.

All multiple variable co-owner event templates belong to the system-defined {All Development Events} event group.

To create event groups:

1. Click the New button in the Event Group page of the Event Group subfolder in the Events folder.

   The Add Event Group dialog box appears.

2. Define a unique, descriptive name for the event group.
3. Select the event type.

   Project administrators may create two types of event groups: regular event groups and co-owner event groups.

   - To create a regular event group, select the Regular Event Group radio button.
   - To create a co-owner event group, select the Co-Owner Event Group radio button.

4. Enter a brief description of the event group in the Note control.
5. Click the OK button.

5.2.2 Ordering Event Groups

Project administrators may order administrator-defined event group folders within the parent regular event group folder and co-owner group folders within the parent co-owner event group folder.

The Event Group tree list control organizes event groups and event templates into a hierarchical tree structure that has*two*root folders. the Regular Event Group folder and the Co-Owner Event Group folder.

To order event groups:

1. Select an event group in the Event Group tree list the Event Group subfolder in the Events folder.

   The Event Group tree list displays all administrator-defined event groups organized by event group type. Project administrators may create and edit two types of event groups:regular event groups and co-owner event groups.

2. Move the event group higher or lower in the event group type folder.

   - To move an event group higher, select the Up button.
   - To move an event group lower, select the Down button.

5.2.3 Ordering Event Templates within Event Groups

Project administrators may order administrator-defined event templates within an event group in the Event Group tree list control of the Event Group page.

To order event templates within event groups:

1. Select an event template in the Event Group tree list the Event Group subfolder in the Events folder.

   The Event Group tree list displays all administrator-defined event groups organized by event group type.

2. Move the event template higher or lower in the event group folder.

   - To move an event group higher, select the Up button.
   - To move an event group lower, select the Down button.

5.3 Administering Regular Development Event Templates

In DevTrack, an event template is a set of business rules that defines how a development subtask is managed in a project. Every event template is defined by its applicable owners, account type-based access controls, attachment rules, as well as standard and advanced workflow rules.

A regular event template is a template for creating and managing regular development events. A regular event instance inherits all of its properties from the event template used to create it. The owner of a regular event is limited to a list of applicable owners that are identified in the event template.

Project administrators may create and define event templates in the Event Template page of the Events subfolder in theEnterpriseEdition folder. For each event template created in the Event Template page, the project administrator must define appropriate properties, attachment rules, access controls, an event group, event workflow rules, event

notification rules, and event escalation rules

5.3.1 Defining Regular Event Templates

A regular event template is a template for creating and managing regular development events. Regular event templates define the business rules that are used to manage regular development events in DevTrack projects. Every development event is based on an event template and is managed by the rules defined in that template.

For each regular event template created in the Event Template page, the project administrator must also define appropriate properties, a set of applicable owners, attachment rules, access controls, an event group, event workflow rules, event notification rules, and event escalation rules.

To define regular event template properties:

1. Click the New button in the Event Template page of the Development Event subfolder.

   The Add a New Event Template dialog box appears.

2. Define a unique, descriptive name for the event template in the Name text box control.

   The title of the event template is inherited by every development event based on that template and is the default title for those event instances. For more information see “Defining Default Event Titles and Descriptions”.

3. Select the Regular Event option from the Event Type dropdown list.

   The Event Type dropdown list displays two options:

   - The Regular Event option enables the administrator to create a regular event template.
   - The Co-owner Event option enables the administrator to create a co-owner event template.

4. **Optional:** To assign the event template to an event group, select an event group from the Event Group dropdown list.

   The Event Group dropdown list displays all regular event groups and the {Ungrouped} system option.

   - To assign the event template to an event group, select a regular event group in the Event Group dropdown list. Each regular event template may be assigned to one regular event group.
   - To not assign the event template to an event group, select the {Ungrouped} option in the Event Group dropdown list. Ungrouped standard event templates are not subject to group advanced workflow rules.

   Event groups are an optional DevTrack Enterprise Edition feature that enable development organizations to define more powerful and flexible rules for managing development events. For more information about event groups and group advanced workflow rules see “Administering Event Groups”.

5. Define a brief description of the event in the Description field.

   Event template descriptions enable project manager to describe the purpose and functionality of an event template.

6. Click the OK button.

   The Add a New Event Template dialog box closes. The regular event template is displayed in the Regular Events folder of the Event Templates tree control.

5.3.2 Editing Event Templates

Project administrators may edit basic event template properties in the Properties tab of the Event Templates page in the Development Events folder.

Two key event template properties cannot be edited or updated.

- The Event Type property determines whether the event template may be used to create standard development events or co-owner development events.
- The Event Group property defines how events based the event template are managed using group advanced event workflow rules.

Both event template properties are mandatory and can only be defined when the project administrator initially creates the event template.

5.3.3 Defining Default Event Titles and Descriptions

Every development event is defined by a title and a description of the event. The title and description of development events are tracked in two data-entry controls: the Title control and the Description control.

Every development event that is based on that event template inherits the predefined title and description.

To define default event titles and descriptions:

1. Select an event template in the Event Templates tree list control of the Event Templates page.
2. Define the title of the event in the Title control.
3. Define the description of the event in the Description control.
4. Click the OK button.

5.3.4 Defining Event Description Timestamping Rules

Event description timestamping is an optional feature that may be enabled on a template-by-template basis in DevTrack projects. Every event instance inherits timestamping rules from the event template that was used to create it.

The Description control is data-entry control that enables project members to manage and track detailed information about a specific event. The text displayed in the Description control may be predefined by the project administrator in the event template.

Event description timestamps show the date and time of every entry or edit made to text in the Description control, as well as the name of the project member that made the change.

Project administrators may choose between three different event template description timestamping rules for each event template:

Do Not Support Time Stamp: The Do Not Support Time Stamp rule disables timestamping and enables project members to freely edit Description control text.

Support Time Stamp and Most Recent: The Support Time Stamp and Can Edit Owner's Most Recent Changes rule enables timestamping,Change but restricts edits to the previous entry in the Description control.

Support Time Stamp and Cannot: The Support Time Stamp and Can Not Change Once Entered rule enables timestamping and Change disables the editing of all previous Description control entire.

Project administrators may customize the title of the Description control, define default values, and enable timestamping whenever they edit an event template.

To define event template description fields:

1Select an event template in the Event Templates tree list control of the Event Templates page.

2Double-click the Description control in the Property tab of the Event Template page.

The Event Templates dialog box appears.

3Select a timestamping option in the Time Stamp Options area of the Event Templates dialog box.

The Time Stamp Options area of the Event Templates dialog box displays three options:

- To disable timestamping, select the Do Not Support Time Stamp radio button.
- To enable timestamping and the editing of previous Description control entries,select the Support Time Stamp and Can Edit Owner's Most Recent Changes radio button.
- To enable timestamping and to disable Description control editing, select the Support Time Stamp and Can Not Change Once Entered radio button.

4Click the OK button.

5.3.5 Defining Default Event Workflow States

The life cycle of development events is defined by two or more administrator-defined event workflow states. In the DevTrack client, project members may track the current workflow state of development events in the State dropdown list control. The default workflow state is the workflow state that automatically defines a development event as soon as it is created.

Project administrators may define default event workflow states for each event template. Every development event that is based on that event template inherits all administrator-defined event settings including the default event workflow state Project administrators may customize the title of data-entry control, define a default state, and define the options that are available in the State dropdown list in each event template.

To define default event workflow states:

1Select an event template in the Event Templates tree list control of the Event Templates page.

2Double-click the State control in the Properties tab. The Event Templates dialog box appears.

3Select an option from the Default Value dropdown list.

4Click the OK button. The Event Templates dialog box closes.

5.3.6 Defining Default Event Owners

In the DevTrack client, project members may track the current owner of development events in the Owner dropdown list control. The default event owner is the project member that is automatically assigned ownership of the event as soon as it is created.

Project administrators may define default event owners for each event template. Every development event that is based on that event template inherits all administrator-defined event settings including the default event owner from that template.

Project administrators may define default event owners by project member, group folder, or system user variable.

- The project members options represent each project member.
- The group folder options represent an administrator-defined group folder.
- The {Parent Issue Owner} system user variable defines the project member that currently owns a development issue as an applicable owners.
- The {Parent Issue Submitter} system user variable defines the project member that submitted a development issue as an applicable owners.

Default event owners cannot be defined by account type or team group.

To define default event owners:

1Select an event template in the Event Templates tree list control of the Event Templates page.

2Double-click the Owner control in the Properties tab. The Event Templates dialog box appears.

3Select an option from the Default Owner dropdown list.

The Default Owner dropdown list display all project members and administrator-defined group folders, as well as the {Parent Issue Owner} and {Parent Issue Submitter} system user variables.

4 Click the OK button.

The Event Templates dialog box closes.

5.3.7 Defining Default Event Start and Due Dates

In the DevTrack client, project members may track the start date and due date of development events in the Start Date and Due Date controls. Project administrators may define rules for predefining a default start date and default due date in development events.

- The default start date the date that is displayed automatically in the Start Date control as soon as a development event is created.
- The default due date the date that is displayed automatically in the Due Date control as soon as a development event is created.

Project administrators may define default start and due dates in each event template. Every development event that is based on that event template inherits all administrator-defined event settings including the default start and due dates.

To define default start and due dates:

1 Select an event template in the Event Templates tree list control of the Event Templates page.

2 Double-click the Start Date control in the Properties tab. The Event Templates dialog box appears.

3 Select a default start date option.

- To define no default start date, select the Empty radio button.
- To define the current date and time as the default start time, select the Current Date/Time radio button.

4 **Optional:** To define an offset time for the default start date, enter the number of days and hours in the Offset controls.

5 Click the OK button.

The Event Templates dialog box closes.

6 Double-click the Due Date control in the Properties tab.

The Event Templates dialog box appears.

7 Select a default due date option.

- To define no default start time, select the Empty radio button.
- To define the current date and time as the default start time, select the Current Date/Time radio button.

8 **Optional:** To define an offset time for the default start date, enter the number of days and hours in the Offset controls.

9 Click the OK button.

The Event Templates dialog box closes.

5.3.8 Defining Regular Development Event Attachment Rules

During the course of project workflow, project members may add attachments (documents, URLs, knowledge topics) to events whenever they create or update an event in the DevTrack client.

Event attachment rules determine whether attachments are allowed, optional, or mandatory, which knowledge items may be attached to an event, and how the attached knowledge items are named and described.

Every event template is defined by a set of attachment rules that determine whether documents may or must be linked to development events. Development events inherit event attachment rules from the event template used to create them.

Project administrators may define event attachment rules for each event template in the Attachment tab.

To define development event template attachment rules:

1 Select an event template in the Event Templates list of the Event Template page in the Development Events subfolder.

The Event Templates list displays every development event template defined in the DevTrack project. Development event attachment rules are defined on a template-by-template basis.

2 Select an attachment option in the Attachment tab.

- To bar project members from attaching knowledge items to development events based on the selected event template, select the None radio button.
- To enable project members to optionally attach knowledge items to development events based on the selected event template, select the Optional radio button.
- To mandate that project members attach knowledge items to development events based on the selected event template, select the Mandatory radio button.

3 **Optional:** To enable project members to attach specific types of knowledge items to events based on a development event template, select one or more attachment type options in the Attachment Type field.

- The New Document option enables project members to attach new files to development events based on the event template.
- The New HTML Link option enables project members to attach new URLs to development events based on the event template.

- The Existing Knowledge Document option enables project members to attach files saved in the project knowledge base development events based on the event template.
- The Existing HTML Link option enables project members to attach URLs saved in the project knowledge base development events based on the event template.
- The Existing Knowledge Topic option enables project members to attach knowledge topics saved in the project knowledge base development events based on the event template.

Project administrators may enable one or more types of knowledge items to be attached to development events only if they have selected the Optional or Mandatory attachment option.

4 Select a title option in the Title Option area. In the DevTrack client, the Edit Event Template page enables project members to define the title and description of an event. Project administrators may define the default options displayed in the Title field and the Description field using controls in the Attachment tab.

Title Option settings represent the default title for an event. With the appropriate access users can edit the title of an event at any time.

- The Same as Event Template option makes the title of the parent event template the default title of every event based on that event template. If the title of the parent event template is x, the default title of the event will also be x.
- The Copy from Linked Knowledge Item option makes the title of a knowledge item the title of the event to which it is linked. If the title of the attached knowledge item is y, the default title of the event will also be y.
- The Append from Linked Knowledge Item option appends the title of a linked knowledge item to the title of the parent event template to create the default title of an event. If the title of the parent event template is New Event and the title of the attached knowledge item is Unix Usability Testing, the default title of the event will be x, y.

5 Select a description option in the Description area.

In the DevTrack client, the Edit Event Template page enables project members to define the title and description of an event. Project administrators may define the default options displayed in the Title field and the Description field using controls in the Attachment tab.

Description Option settings represent the default title for an event. With the appropriate access users can edit the description of an event at any time.

- The Same as Template Description option makes the description of the parent event template the default description of every event based on that event template. If the description of the parent event template is x, the default title of the event will also be x.
- The Copy from Linked Knowledge Item option makes the body of a knowledge item the description of the event to which it is linked. If the description of the attached knowledge item is y, the default description of the event will also be y.
- The Append from Linked Knowledge Item option appends the title of a linked knowledge item to the title of the parent event template to create the default title of an event. If the title of the parent event template is x and the title of the attached knowledge item is y, the default title of the event will be x, y.

- The project members options represent each project member.
- The group folder options represent an administrator-defined group folder.
- The {Parent Issue Owner} system user variable defines the project member that currently owns a development issue as an applicable owners.
- The {Parent Issue Submitter} system user variable defines the project member that submitted a development issue as an applicable owners.

Default event owners cannot be defined by account type or team group.

To define default event owners:

1 Select an event template in the Event Templates tree list control of the Event Templates page.

2 Double-click the Owner control in the Properties tab. The Event Templates dialog box appears.

3 Select an option from the Default Owner dropdown list.

The Default Owner dropdown list display all project members and administrator-defined group folders, as well as the {Parent Issue Owner} and {Parent Issue Submitter} system user variables.

4 Click the OK button.

The Event Templates dialog box closes.

5.4 Administering Co-Owner Event Templates

A co-owner event template is a template that enables development teams to create multiple events simultaneously in a project. Each co-owner event instance inherits its properties from the event template used to create it. The owner of each co-owner event instance is determined by the event variable that is built into the co-owner event template.

An event template is a set of default event properties, attachment rules, and access controls that may be used to create development events in DevTrack projects. Every event is an instance of an administrator-defined event template and inherits its properties from the template used to create it.

A co-owner event template is a template that enables development teams to create multiple events simultaneously in a project. Each co-owner event instance inherits its properties from the event template used to create it. The owner of each co-owner event instance is determined by the event variable that is built into the co-owner event template.

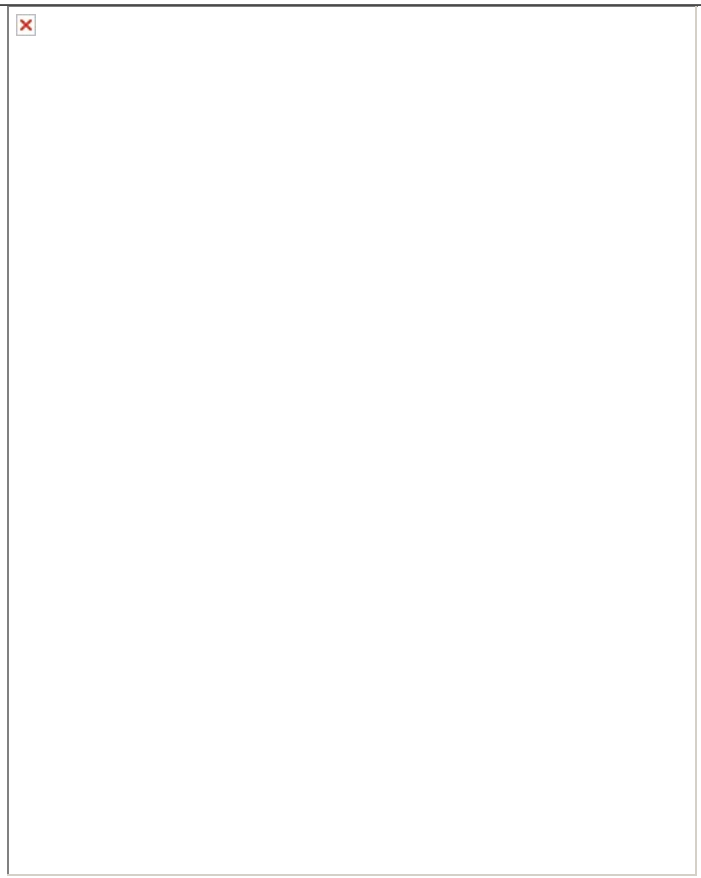


Table 5-1: Issue Owners and Event Owners

Project administrators may create, edit, and delete event templates in the Event Templates page of the Development Event subfolder in the Enterprise Edition folder.

For each co-owner event template created in the Event Template page, the project administrator must define appropriate properties, a set of applicable owners, attachment rules, access controls, an event group, event workflow rules, event notification rules, and event escalation rules."

5.4.1 Creating Co-Owner Event Templates

The Event Templates tree list control displays high-level information about every regular event template and co-owner event template that have been created in a DevTrack project.

For co-owner regular event template created in the Event Template page, the project administrator must also define appropriate properties, a set of applicable owners, attachment rules, access controls, an event group, event workflow rules, event notification rules, and event escalation rules.

To create co-owner event templates:

1Click the New button in the Event the Event Template page of the Development Event subfolder.

The Add a New Event Template dialog box appears.

2Define a unique, descriptive name for the event template in the Name text box control.

3Select the Co-owner Event option from the Event Type dropdown list.

Co-owner events are event templates in which applicable owners are represented by an event variable. A co-owner event template may be used to generate multiple event instances and assign those events to many different applicable owners.

4Select an option from the Variable dropdown list.

The Variable dropdown list displays a list of administrator-defined co-owner event variables and the {No Variable} variable.

- All co-owner event variables are defined by the project administrator in the Event Variable page of the Development Event subfolder.
- The {No Variable} variable

5Select an option in the choice number area.

6 Optional: To assign the event template to an event group, select an event group from the Event Group dropdown list.

The Event Group dropdown list displays a list of administrator-defined co-owner event groups and the {Ungrouped} system option.

- To assign the event template to an event group, select a co-owner event group in the Event Group dropdown list. Each co-owner event template may be assigned to one co-owner event group.
- To not assign the event template to an event group, select the {Ungrouped} option in the Event Group dropdown list. Ungrouped co-owner event templates are not subject to group advanced workflow rules.

Event groups are an optional DevTrack Enterprise Edition feature that enable development organizations to define more powerful and flexible rules for managing development

events. For more information about event groups and group advanced workflow rules see “Administering Event Groups”.

7Define a brief description of the event in the Description field.

8Click the OK button.

5.4.2 Administering Event Variables

An event variable is an administrator-defined variable that represents the applicable owners of a DevTrack development event. Every event variable represents one or more event variable values, and each of these value is tied to a project member that may own the events.

Event variables enable project administrators to save time and effort by creating more powerful and versatile event templates called co-owner event templates. In a co-owner event template the applicable owners are not defined directly in the template, but are represented in that template by an event variable.

Each event variable represents one or more event variable values and each event variable value is tied to one or more applicable owners, project members who may own event instances based on the event template.

Project administrators may create two different types of event variables: linked event variables and unlinked event variables.

Each administrator-defined event variable may be added to many different event co-owner event templates.

Unlinked Event Variable: Unlinked event variables represent one or more administrator-defined event variable values, which in turn are linked to one or more applicable owners. Event templates that incorporate unlinked event variables may be used to generate multiple co-owner event instances simultaneously. One event is generated for each administrator-defined event variable value.

Linked Event Variable: Linked event variables represent a set of variable values that are mapped to issue property values. Each issue property value is linked to a different set of applicable owners. One event instance is generated for each issue property value.

5.4.3 Enabling Co-owner Events and Event Variables

Event variables enable project administrators to*co-owner event templates*. Co-owner event templates are templates that use an event variable to represent the applicable owners of the events generated from the event template.

Project administrators may enable support for event variables and co-owner events before they can define event variables in DevTrack projects. Event variables and co-owner events may be enabled in the General Settings page of the Events subfolder in the Enterprise Edition folder.

To enable co-owner events and event variables, select the Enable Co-Owner Variable Support in the General Settings page of the Events subfolder.

5.4.4 Creating Unlinked Event Variables

Unlinked event variables are administrator-defined variables that may be incorporated into co-owner event templates. The unlinked event variable represents one or more administrator-defined event variable values. Each event variable value is linked to one or more applicable owners.

Co-owner event templates may be used to generate multiple events at a time. Whenever events are generated from a co-owner event template, one event instance may be created for every event variable value built into the event template.

For example, an administrator may create an unlinked variable event called Requirements Sign Off that represents six different administrator-defined event variable values: Manager 1, Manager 2, Manager 3, Manager 4, Manager 5, and Manager 6. Each event variable value is linked to a different project manager.

The administrator then creates a co-owner event template, and incorporates that event variable into the event template.When the administrator-defined autocreation rule (built into the co-owner event template) is triggered, DevTrack generates six events from the event template, one for each event variable value.

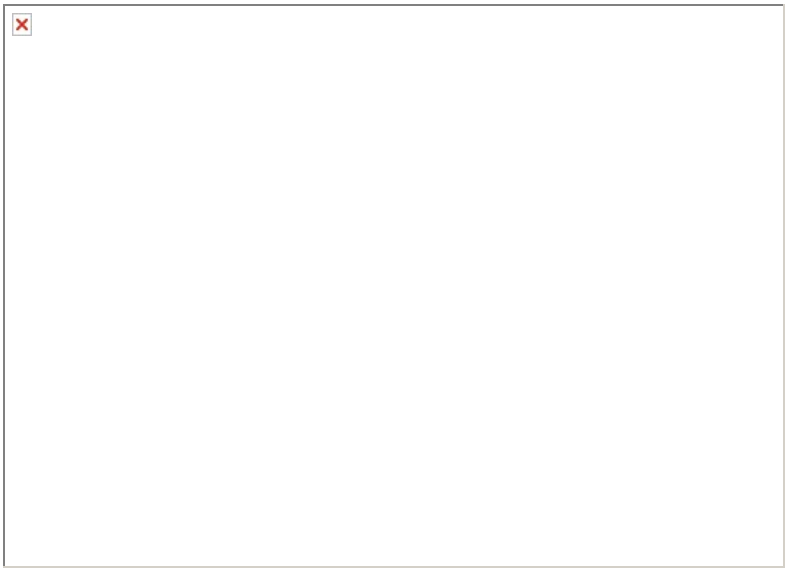


Figure 5-1: Linked Co-Owner Event Variables

Each of the six events generated would be assigned to a different applicable owner." If the project administrator linked event variable values to multiple applicable owners, then each event would have multiple applicable owners. Project members may then assign the event to one of the names displayed in the Owner dropdown list.

Project administrators may create, edit, and delete event variable and define applicable variable values in the Variable area of the General Settings page.

To create unlinked event variables:

1. Click the New button in the General Settings page of the Events subfolder.

   The Add a New Variable dialog box appears.

2. Define a unique title for the event variable in the Variable Name field.

   The title of the event variable should identify the type of data represented by the event variable.

3. Select the {Unlinked} variable option in the Linked Field dropdown list.

   The Linked Field dropdown list displays the {Unlinked} variable and multiple selection controls.

   - To create an unlinked event variable, select the {Unlinked} variable in the Linked Field dropdown list.
   - To create a linked event variable, select a selection control in the Linked Field dropdown list. Selection controls include administrator-defined dropdown list controls, combo box controls, autogrow combo box controls, and multiple selection list controls.

   If the {Unlinked} variable is selected in the Linked Field dropdown list, the event variable is not linked to any specific field. The Co-Owner Event control is automatically set to read-only.

4. Click the OK button.

   The Add a Variable dialog box closes.

   The Variable Value list displays the field choices belonging to the linked field choice selected in the Linked Field dropdown list.

5.4.5 Creating Linked Event Variables

Linked event variables are administrator-defined variables that may be incorporated into co-owner event templates. The linked event variable is always linked to an issue property and each issue property value is linked to a set of one or more applicable owner. The applicable owner of the event generated from the event template is determined by the issue property value of the parent issue.

For example, if event variable is linked to the Component issue property, the project administrator could define a unique set of applicable owners for each Component issue property value.

In the sample project, the Component property represent six property values (Web, GUI, Reporting, Printing, Installation, and Other). Once the event variable is linked to the issue property, the administrator may link each of the issue property values to a set of one or more applicable owners.

The option selected in the Component dropdown list determines which set of applicable owners are assigned to the event.

To define linked event variables:

1. Click the New button in the Variable area in the General Settings page of the Events folder.

2. Define a unique title of the event variable in the Variable Name field.

   The title of the event variable should identify the type of data represented by the event variable.

3. Select an issue property in the Linked Field dropdown list.

   The Linked Field dropdown list displays the {Unlinked} variable and multiple selection controls.

   - To create an unlinked event variable, select the {Unlinked} variable in the Linked Field dropdown list.
   - To create a linked event variable, select a selection control in the Linked Field dropdown list. Selection controls include administrator-defined dropdown list controls, combo box controls, autogrow combo box controls, and multiple selection list controls.

4. Select an event variable option from the Event Co-Owner Value Option.

   The Event Co-Owner Value dropdown list is optional. The Co-Owner Value Option dropdown list enables the project administrator to define options for linked event variables. The Co-Owner Value Option dropdown list displays two options:

   - The Always Same as the Incident Field Choice option
   - The Editable option

5. Click the OK button.

   The Add a Variable dialog box closes.

   The Variable Value list displays the field choices belonging to the linked field choice selected in the Linked Field dropdown list

5.4.6 Defining Co-Owner Event Attachment Rules

During the course of project workflow, project members may add attachments (documents, URLs, knowledge topics) to events whenever they create or update an event in the DevTrack client.

