AnyLink
Studio Guide

AnyLink 7

Document Information

Title: AnyLink Studio Guide

Publication Date: 2022-03-29

Software Version: AnyLink 7

Edition: v2.1.4

Website

http://www.tmaxsoft.com

Table of Contents

About This Document xi

Chapter 1. Introduction 1

1.1. Starting Studio 1

1.2. Screen Layout 5

1.2.1. Menus 6

1.2.2. Views 12

1.2.3. Navigator 13

1.3. Resources 15

1.4. Creating a Project 17

Chapter 2. BizTx/BizTx Group Editor 19

2.1. Creating a BizTx/BizTx Group 19

2.2. BizTx/BizTx Group Editor 21

2.2.1. BizTx/BizTx Group Info 21

2.2.2. BizTx/BizTx Group Options 32

2.2.3. Parsing Info 36

2.2.4. Parsing Options 38

2.2.5. XML Viewer 40

Chapter 3. Message Editor 41

3.1. Creating a Message 41

3.2. Message Editor 42

3.2.1. Message Configuration 43

3.2.2. Message Format 47

Chapter 4. Outbound Rule Editor 55

4.1. Overview 55

4.2. Custom Log Outbound Rule 55

4.2.1. Creating a Custom Log Outbound Rule 55

4.2.2. Configuring a Custom Log Outbound Rule 56

4.2.3. Custom Log Example 61

4.3. Batch Outbound Rule 66

4.3.1. Creating a Batch Outbound Rule 66

4.3.2. Configuring a Batch Outbound Rule 68

Chapter 5. Multi-Binding Rule Editor 89

5.1. Multi-Binding Rule 89

5.2. Creating a Multi-Binding Rule 89

5.3. Configuring a Multi-Binding Rule 90

5.4. Mapping 92

5.4.1. Adding/Deleting a Mapping 92

Chapter 6. Service Flow Editor 95

6.1. Overview 95
6.2. Creating a Service Flow 95
6.3. Service Flow Diagram 96
6.4. Activity 99
6.4.1. Configuring an Activity 100
6.4.2. Configuring a Specific Activity Type 116
6.5. Event 122
6.5.1. Configuring an Event 123
6.5.2. Event Configuration by Type 124
6.6. Gateway 127
6.7. Swimlane/Block/Annotation 129
6.7.1. Block Activity 129
6.8. Utility 130
6.8.1. SESSION CLOSE ACTIVITY 131
6.8.2. SESSION ABORT ACTIVITY 132
6.8.3. NETWORK MANAGEMENT ACTIVITY 132
6.8.4. MANAGER APPROVAL ACTIVITY 134
6.8.5. HTTP MULTIPART ACTIVITY 136
6.8.6. BATCH PROGRESS ACTIVITY 138
6.8.7. TCP MESSAGE SPLIT ACTIVITY 141
6.8.8. CONTROL SESSION ACTIVITY 143

Chapter 7. User Class/Handler Editor 145
7.1. User Class 145
7.1.1. Creating a User Class 145
7.1.2. User Class Implementation 145
7.1.3. getEnv Function 146
7.2. Handler 147
7.2.1. Creating a Handler 147
7.2.2. Handler Class Implementation 148

Chapter 8. Common Business Project 163
8.1. Creating a Common Business Project 163
8.2. Creating a Common BizTx/BizTx Group 165

Chapter 9. Deployment & Undeployment 167
9.1. Overview 167
9.2. Resource Deployment 167
9.2.1. Deployment Unit 167
9.2.2. Deployment 168
9.3. Resource Undeployment 170
9.3.1. Undeployment Unit 170
9.3.2. Undeployment 170
9.4. Shared Library Deployment & Undeployment 172
9.4.1. Deploying a Shared Library 172

9.4.2. Undeploying a Shared Library 172
9.4.3. Adding a Library 173
9.4.4. Downloading a Library 174

Chapter 10. Resource Search & Download 175
10.1. Overview 175
10.2. Resource Search & Download 175