Event attachment rules determine whether attachments are allowed, optional, or mandatory, which knowledge items may be attached to an event, and how the attached knowledge items are named and described.

Every event template is defined by a set of attachment rules that determine whether documents may or must be linked to development events. Development events inherit event attachment rules from the event template used to create them.

Co-owner development event templates are defined by two distinct rules:

- Attachment rules determine whether attachments are mandatory, optional, or forbidden for development events based on an event template.

- Applicable knowledge item type rules determine which types of knowledge items may be attached to development events based on an event template.
  Project administrators may define applicable knowledge item type rules only if they have defined event attachments as mandatory or optional for an event template.

Unlike regular development event templates, co-owner event templates are not defined by knowledge item title or knowledge item description rules.

Project administrators may define event attachment rules for each event template in the Attachment tab.

To define co-owner event template attachment rules:

1. Select an event template in the Event Templates list of the Event Template page in the Development Events subfolder.

   The Event Templates list displays every development event template defined in the DevTrack project. Development event attachment rules are defined on a template-by-template basis.

2. Select an attachment option in the Attachment tab.

   - To bar project members from attaching knowledge items to development events based on the selected event template, select the None radio button.
   - To enable project members to optionally attach knowledge items to development events based on the selected event template, select the Optional radio button.
   - To mandate that project members attach knowledge items to development events based on the selected event template, select the Mandatory radio button.

3. **Optional:** To enable project members to attach specific types of knowledge items to events based on a development event template, select one or more attachment type options in the Attachment Type field.

   - The New Document option enables project members to attach new files to development events based on the event template.
   - The New HTML Link option enables project members to attach new URLs to development events based on the event template.
   - The Existing Knowledge Document option enables project members to attach files saved in the project knowledge base development events based on the event template.
   - The Existing HTML Link option enables project members to attach URLs saved in the project knowledge base development events based on the event template.
   - The Existing Knowledge Topic option enables project members to attach knowledge topics saved in the project knowledge base development events based on the event template.

   Project administrators may enable one or more types of knowledge items to be attached to development events only if they have selected the Optional or Mandatory attachment option.

5.5 Administering Applicable Owners of Development Events

Applicable owners are project members that may own events that are created from a particular event template. In the DevTrack client, the name of the project member is displayed as an option in the appropriate dropdown list control (usually, the Owner dropdown list control). The event submitter may assign that event to any project member displayed in the list.

account type: the account type options represent every project member belonging to that account type.

Group: the group options represent every project member belonging to that team group.

group folder: the group folder options represent every project member belonging to that group folder.

{Parent Issue Owner}: the {Parent Issue Owner} system user variable defines the project member that currently owns a development issue as an applicable owners.

{Parent Issue Submitter}: the {Parent Issue Submitter} system user variable defines the project member that submitted a development issue as an applicable owners.

Project administrators may define rules that determine which project members may own development events based on their account type, group, group folder, or one or more system user variables.

All applicable event owner rules are defined on a template-by-template basis.

5.6 Administering Event Management Privileges and Access Controls

Event access controls define which project members may view, submit, edit, and delete development events in a DevTrack project. Access controls enable project administrators to permit or deny a project member to perform a certain task based on their role in the project.

The ability of project members to perform tasks in DevTrack projects is usually determined by their *account type*, but the privilege to specific perform tasks may also be defined by team group membership or relationship between the project member and the event or issue in question.

All event access controls are defined on a template-by-template basis. Every event instance inherits the event access control settings from the event template that was used to create it.

5.6.1 Defining Event Access Controls

Event access controls determine which project members may manage development events based two factors: the event template that was used to create the event and the role of that project member in the project.

All event access controls are defined on a template-by-template basis. Project administrators may define event access controls for every event template in the project. Event instances inherit access control settings from the template that was used to create them.

The foundation of all DevTrack access controls are account types. The account type defines the role (responsibilities and privileges) that are granted to a project member in a project. Project administrators may grant event access controls to project members based on their account type, team group membership, or one of four system user variables:

{Event Submitter}: the {Event Submitter} system user variable represents the project member that submitted the development event.

{Event Owner}:the {Event Owner} system user variable represents the project member that currently owns the development event.

{Parent Issue Submitter}:the {Parent Issue Submitter} system user variable represents the project member that submitted the parent event of the development event.

{Parent Issue Owner}:the {Parent Issue Owner} system user variable represents the project member that submitted the current development event.

Project administrators may define event access controls for each event template in the User Access tab of the Event Template page in the Events subfolder.

To define event access controls:

1Select an event template in the Event Templates list.

2Select the User Access tab.

3Select an access control option in the Applicable Account Types list of the User Access tab.

4Select an option in the in the Applicable Account Types list.

- The Select the All Account Types radio button enables administrators to select all of the account types displayed in the list.
- The Select the Selected Account Types option radio button enables administrators to select only some of the account types displayed in the list.

5.6.2 Understanding Event Privileges

All event access controls are defined in the User Access tab of the Event Template page in the Events subfolder.

Can View privilege:the*Can View privilege*enables project members to view event based on an event template.

Can Submit privilege:the*Can Submit privilege*enables project members to submit events based on an event template.

Can Edit Open Event privilege:the*Can Edit*Open Event privilege enables project members to edit closed events based on an event template.

Can Edit Closed Event privilege:the*Can Edit*Closed Event privilege enables project members to edit closed events based on an event template.

Can Delete privilege:the*Can Delete*Event privilege enables project members to delete events based on an event template.

Each event template is defined by five event access controls. Project members may view, submit, edit, or delete development events based on the privileges that are granted to their account type in the event template that was used to create the development event.

5.7 Administering Customer Access to Development Events

Project administrators may use controls in the Customer Access tab of the Event Template page to define event privileges for customers through the Beta Customer Portal.

The event privileges that the administrator defines for each event template are inherited by every event based on that template.

5.7.1 Understanding Event Customer Access Rights

The Beta Customer Portal enables customer contacts to submit and view DevTrack issues to DevTrack projects. Project administrators may use controls in the Customer Access tab to grant customer contacts the ability to view and edit events as well.

Project administrators may grant the Who Can View privilege to DevTrack customer contacts based on their access type or one of two system-defined variables: the {Customer - Issue Submitter} variable and the {Customer - Event Submitter} variable.

Who Can View:the Who Can View access right determines which customer contacts may view events based on the current event template in the Beta Customer Portal.

Who Can Edit The: Who Can Edit access right determines which customer contacts may edit events based on the current event template in the Beta Customer Portal.

5.7.2 Defining Event Customer Access Controls

Administrator may use controls in the*Customer Access tab*of the Event Template page to grant the Who Can View privilege to customer contacts based on their account type.

The privileges that an administrator grants in each event template are inherited by every event that is based on that event template.

To define event template customer access controls:

1Select an event template in the Event Templates list.

2Select the Contacts Access tab.

3Select the Who Can View privilege option in the Applicable Contact Types list of the Contacts Access tab.

4Select an option in the in the Applicable Account Types list.

- The Select the All Account Types radio button enables administrators to select all of the account types displayed in the list.
- The Select the Selected Account Types option radio button enables administrators to select only some of the account types displayed in the list.

5.8 Administering Event Workflow

Workflow is a business process used to manage tasks within a project. Workflow rules enable project managers to define and track the flow of work between individuals and

teams. DevTrack standard event workflow enable administrators to define sophisticated workflow rules to manage and track issue subtasks. Event workflow defines the life cycle of a development event.

Event workflow states represent the various stages of the event life cycle from its initial creation to its closure and one or more intermediary steps.

Every development event is the child of a parent issue and represents a subtask of that development issue that must be performed within an issue workflow state.

Development event workflow enables organizations to independently manage the various subtasks that must be completed before a development issue can be forwarded from one issue workflow state to the next.

Event workflow rules enable project teams to manage and track development subtasks independently of one another, assign those tasks to different owners, and define separate schedules (start and due dates) for each event.

Like development issues, development events are managed within a set of administrator-defined workflow rules. The life cycle of a development event is composed of open and closed workflow states.

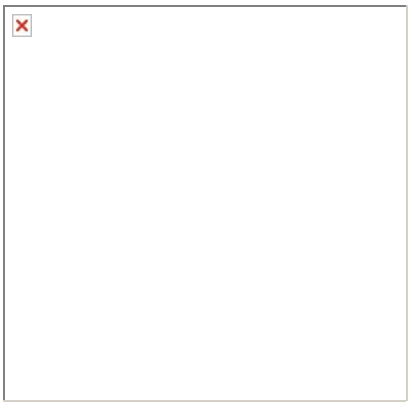


Figure 5-2: Event Workflow within Development Issue Workflow

The workflow of development events is typically much simpler than that of their parent development issues."

For example, the life cycle of a development event frequently consists of no more than two event workflow states: an open state and a closed state. In this scenario, the event merely tracks whether a specific subtask has been completed.

The event has two workflow states:

- To be done (open)

Chapter 6 Event Notification & Escalation Adminis

6 Chapter 6 Event Notification & Escalation Administration

In this chapter:

- Administering Event Notifications and Escalations
- Administering Event Notifications
- Administering Event Escalations
- Administering Event Notification and Escalation Conditions
- Administering Event Notification and Escalation Actions
- Administering Potential Recipients and Subscriptions
- Select the Yes or No option in the Option dropdown list.

6.1 Administering Event Notifications and Escalations

DevTrack event notifications and event escalations provide development organizations the means to ensure that event stakeholders are automatically notified at key junctures in the event lifecycle, enforce accountability, and drive the quality of their products.

All event notifications and event escalations are based on administrator-defined rules which identify the triggers, conditions, notification methods and messages, and recipients of every notification or escalation action.

Event notifications and escalations enable to ensure that event stakeholders are kept abreast of the status of their events. The primary difference between DevTrack event notification and event escalation is how these rules are triggered:

Event Notification: Event notifications alert event stakeholders whenever key changes are made to their development events-subscribers may be notified whenever events are created, updated, or closed in a development project.

Event Escalation: Event escalations alert event stakeholders at key points in the lifecycle of development events or when insufficient action is taken on an event in a defined time period and provide for the reassignment of escalated events to other project members and workflow states.

Both DevTrack event notification and event escalation support:

- Customized notification triggers. The trigger defines the specific*action*(event notification) or*lack of action*(event escalation) that prompts the delivery of the notification message.
- Customized conditions that limit the scope of the trigger to a subset of events based on an administrator-defined query.
- Customized email notification messages and notification methods (email, pager,mobile phone, and personal folder)
- Customized subscription rules including mandatory and optional notification subscriptions.

6.1.1 Notification and escalation triggers

The primary difference between DevTrack event notifications and event escalations is how these rules are triggered.

6.1.2 Push and pull notification subscriptions

TechExcel DevTrack supports both*push notification subscriptions*and*pull notification subscriptions*.

All event notifications and escalations are defined by administrator-defined*subscription rules*. Subscription rules determine whether a project team member must, can, or cannot receive a notification.

Subscriptions rules may mandate that certain project members receive event notifications for certain events while others may*opt in*or*opt out*of individual subscriptions. Project administrators may also mandate that certain project member*never*receive event notifications based on event notification rules.

- Push notifications are “instantly and actively transferred (pushed)” to subscribers. A project member is subscribed to all events and is always notified when changes are made to events based on that rule.
- Pull notifications are optionally sent. A project member may choose to subscribe to a notifications on an event-by-event basis.

6.2 Administering Event Notifications

In DevTrack, event notification is the process by which stakeholders are notified (by email, pager, or mobile phone) whenever a specific change—such as the creation or deletion—is made to a development event. All event notifications are based on administrator-defined event notification rules.

An event notification rule is defined by a*notification trigger*, one or more*notification actions*, and one or more*subscribers*. The scope of an event notification rule may be defined by a*notification condition*.

Using controls in the Add Notification Rule manager and the Properties tab of the Event Notifications page, project administrators may define and update many notification rule properties.

Notification Trigger:A notification trigger identifies the event management task that triggers a notification action.

Notification Condition:A notification condition limits the limit the scope of the trigger to a subset of events based on an administrator-defined query.

Notification Action:A notification action defines the notification message and the notification method (email, pager, mobile phone).

Subscribers

6.2.1 Enabling Event Notification in a DevTrack Project

Event notification is an optional DevTrack feature that must be enabled on a project-by-project basis.

To enable event notification select the Enable Auto Email Notification check box in the Event Notifications page of the Event Notification subfolder in the Advanced Settings folder.

6.2.2 Defining Event Notification Rules

An event notification rule identifies which project team members are notified whenever a specific change is made to a development event.

An event notification rule is defined by a*notification trigger*, one or more notification actions, and*notification subscription rules*. Optional event notification conditions limit the scope of the rule to a subset of development events

Any number of notification rules may be defined in a project.

To define an event notification rule:

1Select Advanced Features > Notifications/Escalations > Event Notifications.

2Click the New button.

The Add Notification Rule dialog box appears.

3Provide a name and description for the event notification rule.

A unique name and note that describe the logic of the event notification rule enables project administrators to save time when they review or update event notifications.

Notification trigger

4Select a trigger in the Trigger dropdown list.

The Trigger dropdown list displays six system triggers and all administrator-defined status change and field change triggers.Every event notification rule is defined by a single notification trigger.

Notification conditions

5 Optional: To limit the scope of a notification rule, select an event notification condition in the Condition dropdown list.

The Condition dropdown list displays all administrator-defined conditions.

Custom notification conditions may be defined in the Notification & Escalation Conditions window. For step-by-step instructions see “Administering Event Notification and Escalation Conditions”.

Applicable event templates

6Select one or more event templates in the Applicable Regular or Co-Owner Event Template list.

The scope of the event notification rule is restricted to development events based on the selected regular and co-owner event templates.

7Click the OK button.

The Add Notification dialog box closes and the event notification rule is displayed in the Notification Rules list.

Notification actions

8Define one or more notification actions in the Actions tab.

A notification action is defined by a*notification method*(email, pager, or mobile phone) and a*notification message*.

- To define an email notification action, select the Notify By Sending Email option and select appropriate*internal*(project team member) and*external* (customer) messages.
- To define a pager notification action, select the Notify By Pager option and select an appropriate message.
- To define a mobile phone notification action, select the Notify By Mobile Phone option and select an appropriate message.

Custom notification messages may be defined in the Define Email window. For step-by-step instructions see “Select the Yes or No option in the Option dropdown list.” .

Potential recipients

9Define potential recipients in the Potential Recipient tab.

A potential recipient is a project team member who*may subscribe*to an event notification rule. A project team member may be identified as a potential recipient based on their account type, team group, or potential recipient variable.

Potential recipient subscription rules

10Define subscription rules in the Subscription Option tab.

Potential recipient subscription rules define whether a potential recipient can,cannot, or must subscribe to an event notification.

Subscription rules may be defined for each account type, team group, or user account and affect only those project team members that have been designated as potential recipients of event notifications.

Subscribers

11Define subscriptions in the Potential Recipient tab.

Event notification subscriptions define which project team members are subscribed and which project team members are not subscribed to each event notification rule.Event notification rules are specifically defined for each user account and override all potential recipient subscription rules.

To rename an event notification rule:

To rename an event notification rule, select the rule in the Notification Rules list and click the Rename button. The name of the rule may be changed in the dialog box.

To delete an event notification rule:

To delete an event notification rule, select the rule in the Notification Rules list and click the Delete button.

6.2.3 Administering Event Notification Triggers

In DevTrack, an event notification trigger is an*action*—such as the creation, edit, or deletion of an event—that prompts the DevTrack Mail Service to deliver a notification message to a list of event stakeholders.

An event notification is sent every time that the triggering action is performed. If the action is not performed, the event notification rule is not triggered and the notification message is not sent.

DevTrack supports both*system triggers*and*custom triggers*.

System triggers

DevTrack system triggers prompt the DevTrack Mail Service to send notification messages whenever an event management task is performed on events meeting the conditions of the event notification rule.

DevTrack supports six system triggers:

New: The New Event trigger generates an event notification whenever a new development event is created in a project.

Close: The Closed Event trigger generates an event notification whenever a development event is closed in a project.

Status Change: Status change triggers generate event notifications based on changes to the workflow state of development events.

Owner Change: Owner change triggers generate event notifications based on changes in event ownership.

6.3 Administering Event Escalations

Event escalations alert event stakeholders at key points in the lifecycle of development events or when insufficient action is taken on an event in a defined time period and provide for the reassignment of escalated events to other project members and workflow states.

Event escalation enables project teams to ensure that DevTrack events do not “fall through the cracks”. Project teams may be notified by email, pager, or mobile phone based on an administrator-defined escalation schedule. If still no action is taken, events may be automatically reassigned to another project member, group, or group folder.

An event escalation rule is defined by an*escalation trigger*, one or more optional*escalation condition*s, one or more*escalation actions*, and its*subscribers*—the project team members that may be notified when an event is escalated based on the rule.

6.3.1 Escalation triggers

Unlike event notifications (which are based on the changes that project members make to DevTrack events), event escalation rules are based on changes that are not made to events.

Every event escalation rule is defined by one of nine, system-defined*escalation triggers*: No Acknowledge, Open Too Long, No Progress, No Status Change Only, No Owner Change Only, Before Start Date, After Start Date, Before Finish Date, and After Finish Date.

6.3.2 Escalation conditions

The scope of an escalation rule may be limited to development events that meet a set of criteria—the*event escalation condition*.

An escalation condition is essentially a query that filters events by their event property values. For example, a condition may limit the scope of an escalation rule to events that have a High or Urgent status.

6.3.3 Escalation actions

All escalation triggers are “time triggers”. DevTrack executes an escalation action when the “status” defined by the escalation trigger remains unchanged for the escalation wait period.

Event escalation actions define the time period that must pass before notifications are sent, the method used to notify subscribers (email, pager, or mobile phone), the message delivered by the notification, and the number of times that notification are sent before an escalated event is reassigned to another project team member.

6.3.4 Subscribers

DevTrack enables administrators to manage both push email and pull email for each event escalation rule using subscription rules.

Project administrators may mandate that certain project members receive event escalation notifications for certain events while others may opt in or opt out of individual subscriptions. Project administrators may also mandate that certain project member never receive event escalation notifications based on event escalation rules.

Enabling Event Escalations and Start Dates

6.3.5 Administering Event Escalation Rules

Event escalations alert event stakeholders at key points in the lifecycle of development events or when insufficient action is taken on an event in a defined time period and provide for the reassignment of escalated events to other project members and workflow states.

An event escalation rule is defined by an*escalation trigger*, one or more notification actions, and*escalation subscription rules*. Optional event*escalation conditions*limit the scope of the rule to a subset of development events.

Any number of notification rules may be defined in a project.

To create event escalation rules:

1. Select Advanced Features > Notifications/Escalations > Event Escalations.
2. Click the New button.

   The Add Escalation Rule dialog box appears.
3. Provide a name and description for the event escalation rule. A unique name and note that describe the logic of the event notification rule enables project administrators to save time when they review or update event notifications.

Escalation triggers

4. Select an option from the Trigger dropdown list. The Trigger dropdown list displays two system-defined triggers: the On Start Date trigger and the On Finish Date trigger. For more information see “Understanding Event Escalation Triggers”.

Escalation conditions

5. **Optional**: To limit the scope of an escalation rule, select an event escalation condition in the Condition dropdown list.

   The Condition dropdown list displays all administrator-defined conditions.

   Custom escalation conditions may be defined in the Notification & Escalation Conditions window. For step-by-step instructions see “Administering Event Notification and Escalation Conditions”.

Applicable event templates

6 Select one or more event templates in the Applicable Regular or Co-Owner Event Template list.

The scope of the event escalation rule is restricted to development events based on the selected regular and co-owner event templates.

7 Click the OK button.

The Add Escalation dialog box closes and the event escalation rule is displayed in the Escalation Rules list.

Escalation actions

An escalation action is defined by an *escalation schedule*, an *escalation method* (email, pager, or mobile phone) and an *escalation message*.

8 Define the escalation schedule. An *escalation schedule* is defined by an ***escalation wait period***, a *repeat schedule*, and a *repeat number*.

- To define the escalation wait period, enter the number of hours and minutes in the Start Escalation controls.
- To define the repeat schedule, enter the number of hours and minutes in the Repeat Escalation controls.
- To define the repeat number, enter a digit in the Repeat Number text box.

9 Select an appropriate notification message for each notification method:

- To define an email notification action, select the Notify By Sending Email option and select appropriate internal and external messages.
- To define a pager notification action, select the Notify By Pager option and select an appropriate message.
- To define a mobile phone notification action, select the Notify By Mobile Phone option and select an appropriate message.

Custom notification messages may be defined in the Define Email window. For step-by-step instructions see “Select the Yes or No option in the Option dropdown list.” .

Event reassignment actions

10 **Optional**: To enable development events to be automatically forwarded to a different workflow state or owner if no action is taken on that event, define event reassignment rules.

Event reassignment actions may be defined in the Escalation Action window. For step-by-step instructions see “Administering Event Reassignment Actions”.

Potential recipients

11 Define potential recipients in the Potential Recipient tab.

A potential recipient is a project team member who *may subscribe* to an event notification rule. A project team member may be identified as a potential recipient based on their account type, team group, or potential recipient variable.

Potential recipient subscription rules

12 Define subscription rules in the Subscription Option tab.

Potential recipient subscription rules define whether a potential recipient can, cannot, or must subscribe to an event notification. Subscription rules may be defined for each account type, team group, or user account and affect only those project team members that have been designated as potential recipients of event notifications.

Subscribers

13 Define subscriptions in the Potential Recipient tab.

Event notification subscriptions define which project team members are subscribed and which project team members are not subscribed to each event notification rule.Event notification rules are specifically defined for each user account and override all potential recipient subscription rules.

Project administrators may use controls in the Email Escalations page to rename, event escalation rules.

6.3.6 Understanding Event Escalation Triggers

In DevTrack, all event escalation triggers are “time triggers” which prompt the DevTrack Mail Service to send notification messages to event stakeholders at key points in the event lifecycle—for example, its start and finish dates.

An *escalation trigger* identifies the *unperformed* task that triggers an administrator-defined action (event notification or event re-assignment). Unlike event notification triggers which represents specific acts performed on an event, event escalation triggers represents acts that *are not* performed on an event.

DevTrack supports four event escalation triggers:

On Start Date: The On Start Date trigger prompts escalation actions when work on an event has not started within the period defined in the escalation schedule.

On Finish Date: The On Finish Date trigger prompts escalation actions when work on an event has not ended within the period defined in the escalation schedule.

In DevTrack, every escalation rule is defined by a single trigger. The scope of the escalation rule may be limited by one or more escalation conditions which limit the number of development events that are subject to it.

6.3.7 Administering Event Escalation Actions

Event escalation actions define the time period that must pass before notifications are sent, the method used to notify subscribers (email, pager, or mobile phone), the message delivered by the notification, and the number of times that notification are sent before an escalated event is reassigned to another project team member.

An escalation action is defined by an *escalation schedule*, a *notification method* and, optionally, an *event reassignment action:*

6.3.8 Escalation schedules

An escalation schedule defines the time that must pass before notifications are delivered to escalation subscribers, the number of times that notifications are sent, and the time between notifications.

6.3.9 Notification methods

A notification method defines the method by which subscribers are notified that an event has been escalated.

DevTrack supports three event escalation notification methods:

Email Notification: Email notification actions notify subscribers by email whenever an event notification rule is triggered.

Page Notification: Pager notification actions notify subscribers by pager whenever an event notification rule is triggered.

Mobile Phone Notification: Mobile phone notification actions notify subscribers by mobile phone whenever an event notification rule is triggered.

A different email message may be sent for every notification method.

6.3.10 Event reassignment actions

An event reassignment action defines rules for routing escalated development events to specific workflow states or project team members at the end of the event escalation schedule.

Event reassignment actions are an optional feature that must be enabled and defined in each event escalation rule.

6.3.11 Administering Event Escalation Schedules

An escalation schedule defines the time that must pass before notifications are delivered to escalation subscribers, the number of times that notifications may be sent, and the time between the delivery of each notification.

All escalation triggers are “time triggers”. DevTrack executes an escalation action when the “state” defined by the escalation trigger remains unchanged for the escalation wait period.

Using controls in the Escalation Action window, project administrators may define an escalation wait period, an escalation schedule, and a repeat number for each event escalation rule.

An*escalation schedule*is defined by an*escalation wait period*, a*repeat schedule*, and a*repeat number*.

- The escalation wait period defines the time that must pass before the subscribers are notified of the event escalation. The escalation wait period is defined in hours and minutes.
- The escalation repeat schedule defines time between escalation actions. The escalation repeat period is defined in hours and minutes.
- The repeat number defines the number of times that events may be escalated before the event is reassigned or the event expires.

After the final escalation action is performed, escalated events may be automatically reassigned to another project member, group, or group folder based on administrator-defined reassignment settings. For more information see “Administering Event Reassignment Actions”.

To define an escalation schedule:

1. Define an event escalation rule, escalation trigger and trigger conditions in the Event Escalations page.
2. Click the Add button in the Action tab.

   Escalation actions may be defined whenever you create or edit an event escalation rule.

   The Escalation Action window appears.
3. Define the escalation wait period in the Start Escalation if Event Remain Open After text box. The escalation wait period defines the time that must pass before the subscribers are notified of the event escalation. The escalation wait period is defined in hours and minutes.
4. Define the escalation repeat schedule in the Repeat Escalation Every text box.

   The escalation repeat schedule defines time between escalation actions. The escalation repeat period is defined in hours and minutes.
5. Define the repeat number in the Repeat Number text box.

   The repeat number defines the number of times that events may be escalated before the event is reassigned or the event expires.

6.3.12 Administering Event Reassignment Actions

An event reassignment action defines rules for routing escalated development events to specific workflow states or project team members at the end of the event escalation schedule.

If event reassignment is enabled, the escalated event is reassigned and its workflow state is changed*only after*every scheduled escalation action is performed.

Using controls in the Escalation Action window, project members may enable event reassignment and routing rules, designate a project team member or group folder as the event owner, and define workflow state changes for each event escalation rule.

The target event owner of an event reassignment action may be identified by user account, group folder, or the {Submitter} potential recipient variable.

To enable event reassignment:

1. Define event escalation actions in the Actions tab of the Event Escalation page.

Event reassignment actions may be defined whenever you create or edit event escalation actions in the Actions tab of the Event Escalation page.

2Select the Do Event Re-assignment After the Actions Above check box.

Event reassignment actions must be specifically enabled for each event escalation rule.

3Select a project team member in the Re-assign To dropdown list.

The Re-assign To dropdown list displays all user accounts and group folders as well as the {Submitter} potential recipient variable.

6.4 Administering Event Notification and Escalation Conditions

In DevTrack, an event condition is a query that defines the scope of an event notification or event escalation rule. Escalation conditions may be based on event property values or on the property values of its parent development issue.

Development events that do not met the condition criteria are not subject to the notification or escalation rule and no action is taken when that event “triggers” an escalation or notification action.

Using controls in the Event Notification & Escalation Condition window, project administrators may define and manage event conditions.

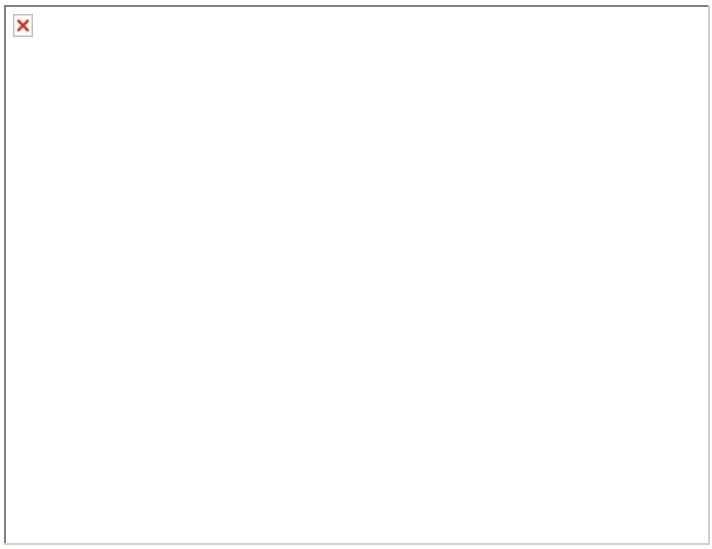


Figure 6-1: Event Notification & Escalation Condition Window

An event condition query is a set of instructions for identifying a subset of development events that are subject to an event notification or escalation rule." Every condition query consists of one or more search conditions, which identify the data fields searched and the field value criteria for each field. Multiple search conditions may be organized into search terms and filtered by using the AND, NOT, and OR operators.

Escalation condition queries may be based on event property values or parent issue property values tracked in system fields, custom fields, and seven system-defined variables.

Event Fields: Event conditions may be defined by event property definitions that tracked in six fields: the Event Owner, Created By, Event Status, Start Date, Due Date, Create Date system fields. Conditions may also be defined based on keywords in the event title or description.

System Fields: Event conditions may be defined by issue property definitions that tracked in 13 system fields.

Custom Fields: Event conditions may be defined by issue property definitions that tracked in all custom fields.

System Variables: Event conditions may be defined by issue property definitions based on dynamic field or user variables. System variables include the issue ID, the issue owner, the issue submitter, and the issue subproject.

6.5 Administering Event Notification and Escalation Actions

In DevTrack, an event notification or event escalation action identifies the*notification methods*(by email, pager, or mobile phone) and the*notification messages*that are sent by each method to the notification or escalation rule subscribers.

Event notification and event escalation rules share a common repository of notification messages and use the same notification methods. However, there are important distinctions between the two based on how each is triggered.

- An event notification action is defined by a*notification method*(email, pager, or mobile phone) and a*notification message*. Notification actions are triggered by the execution of a specified issue or event management task.
- An escalation action is defined by an*escalation schedule*, an*escalation method* (email, personal folder, pager, or mobile phone) and an*escalation message*.Escalation actions are triggered when a specified issue or event management task is not performed within a specified time period as identified by escalation schedule.

Also, DevTrack event notification supports email notification actions that are specifically targeted to*external subscribers*. An external subscriber is a customer that has been designated as an issue stakeholder and subscribed to a notification rule.

DevTrack supports three notification methods for both event notifications and event escalations:

Email Notification: Email notification actions notify subscribers by email whenever an event notification or event escalation rule is triggered. Event notification actions may be targeted to both internal and external subscribers.

Page Notification: Pager notification actions notify subscribers by pager whenever an event notification or event escalation rule is triggered.

Mobile Phone Notification: Mobile phone notification actions notify subscribers by mobile phone whenever an event notification or event escalation rule is triggered.

Personal folder notifications are only supported for issue notifications and escalations.

If an administrator enables all three notification actions for a event notification rule, every subscriber to that event notification will receive notification by email, pager, and mobile phone when that notification rule is triggered.

6.5.1 Identifying Notification Action Email Addresses

DevTrack enables project administrators to define event notification rules that automatically notify many different subscribers of the status of DevTrack events.

Event notification email may be sent to project members, customer contacts, and administrator-defined email address lists.

The email addresses of event notification subscribers are managed in different tools in DevTrack applications.

- Project member email addresses, pager addresses, and mobile phone addresses are all managed in the User Manager.Project member addresses must be defined in the User Manager for project members to receive event notifications. All DevTrack project members addresses are managed at the system level.
- Customer contact email addresses are managed in the Customer Manager of the DevTrack Windows client.
- Project administrators may create a unique list of email addresses for each notification rule. The email addresses on that list are notified automatically whenever the event notification rule is triggered.

6.5.2 Defining Event Email Notifications

Email notification actions send email notifications to project members and customers whenever an event notification rule is triggered.

Distinct notification messages may be sent to internal subscribers (project team members) and external subscribers (customers).

Using controls in the Escalation Action window, project administrators may define the email messages to be sent to rule subscribers whenever the event escalation rule is triggered.

Figure 6-2: Escalation Action Window (Detail)

Email notifications may also be sent to an administrator-defined list of email addresses and user-defined cc lists."

- The administrator may define a list of email addresses that receive all email notifications relevant to a specific event notification rule.
- Individual project members may define a cc list of email addresses for events that they subscribe to in the DevTrack client. Project administrators must enable event specific cc lists for each event notification rule.

Email addresses are managed in different DevTrack applications depending on the subscriber.

- Project member email addresses are managed in the User Manager in DevSuite Admin.
- Customer email addresses are managed in the Customer Manager of the DevTrack Windows client.

To define an event escalation email notification:

1Define an event escalation rule, escalation trigger and trigger conditions.

2Click the Add button in the Action tab.

Escalation actions may be defined whenever you create or edit an event escalation rule.

The Escalation Action window appears.

3Select an email message in the Email Format dropdown list.

4Select an email message in the Last Email Format dropdown list.

The Email Format and Last Email Format dropdown lists display all administrator-defined email messages.

- The option selected in the Email Format identifies the email message that is sent to rule subscribers when an event is first escalated and for every subsequent notification prior to the final notification.
- The option selected in the Last Email Format identifies the last email message that is sent to rule subscribers before the escalated event is reassigned.

To define email notification actions:

1Select an email notification rule in the Notification Rule list of the Event notifications page.

2 Select the Actions tab.

3 Select the By Sending Email check box.

Project administrators must enable email notification.

4 Select an option from the Email Format dropdown list.

Project administrators may select the email format used to send email notifications to (internal) project members.

The Email Format dropdown list displays all of the email formats defined in the Email Format manager. Project administrators may define new email formats by clicking the Setup button.

5 Select an option from the Email Format for Customer dropdown list.

Project administrators may select the email format used to send email notifications to (external) customers.

The Email Format for Customer dropdown list displays all of the email formats defined in the Email Format manager.Project administrators may define new email formats by clicking the Setup button.

6.5.3 Defining Customer Email Notifications

Customers may be notified by email whenever an issue or event notification rule (to which they are subscribed) is triggered. Customers cannot subscribe to issue escalation or event escalation rules.

All customer subscriptions are based on potential recipient variables. In DevTrack, a potential recipient variable identifies potential subscribers based on their relationship to an individual issue. If a project member or customer submitted or previously owned an issue that user may be designated as a potential recipient of notification messages.

Distinct notification messages may be defined for internal (project member) and external (customer) issue and event notification subscribers.

To define a customer email notification action:

1 Select an event notification rule in the Event Notification page.

Customers cannot subscribe to issue escalation or event escalation rules.

2 Select the By Sending Email check box in the Actions tab.

3 Select a notification message in the Email Format for Customer dropdown list.

The Email Format for Customer dropdown list displays all of the email formats defined in the Email Format manager. Project administrators may define new email formats by clicking the Setup button.

Every customer that is subscribed to the selected event notification rule is notified by email when the rule is triggered. Customers may be identified as subscribers based two potential recipient variables. For more information see “Customers as potential recipients”.

6.5.4 Defining Pager and Mobile Phone Notifications

Pager and mobile phone notifications are automatically sent to the subscribers pager or mobile phone address that is saved in the User Manager.

All pager and mobile phone numbers are defined at the site-level in the System Settings project.

To define a pager or mobile phone notification:

1 Define an event escalation rule, escalation trigger and trigger conditions.

2 Click the Add button in the Action tab.

Escalation actions may be defined whenever you create or edit an event escalation rule.

The Escalation Action window appears.

3 Click the Pager & Mobile Phone Notification button.

The Pager & Mobile Phone Notification window appears.

4 **Optional**: To define an event escalation pager action, select the Enable Pager Notification check box and select an email message in the Pager dropdown list.

5 **Optional**: To define an event escalation mobile phone action, select the Enable Pager Notification check box and select an email message in the Mobile Phone dropdown list.

The option selected in the Email Format identifies the email message that is sent to rule subscribers when an event is first escalated and for every subsequent otification prior to the final notification.

6.6 Administering Potential Recipients and Subscriptions

For each event notification rule, recipients are defined at three levels:(1)*potential recipients*are identified by account type, team group, or potential recipient variable; (2) s*ubscription rules*are defined for each account type and team group—these rules determine whether project members may, may not, or must receive event notifications; (3) *subscription*s are defined on an account-by-account basis.

Potential Recipient:A potential recipient is a project team member who*may subscribe*to an event notification or event escalation rule. To be notified by a notification action, the project member must be subscribed to an event notification rule.

Subscription Rule: Subscription rules enable the project administrator to define whether a potential recipient can, cannot, or must subscribe to an event notification based on his or her account type or team group.

Subscription:

The email notifications sent to the addresses on the list use the same email format selected in the Email Format dropdown list in the Actions tab. Project teams that are sending two sets of email notifications using different email formats: an internal email to project members and an external email to customers, must bear in mind that the email sent to addresses in the User Defined list use the internal email format.

6.6.1 Identifying Potential Recipients of Notification Actions

A potential recipient is a project team member who*may subscribe*to an event notification rule. A project team member may be identified as a potential recipient based on their account type, team group, or potential recipient variable.

A project team member may be designated as a potential recipient based on his or her account type, team group, or*relationship with an event.*In DevTrack, these relationships are represented by*potential recipient variables.*

Most potential recipients are identified by account type or team group membership.

6.6.2 Potential recipient variables

A potential recipient variable is a dynamic placeholder that represents an issue stakeholder—a project member or customer that has worked on an issue. For example, a user may be designated as potential recipients based on having submitted or previiously owned the development issue or event.

{Issue Owner}: The {Event Owner} system variable enables administrators to notify the owner of the parent event by event notification actions.

{Issue Submitter}: The {Event Submitter} system variable enables administrators to notify the submitter of the parent event by event notification actions.

{Event Owner}: The {Event Owner} system variable enables administrators tonotify the owner of the parent event by event notification actions.

{Event Submitter}: The {Event Submitter} system variable enables administrators to notify the submitter of the parent event by event notification actions.

{Event Owner's Group Email Address}: The {Event Owner's Group Email List}system variable represents an administrator-defined list of email addresses that may designated as potential recipients of email notification actions.

{Last Previous Owner}: The {Last Previous Owner} system variable enablesadministrators to notify previous owner of the event by event notification actions.

{All Previous Owners}: The {All Previous Owners} system variable enablesadministrators to notify all previous owners of the event by event notification actions.

{User Defined Address}: The {User Defined Address} system variable enablesadministrators to define a list of email addresses. Email notifications are sent to all email addresses on that list.

{Issue Specific CC List}: The {Event Specific CC List} user variable represents a user-defined cc list that defines the potential recipients of email notification actions.

A event-specific cc list is a list of email addresses that identifies the stakeholders who must be kept up to date in regards to a specific event. Every event-specific cc list is defined on an event-by-event basis by a project team member in the DevTrack client.

{Linked Issue Owner}: The {Linked Event Owner} system variable enablesadministrators to notify the owners of any events that are linked to the parent event by event notification actions.

{Linked Issue Submitter}: The {Linked Event Submitter} system variable enables administrators to notify the submitters of any events that are linked to the parent event by event notification actions.

The scope of most potential recipient variables may be further limited by account type.. Once defined, only project team members that belong to a specifc account type may be designated as potential recipients based on their relationship to the issue or event. For more information see "Defining the Scope of Potential Recipient Variables".

6.6.3 Customers as potential recipients

In DevTrack, customers may be identified as issue and event notification rule subscribers based on their relationship to development issues.

Relationships between users and issues are represented by*user variables*. A user variable is a dynamic placeholder that represents an issue stakeholder. Two user variables identify customers:

{Customer - Issue Submitter}: The {Customer - Issue Submitter} system variable enables administrators to notify the customer that submitted the parent issue by email notification.

{Selected Issue Contacts}: The {Selected Issue Contacts} system variable enables administrators to notify a list of parent issue contacts by email notification.

To define customers as potential recipients of email notifications, the project administrator must first enable and configure project-level support for a customer base project and the Beta Customer Portal. No customer variables are displayed in the Potential Recipients tab until these features are enabled.

Customers may be notified by email whenever an issue or event notificaiton rule to which they are subscribed is triggered. Customers cannot subscribe to issue escalation or event escalation rules.

Using controls in the Potential Recipient tab, the project administrator may define which DevTrack project members may be notified by a notification action for each event notification rule.

To define a user as a potential recipient:

1Select an event notification in the Event Notification page or an escalation rule in the Event Escalation page.

2Select the Potential Recipients tab.

3Select an account type, team group, or potential recipient variable in the Notification Recipients list.

Potential recipients may be identified by account type, team group, or potential recipient variable.

6.6.4 Defining the Scope of Potential Recipient Variables

In DevTrack, a potential recipient variable represents a project team member that may be designated as a*potential recipient*of an event notification or event escalation action based on their relationship with a development event or its parent issue.

Using one or more potential recipient variables, a project administrator may ensure that every event or issue stakeholder is kept up-to-date with their issues and events.

The scope of potential recipient variables may be limited by account type or project:

- The scope of most potential recipient variables may be limited by*account type*.Only those project members that belong to the selected account types are defined as potential recipients of notification actions.
- The scope of the {Linked Event Submitter} and {Linked Event Owner} potential recipient variables may be defined by project. Only those project members that submitted or currently own a linked event in the selected projects are defined as potential recipients of notification actions.

To define the scope of a potential recipient variable by account type:

1. Select an event notification rule.
2. Double-click a user variable in the Potential Recipient tab.

   The Email Notification/Escalation window appears.
3. Select one or more account types in the Account Types list.

   The Account Types list displays the name of every administrator-defined account type.
4. Click the OK button.

To define the scope of a user variable by project:

1. Select an event notification rule.
2. Double-click a user variable in the Potential Recipient tab.

   The Email Notification/Escalation window appears.
3. Select one or more account types in the Account Types list.

   The Account Types list displays the name of every project in the DevSuite site.

In an integrated DevSuite site DevTrack development events may be linked to development events in oth

Chapter 7 Beta Customer Portal Administration

7 CHAPTER 7 Beta Customer Portal Administration

In this chapter:

- Managing the DevTrack Beta Customer Portal
- Managing Customer Base Projects
- Customizing the Address Page of the Customer Manager
- Customizing Contact Info in the Customer Manager
- Managing DevTrack Customer Manager Privileges
- Managing Beta Customer Portal Page and Field Access
- Managing Customer Issue Submission Settings
- Managing the Beta Customer Portal

7.1 Managing the DevTrack Beta Customer Portal

Project administrators may use tools in the Beta Customer Portal folder of the DevTrack Tree window to associate a customer base project with a DevTrack project, customize the Customer Manager, and manage the DevTrack Beta Customer Portal.

The*Beta Customer Portal*enables customers and customer contacts to submit and edit DevTrack issues through the Internet.

The*Customer Manager*enables DevTrack project members to create customers and customer contacts. Customer and contact information is stored in a customer base project. Each customer is assigned an administrator-defined customer type. Each contact is assigned an administrator-defined contact type.

A*customer base project*enables project teams to share a common database of customers and customer contacts. Each DevTrack project may be associated with one customer base project.

Project administrators may use tools in DevSuite Admin to perform the following tasks:

- Identify the customer base project associated with a DevTrack project. DevTrack uses a customer base project to manage customer data.
- Customize the graphical user interface of the Customer Manager.
- Assign Customer Manager privileges to DevTrack project account types.
- Define customer types and contact types that
- Define privileges for each customer type
- Define page-level and field-level access controls.

The ability of customers and customer contacts to access to the DevTrack project through the Beta Customer Portal is regulated by administrator-defined customer accounts and contact accounts.

The Beta Customer Portal enables user-defined customers and customer contacts to submit, view, and edit DevTrack issues through the Internet. All customers and customer contacts are created and managed by DevTrack project members in the Customer Manager in a DevTrack client.

Administrators may customize the graphical user interface (GUI) of the Customer Manager, define customer types and contact types, assign access privileges to those customer types and contact types, and perform other administrative tasks using tools in the Customer Web Portal folder of the Admin Tree window.

Once the administrator configures and customizes the Beta Customer Portal, project members may create and manage customers and customer contacts using the Customer Manager in the DevTrack Windows client.

7.1.1 Understanding the Beta Customer Portal

The DevTrack*Beta Customer Portal*is a secure, interactive Web site through which selected customer contacts may access customer-specific issues related to development processes.

Customer contacts may use the Beta Customer Portal to create, edit, and delete DevTrack issues through the Internet. Typically, those contacts that are granted access to the Beta Customer Portal represent a*beta group*that is assisting the development team in the development and QA processes.

- Each*beta customer*(company) may access to one or more development projects, depending on their customer type.
- Each contact is generally an employee of a customer and may perform various issue management tasks within the Beta Customer Portal depending on their contact type.

7.1.2 Understanding Beta Customer Portal Security

Beta Customer Portal security is managed using two administrator-defined*accounts*: the customer type and the contact types. Project members may use these administrator-defined accounts to define which projects, pages, and fields customer contacts may access through the Beta Customer Portal.

Using the Customer Manager in a DevTrack client, project members may create customers and customer contacts and assign each customer a customer type and each contact a contact type.

Each customer (a company) created in the Customer Manager may include any number of customer contacts (employees).

- A customer type determines which DevTrack projects a customer may access through Beta Customer Portal. Every customer is assigned a customer type and inherits all of the privileges assigned to that customer type. Administrators define customer types in the Address page of DevTrack Admin.
- A contact type determines whether a customer contact may create, edit, delete DevTrack issues in the Beta Customer Portal. Every contact is assigned a contact type and inherits all of the privileges assigned to that contact type. Contact types are, therefore, similar to the account types used to manage project-level privileges within the internal development team. Contact types are set up with defined privileges, and contacts are assigned contact types that best represent their responsibilities in the development process.

Administrators may use page-level and field-level access rules to add an additional layer of security within the Beta Customer Portal. Administrators may restrict the ability of Beta Customer Portal user to view or edit pages or fields when they create or update DevTrack issues based on their contact type or on the customer type assigned to their employer. Administrators may define page-level and field-level access in the Page and Field Access page. For more information see “Managing Beta Customer Portal Page and Field Access” on page 100.

Project members may use the Customer Manager in the DevTrack client to define customers and customer contacts and to assign administrator-defined customer types to their customers and contact types the contacts.

7.1.3 Understanding the Customer Manager

DevTrack project members may use the Customer Manager to manage customers and customer contacts and the links between customers and DevTrack issues.

Although the administrator defines customer types and contact types in DevTrack Admin, the customers and customer contacts for each project are managed by project members in the DevTrack Windows client.

The Customer Manager consists of the Customer list and four Customer Detail tabs.

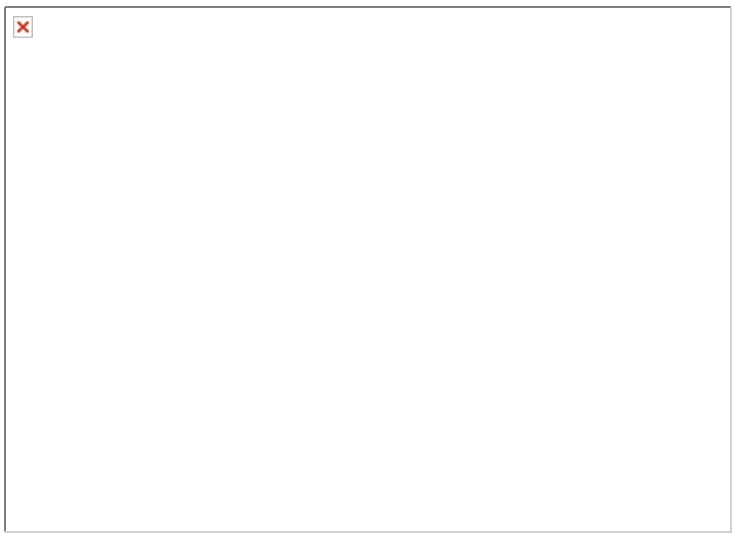


Figure 7-1: Customer Manager in the DevTrack Windows Client

The*Customer list window*displays high-level information about every customer defined in the DevTrack project including their name and their customer type." Project members may filter the Customer List window based on whether a customer is linked to the current project.

The*Customer Detail window*consists of four tabs. Each tab in the Customer Detail window enables project members to manage a customer account.

- The*Customer Info page*enables project members to define basic information about a customer. Administrators may customize the Customer Info page using controls in DevTrack Admin. For more information see “Customizing the Address Page of the Customer Manager”.
- The*Contact page*enables project members to view high-level information about customer contacts. Controls in this tab enable project members to create, edit, and delete customer contacts. Administrators may customize the Add a New Contact page that project members use to create new contacts. For more information see “Customizing Contact Info in the Customer Manager” .
- The*Parent-Child Relation page*enables project members to define which options are displayed in a child multiple-selection list control when the parent multiple- selection list control displays customer type or customer options. For more information see “Defining Parent-Child Relationship Support” .
- The*Related Issues page*enables project members to view issues submitted by a customer.

7.2 Managing Customer Base Projects

Administrators may use the Customer Base Settings page to associate a DevTrack project with a customer base project, create or manage customer base projects, and define general general settings that enable customer contacts to access the Beta Customer Portal.

DevTrack project members may use a*customer base project*to manage all customer-related data including the names, numbers, and addresses of customer contacts. Project members may manage customer data using the Customer Manager in the DevTrack client.

Customer base project management tasks include:

- Understanding Customer Base Projects
- Understanding the Customer Base Settings Page
- Associating Custom Base Projects with DevTrack Projects
- Defining Parent-Child Relationship Support
- Defining Default Subprojects for Customer Base Projects
- Creating New Customer Base Projects

7.2.1 Understanding Customer Base Projects

All DevTrack customer and customer contact data is managed in a customer base project in the DevTrack Windows client. A*customer base project*is a central repository of customer data that that may be shared across multiple projects.

A customer base project enables project teams to share a common database of customers and customer contacts.

- Each DevTrack project may be associated with one customer base project.
- Each customer base project may be used by multiple DevTrack projects as well as TechExcel CRM projects and TechExcel HelpDesk projects.

Figure 7-2: DevTrack Projects and Customer Base Projects

Each DevTrack project (and TechExcel CRM projects and TechExcel HelpDesk projects) may be associated with a single customer base project." A customer base project may, however, store customer information for any number of DevTrack, CRM, or HelpDesk projects.

Administrators may use controls in the Customer Base Setting page of DevTrack Admin to associate a DevTrack project with one customer base project. If the administrator defines multiple customer base projects, each DevTrack project using the Beta Customer Portal may access its own customer base project.

Note: If the DevTrack project shares a customer base project with a CRM or HelpDesk project, customizations to the customer and contact pages must be defined in CRM Admin or HelpDesk Admin. The Address page and Contact Info page in DevTrack Admin are read-only.

7.2.2 Understanding the Customer Base Settings Page

Administrators may use controls in the*Customer Base Settings page*to select a customer base project that is used to manage customer contact information for the DevTrack project.

The Customer Base Settings page enables administrators to select or create customer base projects and to define the general settings that enable the customer contacts to access the DevTrack project issues through the Beta Customer Portal.

Figure 7-3: Customer Base Settings Page

The Customer Base Settings page enables adminsitrators to define general settings for the Beta Customer Portal."

- The Enable Supporting Customer Web Portal option must be selected for customers to access the custom base project through the Beta Customer Portal.
- The Refer Customer As control enables the administrator to change the Customer label used in DevTrack. The text entered in this control is displayed when a team member creates a new customer in the New Customer dialog box and in the Beta Customer Portal.
- The Support Login Alias option enables the administrator to allow customer contacts to log into the Customer Web Portal using either an e-mail address or a login alias. The login alias is defined for each contact in the DevTrack Windows client. If this control is not checked ,all customer contacts must log in using their e-mail address.
- The Multiple Users Can Share the Same E-mail Address option enables the administrator to allow individual customer contacts to share the same e-mail address. If two contacts share the same e-mail address, their login aliases must be different. This option is only available if login aliases are supported.

7.2.3 Associating Custom Base Projects with DevTrack Projects

Administrators may use the Customer Base Settings page to associate a customer base project with the current DevTrack project.

Each DevTrack project may be associated with a single customer base project. DevTrack project members may use the Customer Manager to manage customer data stored in the associated customer base project.