Appendix A. Dialog Boxes 177
A.1. Message Dialog Boxes 177
A.1.1. Select Message 177
A.1.2. Select Message Field 178
A.2. Mapping Dialog Boxes 178
A.2.1. Mapping Wizard 179
A.2.2. Using Mapping Wizard 182
A.3. Resource Dialog Boxes 190
A.3.1. Create Resource 190
A.3.2. Search Resource 191

Index 193

List of Figures

[Figure 1.1] Studio Loading Window 1
[Figure 1.2] Workspace Launcher 2
[Figure 1.3] Login Window 2
[Figure 1.4] Edit Login Dialog Box 3
[Figure 1.5] Studio Main Screen 4
[Figure 1.6] Studio Main Screen - Login Information 4
[Figure 1.7] Studio Initial Screen 5
[Figure 1.8] [Debug] Tab 13
[Figure 1.9] [Relationships] Tab 13
[Figure 1.10] AnyLink Navigator 13
[Figure 1.11] Navigator Options - Resource Display Format 15
[Figure 1.12] Creating a Project 17
[Figure 2.1] Create BizTx/BizTx Group Window - Root BizTx Group 20
[Figure 2.2] Create BizTx/BizTx Group Window - BizTx Group (Interim) & BizTx (Leaf) 21
[Figure 2.3] [BizTx (Group) Info] - [Basic Info] 22
[Figure 2.4] [BizTx (Group) Info] - [Invocation Service (Flow)] 23
[Figure 2.5] [BizTx (Group) Info] - [Invocation Service (Outbound Rule)] 24
[Figure 2.6] [BizTx (Group) Info] - [Invocation Service (Multi-binding)] 25
[Figure 2.7] [BizTx (Group) Info] - [Request Message] 26
[Figure 2.8] Select Request Message Dialog Box 27
[Figure 2.9] [BizTx (Group) Info] - [Error Response Mode (FLOW)] 29
[Figure 2.10] Search Mapping ID (Search Resource) 31
[Figure 2.11] [BizTx (Group) Info] - [Error Response Mode (MAPPING)] 31
[Figure 2.12] [BizTx (Group) Info] - [BizTx (Group) Options] 32
[Figure 2.13] [BizTx (Group) Options] - [BizTx Priority] - Message Options 35
[Figure 2.14] Select Message Field Dialog Box 35
[Figure 2.15] [Parsing Info] - [Parsing Definition] - BizTx Identifier Code 36
[Figure 2.16] [Parsing Info] - [Parsing Definition] - Child BizTx Identification Method 37
[Figure 2.17] Encoding Type 38
[Figure 2.18] Message Type 39
[Figure 2.19] XML Self-Closing Setting 40
[Figure 3.1] Create Message Dialog Box 41
[Figure 3.2] Message Editor 42
[Figure 3.3] Create Message Dialog Box 43
[Figure 3.4] Message Editor - Message Configuration 44
[Figure 3.5] Message Editor - Message Format 47
[Figure 4.1] Create Custom Log Outbound Rule Window 56
[Figure 4.2] Define Outbound Rule 56
[Figure 4.3] Request Message 57
[Figure 4.4] [Log Table] - Log Table Name 58