To select a custom base project:

1 Select the Customer Base Setting page in the Beta Customer Portal folder of the Admin Tree window.

2 Click the Select button.

The Select Customer Base Project dialog box appears.

3 Select a project in the Customer Base Project field.

Customer base projects created in TechExcel HelpDesk or TechExcel FrontOffice are displayed in the Project Name field.

4 Click the OK button.

The Select Customer Base Project dialog box closes.

7.2.4 Creating New Customer Base Projects

Administrators may use the New Customer Base Project manager to create new customer base projects.

Administrators may select the Copy Settings from an Existing Project to copy customer base project settings from an existing customer base project. Selecting this option automatically defines customer types and access types in the new customer base project basedon defintions in a selected customer base project.

To define a custom base project:

1 Select the Customer Base Setting page in the Beta Customer Portal folder of the Admin Tree window.

2 Click the Select button.

The Select Customer Base Project dialog box appears.

3 Click the New button.

4 The New Customer Base Project dialog box appears.

5 Enter a title in the Project Name field.

Avoid using the following characters when naming the project/ \ : ? " < > |.

6 To copy settings from an existing project select the Copy Settings from an Existing Project and highlight the project in the project list.

The list of existing customer base projects is only displayed in Customer Base Project field once the Copy Settings from an Existing Project check box is selected.

7 Click the OK button.

7.2.5 Defining Beta Customer Portal Login Settings

Administrators may use controls in the Customer Base Setting page to enable Beta Customer Portal support for a DevTrack project and to define rules for managing customer contact logins to the Beta Customer Portal.

Administrators may define four different settings in the Customer Base Settings page.

- The Enable Beta Customer Portal Setting option enables Beta Customer Portal Support for a DevTrack project.
- The Refer Customer As field enables the administrator to define a name that is used to refer to the customers throughout the Beta Customer Portal. The default name is *Customer*.

- The Login Alias option enables customer contacts to log into the Beta Customer Portal using either an e-mail address or a login alias. Project members may define login aliases for each customer contact in Customer Manager. If this control is not checked ,all customer contacts must log in using their e-mail address.
- The Multiple Users Can Share the Same E-mail option enable multiple customer contacts to share the same e-mail address. If two contacts share the same e-mail address, their login aliases must be different. This option is only available if login aliases are supported.

To enable the Customer Web Portal

1. Select or create a customer base project in the Customer Base Project area of the Customer Base Setting page.

   For more information see “Associating Custom Base Projects with DevTrack Projects” on page 85.

2. Select the Enable Supporting Customer Web Portal check box.

   If this option is not selected, customers cannot access the selected customer base project.

3. Enter a title for the Customer Web Portal in the Refer Customer As field.

   The text entered the Refer Customer As field is displayed in the New Customer dialog box and in the Beta Customer Portal.

4. To enable login aliases, select the Login Alias checkbox in the User Login Control area.

   The Login Alias option enables customer contacts to log into the Beta Customer Portal using either an e-mail address or a login alias. Project members may define login aliases for each customer contact in Customer Manager. If this control is not checked,all customer contacts must log in using their e-mail address.

5. To enable customers to share a single e-mail address, select the Multiple Users Can Share the Same E-mail check box in the User Login Control area.

   This option enables individual customer contacts to share the same e-mail address. If two contacts share the same e-mail address, their login aliases must be different. This option is only available if login aliases are supported.

7.2.6 Defining Parent-Child Relationship Support

Administrators may use controls in the Customer Base Setting page to enable support for a parent-child relationships between the data-entry controls displayed in the DevTrack Customer Manager.

DevTrack project members may create and define customers and customer contacts in the Customer Manager. Using controls in the Parent Child Relation tab of the Customer Manager, project members may determine which options are available for selection in the Applicable Selections area of the Customer Web Portal for each customer.

The Enable Parent-Child Support control enables support for parent-child relationships and allows project teams to determine which options are available to customers in the Beta Customer Portal.

If this option is selected the {Customer} system field may be selected as the parent control for selection list controls displayed in the Description page, and Current Status pages in the DevTrack client.

Administrators may choose between two parent-child relationship options:

- The Based on Specific Customer enables the administrator to define options for each individual customer.
- The Based on Customer Type enables the administrator to define options for each customer type defined in the Type Setup manager.

To enable parent-child relationship support:

1. Select the Enable Parent-Child Relation Support between Customer and Multple Selection Field check box in the Customer Bse Setting page.
2. Choose a parent-child relationship option:
   - To enable administrators to define options for each customer, select the Based on Specific Customer radio button
   - To enable administrators to define options for each customer type, select the Based on Customer Type radio button.

7.2.7 Defining Default Subprojects for Customer Base Projects

Administrators may use controls in the Customer Base Setting page to define a default subproject for the customers created in the selected customer base project.

If subprojects are enabled within the current DevTrack project, every issue submitted to a DevTrack project through the Beta Customer Portal would by default be created in the selected subproject.

The Default Subproject for the Issues Submitted by the Customers dropdown list displays every subproject created in the DevTrack project as an option.

If subprojects are not enabled in the DevTrack project, no option need be selected in this control.

Subprojects enable project teams to divide a project into smaller, more manageable grouping of issues. Like a regular project, a subproject is a tool for organizing and managing DevTrack issues. Development teams may wish to route all issues submitted by beta customers to a common folder where they can be reviewed and routed. For more information see “Understanding DevSuite Subprojects” on page 167.

7.3 Customizing the Address Page of the Customer Manager

Administrators may use the*Address page*to customize the Address page displayed in the Customer Manager of the DevTrack client and to define the Customer Type accounts.

DevTrack project members may use the data-entry controls in these pages to define the customers and the customer contacts who may access the project through the Customer Web Portal.

Note: If the current DevTrack project shares the base project with a TechExcel CRM or TechExcel HelpDesk project, the Address page and the Contact Info page are read-only in DevTrack Admin. All customizations must be managed in FrontOffice or HelpDesk Admin.

Customer and contact management tasks include:

- Customizing the Address Page
- Defining Customer Types
- Customizing the Contact Info Page

7.3.1 Customizing the Address Page

Administrators may use controls in the*Address page*to customize the Address tab displayed in the Customer Manager in the DevTrack client.

Administrators may customize every control displayed in the Address page. The customizations available vary from control to control.

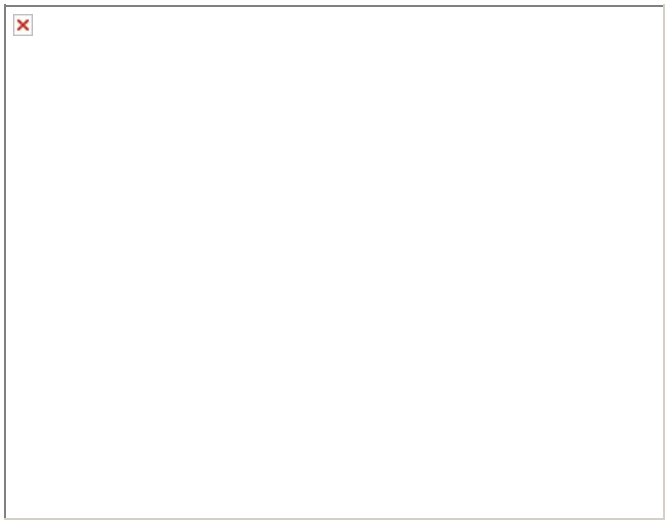


Figure 7-4: Address in DevTrack Admins

In the DevTrack client project members use the controls in the Address page to create and edit customers in DevTrack."

Administrators may customize each of the controls displayed in the Address page. The customizations that the administrator may make vary from control to control. Administrators may make one or more of the following customizations to each control displayed in the Address tab of the Customer Manager:

- *Field Label customizations*: The administrator may change the title of the control displayed in the field labels.
- *Visibility customizations*: The administrator may make many of the controls invisible in the Address page.
- *List options customizations*: The administrator may define the list options displayed in multiple line selection controls.
- *Mandatory on Submission customizations*: The administrator may define certain controls as mandatory. Mandatory controls must be completed for the issue to be submitted.
- *Special customizations*: Administratrators must make special customizations to two controls displayed in the Address page: Customer ID control and the Type control.

Administrators may make the following customizations to each control displayed in the Address tab of the Customer Manager.:

Control	Customizations
Address	Field label
• Name • Address • Zip • Phone1	Field label
• City	List options
• State • Country • Custom Field	• Field label • Visibility • List options
• Fax • Web Address • Phone2 • Cell Phone	• Field label • Visibility
• Customer ID	• Special
• Type	• Special
• Primary Engineer • Secondary Engineer	• Field label • Visibility • Mandatory on submission • List options

Table 7-1: Customer Info Controls

Note: Any addition or changes to available values for the Type, City, State, or Country dropdown fields are at the base project level. These values will be reflected in all DevTrack projects referring to this base project. All other customizations (label changes, etc.) are specific to the project.

7.3.2 Defining Customer Types

Adminsitrators may use the*Customer Type Setup manager*to customize the Customer control in the Address page and to define the customer types that project members may be use to identify different categories of customers in DevTrack projects.

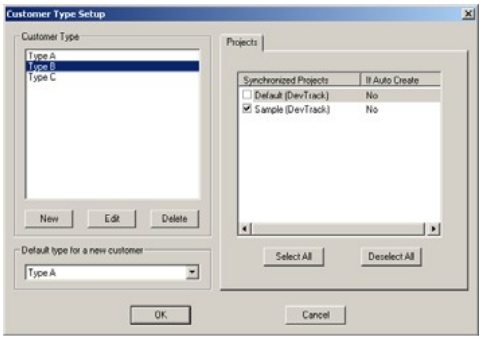


Figure 7-5: Customer Type Setup Manager

A Customer Type account determines the DevTrack project or projects that the contacts belonging to that customer type may access through the Beta Customer Portal.

In the DevTrack client project members may create customers in the Customer manager and assign each customer an administrator-defined customer type. The customer types defined by the administrator in the Type Setup manager are displayed in the Type field of the Address page in the Customer Manager.

The customer type definition enables DevTrack to support the following real world situations:

- Two types of beta customers, Group A testers and Group B testers, are accessing three different development projects. The Group A testers can have access to any of the three development projects, while the Group B testers have access to just one project.
- The Group A testers have edit privileges for all pages and fields.
- The Group B testers have more restricted access and can only view the Description and Current Status page.

Note: No customer contacts may access a DevTrack project through the Customer Web Portal until the customer is linked (automatically or manually) to the DevTrack project.

To create a customer type:

1Double-click the Type control in the Address page.

The Setup UI Field Address dialog box appears.

2Click the Setup button.

The Customer Type Setup manager appears.

3Click the New button.

The New Type dialog box appears.

4Enter a name for the type in the Type Name field.

5Click the OK button.

The New Type dialog box closes.

6Select one or more project in the Project area.

- To select all projects displayed in the Projects area click the Select All button.
- To Deselect all selected projects displayed in the Projects area click the Deselect All button.

7Double-click a selected project in the Projects area.

The Set Property dialog box appears.

8Define project properties for the selected customer type:

- To enable the customer type to access a project select the If Applicable check box.
- To select the Auto Create check box.

9Click the OK button.

The Set Property dialog box closes.

10Select an option from the Default Type for a New Customer dropdown list.

11Click the OK button.

To edit a customer type

1Double-click the Type control in the Address page.

The Setup UI Field Address dialog box appears.

2 Click the Setup button.

The Customer Type Setup manager appears.

3 Select a customer type option in the Customer Type field.

4 Click the Edit button.

The Edit Customer Type field appears.

Update the title of the customer type in the Customer Type Name field.

5 Click the OK button.

The Edit Customer Type field closes.

6 Select one or more project in the Project area.

7 Select an option from the Default Type for a New Customer dropdown list.

8 Click the OK button.

To delete a customer type

1 Double-click the Type control in the Address page.

The Setup UI Field Address dialog box appears.

2 Click the Setup button.

The Customer Type Setup manager appears.

3 Select a customer type option in the Customer Type field.

4 Click the Delete button.

The Warning dialog box appears.

5 Click the Yes button.

The selected customer type is deleted and disappears from the Customer Type field.

7.3.3 Defining Project Linking Options

Administrators may use controls in the Customer Type Setup manager to defined project linking settings between customers and DevTrack projects based on customer types.

Administrators may enable two options for each applicable project:

- The If Applicable option enables project members to link customers belonging to the selected customer type to the selected DevTrack project using the Link control in the Customer Manager. Customers must be manually linked to each applicable project.
- The If Auto Create option automatically links customers belonging to the selected customer type to the selected DevTrack project. Customers are automatically linked to each applicable project based on their customer type.

To define project properties:

1 Double-click a project in the Project tab of the Customer Type Setup manager.

The Set Property dialog box appears.

2 To enable the customer type to access a project select the If Applicable check box.

3 To select the Auto Create check box.

4 Click the OK button.

7.3.4 Customizing the Customer ID Control

Administrators may make numerous customizations to the Customer ID control displayed in the Address page of the Customer Web Portal. Administrators may enable up to five different field attribute options for the Customer ID control displayed in the Address tab of the Customer Manager.

External Customer ID from Third Party Application: The External Customer ID from Third Party Application option makes the Customer ID control read-only in the Customer Manager. Customer ID numbers are not entered manually, but are handled by an external application.

Visible: The Visible option makes the Customer ID control visible to DevTrack project members in the Customer Manager.

Visible to Customers on Web: The Visible to Customers on Web option makes the Customer ID control visible to customer contacts in the Customer Web Portal.

Mandatory on Submission: The Mandatory on Submission makes the Customer ID control mandatory on issue submission.

Use this Field as External Customer ID: The Use this Field as External Customer ID option forces the data entered into the Customer ID control to be unique. If the User this Field as External Customer ID option is not checked, multiple customers may share the same customer ID.

To define Customer ID control settings:

1 Double-click the Customer ID control in the Address page.

The Setup UI Field Address dialog box appears.

2 To update the title of the Customer ID control, enter a name in the Current field.

3 To automatically generate customer ID numbers, select the External Customer ID from Third Party Application check box.

4 To make the Customer ID control visible in the Address tab of the Customer Manager, select the Visible check box.

5 To make the Customer ID control visible in the Customer Web Portal, select the Visible to Customers on Web check box.

6 To make the Customer ID control mandatory, select the Mandatory on Submission check box.

7 To ensure that unique customer ID numbers are created for each customer, select the Use this Field as External Customer ID check box.

8 Click the OK button.

7.4 Customizing Contact Info in the Customer Manager

Administrators may use the Contact Info page in DevTrack Admin to customize the Customer Info tab displayed in the Customer Manager in the DevTrack clients.

Project members may use controls in the Customer Info page to define the customer contacts that may access the DevTrack project through the Beta Customer Portal.

Note: If the current DevTrack project shares the base project with a TechExcel CRM or TechExcel HelpDesk project, the Address page and the Contact Info page are read-only in DevTrack Admin. All customizations must be managed in TechExcel CRM or TechExcel HelpDesk Admin.

Customer and contact management tasks include:

- Customizing the Contact Info Page
- Defining Contact Access Types
- Defining Access Type Privileges

7.4.1 Customizing the Contact Info Page

Administrators may use controls in the Contact Info page to customize the Contact Info tab displayed in the Customer Manager in the DevTrack client. Project members use the controls in the Contact Info page to create and edit customer contacts in DevTrack.

The Contact Info Page shows all of the controls displayed in the Contact Info page displayed in the Customer Manager.

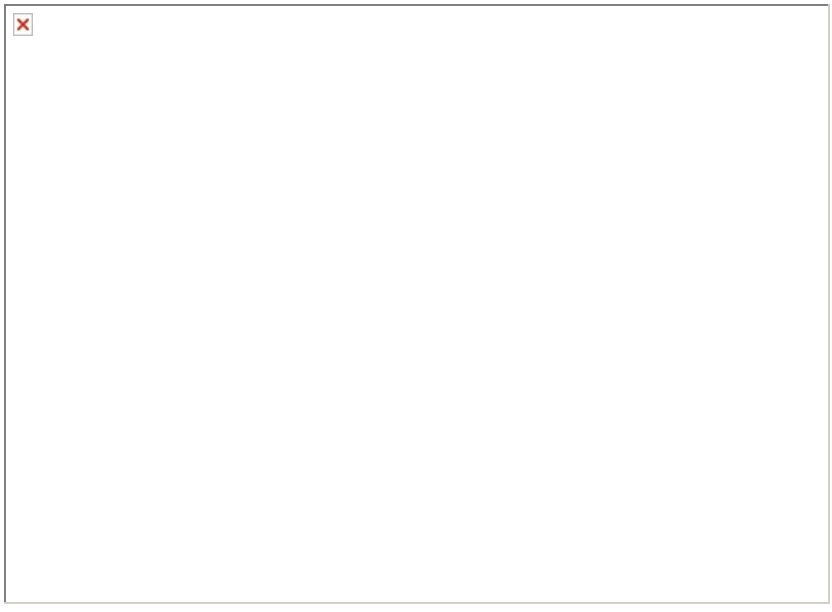


Figure 7-6: Contact Info Page in DevTrack Admin

Administrators may customize each of the controls displayed in the Contact Info page." The Customizations that the administrator may make vary from control to control. Administrators may make one or more of the following customizations to each control displayed in the Contact Info tab of the Customer Manager:

- *Field Label customizations*: The administrator may change the title of the control displayed in the field labels.
- *Visibility customizations*: The administrator may make many of the controls invisible in the Address page.
- *List options customizations*: The administrator may define the list options displayed in multiple line selection controls.
- *Mandatory on Submission customizations*: The administrator may define certain controls as mandatory. Mandatory controls must be completed for the issue to be submitted.
- *Special customizations*: Administrators must make special customizations to two controls displayed in the Address page: Customer ID control and the Type control. All of the controls in the Contact Info page may be customized.The following ustomizations can be made (field names referred to in this section are the default field names):

Control	Customizations
Contact Info	Field label
• Last Name • First Name • Company • Address • Zip • Phone • Fax • Phone 2 • E-mail • Login alias • Password • Active Contact	Field label.
• Title • City • State • Country	• Field label • List options
• State • Country • Custom Field	• Field label • Visibility • List options
• Web Address • Mobile • Pager	• Field label • Visibility
• Access Type	• Special

Table 7-2: Contact Info Controls

7.4.2 Defining Contact Access Types

Administrators may use the Access Type Setup manager to create, edit, and define contact access types. An access type represents a set of privileges that may be assigned to a customer contact in a customer base project.The Access Type control in the Contact Info page of the Customer Web Portal enables team members to differentiate between different categories of contacts.

Figure 7-7: Setup UI Field- Access Type Manager

DevTrack project members may use controls in the Customer Manager to create customer contacts and assign a contact access type to each customer.

Whenever a DevTrack project member creates or edits a customer contact in the Contact Info page of the Customer Manager, that project member assigns an access type to that contact." The access type determines which tasks a customer contact may perform in the Customer Web Portal.

Contact access types are similar to the account types that an administrator assigns to internal DevTrack project members. Access types are set up with defined privileges, and each contact is assigned an access type that best represents that project members responsibilities within the Customer Web Portal.

To create an access type:

1 Double-click the Access Type control in the Contact Info page.

The Setup UI Field: Access Type manager appears.

2 Click the Setup button.

The Access Type Setup manager appears.

3 Click the New button.

The New Access Type dialog box appears.

4 Enter a name for the type in the Access Type Name field.

5 Click the OK button.

The New Access Type dialog box closes.

6Define access type privileges in the Privileges tab.

- To grant the Can Submit New Issue privilege to the access type select the Can Submit New Issue check box.
- To grant the View All Issues within the Customer privilege to the access type select the View All Issues within the Customer check box.

7Define project access by selecting one or more projects in the Project tab.

- To select all projects displayed in the Projects area click the Select All button.
- To Deselect all selected projects displayed in the Projects area click the Deselect All button.

8Double-click a selected project in the Projects area.

The Set Property dialog box appears.

9Define project properties for the selected customer type:

- To enable the customer type to access a project select the If Applicable check box.
- To select the Auto Create check box.

10Click the OK button.

The Set Property dialog box closes.

11Select an option from the Default Type for a New Customer dropdown list.

12Click the OK button.

To edit access types:

1Double-click the Access Type control in the Contact Info page.

The Setup UI Field: Access Type manager appears.

2Click the Setup button.

The Access Type Setup manager appears.

3Select an access type in the Access Type field.

4Click the Edit button.

The Edit Access Type dialog box appears.

5Update the title of the access type in the Access Type Name field.

The Edit Access Type dialog box closes.

To delete access types:

1Double-click the Access Type control in the Contact Info page.

The Setup UI Field: Access Type manager appears.

2Click the Setup button.

The Access Type Setup manager appears.

3Select an access type in the Access Type field.

4Click the Delete button. A warning dialog box appears.

5Click the OK button.

7.4.3 Defining Access Type Privileges

Administrators may use the Privileges tab of the Access Type Setup manager to define the privileges that are granted to each administrator-defined access type.

Can Submit New Issue privilege: The Can Submit New Issue privilege enables contacts of the selected access type to submit new issues.

Can View All Issues Within The Customer privilege: The Can View All Issues Within The Customer privilege enables contacts of the selected access type to view any issue submitted by any contact in their company.

To define access types privileges:

1Double-click the Access Type control in the Contact Info page.

The Setup UI Field: Access Type manager appears.

2Click the Setup button.

The Access Type Setup manager appears.

3Select an access type in the Access Type field.

4Click the Privileges tab.

5Define access type privileges in the Privileges tab.

- To select all privileges displayed in the Privileges area click the Select All button.

- To deselect all selected privileges displayed in the Privileges area click the Deselect All button.

7.4.4 Defining Project Access Controls

Administrators may define project properites for each customer type created in the Type Setup manager. Project properties define the access that each customer type has within a Beta Customer Portal project.

Administrators may grant two privileges for each project:

- The If applicable option enables customers of the selected customer type to access the selected project.
- The If auto-create option automatically links new customers of the selected customer type to the selected project.

A customer must be linked to a project in order for contacts to access the project through the Beta Customer Portal. Without this auto-create option selected, a DevTrack team member must manually link the customer to a DevTrack project before the customer can access it.

No customer contacts can access a DevTrack project until the contact is linked (automatically or manually) to the DevTrack project.

Note: The customer type applicable projects definition overrides the access type applicable projects definition.

7.5 Managing DevTrack Customer Manager Privileges

Administrators may use controls in the Account Type Access Control page to determine which DevTrack project members may create, edit, and delete customers and customer contacts in the Customer Manager.

All Customer Manager access privileges are assigned to DevTrack account types. Account type access privileges enable administrators to control which internal team members may create, delete, and view customer and contact information.

Page-level and field-level access privilege management tasks include:

- Understanding Customer Manager Privileges
- Defining Account Type Privileges

7.5.1 Understanding Customer Manager Privileges

Administrators may use tools in the Account Type Access Control page to assign Customer Manager privileges to DevTrack project account types.

Administrators may grant five different privileges to each account type:

Can Create Customer/Contact Information privilege: The Can Create Customer/Contact Information privilege enables account types to create new customers and customer contacts.

Can Edit Customer Information privilege: The Can Edit Customer Information privilege enables account types to edit customer settings.

Can Delete Customer Information privilege: The Can Delete Customer Information privilege enables account types to delete customers from the project.

Can Associate Customer/contact With An Issue: The Can Associate Customer/Contact With An Issue privilege enables account types to associate a customer type or contact with an issue.

Can View Customer Information privilege: The Can View Customer Information privilege enables account types to view customer related information

The Can View Customer Information privilege overrides all other privileges. If an account type is not granted the Can View Customer Information privilege, project members belonging to that account type may not delete, edit, or create customers, regardless of other settings.

7.5.2 Defining Account Type Privileges

Administrators may use tools in the Account Type Access Control page to assign Customer Manager privileges to DevTrack project account types.

To define account type privileges:

Chapter 8 Interproject Actions and Linking Admini

8 CHAPTER 8 Interproject Actions and Linking Administration

Interproject actions provide development organizations with greater power and flexibility to manage business processes across multiple projects. DevTrack enables development organizations to define and manage two levels of interproject actions: standard interproject actions and advanced interproject actions.

In this chapter:

- Administering Interproject Actions and Linking
- Administering Standard Interproject Actions and Linking
- Administering System Interproject Actions
- Administering Custom Interproject Actions
- Administering Custom Interproject Action Automations

8.1 Administering Interproject Actions and Linking

Interproject actions and linking provide development organizations with greater flexibility and power to manage development issues across multiple projects so that they may close gap between their development processes and other business processes.

- An interproject action is a manual, automated, or semi-automated procedure by which a development issue is copied, cloned, or submitted to an integrated DevTrack, ServiceWise, or CustomerWise project.
- Optional interproject linking associates the development issues in one project with another issue (or incident) in another project. Issues may be automatically linked whenever an issue is copied in one project and submitted to another project.

DevTrack enables development organizations to define and manage two levels of interproject actions: standard interproject actions and advanced interproject actions.

Standard Interproject Actions: Standard interproject actions enable development organizations to submit and copy issues from one project to another project, create links between issues in different projects, and define workflow rules that tie the workflow state of linked issues in different projects. All standard interproject settings are defined in the Advanced Features folder in the DevSuite Admin client.

EnterpriseEdition Actions: EnterpriseEdition interproject actions enable administrators to define system and custom interproject actions to streamline workflow and define best practices for submitting or copying issues between projects. DevTrack Enterprise Edition interproject actions build upon standard interproject action settings.

Interproject actions and linking provide development organizations with the following benefits:

- Interproject issue submission: Project members may submit new issues to a target DevTrack, TechExcel CustomerWise, or TechExcel ServiceWise project from within a DevTrack project.
- Interproject issue copying: Project members may copy issues in a DevTrack project and submit those issues to a target project.
- Interproject linking: Project members may create referential or parent-child links between issues submitted or copied to a target project and issues in the originating DevTrack project.
- Tracking and editing linked issues: Interproject links may enable project members to view, edit, and manage the linked issues from with the DevTrack client.
- Interproject automation rules: The issue workflow state of an issue may drive changes in the issue workflow state a linked issue based on administrator-defined state automation triggers.

8.2 Administering Standard Interproject Actions and Linking

In DevTrack, standard interproject actions and linking enables development teams to submit, copy, and link issues across development projects. Distinct interproject integration rules may be defined for each development project in a DevSuite site.

To enable standard interproject actions, an administrator must define one or more projects as “applicable” target projects, enable interproject actions, enable interproject linking, define applicable link types for target projects, map fields between issues in different projects, and define appropriate workflow automation rules.

A target project is a DevTrack project, TechExcel ServiceWise project, or TechExcel. CustomerWise project that has been identified and enabled to receive new issue submissions through interproject actions. All interproject action settings are defined at the project-level and apply to all enabled interproject actions.

8.2.1 Enabling Project-to-Project Integration

In DevTrack, a DevTrack development project may be integrated with multiple TechExcel DevTrack, ServiceWise, and CustomerWise projects.

All standard interproject integration rules (actions and linking) between the current DevTrack project and another defined on a project-by-project basis.

Distinct interproject workflow rules may be defined for each integration. Enabling project-to-project integration merely allows the administrator to enable support for standard actions and linking. All standard interproject settings must be enabled and defined on a project-by-project basis.

Using controls in the Interproject Setting page, project administrators may enable the current DevTrack project to be enabled by one or more DevTrack, ServiceWise, or CustomerWise projects.

To enable integration with another project:

1. Select Advanced Features > Interproject Setting.
2. Select a “target” project in the Project List.

   Standard interproject integration settings may be defined for a project if a red check mark is displayed next to a project.

8.2.2 Enabling Standard Interproject Action Support

All standard interproject integration rules (actions and linking) between the current DevTrack project and another defined on a project-by-project basis. Distinct interproject workflow rules may be defined for each integration.

The Submit/Copy check box must be selected for project members to submit or copy issues from the DevTrack project to the target project.

To enable interproject actions between two projects:

1. Select Advanced Features > Interproject Settings of the tree panel.
2. Select an enabled target project in the Project list.
3. Select the Interproject Submit/Copy check box.

8.2.3 Enabling Interproject Link Support

All standard interproject integration rules (actions and linking) between the current DevTrack project and another defined on a project-by-project basis. Distinct interproject workflow rules may be defined for each integration.

The Link check box must be selected for project members to submit or copy issues from the DevTrack project to the target project.

To enable interproject linking between two projects:

1. Select Advanced Features > Interproject Settings of the tree panel.
2. Select an enabled target project in the Project list.
3. Select the Interproject Link check box.

8.2.4 Defining Interproject Standard Action Settings

An interproject action is a procedure by which a work item (issue or incident) created in one project is submitted to another project.

A standard interproject action is defined by a*target project account type*,*default owner*,*title prefix*, and a*default issue template*. Standard interproject action rules define who may submit and own issues submitted by interproject actions, the templates that may be used to create those issues, and how those issues may be identified in the target project.

Using controls in the General Setting tab, the project administrator may enable project-to-project integration, enable interproject action support, and define interproject action settings.

Figure 8-1: State Automation Tab in the Interproject Actions Page

To enable interproject linking between two projects:

1. Select Advanced Features > Interproject Settings of the tree panel."
2. Select an enabled target project in the Project list.
3. Select the Interproject Submit/Copy check box.

   The Submit/Copy check box must be selected for project members to submit or copy issues from the DevTrack project to the target project.
4. Select a target project account type in the Default Account Type dropdown list.

   The Default Account Type dropdown list displays account types available in the*target project*.

   The*target project account type*enables project members that are not members of the target project to submit, copy, or link issues between their DevTrack project and the target project. If the user that submits an issue to the target project by an interproject action is not a member of the target project, that user is treated as though they were assigned the target project account type in that project.
5. Select a user account or group folder in the Default Folder/Members To Be Assigned dropdown list.

   The Default Folder/Members To Be Assigned dropdown list displays user accounts and group folders in the integrated project.

   The default owner is a project team member or group folder that is designated as the owner of issues copied or submitted to that project by an interproject action.
6. Define a default title prefix in the Title Prefix Concatenated text box.

   Concatenated title prefixes enable project teams to quickly identify issue that have been submitted from other projects. Project members may edit the title prefix when they submit or copy an issue in the DevTrack client.
7. Select an issue template or incident template in the Default Issue Template dropdown list.

   The Default Issue Template dropdown list displays the issue templates or incident templates available in the*target project*—issue templates for integrated DevTrack projects; incident templates for integrated CustomerWise or ServiceWise projects.

   The selected template serves as the default template for all work items submitted to the target project from the current project by an interproject action. For more information on issue templates see “Administering Issue Templates” on page 119.

8.2.5 Defining Interproject Editing Rules

Interproject links define relationships between issues in different DevTrack projects, or between DevTrack issues and incidents in TechExcel ServiceWise and TechExcel CustomerWise projects. Project members may create links between issues whenever they submit or copy issues from a DevTrack project and a target project.

Interproject linked issue editing rules determine which changes project members may make to issues in other projects that are linked to their project through an interproject link.

Project members may make the following edits to linked issues in target projects:

- Change the issue workflow state of linked issues.
- Change the owner of the linked issues.
- Edit the title and description of linked issues.
- Edit the work description of linked issues.

Project administrators may enable or restrict the editing of linked issues on a project-by-project basis.

To define interproject editing rules:

1. Select Advanced Features > Interproject Settings of the tree panel.
2. Select an enabled target project in the Project list.
3. Select the Interproject Link check box in the General Setting tab.

   The Link check box must be selected for project members to link issues between the current DevTrack project to the target project.

4. **Optional**: To enable project team members to update the issue workflow state of the issue, select the Can Edit Progress Status check box.
5. **Optional**: To enable project team members to update the issue owner of the issue, select the Can Edit Owner check box.
6. **Optional:** To enable project members to edit the work description of a linked issue, select an option from the Can Edit Work Description dropdown list.

   The Can Edit Work Description dropdown list displays two options:

   - The Edit Work Description with Timestamp option enables project members to add text to the work description. All additions are accompanied by a the date and time of the change.
   - The Edit Work Description Freely option enables project members to edit all text in the work description.

7. **Optional:** To enable project members to edit the title and issue description of a linked issue, select an option from the Title and Description dropdown list.

   The Title and Description dropdown list displays two options:

   - The Read-only option makes the linked issue Title and Description fields read-only.
   - The Edit Description with Timestamp option makes the Description field editable, but timestamps all changes.
   - The Edit Description Freely option makes the Description field editable.

8.2.6 Defining Interproject Field and Field Value Mapping

To enable the submission, copying, and linking of issues across heterogeneous systems, project administrators must first map the fields in one project to fields in other projects. Once the two fields are mapped, the data that is tracked in the mapped field in one project is equated with the data that is tracked in the mapped field in the other project.

Field mapping and field value mapping are prerequisites for interproject copying, interproject submission, and interproject state automation rules between*heterogeneous projects*—projects in which work items are defined differently.

- Interproject copying
- Interproject state automation

Each DevTrack project may use distinct pages, controls, and fields to manage and track development data. And TechExcel ServiceWise projects and TechExcel CustomerWise projects are designed to enable businesses to track entirely different types of data.

Issue field mapping and field value mapping enable development organizations to equate the issue properties tracked in one project with the issue properties (or incident properties) tracked in another project.

The following rules apply to mapping fields across projects:

- Mapped fields must belong to the same data type.
- Mapped fields are defined on a project-by-project basis. The same field may be mapped to different fields in different projects.

To define interproject field and field value mappings:

1. Select Advanced Features > Interproject Settings of the tree panel.
2. Select an enabled target project in the Project list.
3. Select the Field Map tab.
4. Map the fields that are used to define issue properties in each project.

- To manually map a DevTrack project field to a target project field, select a field in the Field Map tab and click the Map button. Identify the target project page and field in the dialog box.
- To automatically map DevTrack project fields to a target project fields, click the Auto Map button. You may need to manually edit interproject field mappings.

5. Select the Field Choice Map tab.
6. Select a mapped multiple choice control in the Mapped Field with Multiple Choice list.

   The Mapped Field with Multiple Choice list displays every selection control in the current project that is mapped to a multiple choice control in the target project. The selection control selected determines the options that are displayed in the Map Field Choice Map list.

7Double-click a field choice in the Map Field Choice Map list.

The Choice Map dialog box appears.

8Select an option from the Map To dropdown list.

The options displayed in the Map To dropdown list represent the list options displayed in the mapped multiple choice control in the target project.

9Click the OK button.

The Choice Map dialog box closes.

8.2.7 Defining Issue History for Linked Issues

In DevTrack, the Issue History report displays the history of actions performed on an issue in a DevTrack project or other TechExcel project. Using controls in the History tab of the Interproject Settings page, project administrators may define which sections of the history report are displayed in the Linked Issue Info tab of the DevTrack client.

The history of the original issue in a DevTrack, ServiceWise, or CustomerWise project may be copied to the history report of a copied issue and displayed in the linked target project. Issue history inheritance rules are defined in the Issue History tab of the Interproject Setting page.

Using controls in the History tab, project administrators may determine which sections of the History report are copied over to the new issue and are displayed in the DevTrack client.

To display issue history of linked issues:

1Select Advanced Features > Interproject Settings of the tree panel.

2Select an enabled target project in the Project list.

3Select the Interproject Link check box in the General Setting tab.

Standard interproject linking must be enabled to view linked issue histories.

4Add or remove issue history report sections in the History tab.

- To add a section to the History report select an option in the Available History Sections list and click the Left Arrow button.
- To remove a section from the History report select an option in the History Sections to Display list and click the Left Arrow button.

5 Optional: To change the display order of the sections in the report, select a section and click the Up button or the Down button.

8.2.8 Defining Standard Interproject State Automation Rules

An interproject state automation rule is defined by its title, an*issue workflow state*, a*state automation rule condition*, a*target state*, a*default owner*in the target state, and one or more*link type conditions*.

Using controls in the State Automation tab, project administrators may define triggers that change the issue workflow state of linked issues (in the target project) based on changes made to linked issues in the current DevTrack project.

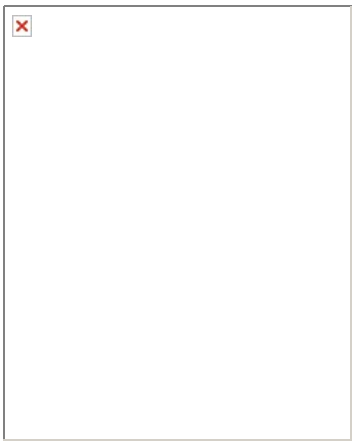


Figure 8-2: State Automation Tab in the Interproject Actions Page

The State Automation tab enables administrators to define triggers for each workflow state in the target project."

The state automation rules defined are applied to every link type selected in the list. The Applicable Link Type area displays every link type defined within the DevTrack project

To defining an interproject state automation rule:

1Select Advanced Features > Interproject Settings.

2Define or edit interproject integration settings.

3Enable support for interproject issue automationt.

4 Optional: To enable cascading interproject copying, select the Enable Cascaded Interproject Copy Automation check box.

5Click the Add button.

The Interproject State Automation Rule dialog box appears.

6Define the title of the interproject state automation rule in the Rule Name text box.

7Select an issue workflow state in the State Changed To list.

The option selected represents the workflow state that the issue in the target project is changed to when the linked issue in the current project is changed to an administrator-defined trigger state.

8 Optional: To define the scope of the interproject state automation rule, select an interproject state automation rule condition in the Condition dropdown list.

9Select a trigger state in theTriggerStatedropdown list.

The option selected represents the workflow state of the issue in the current project. When the issue in the current project is changed to the selected trigger state, the linked issue in the target project is change to the administrator-defined state.

10Select an issue workflow state in theTargetStatedropdown list.

TheTargetStatedropdown list displays the workflow states that define the issue lifecycle in the target project. The target state is the workflow state assigned to the issue that is copied or submitted to the target project.

11Select a target default owner from the Target Default Owner dropdown list.

The Target Default Owner dropdown list displays the user accounts and group folders that may own issues in the target project. The target default owner identifies the project team member or group folder that by default is designated as the owner of the issue that is copied or submitted to the target project.

The option selected represents the default owner of the linked issue when the issue undergoes a state change.

12To define the scope of the rule by issue type, select one or more link types in the Applicable Link Type area.

In DevTrack, a link types is a set of business rules that defines the relationship between two linked development issues. A*link type condition*defines the scope of issues that are subject to the state automation rule.

13 Optional: To define closed issues applicable to the interproject state automation rule, select the Include Closed Issues check box.

8.2.9 Defining Interproject State Automation Rule Conditions

An interproject state automation rule condition defines the scope of an interproject state automation rule by identifying the development issues that are subject to that rule based on issue property values.

A condition is an administrator-defined query that “filters” development issues based on issue properties. Development issues not returned by the query are not subject to the business rule.

Using controls in the Interproject State Automation Rule window, project administrators may define and manage any number of interproject state automation conditions.

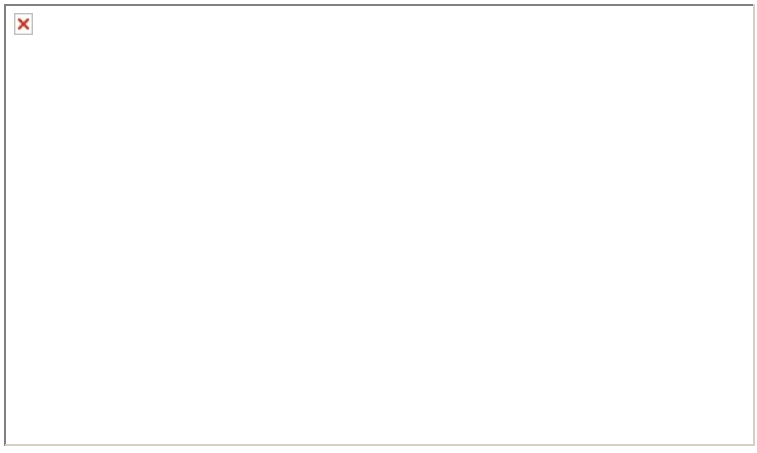


Figure 8-3:InterprojectStateAutomation Rule Condition Window

Every query consists of one or more*search conditions*, which identify the data fields searched and the field value criteria for each field." Multiple search conditions may be organized into*search terms*and filtered by using the AND, NOT, and OR logical operators.

Every tab is organized into four distinct areas: the Field list, Field Values area, Search Condition list, and Search Logic control.Using the controls in each area, project members may define the three components of a DevTrack query: search conditions, search terms, and search logic.

Condition List: In the Field List area the user may select the field (usually an issue property) to be queried.

Field Value: In the Field Value area the user may identify the field values (issue property definitions) that define the*search condition*.

Search Condition: The Search Condition displays all user-defined search conditions and organizes those search conditions into discreet groups called*search terms*.

Search Logic: In the Search Logic area the user may define logical relationships between search terms using three operators: AND, OR, and NOT.

An interproject state automation rule condition may be used to define the scope of many different interproject state automation rules. Although all interproject state automation rules are defined on a project-by-project basis, the conditions that define the scope of these rules are applicable to all rules regardless of project.

Unlike standard database queries, which require knowledge of a search language— such as SQL—to define the search criteria, DevTrack enables the user to define complex

queries that incorporate multiple conditions and search logic using controls in a graphical interface.

To define an interproject state automation rule condition:

1 Select Advanced Features > Interproject Settings.

2 Define an interproject state automation rule.

3 Click the Condition button in the Interproject State Automation Rule window.

The Interproject State Automation Condition window appears.

4 Define or edit an interproject state automation rule condition.

- To define a new rule condition, click the New button and define the title of the rule condition.
- To edit an existing rule condition, select a rule and click the Edit button. Redefine the title in the dialog box.

5 Define one more search conditions based on system fields and custom fields.

- To define a search condition based a system field, select a field in the System Field tab and define search criteria in the Field Value area.
- To define a search condition based a custom field, select a field in the Custom Field tab and define search criteria in the Field Value area. Search conditions are represented by capital letters in the Search Condition area.

6 Define the logical relationships between search condition criteria in the Search Logic Area.

DevTrack supports three logical operators: the AND operator, the OR operator, and the NOT operator. Project members may also use parentheses (the grouping function) to define complex search conditions.

7 Click the Close button.

The Interproject Automation Condition window closes.

8.3 Administering System Interproject Actions

Enterprise Edition interproject actions enable administrators to predefine system and custom interproject actions to streamline workflow and define best practices for submitting or copying issues between projects.
DevTrack Enterprise Edition interproject actions build upon standard interproject action settings.

DevTrack supports three system-defined interproject actions: clone, interproject submit, and interproject copy actions.

- A clone interproject action defines rules for cloning an existing development issue and submitting a copy of that issue to another DevTrack development project.
- A submit interproject action defines rules for submitting a new development issue to another DevTrack development project.
- A copy interproject action defines rules for copying an existing development issue and submitting a copy of that issue to another DevTrack development project.

Using controls in the Define System Actions page, project administrators may create, edit, delete clone, copy, and submit interproject actions.

Figure 8-4: Define System Actions Page

Every system interproject action is defined by its title, one or more applicable projects, and, optionally, one or more link types."

8.3.1 Defining Clone System Interproject Actions

A clone interproject action defines rules for cloning an existing development issue and submitting a copy of that issue to another DevTrack development project.

To define clone system interproject actions:

1 Select Enterprise Edition Features > Interproject Actions > System Actions in the tree panel.

2 Select the Interproject Clone action in the list.

3 **Optional:** To update the name of the system action, enter a name in the Action Name control of the Properties tab.

4 **Optional:** To update the name of the system action, enter a name in the Action Name control of the Properties tab.

5 **Optional:** To define a brief description of the system action, enter text in the Action Descriptions control.

6 Enable interproject actions and links between the current project and a target project by selecting that project in the tree menu.

Target projects are only displayed in the tree menu if they have been mapped in the Interproject Action Settings page of the Advanced Settings folder.

7 Define the applicable link types for the selected action.

- To define all link types as applicable to the system action, select the Any radio button.
- To specify the applicable link types for the system action, select the Selected radio button and one or more link types or links in the Link Type tree control.

8.3.2 Defining Submit System Interproject Actions

A submit interproject action defines rules for submitting a new development issue to another DevTrack development project.

To define submit system interproject actions:

1 Select Enterprise Edition Features > Interproject Actions > System Actions in the tree panel.

2 Select the Interproject Copy action in the list.

3 **Optional:** To update the name of the system action, enter a name in the Action Name control of the Properties tab.

4 **Optional:** To update the name of the system action, enter a name in the Action Name control of the Properties tab.

5 **Optional:** To define a brief description of the system action, enter text in the Action Descriptions control.

6 Enable interproject actions and links between the current project and a target project by selecting that project in the tree menu.

Target projects are only displayed in the tree menu if they have been mapped in the Interproject Action Settings page of the Advanced Settings folder.

7 Define the applicable link types for the selected action.

- To define all link types as applicable to the system action, select the Any radio button.
- To specify the applicable link types for the system action, select the Selected radio button and one or more link types or links in the Link Type tree control.

8.3.3 Defining Copy System Interproject Actions

A copy interproject action defines rules for copying an existing development issue and submitting a copy of that issue to another DevTrack development project.

To define copy system interproject actions:

1 Select Enterprise Edition Features > Interproject Actions > System Actions in the tree panel.

2 Select the Interproject Copy action in the list.

3 **Optional:** To update the name of the system action, enter a name in the Action Name control of the Properties tab.

4 **Optional:** To define a brief description of the system action, enter text in the Action Descriptions control.

5 Enable interproject actions and links between the current project and a target project by selecting that project in the tree menu.

Target projects are only displayed in the tree menu if they have been mapped in the Interproject Action Settings page of the Advanced Settings folder.

6 Define the applicable link types for the selected action.

- To define all link types as applicable to the system action, select the Any radio button.
- To specify the applicable link types for the system action, select the Selected radio button and one or more link types or links in the Link Type tree control.

8.4 Administering Custom Interproject Actions

Enterprise Edition interproject actions enable administrators to predefine system and custom interproject actions to streamline workflow and define best practices for submitting or copying issues between projects. DevTrack Enterprise Edition interproject actions build upon standard interproject action settings.

A custom interproject action is an automated or semi-automated procedure by which a development issue is copied, cloned, or submitted to an integrated DevTrack, ServiceWise, or CustomerWise project.

Every custom interproject action is defined by a target project, a *target subproject*, an *issue type*, *title prefix*, account type-based privileges, one or more applicable states, and, optionally, a *link type*. DevTrack supports both *automatic* and *interactive* custom interproject actions.

Using controls in the Define System Actions page, project administrators may create, edit, delete clone, copy, and submit interproject actions.

Figure 8-5: Define custom Interproject Actions Page

Custom interproject actions may automated or semi-automated (interactive) processes:

Automated interproject actions: An automated interproject action is automatically submitted whenever the rule is triggered."

Interactive interproject actions: An interactive interproject action prompts the user to submit an action whenever the rule is triggered.

If a custom interproject action is an automated action, the default issue template and other settings cannot be overridden by the user.

8.4.1 Adding, Editing, and Deleting Custom Interproject Actions

Using controls in the Define Interproject Actions page, project administrators may create any number of custom interproject actions and freely edit or delete interproject actions and the rules that define those actions.

To add a custom interproject action:

1Select Enterprise Edition Features > Interproject Actions > Define Interproject Actions in the tree panel.

2Click the Add button.

The Add Interproject Action dialog box appears.

3Define a unique name for the interproject action in the Action Name control.

4Click the OK button.

The Add Interproject Action dialog box closes.

To edit a custom interproject action:

1Select Enterprise Edition Features > Interproject Actions > Define Interproject Actions in the tree panel.

2Select an interproject action in the Interproject Action list.

3Click the Delete button.

The Edit Interproject Action dialog box appears.

4Define a unique name for the interproject action in the Action Name control.

5Click the OK button.

The Edit Interproject Action dialog box closes.

To delete a custom interproject action:

1Select Enterprise Edition Features > Interproject Actions > Define Interproject Actions in the tree panel.

2Select an interproject action in the Interproject Action list.

3Click the Delete button.

A DevTrack dialog box appears.

4Click the Yes button.

The interproject action is deleted.

8.4.2 Defining Custom Interproject Actions

In DevTrack, a custom interproject action is an automated or semi-automated procedure by which a DevTrack development issue is submitted to another DevTrack development

projects, TechExcel ServiceWise projects, or TechExcel CustomerWise project.

Every custom interproject action is defined by a target project, a*target subproject*, an*issue type*,*title prefix*, account type-based privileges, one or more applicable states, and, optionally, a*link type*. DevTrack supports both*automatic*and*interactive*custom interproject actions.

To define a custom interproject action:

1. Select Enterprise Edition Features > Interproject Actions > Define Interproject Actions in the tree panel.
2. Select an interproject action in the Interproject Action list.
3. Select the Properties tab.
4. **Optional:** To provide a brief description, enter text in the Action Description control.
5. Select an option from the Target Project dropdown list control.
6. Select a subproject in the Target Subproject dropdown list control.
7. Select an issue type in the Select Issue Type dropdown list control.
8. Define a default title prefix in the Title Prefix Concatenated field.

   Concatenated title prefixes enable project teams to quickly identify issue that have been submitted from other projects. Project members may edit the title prefix when they submit or copy an issue in the DevTrack client.
9. **Optional:** To enable issue linking and to define the link type used to define the relationship between linked issues, select the Enable Link Type check box and select an option from the Select Link Type dropdown list control.

   The Link Type dropdown list displays the link types defined in the Define Link Type manager. For more information see “Administering Issue Types”.
10. Define the relationship between the DevTrack issue and the linked issue or incident in the other project in the Create New Issue and Link As area:

    - To define the new issue as the parent issue in the link, select the Parent radio button.
    - To define the new issue as the child issue in the link, select the Child radio button.
    - To define a referential link between the new issue and the original issue, select the Referential radio button.
11. Select an option from the Select a New Action dropdown list control.

    The Select a New Action dropdown list displays a list of the “actions” (transitions) that connect the {New} issue workflow state with other workflow states in the target project. The options displayed in the Select a New Action dropdown list depend on the transitions defined in the project.
12. Select an option from the Assign To dropdown list control.

    The Select a Assign To dropdown list displays the names of project members that may own the submitted issue.
13. Select one or more issue workflow states in the Applicable States tab.

8.4.3 Defining Custom Interproject Action Access Controls

To define access controls for a custom interproject action:

1. Select Enterprise Edition Features > Interproject Actions > Define Interproject Actions in the tree panel.
2. Select an interproject action in the Interproject Action list.
3. Select the Who Can Change tab.
4. Select one or more account types in the Who Can Perform list.

8.4.4 Defining Custom Interproject Action Applicable States

To define applicable states for a custom interproject action:

1. Select Enterprise Edition Features > Interproject Actions > Define Interproject Actions in the tree panel.
2. Select an interproject action in the Interproject Action list.
3. Select the Applicable State tab.
4. Select one or more workflow states in the Applicable States list.

8.5 Administering Custom Interproject Action Automations

DevTrack interproject automation defines rules for performing a custom interproject action whenever specific changes are made to a development issue in an issue workflow state or transition.

A custom interproject action is an automated or semi-automated procedure by which a DevTrack development issue is submitted to another DevTrack development projects, TechExcel ServiceWise projects, or TechExcel CustomerWise project.

Interproject action automation rules define conditions and rules for triggering custom interproject actions based on changes to the status of development issues in workflow.

DevTrack supports both transition-dependent and state-dependent interproject automation rules.

- Transition-dependent interproject action automation rules are triggered when a development issue undergoes a specific transition. Transition-dependent rules may be defined

in projects using transition-based workflow.

- State-dependent interproject action automation rules are triggered when a development issue enters a specific workflow state. State-dependent rules may be defined in projects using either transition-based workflow or state-based workflow.

Every interproject action automation rule is defined by its title, an interproject*automation condition*, and an i*nterproject action*.