[Figure 4.5] [Log Table] - [Custom Log Table Info] - Mapping Dialog Box 59
[Figure 4.6] [Log Table] - Custom Log Table Info 59
[Figure 4.7] [Log Table] - Custom Log Items Dialog Box 60
[Figure 4.8] [Log Table] - INSERT SQL Statement 61
[Figure 4.9] Create Custom Log Outbound Rule Window 62
[Figure 4.10] Outbound Rule Definition 62
[Figure 4.11] Request Message 63
[Figure 4.12] Log Table 63
[Figure 4.13] Log Table Mapping 64
[Figure 4.14] Outbound Call - Outbound Rule (Service) 65
[Figure 4.15] Request Custom Log Mapping 65
[Figure 4.16] Custom Log 66
[Figure 4.17] Create Batch Outbound Rule Window 67
[Figure 4.18] Outbound Rule Definition 69
[Figure 4.19] Request Message 69
[Figure 4.20] Response Message 70
[Figure 4.21] File List 70
[Figure 4.22] Create Batch Style Dialog Box 70
[Figure 4.23] Create Main File Dialog Box 71
[Figure 4.24] Input Mapping Dialog Box 72
[Figure 4.25] Output Mapping Dialog Box 73
[Figure 4.26] Batch Style Specifications 73
[Figure 4.27] CMS Style 74
[Figure 4.28] DACOM_EDI Style 76
[Figure 4.29] KIDI Style 77
[Figure 4.30] SSNW Style 78
[Figure 4.31] KLIA Style 79
[Figure 4.32] SONBO Style 80
[Figure 4.33] KNIA Style 81
[Figure 4.34] KFB Style 82
[Figure 4.35] FTP Style 83
[Figure 4.36] KCREDIT Style 84
[Figure 4.37] KSD Style 85
[Figure 4.38] KOTA Style 86
[Figure 4.39] Code Information 87
[Figure 4.40] Output Mapping 88
[Figure 5.1] Create Multi-Binding Rule Window 90
[Figure 5.2] Multi-Binding - Basic Info 90
[Figure 5.3] Multi-Binding - Request Message 91
[Figure 5.4] Multi-Binding - Binding Options 91
[Figure 5.5] Add/Delete Mapping Tab 93
[Figure 6.1] Flow Diagram 95
[Figure 6.2] Service Flow Page 96

[Figure 6.3] Service Flow Page - Toolbar 96
[Figure 6.4] Service Flow - Palette 98
[Figure 6.5] Input/Output of an Activity 100
[Figure 6.6] Activity Preferences Menu 101
[Figure 6.7] Activity Preferences - Basic Info 102
[Figure 6.8] Activity Preferences - Outbound Rule 104
[Figure 6.9] Activity Preferences - Skip Condition 105
[Figure 6.10] Activity Preferences - Select Variable 106
[Figure 6.11] Activity Preferences - Parameter Settings 107
[Figure 6.12] Activity Preferences - Add New Process Variable 108
[Figure 6.13] Activity Preferences - Activity Handler 110
[Figure 6.14] Service List Dialog 111
[Figure 6.15] Activity Preferences - Activity Error Handler 112
[Figure 6.16] Add New Exception Handler 112
[Figure 6.17] Activity Preferences - Activity Loop 113
[Figure 6.18] Activity Preferences - Standard Loop - Standard Loop 114
[Figure 6.19] Activity Preferences - Activity Loop - Multiple Instances 115
[Figure 6.20] Activity Preferences - User Class Activity 117
[Figure 6.21] Activity Preferences - Activity Mapping 118
[Figure 6.22] Mapping Dialog Box 119
[Figure 6.23] Activity Preferences - Reply Message 120
[Figure 6.24] Activity Preference - Subflow 121
[Figure 6.25] Event Preferences Menu 123
[Figure 6.26] Event Preferences - Basic Info 123
[Figure 6.27] Event Preferences - Timer 125
[Figure 6.28] Error Preferences - Error Event 126
[Figure 6.29] Event Preferences - Link Event 127
[Figure 6.30] Gateway Preferences 128
[Figure 6.31] Block Activity Preferences 130
[Figure 6.32] Session Close Preference 131
[Figure 6.33] Select Session ID Dialog Box 132
[Figure 6.34] Network Management Preference 132
[Figure 6.35] Search Network ID (Name) Dialog Box 133
[Figure 6.36] Network Status Mapping 133
[Figure 6.37] Manager Approval Preference 134
[Figure 6.38] Select Variable Field 135
[Figure 6.39] Terminal/User Information Mapping 136
[Figure 6.40] HTTP Multipart Preference 137
[Figure 6.41] Multipart Mapping 138
[Figure 6.42] Batch Progress Preference 139
[Figure 6.43] Batch Progress Mapping 140
[Figure 6.44] Batch Process Result Mapping 141
[Figure 6.45] TCP Message Split Preference 142