The scope of an interproject action automation rule may be defined by an interproject automation condition. An interproject automation condition is an administrator-defined query that “filters” development issues based on issue properties.

8.5.1 Enabling Interproject Action Automation in Transition-Based Projects

In projects using transition-based workflow, project administrators may enable support for transition-dependent or state-dependent interproject action automation.

- Transition-dependent interproject action automation rules are triggered when a development issue undergoes a specified transition. Transition-dependent rules may be defined in projects using transition-based workflow.

- State-dependent interproject action automation rules are triggered when a development issue enters a specified workflow state. State-dependent rules may be defined in projects using either transition-based workflow or state-based workflow.

Once interproject action automation is enabled, the project administrator may define interproject automation rules in the Interproject Action Automation tab of the Workflow Settings page.

To enable interproject action automation in a transition-based project:

1Select Workflow > Workflow Options in the tree panel.

2Select the Based onTargetStateoption button.

8.5.2 Enabling Interproject Action Automation in State-Based Projects

In projects using state-based workflow, project administrators may enable support for state-dependent interproject action automation. State-dependent interproject action automation rules are triggered when a development issue enters a specified workflow state.

State-dependent rules may be defined in projects using either transition-based workflow or state-based workflow.

Once interproject action automation is enabled, the project administrator may define interproject automation rules in the Interproject Action Automation tab of the Workflow Settings page.

To enable interproject action automation in a state-based project:

1Select Workflow > Workflow Options in the tree panel.

2Select a triggering option in the Interproject Action Automation Triggering area.

- To disable interproject automation, select the
- To enable transition-dependent interproject automation, select the Based on Transition option button
- To enable state-dependent interproject automation, select the Based onTargetStateoption button.

8.5.3 Defining State-Dependent Interproject Automation Rules

A state-dependent interproject action automation rule is defined by its title, an*interproject automation condition*, an i*nterproject action*, and an issue workflow state.

State-dependent interproject automation rules are triggered whenever a development issue meeting the rule conditions enters a specific workflow state.

Using controls in the Interproject Automation tab, project administrators may define the issue workflow state, title, interproject automation condition, and interproject action for each interproject automation rule.

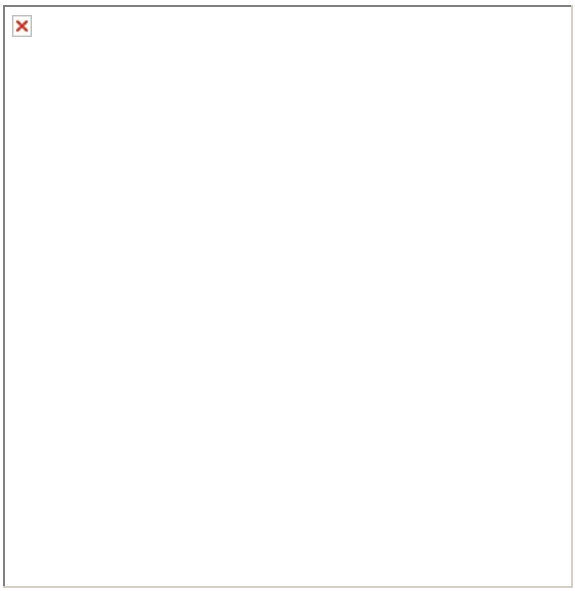


Figure 8-6: Interproject Automation Tab

To define a state-dependent interproject automation rule:

1. Select Workflow > Workflow Settings."
2. Select a workflow state in the States list.
3. Click the New Rule button.

   The Define Interproject Action Automation Rule window appears.

4. Define the title of the rule in the Name text box.
5. Select an interproject automation condition in the Condition dropdown list.

   The Condition dropdown list displays all administrator-defined interproject automation conditions. An interproject automation condition is an administrator-defined query that "filters" development issues based on issue properties.

6. **Optional**: To define an interproject automation condition, click the Setup button and define a query in the Interproject Automation Condition window.

   For step-by-step instructions see "Defining Interproject Action Automation Rule Conditions".

7. **Optional**: To enable the rule to triggered by an automated action, select the Allow Rule to Be Triggered by Automated Actions check box.
8. Select one or more interproject actions in the Interproject Actions to Perform list.
9. Click the OK button.

   The Define Interproject Action Automation Rule window closes.

8.5.4 Defining Transition-Dependent Interproject Automation Rules

A transition-dependent interproject action automation rule is defined by its title, an*interproject automation condition*, an i*nterproject action*, and a transition.

Transition-dependent interproject automation rules are triggered whenever a specific action is performed on a development issue meeting the rule conditions.

Using controls in the Interproject Automation tab, project administrators may define the issue workflow state, title, interproject automation condition, and interproject action for each interproject automation rule

To define a transition-dependent interproject automation rule:

1. Select Workflow > Workflow Settings.
2. Select a workflow state in the States list.
3. Click the New Rule button.

   The Define Interproject Action Automation Rule window appears.

4. Define the title of the rule in the Name text box.
5. Select an interproject automation condition in the Condition dropdown list.

   The Condition dropdown list displays all administrator-defined interproject automation conditions. An interproject automation condition is an administrator-defined query that "filters" development issues based on issue properties.

6. **Optional**: To define an interproject automation condition, click the Setup button and define a query in the Interproject Automation Condition window.

   For step-by-step instructions see "Defining Interproject Action Automation Rule Conditions".

7. **Optional**: To enable the rule to triggered by an automated action, select the Allow Rule to Be Triggered by Automated Actions check box.
8. Select one or more interproject actions in the Interproject Actions to Perform list.
9. Click the OK button.

   The Define Interproject Action Automation Rule window closes.

8.5.5 Defining Interproject Action Automation Rule Conditions

An interproject action automation rule condition defines the scope of an interproject action automation rule by identifying the development issues that are subject to that rule based on issue property values.

A condition is an administrator-defined query that "filters" development issues based on issue properties. Development issues not returned by the query are not subject to the business rule.

Using controls in the Interproject Action Automation Rule window, project administrators may define and manage any number of rule conditions.

Figure 8-7: Interproject Action Automation Rule Condition Window

Every query consists of one or more*search conditions*, which identify the data fields searched and the field value criteria for each field." Multiple search conditions may be organized into*search terms*and filtered by using the AND, NOT, and OR logical operators.

Every tab is organized into four distinct areas: the Field list, Field Values area, Search Condition list, and Search Logic control.Using the controls in each area, project members may define the three components of a DevTrack query: search conditions, search terms, and search logic.

Condition List: In the Field List area the user may select the field (usually an issue property) to be queried.

Field Value: In the Field Value area the user may identify the field values (issue property definitions) that define the*search condition*.

Search Condition: The Search Condition displays all user-defined search conditions and organizes those search conditions into discreet groups called*search terms*.

Search Logic: In the Search Logic area the user may define logical relationships between search terms using three operators: AND, OR, and NOT.

An interproject action automation rule condition may be used to define the scope of many different interproject action automation rules. Although all interproject action automation rules are defined on a project-by-project basis, the conditions that define the scope of these rules are applicable to all rules regardless of project.

Unlike standard database queries, which require knowledge of a search language— such as SQL—to define the search criteria, DevTrack enables the user to define complex queries that incorporate multiple conditions and search logic using controls in a graphical interface.

To define an interproject action automation rule condition:

1Select Workflow > Workflow Settings.

2Define or edit an interproject action automation rule.

3Click the Setup button in the Interproject Action Automation Rule window.

The Interproject State Automation Condition window appears.

4Define or edit an interproject action automation rule condition.

- To define a new rule condition, click the New button and define the title of the rule condition.
- To edit an existing rule condition, select a rule and click the Edit button. Redefine the title in the dialog box.

5Define one more search conditions based on system fields and custom fields.

- To define a search condition based a system field, select a field in the System Field tab and define search criteria in the Field Value area.
- To define a search condition based a custom field, select a field in the Custom Field tab and define search criteria in the Field Value area.

Search conditions are represented by capital letters in the Search Condition area.

6Define the logical relationships between search condition criteria in the Search Logic Area.

DevTrack supports three logical operators: the AND operator, the OR operator, and the NOT operator. Project members may also use parentheses (the grouping function) to define complex search conditions.

7Click the Close button.

The Interproject Automation Condition window closes.

Chapter 9 Product, Version, and Build Administrat

9 CHAPTER 9 Product, Version, and Build Administration

This chapter describes the product tree and how the product, version, and build structures defined within the product tree may be used to define subprojects and branch management processes in DevTrack.

In this chapter:

- Understanding Product Version Tree Administration
- Administering Milestone Date Types
- Administering the Product Tree
- Administering Applicable Products, Versions, and Builds
- Administering Product Categories
- Administering Products
- Administering Versions
- Administering Builds

9.1 Understanding Product Version Tree Administration

In DevSuite, the product tree is a hierarchical structure that provides a business with a framework for organizing the products, versions, and builds under development in every project within the DevSuite site.

The product tree organizes products, versions, and builds into a hierarchical tree structure composed of multiple folders and subfolders.

In a DevTrack development project, the product tree defines the framework in which development issues and branch issues are managed in the project. Each subproject in a DevTrack project is defined as a release folder and linked to a specific product, version, or build.

To use the product, version, and build structures, the project administrator must identify which products, versions, and builds in the product tree are applicable to their DevTrack project. Only those release folders that have been defined as “applicable” may be utilized in the DevTrack project.

Once a product, version, or build has been defined as applicable to a DevTrack project, project members may create subproject folders in the DevTrack client to represent each of those releases.

9.2 Administering Milestone Date Types

A milestone is a point in time that represents an important intermediate event in a project. Milestones commonly represent the completion of key components, reporting requirements, and the beginning or end of project phases.

All version releases may represent one or more milestones. Typical milestones that a product administrator may want to create include:

- Alpha date
- Beta date
- Release date
- Code Freeze date
- Preproduction date
- Feature Freeze date

In DevTrack, every milestone is based on a milestone date type. A milestone date type is an administrator-defined template for creating appropriate milestones in a DevTrack site. All milestone date types are defined and managed in the product tree.

Note: The creation and management of milestones is particularly important to organizations that are using DevPlan to manage development planning.

System administrators may create, edit, and delete milestone date types in Milestone Date Type manager.

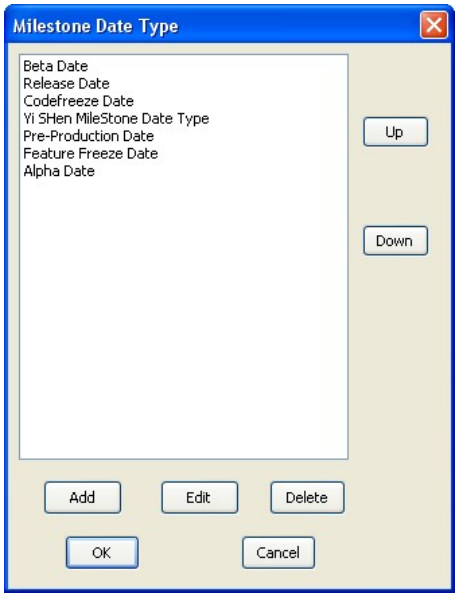


Figure 9-1: Milestone Date Type Manager

9.2.1 Adding Milestone Date Types

To add milestone date types:

1. Select the Define Milestone Date Type command in the Tool menu of the DevSuite Admin client.

   The Milestone Date Type manager appears.

2. Click the Add button.

   The Add Milestone Date Type dialog box appears.

3. Define a unique name for the milestone date type in the Milestone Date Type Name control.

4. Click the OK button.

9.2.2 Editing Milestone Date Types

To edit milestone date types:

1. Select the Define Milestone Date Type command in the Tool menu of the DevSuite Admin client.

   The Milestone Date Type manager appears.

2. Select a milestone date type in the Milestone list.

3. Click the Edit button.

   The Edit Milestone Date Type dialog box appears.

4. Update the name of the milestone date type in the Milestone Date Type Name control.

5. Click the OK button.

9.2.3 Deleting Milestone Date Types

To edit milestone date types:

1. Select the Define Milestone Date Type command in the Tool menu of the DevSuite Admin client. The Milestone Date Type manager appears.

2. Select a milestone date type in the Milestone list.

3. Click the Delete button.

   A DevSuite Admin dialog box appears.

4. Click the Yes button.

9.3 Administering the Product Tree

The product tree is a hierarchical tree structure that represents the product categories, products, versions, and builds under development in a DevSuite site. The product tree is sometimes referred to as the “global product tree” because the products, versions, and builds defined and managed in it are available to every KnowledgeWise, DevSpec, and DevTrack project in the site.

Product tree administration is a system-level administrative task that is generally defined by a system administrator in the system settings project. Once defined, the product tree may be used by every TechExcel KnowledgeWise, DevSpec, and DevTrack project in a DevSuite site.

System administrators must define a product, version, or build in the product tree before the product, version, or build represented by that release can be managed in a DevTrack project.

Product Category: A product category is any grouping of products, such as a product line, that a business uses to organize its products.

Product: A product represents a product in development in DevTrack. Each product may be the parent of many different versions.

Version: A version represents a version of a product. Each version is the child of a product and may be the parent to one or more builds.

Build: A build represents a specific build of a product version. Each build is the child of a single version.

Project administrators may create, edit, and delete product categories, products, versions, and builds in the Define Product/Version/Build Tree manager.

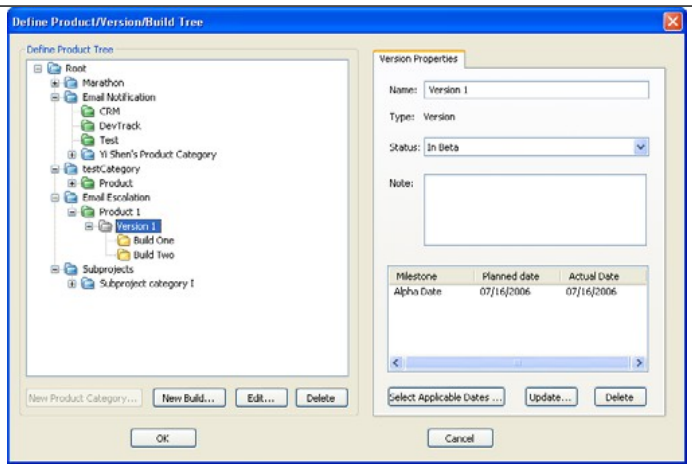


Figure 9-2: Define Product/Version/Build Tree Manager

To define products, versions, and builds in the product tree, a system administrator must first define appropriate product release statuses, version release statuses, and build release statuses.

To define versions in the product tree, a system administrator must first define appropriate milestone date types that may be used to define versions.

9.3.1 Administering Product, Version, and Build Release Cycles

In DevSuite, the development of products, versions, and builds are defined by a unique lifecycle consisting of multiple*release states.*A release state is an administrator-defined workflow state that identifies the progress of that release within its development life cycle.

Administrator-defined workflow rules determine how a release item (product, version, or build) may be managed within each release state in its lifecycle.

The privileges available to project administrators and project members in each release depend on the release type.

Release	Admin Privileges	Client Privileges
Product	Add new versions	Add new issues
	Add new builds	
Version	Add new builds	Add new issues
Build		Add new issues

Table 9-1: Definable Open Release Statuses

9.3.2 Release status

Every release state is defined by its release status—open or closed. Products, versions, and builds in open states (that is, release states with an open release status) are managed in workflow based on administrator-defined rules. Records in closed states must be reopened to be managed in release management workflow.

Open Release Status: An open release status indicates that the product, version, or build is currently under development. Changes may be made to releases with an open release status.

Closed Release Status: An closed release status indicates that the product, version, or build is currently not under development. The product, version, or build may be completed or may still be in the future. Changes may not be made to releases with a closed release status.

Open project are definable, and the system administrator may enable certain privileges (at either administrative or client level) and restrict others. Closed releases cannot be changed at either level.

9.4 Administering Applicable Products, Versions, and Builds

In DevSuite, the product tree identifies the products, product versions, and builds under development site and provides a common framework for organizing and managing the development of products, versions, and builds.

Product tree structures define the framework for DevSuite knowledge management structures, DevTrack/DevPlan subprojects, and DevTrack branch management.

To use the product, version, and build structures, the project administrator must identify which products, versions, and builds in the product tree are applicable to their DevTrack project. Only those release folders that have been defined as “applicable” may be utilized in the DevTrack project.

To select applicable products, versions, and builds:

1Select Product Settings > Applicable Product/Version in the tree panel.

2Click the Select button.

The Select Product dialog box appears. The Product Tree displays the product categories, products, versions, and builds that have been defined in site.

3Select all applicable products, versions, and builds.

The products, versions, and builds selected in the product tree may define DevTrack subproject structures.

4 Click the OK button.

9.5 Administering Product Categories

A product category is any grouping of products, such as a product line, that a business uses to organize its products. A product category the top branch in the product tree and is the parent of multiple child products.

Product categories do not represent a release of a product and cannot be associated with a subproject folder in the DevSuite project. But project administrators may choose which product line is managed in their project by selecting or deselecting product categories.

To add a product category:

1 Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

The Define Product/Version/Build Tree manager appears.

2 Select the Root folder or a product category subfolder in the Product Tree.

The Root folder and all product category subfolders in the Product Tree are blue.

3 Click the New Product Category button.

The New Product Category dialog box appears.

4 Define a unique name for the product category in the Product Category Name control.

5 Click the OK button.

6 Define product category properties in the Product Category Properties tab.

Project administrators may define two product category properties in the Product Category tab.

- The Product Category Name
- The Product Category Notes

7 Click the OK button.

To edit a product category:

1 Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

The Define Product/Version/Build Tree manager appears.

2 Select a product category subfolder in the Product Tree.

All product category subfolders in the Product Tree are blue.

3 Click the Edit button.

The Edit Product Category dialog box appears.

4 Update the name of the product category in the Product Category Name control.

5 Click the OK button.

6 Update product category properties in the Product Category Properties tab.

Project administrators may define two product category properties in the Product Category tab.

- The Product Category Name
- The Product Category Notes

7 Click the OK button.

To delete a product category:

1 Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

The Define Product/Version/Build Tree manager appears.

2 Select a product category subfolder in the Product Tree.

All product category subfolders in the Product Tree are blue.

3 Click the Delete button.

A DevSuite Admin dialog box appears.

4 Click the Yes button.

5 Click the OK button.

9.6 Administering Products

In DevSuite, a product is a representation of a development item. Every DevSuite product is the child of a product category and the parent to one or more versions.

The development of products is defined by the product lifecycle. The product lifecycle is a set of release states and workflow rules for managing products within each release state that enables the development organization to manage and track the development of that product in workflow.

A product may be associated with a subproject folder in the DevTrack client. All issues and events, versions, and builds relevant to that product are stored within that subproject folder in the DevTrack client.

9.6.1 Defining Products

To add a product:

1. Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

   The Define Product/Version/Build Tree manager appears.

2. Select a product category subfolder in the Product Tree.

   All product category subfolders in the Product Tree are blue. Every product is the child of a product category.

3. Click the New Product button.

   The New Product dialog box appears.

4. Define a unique name for the product in the Product Name control.

5. Click the OK button.

   The New Product dialog box closes.

6. Define product properties in the Product Properties tab of the Define Product/ Version/Build Tree manager.

   Project administrators may define three product properties in the Properties tab.

   - The Product Name identifies the product.
   - The Product Note describes the product.
   - The Product Status determines how the product is managed in the DevTrack site. The Product Status dropdown displays the product release statuses defined in the Release Status manager.

7. Click the OK button.

To edit a product:

1. Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

   The Define Product/Version/Build Tree manager appears.

2. Select a product in the Product Tree.

   All product subfolders in the Product Tree are green.

3. Click the Edit button.

   The Edit Product dialog box appears.

4. Update the name of the product in the Product Name control.

5. Click the OK button.

6. Update product properties in the Product Properties tab.

   Project administrators may define three product properties in the Properties tab.

- The Product Name identifies the product.
- The Product Note describes the product.
- The Product Status determines how the product is managed in the DevTrack site. The Product Status dropdown displays the product release statuses defined in the Release Status manager.

7. Click the OK button.

To delete a product:

1. Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

   The Define Product/Version/Build Tree manager appears.

2. Select a product subfolder in the Product Tree.

   All product subfolders in the Product Tree are green.

3. Click the Delete button.

   A DevSuite Admin dialog box appears.

4. Click the Yes button.

5 Click the OK button.

9.6.2 Defining the Product Release Lifecycle

In DevSuite, a product is a representation of a development item. Every DevSuite product is the child of a product category and the parent to one or more versions.

The development of products is defined by the product lifecycle. The product lifecycle is a set of release states and workflow rules for managing products within each release state that enables the development organization to manage and track the development of that product in workflow.

Using controls in the Product Release Status page, the system administrator may define multiple product release states and define rules for administering and managing products within that state.

Figure 9-3: Product Release Status Page

The lifecycle of a product may consist of one or more product release states." Each release state defines a set of rules for managing products when they are in that particular state.

Each product release state is defined by three rules.

Add New Versions: The Add New Versions privilege enables project managers to create new child version subprojects under the product subproject.

Add New Builds: The Add New Builds privileges enables project managers to create new child build subprojects under the product subproject.

Add New Issues: The Add New Issues privilege enables project members to create new issues in a release subproject.

To add a product state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Click the Add button in the Product tab.

The New Product Status dialog box appears.

3 Define the name of the product status in the Product Status Name control.

4 Click the OK button.

The New Product Status dialog box closes.

5 Define Product Status properties.

6 Click the OK button.

To edit a product state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a product release status in the Product list of the Product tab.

3 Click the Edit button.

The Edit Product Status dialog box appears.

4 Update the name of the product status in the Product Status Name control.

5 Click the OK button.

The Edit Product Status dialog box closes.

6 Update Product Status properties in the Product Properties tab.

7 Click the OK button.

To delete a product state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a product release status in the Product list of the Product tab.

3 Click the Delete button.

A DevTrack dialog box appears.

4 Click the Yes button.

9.7 Administering Versions

In DevSuite, a version is a representation of a version of a product release. Every DevSuite version is the child of a product and the parent to one or more builds.

The development of product versions is defined by the version lifecycle. The version lifecycle is a set of release states and workflow rules for managing versions within each release state that enables the development organization to manage and track the development of that version in workflow.

System administrators may create, edit, and delete versions and define version milestones in the Version Properties tab of the Product/Version/Build Tree manager.

Figure 9-4: Product/Version/Build Tree Manager.

A version may represent one or more milestones." A milestone is a point in time that represents an important intermediate event in a project. Milestones commonly represent the completion of key components, reporting requirements, and the beginning or end of project phases.

To add a version:

1 Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

The Define Product/Version/Build Tree manager appears.

2 Select a product subfolder in the Product Tree.

All product subfolders in the Product Tree are green.

3 Click the New Version button.

The New Version dialog box appears.

4 Define a unique name for the version in the Product Name control.

5 Click the OK button.

The New Version dialog box closes.

6 Define version properties in the Version Properties tab.

7 Select a version status option from the Version Status dropdown list.

The version status determines how the version is managed in the DevTrack site. The Version Status dropdown displays the version release statuses defined in the Release Status manager.

8 Provide a brief description of the version in the Version Note control.

9 Define version milestones in the Version Milestones area.

A version may represent one or more milestones. A milestone is a point in time that represents an intermediate event in the life of a project. System administrators may define multiple milestones for each version.

10 Click the OK button.

The Define Product/Version/Build Tree manager closes.

9.7.1 Managing Version Milestones

Every version in the product tree represents a version of a product in development in a DevTrack project. Each version is the child of a product and may be the parent to one or more builds.

A version may represent one or more milestones. A milestone is a point in time that represents an important intermediate event in a project. Milestones commonly represent the completion of key components, reporting requirements, and the beginning or end of project phases.

Typical milestones that a product administrator may want to add to a version include:

- Alpha date
- Beta date
- Release date
- Code Freeze date
- Preproduction date
- Feature Freeze date

System administrators may create, edit, and delete versions and define version milestones in the Version Properties tab of the Product/Version/Build Tree manager.

Figure 9-5: Product/Version/Build Tree Manager.

Every milestone that the system administrator adds to a version is based on a milestone date type." A milestone date type is a template that enables system administrators to define appropriate milestones for each version in the product tree. System administrators may define milestone date types in the Milestone Date Type manager.

To add a version milestone:

1. Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

   The Define Product/Version/Build Tree manager appears.

2. Select a version subfolder in the Product Tree.

   All version subfolders in the Product Tree are silver.

3. Click the Select Applicable Dates button in the Version Properties tab.

   The Applicable Dates dialog box appears.

4. Select one or more milestones in the Applicable Dates dialog box.

   The Applicable Dates dialog box displays all of the milestone defined in the Milestones Date Type manager.

5. Click the OK button.

To update milestone dates:

1. Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

   The Define Product/Version/Build Tree manager appears.

2. Select a version subfolder in the Product Tree.

   All version subfolders in the Product Tree are silver.

3. Select a milestone in the Milestone list and click the Update button.

   The Update dialog box appears.

4. Update the Planned Date and the Actual Date.

5. Click the OK button.

To delete a version milestone:

1. Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

   The Define Product/Version/Build Tree manager appears.

2. Select a version subfolder in the Product Tree.

All version subfolders in the Product Tree are silver.

3 Select a milestone in the Milestone list and click the Delete button.

A DevSuite Admin dialog box appears.

4 Click the Yes button.

9.7.2 Deleting Versions

A version represents a version of a product. Project administrators may define version status, creator, and multiple milestone dates for each version. Each version is the child of a product and may be the parent to one or more builds.

To delete a version:

1 Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

The Define Product/Version/Build Tree manager appears.

2 Select a version subfolder in the Product Tree.

All version subfolders in the Product Tree are silver.

3 Click the Delete button.

A DevSuite Admin dialog box appears.

4 Click the Yes button.

5 Click the OK button.

9.7.3 Managing the Version Release Lifecycle

In DevSuite, a version is a representation of a version of a product release. Every DevSuite version is the child of a product and the parent to one or more builds.

The development of product versions is defined by the version lifecycle. The version lifecycle is a set of release states and workflow rules for managing versions within each release state that enables the development organization to manage and track the development of that version in workflow.

The lifecycle of a product may consist of one or more product release states. Each release state defines a set of rules for managing products when they are in that particular state.

Add New Builds: The Add New Builds privileges enables project managers to create new child build subprojects under the version subproject.

Add New Issues: The Add New Issues privilege enables project members to create new issues in a version subproject.

To add a version state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Click the Add button in the Version tab.

The New Version Status dialog box appears.

3 Define the name of the version status in the Version Status Name control.

4 Click the OK button.

The New Version Status dialog box closes.

5 Define Version Status properties.

6 Click the OK button.

To define version status properties:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 To define the version status as an open or closed status, select an option in the Product Properties tab.

- Select the Open
- Select the Closed

3 To enable, select the Can Add New Build check box.

4 To enable, select the Can Add New Issues check box.

5 Click the OK button.

To edit a version state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a version release status in the Product list of the Product tab.

3 Click the Edit button.

The Edit Version Status dialog box appears.

4 Update the name of the product status in the Version Status Name control.

5 Click the OK button. The Edit Version Status dialog box closes.

6 Update Product Status properties in the Version Properties tab.

7 Click the OK button.

To delete a version state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a version release status in the Version list of the Version tab.

3 Click the Delete button.

A DevTrack dialog box appears.

4 Click the Yes button.

9.8 Administering Builds

Every build in the product tree represents a build of a product version in development in a DevTrack project. Each build is the child of a single version.

A build may be associated with a subproject folder in the DevTrack client. The subproject folder may be used to manage and track development and QA test issues for the associated build.

9.8.1 Defining Builds

To add a build:

1 Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

The Define Product/Version/Build Tree manager appears.

2 Select a version subfolder in the Product Tree.

All version subfolders in the Product Tree are silver.

3 Click the New Build button.

The New Build dialog box appears.

4 Define a unique name for the build in the Build Name control.

5 Click the OK button. The New Build dialog box closes.

6 Define build properties in the Build Properties tab of the Define Product/Version/ Build Tree manager.

7 Select a build status option from the Build Status dropdown list.

The build status determines how the build is managed in the DevTrack site. The Build Status dropdown displays the build release statuses defined in the Release Status manager.

8 Click the OK button.

The Define Product/Version/Build Tree manager closes.

To edit a build:

1 Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

The Define Product/Version/Build Tree manager appears.

2 Select a build in the Product Tree.

All build subfolders in the Product Tree are gold.

3 Click the Edit button.

The Edit Build dialog box appears.

4 Update the name of the build in the Build Name control.

5 Click the OK button.

6 Update build properties in the Build Properties tab.

Project administrators may define three product properties in the Build Properties tab.

- The Build Name
- The Build Status
- The Build Note

7 Click the OK button.

To delete a build:

1 Select the Define Product Tree command in the Tool menu of the DevSuite Admin client.

The Define Product/Version/Build Tree manager appears.

2 Select a build subfolder in the Product Tree.

All build subfolders in the Product Tree are gold.

3 Click the Delete button.

A DevSuite Admin dialog box appears.

4 Click the Yes button.

5 Click the OK button.

9.8.2 Managing Build Release Lifecycle

In DevSuite, a build represents a specific build of a product version. Every build is the child of a version. Every build is defined by its release status. The release status indicates the current status of the build in the DevTrack site.

Every build is assigned either an open release status or a closed release status.

Open releases statuses are definable, and the system administrator may enable or disable the open status privilege: The Add New Issues privilege enables project members to create new issues in the version subproject.

To add a build state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Click the Add button in the Build tab.

The New Build Status dialog box appears.

3 Define the name of the build status in the Product Status Name control.

4 Click the OK button.

The New Build Status dialog box closes.

5 Define Build Status properties.

6 Click the OK button.

To define build release state properties:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Click the Add button in the Build tab.

3 To define the build status as an open or closed status, select an option in the Product Properties tab.

- Select the Open
- Select the Closed

4 To enable, select the Can Add New Test check box.

5 Click the OK button.

To edit a build state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a product release status in the Product list of the Product tab.

3 Click the Edit button.

The Edit Product Status dialog box appears.

4 Update the name of the product status in the Product Status Name control.

5 Click the OK button. The Edit Product Status dialog box closes.

6 Update Product Status properties in the Product Properties tab.

7 Click the OK button.

To delete a build state:

1 Select the Define Product Release Status command in the Tool menu of the DevSuite Admin client.

The Release Status manager appears.

2 Select a product release status in the Product list of the Product tab.

3 Click the Delete button.

A DevTrack dialog box appears.

4 Click the Yes button.

9.9 Administering Product Implementation Modules

In DevSuite, a product implementation module defines a distinct area of development—such as a platform or module—that is distinct from other areas of development.

Every product and version may be defined by one or more applicable product implementation modules.

Chapter 10 Subproject Administration

10 CHAPTER10 Subproject Administration

This chapter describes how development organizations may use subprojects to organize, prioritize, and schedule development issues.

In this chapter:

- Understanding DevSuite Subprojects
- Administering Subproject Settings
- Administering Subproject Statuses
- Administering Subproject Priorities

10.1 Understanding DevSuite Subprojects

In DevSuite, a subproject is a tool for organizing development issues into smaller, more manageable groupings of issues. Development teams may create subproject folders with subproject folders to organize their work and define subproject statuses, prioritize subprojects, and set due dates for all subprojects. Each subproject has its own description, priority, status, and start and end dates.

Every work project represents a distinct area of work and is defined by unique team structures (account types, team groups, and group folders) and workflow rules. Subprojects provide developers the means to organize, schedule, prioritize, and track groups of issues and to implement an additional layer of security within the project.

Subprojects are ideal for development organizations that wish to develop multiple products within a single DevTrack project. Subprojects are also useful to organizations developing products made up of multiple components or modules, or who wish to track many different types of issues in a single DevTrack project.

10.1.1 Understanding the Subproject Tree Structure

The subproject hierarchy is displayed in both the DevTrack Windows and Web client and the DevPlan smart client providing development organizations with a common framework for managing, tracking, and scheduling tasks.

In DevTrack, the subproject hierarchy is displayed in the issue tree panel and organizes development issues by product, version, and build.

Figure 10-1: Subproject Enabled Client GUI

In DevPlan, the subproject hierarchy is displayed in the issue tree panel and organizes development issues by product, version, and build.

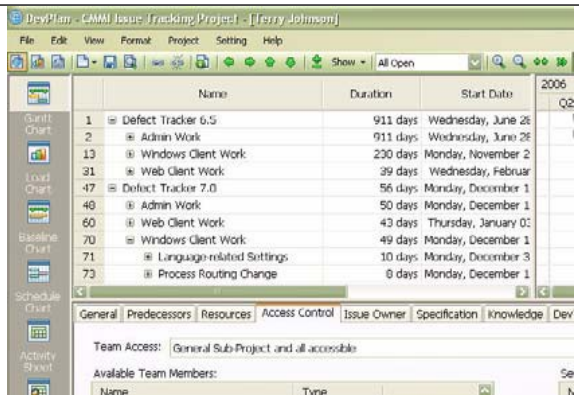


Figure 10-2: Subproject Enabled Client GUI

Subprojects serve three primary purposes in DevTrack projects:

- Subprojects facilitate the management of multiple development projects within a single DevTrack project."
- Subprojects enable development teams to define rules for managing issues within a subproject.
- Subprojects enable development teams to use multiple workflow to manage project issues.

Note: If TechExcel DevPlan integration is enabled for a DevTrack project, all subproject management tasks are performed in the DevPlan smart client. Project team members may create and manage subprojects in the DevTrack client only if DevPlan integration is not enabled.

In DevTrack Admin, project administrators may define rules for defining the status and priority of subprojects, and account type access privileges, which determine which project members may create, edit, and delete subprojects within the DevTrack project.

- The status of a subproject enables project members to indicate the current progress state of a subproject. The status of a subproject may be defined independently or be linked to the progress state of the issues in the subproject.
- The priority of a subproject enables project members to indicate the current priority of a subproject relative to other subprojects.

All subproject properties may be defined in the Subproject manager within the DevTrack client.

Administrators may use tools in the Subprojects subfolder to enable subprojects, define subproject statuses, subproject priorities, and subproject access privileges.

10.1.2 Understanding Subprojects and the Beta Customer Portal

Projects that are using the Beta Customer Portal may wish to route all issues submitted from beta customers through the Beta Customer Portal to a subproject.

Project administrators may define a default subproject for every customer created in a customer base project. All issues submitted by that customer are submitted to a DevTrack project would by default be created in the selected subproject.

10.2 Administering Subproject Settings

In DevSuite, a subproject is a discreet area of work that corresponds to a specific product, feature, version, or build. Using subprojects, development teams may divide a DevTrack project into smaller, more manageable groupings of issues. Project members may schedule, prioritize, and track those issues using subproject properties and, optionally, subproject workflow.

Using controls in the General Settings page, project administrators may enable subproject support in a DevTrack project and to enable team member access controls, applicable issue owners, and applicable issue statuses in subprojects.

Project administrators must enable support for subprojects in each DevTrack project. If subproject support is not enabled no subproject settings can be implemented within the project and the issue tree panel does not appear in the DevTrack Windows client.

10.2.1 Enabling Subproject Support

In TechExcel DevTrack, subprojects are an optional Enterprise Edition feature that must be enabled in each work project. Basic subproject administrative tasks include enabling subproject support, enabling DevPlan integration support, and defining how subprojects are identified and managed in the integrated DevSuite site.

To enable subprojects:

1Select the General Settings page in the Subprojects subfolder.

2Select the Enable Subproject Support check box.

3Enter an alternative title for subprojects the Refer As field.

The Refer As option enables project administrators to change the Subproject label displayed in the DevTrack client. The text entered in the Refer As field is displayed instead of the word*Subproject*throughout the project in both the Windows and Web clients.

The text entered does not change the title of the Project Issues folder in the Issue Tree window of the DevTrack Windows client.

4Select a default subproject from the Default Subproject Creation dropdown list.

10.2.2 Defining Subproject Terminology

The Refer As option enables project administrators to change the Subproject label displayed in the DevTrack, DevSpec, and DevPlan clients. The text entered in the Refer As field is displayed instead of the word*subproject*throughout the project in both the Windows and Web clients.

To define subproject terminology:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Enter an alternative title for subprojects the Refer As text box.

   The text entered does not change the title of the Project Issues folder in the issue tree panel of the DevTrack Windows client.

10.2.3 Defining Default Subprojects

To define default subprojects:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select a default subproject from the Default Subproject Creation dropdown list.

10.2.4 Defining Applicable Issue Types

In DevTrack, an issue type represents a class of development issues share similar properties and which need to be managed in a unique way. De vTrack workflow rules are generally based on issue types.

The issue type of an issue is typically linked to its type definition (for example, bug, new feature, enhancement), but alternatively may be linked to its priority, component, platform, build, version, or any other criteria.

To define applicable issue types:

1. Select Advanced Features > Subproject > Subproject Issue Type in the tree panel.
2. Select the Enable Applicable Issue Type Based on Subproject check box.
3. Select an issue type in the Issue Type list.
4. Click the Edit button.

   The Associate Issue Type with Subproject dialog box appears.
5. Select one or more issue types.
6. Click the OK button.

   The Associate Issue Type with Subproject dialog box closes and the issue type is associated with the subproject.

10.3 Administering Subproject Statuses

In DevSuite, a subproject is a tool for organizing development issues into smaller, more manageable groupings of issues so that development teams may manage and track related de velopment issues as a unit.

Every subproject is defined by its*subproject status*which indicates the current progress of the subproject within the development lifecycle. The lifecycle of subprojects may be uniquely defined in each project or be based on the lifecycle of the development issues managed in that project.

The status of a subproject may be automatically calculated based on administrator-defined rules or manually defined by project members in the DevPlan client.

DevSuite supports four methods of deteriming the status of subprojects in a project: the Independent Of Its Incidents method, Derived from the Highest-ranked Incident Status method, Derived from the Lowest-Ranked Incident Status method, and the Linked to Incident Status Definition method.

Independent of its Incidents Status: The Independent of its Incidents Status method requires that project team members manually define the status of all subprojects using controls in the DevPlan client. Subproject states are fully customizable.

Derived from the Highest-ranked Incident Status: The Derived from the Highest-ranked Incident Status method ties the status of a subproject to the highest ranked issue within the subproject. Subproject statuses are automatically calculated based on administrator-defined issue status rankings.

Derived From The Lowest-ranked Incident Status: The Derived From The Lowest-ranked Incident Status status method ties the status of a subproject to the lowest ranked issue within the subproject. Subproject statuses are automatically calculated based on administrator-defined issue status rankings.

Linked to Incident Status Definition: The Linked to Incident Status Definition status method method requires that project team members manually define the status of all subprojects using controls in the DevPlan client.. Subproject states are identical to development issue states.

Using controls in the Subproject Status page, project administrators may define the method used to define the subproject status of subprojects.

10.3.1 Defining Independent Subproject Statuses

Independent subproject statuses are one of four types of subproject status rules that an administrator may define in the Subproject Status page of DevTrack Admin. Administrators must select the*Independent of its Incidents Status option*in the Status Option dropdown list of the Subproject Status page.

Adminsitrators may add, edit, and delete independent subproject statuses in the Subproject Status page. Although administrators may rank independent subproject statuses the ranks assigned have no effect using this option.

Note: Adminstrator-defined independent subproject status options are overwritten if the administrator selects the Linked to Incident Status option in the Status Option dropdown list.

To add independent subproject statuses:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Status option from the Status Option dropdown list.
3. Click the Add button.

   The Status dialog box appears.
4. Enter the name of the status in the Name field.
5. Click the OK button.

To edit independent subproject statuses:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Status option from the Status option dropdown list.
3. Select a status in the Status list.
4. Click the Edit button.

   The Status dialog box appears.
5. Enter the name of the status in the Name field.
6. Click the OK button.

To delete independent subproject statuses:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Status option from the Status option dropdown list.
3. Select a status in the Status list.
4. Click the Delete button.

   A Warning dialog box appears.
5. Click the Yes button.

10.3.2 Defining Subproject Status Based on Highest Ranking

Administrators may use controls in the Subproject Status page to define rules for determining the status of subprojects in DevTrack projects based on issue status rankings.

Administrators may select the Derived from the Highest-ranked Incident Status option in the Status Option dropdown list to tie the status of a subproject to the status of the highest-ranked issue in that subproject. If this option is selected, the status of subprojects is defined automatically based on the statuses of the issues contained in that subproject.

Each subproject represents multiple issues. The progress status of each issue is represented by its issue status. Administrators may use controls in the Subproject Status page to rank the issue status options used in the project. The issue with the highest-ranking issue status determines the subproject status of the entire subproject. The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1”is ranked higher than a status assigned a “5.”

To rank issue status options:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Derived From The Highest-ranked Incident Status Status option from the Status Option dropdown list.
3. Select an option in the Status list.

   The Status list displays all of the status options that may be selected for a DevTrack issue.
4. Click the Edit button.

   The Status dialog box appears.
5. Enter the rank of the status in the Status field.

   The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1”is ranked higher than a status assigned a “5.”
6. Click the OK button.

   The rank of the issue status is displayed in the Status list.

10.3.3 Defining Subproject Status Based on Lowest Ranking

Administrators may use controls in the Subproject Status page to define rules for determining the status of subprojects in DevTrack projects based on issue status rankings.

Administrators may select the Derived from the Lowest-ranked Incident Status option in the Status Option dropdown list to tie the status of a subproject to the status of the

lowest-ranked issue in that subproject. If this option is selected, the status of subprojects is defined automatically based on the statuses of the issues contained in that subproject.

Each subproject represents multiple issues. The progress status of each issue is represented by its issue status. Administrators may use controls in the Subproject Status page to rank the issue status options used in the project. The issue with the lowest-ranking issue status determines the subproject status of the entire subproject.

The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1”is ranked higher than a status assigned a “5.”

To rank issue status options:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Derived From The Lowest-ranked Incident Status Status option from the Status Option dropdown list.
3. Select an option in the Status list.

   The Status list displays all of the status options that may be selected for a DevTrack issue.
4. Click the Edit button.

   The Status dialog box appears.
5. Enter the rank of the status in the Status field.

   The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1”is ranked higher than a status assigned a “5.”
6. Click the OK button.

   The rank of the issue status is displayed in the Status list.

10.3.4 Defining Subproject Status Based on Issue Status

Administrators may use controls in the Subproject Status page to link subproject status to the options used to define the status of issues in the project.

The Linked to Incident Status Definition Status option enables project teams to use the status options that define DevTrack issues to define DevTrack subprojects as well. Project members may manually define the status of a subproject whenever they create or edit a subproject in the Subproject manager. The options displayed in the subproject Status Option dropdown list are identical to the options displayed in the Status dropdown list that is used to define issue status.

Adminsitrators may edit, but not add or delete status options in the Subproject Status page. Because subproject status options are tied to issue status options, new sub-status options are made available whenever the administrator creates new issue statuses in the Workflow page.

Although administrators may rank subproject statuses in the Subproject Status page, the ranks assigned have no effect using this option.

To link subproject status option to issue status options:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Linked To Incident Status Definition status option from the Status option dropdown list.
3. Select an option in the Status list.

   The status options displayed in the Status list are identical the status options used to define issues in the DevTrack project.
4. Click the Edit button.

   The Status dialog box appears.
5. Click the OK button.

10.4 Administering Subproject Priorities

The subproject priority defines the relative importance of a subproject relative to other areas of development.

The priority of each subproject may be defined manually or defined automatically depending on administrator-defined subproject definition rules.

DevSuite supports four methods of deteriming the priority of subprojects in a project: the Independent Of Its Incidents method, Derived from the Highest-ranked Incident Status method, Derived from the Lowest-Ranked Incident Status method, and the Linked to Incident Status Definition method:

Independent of its Incidents Priority: The Independent of its Incidents Priority option enables administrators to define a unique set of options to indicate the priority of a subproject. Project members may manually define the priority of a subproject by selecting one of these administrator-defined options in the Priority dropdown list in the Subproject Manager.

Derived from the Highest-ranked Incident Priority: The Derived from the Highest-ranked Incident Priority option ties the priority of a subproject to the highest ranked issue within that subproject. The rank of issues in a subproject is based on administrator-define rankings of the options in one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field. The priority of each subproject is determined automatically based on the rank the administrator assigns to the options available in the corresponding issue control. The subproject Priority dropdown list in the Subproject Manager is grayed out and cannot be edited.

Derived from the Lowest-ranked Incident Priority: The Derived from the Lowest-ranked Incident Priority option ties the priority of a subproject to the lowest ranked issue within that subproject. The rank of issues in a subproject is based on Type field, Priority field, the Component field, the Version field, or the Platform field. The priority of each subproject is determined automatically based on the rank the administrator assigns to the options available in the corresponding issue control. The Subproject Priority dropdown list in the Subproject Manager is grayed out and cannot be edited.

Linked to Incident Priority Definition: The Linked to Incident Priority Definition option enables the administrator to tie the priority of a subproject to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field. The options available for selection in the subproject Priority field are

identical to the options displayed in the selected issue control.

10.4.1 Creating Independent Subproject Priorities

Administrators may use controls in the Subproject Priority page to define a unique set of options that DevTrack project members may use to identify the current priority of a subproject.

In the DevTrack client project members may change the priority of a subproject by selecting an option from the Priority dropdown list of the Subproject manager. Administrators may define these priority options using controls in the Subproject Priority page.

Independent subproject priorities are one of four types of subproject priority rules that an administrator may define in the Subproject Priority page of DevTrack Admin. Administrators must select the*Independent of its Incidents Priority option*in the Priority Option dropdown list of the Subproject Priority page.

Adminsitrators may add, edit, and delete independent subproject priority optons in the Subproject Priority page. Although administrators may rank independent subproject priorities the ranks assigned have no effect using this option.

Note: Adminstrator-defined independent subproject priority options are overwritten if the administrator selects the Linked to Incident Priority option in the Priority Option dropdown list.

To add an independent subproject priority:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Priority option from the Priority Option dropdown list.
3. Click the Add button.

   The Priority Dialog dialog box appears.
4. Enter a title for the priority in the Name field.
5. Click the OK button.

   The Priority Dialog dialog box closes and the priority is displayed in the Priority list.

To edit an independent subproject priority:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Priority option from the Priority Option dropdown list.
3. Select a priority in the Priority list.
4. Click the Edit button.

   The Priority Dialog dialog box appears.
5. Enter a title for the priority in the Name field.
6. Click the OK button.

   The Priority Dialog dialog box closes and the updated priority is displayed in the Priority list.

To delete independent subproject priority:

1. Select Advanced Features > Subproject > General Settings in the tree panel.
2. Select the Independent of its Incidents Priority option from the Priority Option dropdown list.
3. Select a priority in the Priority list.
4. Click the Delete button.

   A Warning dialog box appears.
5. Click the Yes button.

10.4.2 Defining Highest Subproject Priorities

Project administrators may use controls in the Subproject Priority page to define rules for determining the priority of subprojects in DevTrack projects based on issue priority rankings.

Administrators may select the Derived from the Highest-ranked Incident Priority option in the Priority Option dropdown list to tie the priority of a subproject to the “priority” of the highest-ranked issue in that subproject. If this option is selected, the priority of subprojects is defined automatically based on the priorites of the issues contained in that subproject.

The relative priority of each subproject may be based on a number of different criteria. Administrators may link the subproject Priority control to one of five issue system fields (the Type field, Priority field, the Component field, the Version field, or the Platform field) and rank each of the options displayed in the selected issue control.

Each subproject represents multiple issues. The issue with the highest-ranking priority determines the subproject priority of the entire subproject.

The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1”is ranked higher than a status assigned a “5.”

To define highest subproject priorities:

1. Select Advanced Features > Subproject > General Settings in the tree panel.

2 Select the Derived From Its Highest-ranked Incident Priority option from the Priority option dropdown list.

3 Select an option in the Link to Priority dropdown list.

Administrators may link subproject priority to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field.

4 Select an option in the Priorities field.

The Priorities field displays options belonging to the list control selected in the Link to Priority dropdown list.

5 Click the Edit button.

The Priority dialog box appears.

6 To edit the rank of the option enter a value in the Rank field.

The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1”is ranked higher than a status assigned a “5.”

7 Click the OK button.

10.4.3 Defining Lowest Subproject Priorities

Administrators may use controls in the Subproject Priority page to define rules for determining the priority of subprojects in DevTrack projects based on issue priority rankings.

Administrators may select the Derived from the Lowest-ranked Incident Priority option in the Priority Option dropdown list to tie the priority of a subproject to the “priority” of the lowest-ranked issue in that subproject. If this option is selected, the priority of subprojects is defined automatically based on the priorites of the issues contained in that subproject.

The relative priority of each subproject may be based on a number of different criteria. Administrators may link the subproject Priority control to one of five issue system fields (the Type field, Priority field, the Component field, the Version field, or the Platform field) and rank each of the options displayed in the selected issue control.

Each subproject represents multiple issues. The issue with the highest-ranking priority determines the subproject priority of the entire subproject.

The lower the number assigned to an issue status, the higher the ranking of that status. For example, an issue that is assigned a “1”is ranked higher than a status assigned a “5.”

To define lowest subproject priorities:

1 Select Advanced Features > Subproject > General Settings in the tree panel.

2 Select the Derived From Its Lowest-ranked Incident Priority option from the Priority option dropdown list.

3 Select an option in the Link to Priority dropdown list.

Administrators may link subproject priority to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field.

4 Select an option in the Priorities field.

The Priorities field displays options belonging to the list control selected in the Link to Priority dropdown list.

5 Click the Edit button.

The Priority dialog box appears.

6 To edit the rank of the option, enter a value in the Rank field.

The lower the number assigned to the issue status, the higher the ranking of that issue status. For example, an issue status that is assigned a “1”is ranked higher than an issue status that is assigned a “5.”

7 Click the OK button.

10.4.4 Defining Linked to Issue Subproject Priorities

Administrators may use controls in the Subproject Priority page to link subproject priorities to options belonging to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform field.

The Linked to Incident Status Definition Priority option enables project teams to use options that define DevTrack issue properties to define DevTrack subproject priorities as well. Project members may manually define the priority of a subproject whenever they create or edit a subproject in the Subproject manager. The options displayed in the subproject Priority Option dropdown list are identical to the options displayed in the linked issue list control (the Type field, Priority field, the Component field, the Version field, or the Platform field).

Adminsitrators may edit, but not add or delete priority options in the Subproject Priority page. Because subproject priority options are tied to system field options, new sub-status priority options are made available whenever the administrator defines new issue options for the corresponding issue control.

Although administrators may rank subproject priorities in the Subproject Status page, the ranks assigned have no effect using this option.

To define linked to incident priority subproject priorities:

1 Select Advanced Features > Subproject > General Settings in the tree panel.

2 Select the Linked to Incident Priority Definition option from the Priority Option dropdown list.

3 Select an option in the Link to Priority dropdown list.

Project administrators may link subproject priority to one of five different system fields: the Type field, Priority field, the Component field, the Version field, or the Platform

field.

4Select an option in the Priorities field.

The Priorities field displays options belonging to the list control selected in the Link to Priority dropdown list.

5Click the Edit button. The Priority dialog box appears.

6Enter the rank of the priority in the Rank field.

7Click the OK button.

Chapter 11 Branch Management Administration

11 CHAPTER11 Branch Management Administration

In this chapter:

- Understanding Multiple Branch Management
- Administering Branch Issues
- Administering Branch Link Types
- Administering Branch Actions

11.1 Understanding Multiple Branch Management

DevTrack multiple branch management enables development organizations to ensure that features and bug fixes managed and tracked as development issues are correctly implemented across product versions and builds.

Branch issues are the means by which developers may confirm that development tasks are completed in all multiple branches of development.

A branch issue is a copy of a development issue that has been copied to a different branch—version or build—of development. Although branch issues inherit basic issue properties from its parent issue, it may be managed within a distinct workflow.