[Figure 6.46] Select Session ID 142
[Figure 6.47] Message Length Mapping 143
[Figure 6.48] Control Session Preference 144
[Figure 6.49] Session Information Mapping 144
[Figure 7.1] Create User Class 145
[Figure 7.2] Create Handler - Activity Handler 148
[Figure 7.3] Adding Message Validation Handler (WebAdmin) 158
[Figure 7.4] Message Validation Result - Dialog Box 158
[Figure 7.5] Message Validation Result - Editor 158
[Figure 8.1] Deploy Biz System 163
[Figure 8.2] Create Project 164
[Figure 8.3] Create Common BizTx/BizTx Group - Root BizTx Group 165
[Figure 8.4] Create Common BizTx/BizTx Group - BizTx Group (Interim), BizTx (Leaf) 166
[Figure 8.5] Checking the Result 166
[Figure 9.1] Warning Message When Assigning a Business System 168
[Figure 9.2] Deploy Biz System 168
[Figure 9.3] Deploy Options 169
[Figure 9.4] Undeploy Options 171
[Figure 9.5] Deploy Shared Library 172
[Figure 9.6] Undeploy Shared Library 173
[Figure 9.7] Download Shared Library 174
[Figure 10.1] Download Resource 176
[Figure A.1] Select Request Message Dialog Box 177
[Figure A.2] Select Message Field Dialog Box 178
[Figure A.3] New Mapping Dialog Box 179
[Figure A.4] User Class Properties Dialog Box 181
[Figure A.5] Java Code Editor Dialog Box 182
[Figure A.6] Source Tree Context Menu 183
[Figure A.7] Manage Indexes Dialog Box 184
[Figure A.8] Target Tree Context Menu 185
[Figure A.9] Edit Context Menu 186
[Figure A.10] Select Function 186
[Figure A.11] Properties Pane 189
[Figure A.12] Create Resource Dialog Box 190
[Figure A.13] Search Resource Dialog Box 191

About This Document

Intended Audience

This document is intended to provide developers and administrators with information about the basic configuration and usage of Tmax AnyLink® (hereafter AnyLink) Studio.

Required Knowledge

To fully understand this guide, users need to have an understanding of the following:

- Basic understanding of AnyLink WebAdmin (Refer to *AnyLink WebAdmin Guide*.)
- Adapter
- Gateway

Document Scope

This guide does not contain information about Java EE or Java specifications. For such information, refer to the relevant Java documents.

This guide only covers the basic features and use of AnyLink studio. For information about adapters and development, refer to each adapter guide.

Document Organization

This guide consists of 10 chapters and one appendix.

Descriptions of each are as follows:

- Chapter 1: Introduction

  Describes the screen layout and basic functions.

- Chapter 2: BizTx/BizTx Group Editor

  Describes the functions and uses of the BizTx/BizTx Group editor.

- Chapter 3: Message Editor

  Describes the functions and uses of the Message editor.

- Chapter 4: Outbound Rule Editor

  Describes the functions and uses of the Outbound Rule editor.

- Chapter 5: Multi-Binding Rule Editor

  Describes the functions and uses of the Multi-Binding Rule editor.

- Chapter 6: Service Flow Editor

  Describes the functions and uses of the Service Flow editor.

- Chapter 7: User Class/Handler Editor

  Describes the functions and uses of the User Class/Handler editor.

- Chapter 8: Common Business Project

  Describes the functions and uses of the Common BizTx/BizTx Group function.

- Chapter 9: Deployment & Undeployment

  Describes how to deploy/undeploy resources to/from DIS.

- Chapter 10: Resource Search & Download

  Describes how to search for and download various resources used in the system.

- Appendix A: Dialog Boxes

  Describes how to use various dialog boxes in Studio.