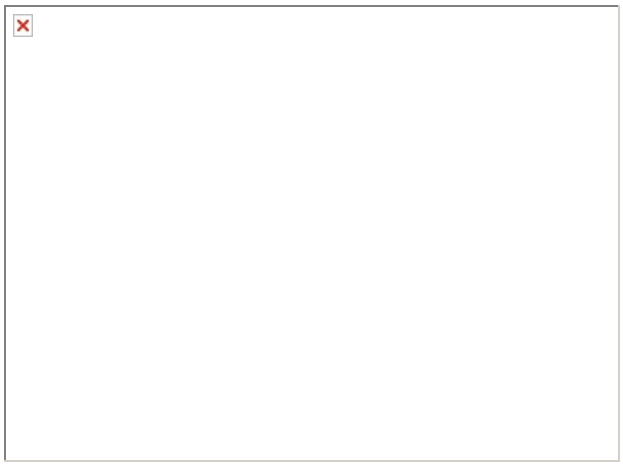


Figure 11-1: Development Issues and Branch Issues

Using branch issues, development teams to create multiple issues and may independently manage and track those issues in different versions and builds of a product in development." Every branch issue may, in turn, be the parent of multiple branch events.

Branch issues are automatically linked to their parent development issue is based on an administrator-defined*branch link types.*A branch link type defines the relationship between the parent development issue and its children and controls the issue lifecycle by means of close*time enforcement rules*.

DevTrack multiple branch management is build upon subproject hierarchy of products, versions, and builds that organize and schedule development tasks within a project.

Every release of a product, version, or build that is managed in a DevTrack project may be associated with a subproject folder in the DevTrack client. The subproject folder represents that release and is used to organize and manage the development issues and branch issues related to that release.

Branch issues enable development teams to create multiple identical issues that may be tested against many different versions or builds of a product.

11.2 Administering Branch Issues

In DevTrack, development organizations may manage and track development tasks across multiple releases as*branch issues*.

A branch issue is a copy of a development issue that has been submitted to a different branch—version or build—of development. Although branch issues inherit basic issue properties from its parent issue, it may be managed within a distinct workflow.

Each branch issue is linked to its parent development issue by a*branch link*enabling developers to manage and track the issues together so that they may ensure that changes to code are properly implemented across all branches of development.

Figure 11-2: Development Issues and Branch Issues

Branch issue administration is the process of enabling multiple branch management support in a DevTrack development project and creating one or more branch issue types.

An issue type is a blueprint for creating issues that are governed within a distinct workflow." The branch issue type does not define issue property values, rather it defines how that issue is managed and tracked in the project.

Every branch issue type is defined by its lifecycle (a subworkflow) and a set of read-only fields.

11.2.1 Enabling Branch Management

Multiple branch management is an optional DevTrack Enterprise Edition feature that must be enabled in each DevTrack project.

To enable branch management:

1. Select Enterprise Edition Features > Multi-Branch Development > General Settings in the tree panel.
2. Select the Enable Multi-Branch Development check box.

11.2.2 Defining Branch Issue Types

In DevTrack, every branch issue is based on a branch issue type. The branch issue type defines the workflow rules by which the branch issues based on the branch issue type are managed in the project.

Branch issues in themselves are not defined by any properties, but inherit issue property definitions from a parent development issue.

To define a branch issue type:

1. Select Enterprise Edition Features > Issue Type in the tree panel.
2. Select a branch issue type in the Issue Types list.

   The Issue Types list displays the all of the administrator-defined development and branch issue types available in the project.
3. Select workflow in the Assigned Workflow dropdown list.

   The Assigned Workflow dropdown list displays the system-defined {Master Workflow} and all administrator-defined “subworkflows” available in the project.
4. **Optional**: To define an issue property field as read-only in child issues based on this issue type, select the issue property field in the Edit Fields list and press the Right Arrow button.

11.2.3 Adding, Editing, and Deleting Branch Issue Types

A project administrator may define any number of branch issue types in a project.

To add a branch issue type:

1. Select Enterprise Edition Features > Issue Type in the tree panel.
2. Click the Add button.

   The New Issue Type dialog box appears.
3. Enter the title of the issue type in the Issue Type Title text box.
4. Select the Branch Issue Type option button.
5. Click the OK button.

   The New Issue Type dialog box closes.

11.2.4 Defining Default Branch Issue Types

To define the default branch issue type:

1. Select Enterprise Edition Features > Issue Type in the tree panel.
2. Select a branch issue type in the Default Branch Issue Type dropdown list.

   The Default Branch Issue Type dropdown list displays all administrator-defined issue types available in the project.

11.3 Administering Branch Link Types

A branch link is a special link type that enables developers to easily track and control the implementation of features and bug fixes across multiple branches of development.

Branch link types define the relationship between a development issues and child branch issues and determine how these issues are managed and tracked in the project. A branch link is automatically created between a parent development issue and a child branch issues whenever a branch issue is created using a branch action.

Branch links tie development issues to their parent development issue and control the lifecycle of both the parent and child issues by means of*close time enforcement rules*.

Close time enforcement rules determine whether the project members may close either issue that is linked using the branch link type. DevTrack supports five branch link close

time enforcement rules:

No Enforcement: The No Enforcement close time enforcement rule defines a relationship in which either the parent development issue or the child branch issue may be closed at any time.

Warning: Do Not Close Parent Issue: The Warning: Do Not Close Parent Issue close time enforcement rule defines a relationship in which a parent development issue should not be closed before its child issues are closed. A warning dialog box is displayed whenever a project member attempts to close a parent development issue that is linked to open child branch issues.

Enforce: Cannot Close Parent Issue: The Enforce: Cannot Close Parent Issue close time enforcement rule defines a relationship in which a parent development issue cannot be closed unless its linked child branch issues are closed.

Warning: Do Not Close Child Issue: The Warning: Do Not Close Child Issue close time enforcement rule defines a relationship in which a child branch issue should not be closed before its parent development issue is closed. A warning dialog box is displayed whenever a project member attempts to close a child branch issue that is linked to an open parent development issue.

Enforce: Cannot Close Child Issue: The Enforce: Cannot Close Child Issue close time enforcement rule defines a relationship in which a child branch issue cannot be closed unless its parent development issue is has already been closed.

Branch link administration is the process of defining branch link types. Every branch link type is defined by its title, link category, link description, and*close time enforcement rules*.

11.3.1 Adding, Editing, and Deleting Branch Link Types

Every branch link type is defined by its title, link category, link description, and*close time enforcement rules*.

To add Branch link types:

1Select the Add button in the Define Release Link Type page of the Branch Management subfolder.

The New branch Link Type dialog box appears.

2Define a unique name for the release link type in the Link Type Name dialog box.

3Click the OK button.

To delete Branch link types:

1Select a link type in the Link Type list of the Define Release Link Type page of the Branch Management subfolder.

2Select the Delete button.

A DevTrack dialog box appears.

3Click the Yes button.

The link type is deleted.

11.3.2 Defining Branch Link Type Properties

To define Branch link type properties:

1Select a link type in the Link Type list of the Define Release Link Type page of the Branch Management subfolder.

2 Optional: To edit the name of the link type, enter a name in the Link Type Name control of the Properties tab.

3 Optional: To provide a brief description of the link type, enter text in the Description control.

4 Optional: To define close time rules for the link type, select an option in the Close Time Enforcement area of the Properties tab.

5The Properties tab displays five options:

- No enforcement: freely close parent or child issue
- Warning: do not close a parent issue until all the child issues get closed
- Enforce: can not close a parent issue until all the child issues get closed
- Warning: do not close child issue until the parent issue gets closed
- Enforce: can not close child issues until the parent issue gets closed

11.4 Administering Branch Actions

In DevTrack, every branch issue is created by means of a*branch action*. A branch action is the task of submitting copies of a development issue to different branches of development—product versions and builds.

The branch action performed defines the relationship between the parent development issue and its child branch issues (the*branch link type*), the lifecycle of the child branch issues (the*branch issue type*), and versions and builds in which branch issues may be submitted (the*release scope*).

In DevTrack, every branch issue is created by means of a branch action whereby copies of a parent development issue are submitted to different versions and branches of a product.

Figure 11-3: Development Issues and Branch Issues

Branch action administration is the task of creating branch actions and identifying the branch issue types, branch link types, and release folders that are applicable to that rule."

The ability to perform a branch action and, therefore, submit branch issues associated with that branch action is controlled by account type-based privileges. All branch action privileges are defined independently for each branch action.

11.4.1 Adding, Editing, and Deleting Branch Actions

Every branch action is defined by its branch issue type and default issue settings, branch link type, release folder scope, and applicable workflow states.

To add branch actions:

1. Select Enterprise Edition Features > Multi-Branch Development > Multi-Branch Link Actions in the tree panel.
2. Click the Add button.

   The New Branch Action dialog box appears.
3. Define a unique name for the action in the Action Name control.
4. Click the OK button.

   The action is displayed in the Branch Actions list. Define branch action properties, privileges, and applicable states in the tabbed pages.

To delete branch actions:

1. Select Enterprise Edition Features > Multi-Branch Development > Multi-Branch Link Actions in the tree panel.
2. Select a branch action in the branch Actions list of the Branch Management subfolder.
3. Select the Delete button.

   A DevTrack dialog box appears.
4. Click the Yes button.

   The action is deleted from the branch Actions list.

11.4.2 Defining Branch Actions

In DevTrack, every branch issue is created by means of a branch action whereby copies of a parent development issue are submitted to different versions and branches of a product.

Every branch action is defined by its branch link type, a branch issue type, release folder scope, default issue settings, and a set of applicable workflow states.

To define branch actions the project administrator must first define the product version tree, subproject hierarchy, one or more branch issue types, and one or more branch link types.

The release folder scope identifies the release folders in which the branch action may be performed. Every branch action is defined by one of three release folder scopes: the Release Folder for Same Version scope, the Release Folder For Same Product scope, and the Any Release Folder scope.

To define branch action properties:

1. Select Enterprise Edition Features > Multi-Branch Development > Multi-Branch Link Actions in the tree panel.

1. Select a branch action in the Branch Actions list.
2. Define or update the name of the branch action in the Name control.
3. Define a brief description of the branch action in the Description control.
4. To define the scope of the branch action, select an option in the Release Folder Scope area of the Properties tab.

   The Release Folder Scope area displays three options: the Release Folder for Same Version scope, the Release Folder For Same Product scope, and the Any Release Folder scope.

   The release folder scope defines the applicable release folders for the branch action.

   Release Folder for Same Version: The Release Folder for Same Version limits the scope of the branch action to the same version folder (or child folder) as that of its parent development issue.

   Release Folder for Same Product: The Release Folder for Same Product limits the scope of the branch action to the same product folder (or child folder) as that of its parent development issue.

   Any Release Folder: The Any Release Folder enables the branch action to submit branch issues to any release folder.
5. Select a branch link type form the Link Type dropdown list.

   A branch link type defines the relationship between a parent development issue and a child branch issue and determines how these issues are managed and tracked in the project.

   The Link Type dropdown list displays all administrator-defined branch link types defined in the DevTrack project.
6. Select a branch issue type in the Issue Type dropdown list.

   The branch issue type defines the issue type of the child branch issue. The administrator-defined branch issue type acts as a template for creating multiple branch issues.

The Issue Type dropdown list displays all administrator-defined branch link types defined in the DevTrack project.

7 Select a default “action” (workflow transition) for the new branch issue.

The Select a New Action dropdown list displays the actions (transitions) that are applicable to the selected branch issue type.

8 Select a default owner for the new branch issue.

The Assign To dropdown list displays the project team members and group folders as well as the user variables that may own the branch issue based on its workflow rules.

11.4.3 Defining Branch Action Privileges

In DevTrack, the ability to perform a branch action and, therefore, submit branch issues associated with that branch action is controlled by account type-based privileges.

All branch action privileges are defined independently for each branch action.

To grant the branch action privilege:

1 Select Enterprise Edition Features > Multi-Branch Development > Multi-Branch Link Actions in the tree panel.

2 Select a branch action in the Branch Actions list.

The Branch Actions list displays all administrator-defined branch actions.

3 Select the Who Can Perform tab.

4 Select one or more account types in the Account Type list.

Project team members belonging to the selected account types may perform the selected branch action.

11.4.4 Defining Branch Applicable States

To define branch action applicable states:

1 Select Enterprise Edition Features > Multi-Branch Development > Multi-Branch Link Actions in the tree panel.

2 Select a branch action in the branch Actions list of the Branch Management subfolder.

3 Select the Applicable State tab.

4 Select one or more applicable workflow states in theApplicableStatelist.

Chapter 12 Branch Event Administration

12 CHAPTER12 Branch Event Administration

In this chapter:

- Understanding Branch Events
- Administering Advanced Branch Event Workflow

12.1 Understanding Branch Events

In DevTrack, a branch event is a “subtask” of a branch issue. Branch events enable organizations to divide a development task into many different subtasks, assign each of those subtasks to a different project member, define separate start and due dates for each subtask, and manage and track each of those tasks independently in its own workflow. Every branch issues the child of a parent development issue. Branch issues may, turn be the parent to multiple branch events.

Figure 12-1: Development Issues and Branch Issues

Using branch events, development teams may quickly create multiple events that are identical with one another except for one or two event properties: the event owner."

All development events are based on administrator-defined branch event templates. A branch event template is a set of rules for managing events in project workflow. The life cycle of a branch event and the rules that determine how that event is managed in workflow are defined by the event template that was used to create it.

branch event administration, therefore, is the largely a matter of creating branch event templates, defining the rules that manage events based on that template, and determining when events based on that event template may be created in the project.

Using controls in the Branch Events folder, project administrators may define branch event templates, branch event workflow and workflow rules, advanced branch event workflow rules, branch event notification rules, and branch event escalation rules.

12.1.1 Enabling Branch Events

To enable branch event support:

1Select Enterprise Edition Features > Multi Branch Development Management > Branch Events > General Settings in the tree panel.

2Select the Branch Event Support check box.

12.1.2 Enabling In-frame Editing of Branch Events

To enable in-frame editing, select the Enable In-Frame Editing for Branch Events check box in the General Settings page of the branch event page in the branch event subfolder of the Release Management folder.

Advanced branch event workflow may be enabled and defined in a DevTrack project only if the in-frame editing feature is disabled.

12.1.3 Defining Branch Event Templates

To define branch event template properties:

1Select an event template in the branch event Template list of the Branch Event Template page of the branch event subfolder.

2To update the title of the branch event template, enter text in the Name control.

3To update the description of the branch event template, enter text in the Description control.

4To update the event variable of the branch event template, click the Change button and select an option in the Variable dropdown list of the dialog box.

The Variable dropdown list displays all of the administrator-defined event variables created in the Event Variable page of the branch event subfolder. For more information see “Administering Event Variables”.

5To define the workflow state settings, double-click the State control and define the control label, visibility, and default owner options:

- To update the title of the State control, enter a new title in the Label control.
- To define the State control as visible or invisible, select or deselect the Visibility check box.
- To define the default workflow state, select an option from the State dropdown list.

6To define the Owner settings, double-click the Owner control and define the control label, visibility, and default owner options:

- To update the title of the Owner control, enter a new title in the Label control.
- To define the Owner control as visible or invisible, select or deselect the Visibility check box.
- To define the default owner, select an option from the Owner dropdown list.

7To define the Start Date control settings, double-click the Start Date control and define the default value, visibility, and time stamp options:

- To define default time settings and offset settings, select the Current Date/ Time radio button, and define the offset in days and hours.
- To define the Start Date control as visible or invisible, select or deselect the Visibility check box.
- To ensure that the Start Date control displays the time, select the With Time check box.

8 To define the Due Date control settings, double-click the Due Date control and define the default value, visibility, and time stamp options:

- To define default time settings and offset settings, select the Current Date/ Time radio button, and define the offset in days and hours.
- To define the Due Date control as visible or invisible, select or deselect the Visibility check box.
- To ensure that the Due Date control displays the time, select the With Time check box.

9 To define the Last Date Modified settings, double-click the Last Date Modified control and define the control label and visibility options:

- To update the title of the Last Date Modified control, enter a new title in the Label control.
- To define the Close Date control as visible or invisible, select or deselect the Visibility check box.

10 To define the Close Date settings, double-click the Close Date control and define the control label and visibility options:

- To update the title of the Close Date control, enter a new title in the Label control.
- To define the Close Date control as visible or invisible, select or deselect the Visibility check box.

12.1.4 Adding, Editing, and Deleting Branch Event Templates

Every branch event is based on an administrator-defined branch event template. An branch event template is a set of default event properties, attachment rules, and event access controls that are copied over to every event based on that event template.

Built into each branch event template is an administrator-defined event variable. branch event templates may be used to generate multiple events at a time. Each branch event may be assigned to a different set of applicable owners and applied to different product versions or builds.

To add a branch event template:

1 Click the New button in the Branch Event Template page of the branch event subfolder.

The Add a New Event Template dialog box appears.

2 Define a unique, descriptive name for the event template in the Name text box control.

The branch event option is automatically selected in the read-only Event Type dropdown list.

3 Select an event variable from the Variable dropdown list.

The Variable dropdown list displays all of the administrator-defined event variables created in the Event Variable page of the Development Event subfolder.

4 Define a brief description of the event in the Description control.

5 Click the OK button.

To delete event templates:

1 Select an event template in the branch event Template list of the Branch Event Template page of the branch event subfolder.

2 Click the Delete button in the Event Templates area.

The DevSuite Admin dialog box appears.

3 Click the Yes button.

The event template disappears from the Event Templates list.

12.1.5 Defining Branch Event Template Attachment Rules

Branch event template attachment rules determine whether project members may add attachments to branch event instances based on an event template, define the types of knowledge topics that may be attached to those test events, and define inheritance rules for branch event titles and descriptions.

Every branch event is based on an administrator-defined branch event template and inherits its properties from the event template used to create it. branch event template properties include four attachment rules:

- Attachment requirement rules
- Knowledge topic type attachment rules
- Title inheritance attachment rules
- Description inheritance attachment rules

In the DevTrack client the Edit Event Template page enables project members to define the title and description of an event. Project administrators may define the default options displayed in the Title control and the Description control using controls in the Attachment tab.

Project administrators may define attachment rules for each event template in the Attachments tab of the Event Template page in the branch events subfolder of the Release Management folder.

To define event template attachment rules:

1 Select a branch event template in the Event Templates list of the Event Templates page of the branch events subfolder.

Attachment rules must be defined separately for each branch event template.

2 Select the Attachments tab.

3 Select an attachment option in the Attachment tab of the Event Template page.

Project administrators may define attachment rules that require, forbid, or allow project members to add attachments to events for each event template.

- The *Mandatory option* requires project members to add attachments to events based on the event template.
- The *Optional option* enables that project members to add attachments to events based on the event template.
- The *None option* disables attachments in events based on the event template.

Attachments are disabled by default in all branch event templates.

4 Select all appropriate attachment type options in the Attachment Type control.

Attachment types include:

- New document
- New HTML link
- Existing Knowledge document
- Existing Knowledge topic
- Existing HTML link

5 Select a title option in the Title Option area.

Title settings represent the default title for an event. With the appropriate access users can edit the title of an event at any time.

- The Same as Event Template option makes the title of the parent event template the default title of every event based on that event template. If the title of the parent event template is New Event, the default title of the event will also be New Event.
- The Copy from Linked Knowledge Item option makes the title of a knowledge item the title of the event to which it is linked. If the title of the attached knowledge item is Unix Usability Testing, the default title of the event will also be Unix Usability Testing.
- The Append from Linked Knowledge Item option appends the title of a linked knowledge item to the title of the parent event template to create the default title of an event. If the title of the parent event template is New Event and the title of the attached knowledge item is Unix Usability Testing, the default title of the event will be New Event, Unix Usability Testing.

6 Select a description option in the Description area.

Description Option settings represent the default title for an event. With the appropriate access users can edit the description of an event at any time.

- The Same as Template Description option makes the description of the parent event template the default description of every event based on that event template. If the description of the parent event template is Test Event Description, the default title of the event will also be Test Event Description.
- The Copy from Linked Knowledge Item option makes the body of a knowledge item the description of the event to which it is linked. If the description of the attached knowledge item is Test Plan Description, the default description of the event will also be Test Plan Description.
- The Append from Linked Knowledge Item option appends the title of a linked knowledge item to the title of the parent event template to create the default title of an event. If the title of the parent event template is x and the title of the attached knowledge item is y, the default title of the event will be Test Event Description, Test Plan Description.

12.1.6 Defining Branch Event Workflow States

Branch event workflow defines the life cycle of a branch event. Event workflow states represent the various stages of the event life cycle from its initial creation to its closure and one or more intermediary steps.

All event workflow states are defined on a template-by-template basis. The event life cycle and event workflow rules used to manage and track a branch event instance are determined by the event template used to create that branch event.

The life cycle of branch event may consists of any number of workflow states. Every workflow state is defined by one of three statuses:

- A Open status indicates that the test event workflow state is open.
- A Closed Successfully status indicates that the test event workflow state is closed and the feature tested in that event passed.
- A Closed Failed status indicates that the test event workflow state is closed and the feature tested in that event failed.

Project administrators may define the event life cycle of an event template in the State tab of the Event Template page in the branch events subfolder.

To create event states:

1 Click the New button in the State tab of the Event Template page.

The EventState dialog box appears.

2 Define a unique name for the event state.

3 Define the event state status.

Every event state has one event state status:

- To define an event state as an open event, select the Open status option.

- To define an event state as a successfully closed event, select the Closed Successfully status option.
- To define an event state as a failed closed event, select the Closed Failed status option.

4Click the OK button.

12.1.7 Managing Branch Event Access Rules

Event access controls determine which project members may view, submit, edit, and delete branch events in a DevTrack project. Project administrators may define event access rules for every event template they create in a DevTrack project.

The event access rules that a project administrator defines in an event template are inherited and enforced by every event instance based on that event template.

Project administrators may grant event access controls to project members based on their account type, group, or one of four system user variables:

- The {Event Submitter} system user variable represents the project member that submitted the branch event.
- The {Event Owner} system user variable represents the project member that currently owns the branch event.
- The {Parent Issue Submitter} system user variable represents the project member that submitted the parent event of the branch event.
- The {Parent Issue Owner} system user variable represents the project member that submitted the current branch event.

Project administrators may define five event access control rules in each event template.

Can View privilege: The*Can View privilege*enables project members to view event based on an event template.

Can Submit privilege: The*Can Submit privilege*enables project members to submit events based on an event template.

Can Edit Open Event privilege: The*Can Edit*Open Event privilege enables project members to edit closed events based on an event template.

Can Edit Closed Event privilege: The*Can Edit*Closed Event privilege enables project members to edit closed events based on an event template.

Can Delete privilege: The*Can Delete*Event privilege enables project members to delete events based on an event template.

Project administrators may define event access controls for each event template in the User Access tab of the Event Template page in the branch event subfolder.

To define event access controls:

1Select an event template in the Event Templates list.

2Select the User Access tab.

3Select an access control option in the Applicable Account Types list of the User Access tab.

4Select an option in the in the Applicable Account Types list.

- The Select the All Account Types radio button enables administrators to select all of the account types displayed in the list.
- The Select the Selected Account Types option radio button enables administrators to select only some of the account types displayed in the list.

12.1.8 Administering Branch Event Workflow

DevTrack branch event workflow enables project administrators to define sophisticated workflow rules for managing the child test events of a DevTrack QA test issue.

Event workflow utilizes the DevTrack workflow engine to control the applicable events for each issue workflow state and to close events based on issue workflow states.

In each QA test issue trigger state, the child branch events are defined by an auto close rule:

- Can be Open rule
- Must be Closed rule
- Must be Closed (Auto Close) rule

Project administrators may optionally define advanced event workflow rules using tools in the Branch Event Advanced Workflow page.

12.1.9 Defining Branch Event Automatic Close Workflow Rules

To define close rules:

1Select a subworkflow page in the branch event subfolder of the DevSuite Admin tree window.

One QA test workflow page is displayed for each administrator-defined issue subworkflow.

2Select a trigger state in theTriggerStatelist of the Event Workflow page in the branch event subfolder in the Release Management folder.

3Select an event template in the Event Template list.

4Click the Define Event Close Rule button.

The Define Event Close Rule appears.

5Select an option from the Auto Close Rule dropdown list control.

The Auto Close Rule dropdown list displays three options:

- Can be Open
- Must be Closed
- Must be Closed (Auto Close)

6Click the OK button.

12.2 Administering Advanced Branch Event Workflow

Advanced branch event workflow rules define changes to the issue worKflow state of parent QA test issues based on system-defined event status triggers.

System-defined event triggers include:

- {All events of this template closed -- Success}
- {All events of this template closed -- Any Fail}
- {First event of this template closed -- Fail}
- {Any event of this template closed -- Fail}
- Not started {Open state}
- In progress {Open state}
- Closed - test passed {Close--Successfully}
- Closed - test failed {Close--Failed}

Note: Advanced QA test workflow rules override issue workflow rules. Changes to the issue workflow state of a QA test issue are not restricted by issue workflow rules when those changes are triggered by an advanced branch event workflow rule.

Project administrators may define advanced event workflow rules in the Advanced Event Workflow page of the branch event subfolder in the Release Management folder.

12.2.1 Disabling In-frame Editing

Advanced branch event workflow may be enabled and defined in a DevTrack project only if the in-frame editing feature is disabled.

To disable in-frame editing, deselect the Enable In-Frame Editing for Branch Events check box in the General Settings page of the branch event page in the branch event subfolder of the Release Management folder.

12.2.2 Defining branch event Advanced Workflow Rules

To define branch event advanced workflow rules:

1Select a branch event template in the branch event tree control in the Branch Event Advanced Workflow page of the branch event subfolder.

2Select an event trigger in the Event Trigger list.

3Select an option from the Issue State Change dropdown list.

4The Issue State Change dropdown list displays all applicable issue workflow state changes for the selected branch event template and event trigger.

- To define the change as mandatory, select the Mandatory Change radio button.
- To define the change as optional yes, select the Optional Change (Yes) radio button.

To define the change as optional no, select the Optional Change (No) radio button.