Conventions

Convention	Meaning
<AaBbCc123>	Program source code file name
<Ctrl>+C	Hold down the Ctrl key and press the C key
[Button]	Name of a GUI button or menu
Bold	Emphasis
Italics	Reference to another guide
" " (Double quotation marks)	Reference to a chapter or section in this or another guide
'Input'	User input on the screen
`Hyperlink`	Email account, website, or a reference to other chapters or sections
>	Proceeding order of menu
+----	Files or directories exist in this directory
|----	Files or directories do not exist in this directory
Note	Reference or caution
[Figure 1.1]	Figure caption
`AaBbCc123`	Commands, screen output after executing command, or sample code
{ }	Required items
[ ]	Optional items
|	Selective items

System Requirements

Category	Requirement
Platform	Solaris 9-11
	HP-UX 11.x, 11i, 11iV2
	AIX 5L, 6L, 7L
	Linux Kernel 2.6 or later
	Windows 7 (32-bit, 64-bit)
Server	More than 1 GB RAM recommended (At least 512 MB)
	At least 500 MB hard disk space
Studio	Windows 7 (64-bit)
	1 GB RAM recommended (At least 512 MB)
	At least 512 MB hard disk space
Remote Agent	512 MB RAM recommended (At least 256 MB)
	At least 512MB hard disk space
Software	JDK 7.0
	JEUS 7 (Fix#3)
Supported Browsers for WebAdmin	IE 10 or later
	Chrome 41 or later
Supported Databases	Oracle 10g, 11g, 12c
	Tibero 6 FixSet03 or later

Chapter 1. Introduction

This chapter describes the screen layout and basic functions.

1.1. Starting Studio

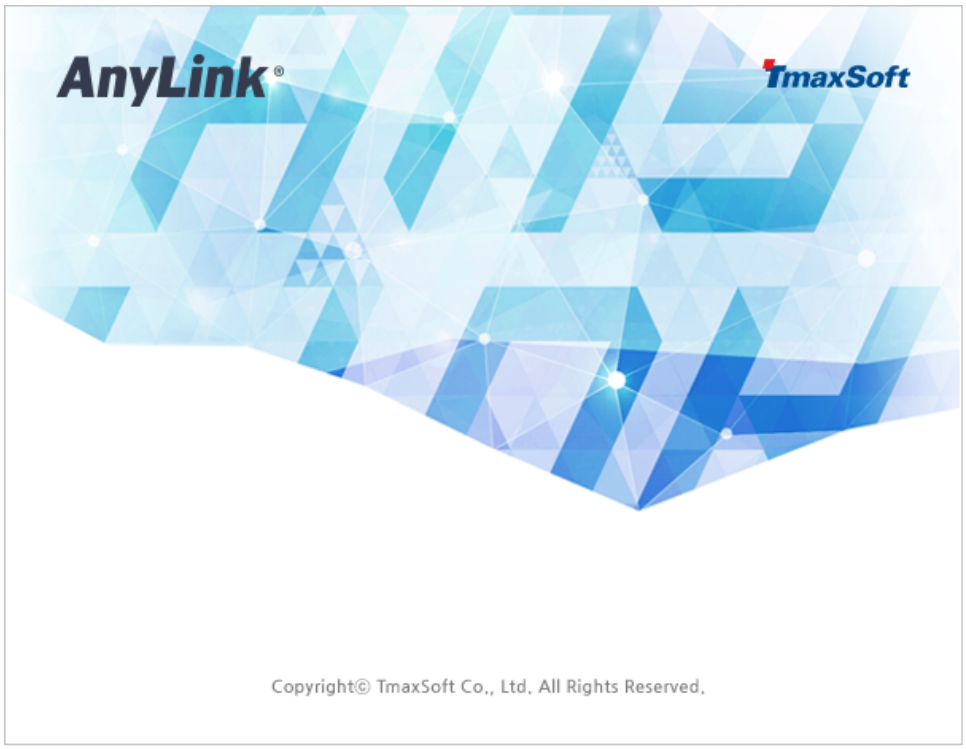
AnyLink Studio, a product line of AnyLink, can be used to easily create and deploy service flows and various adapter rules. Users can easily edit and manage flows and rules using the Studio GUI. Refer to the relevant guides for information about flow services and various types of adapters.

The following are the steps for starting AnyLink Studio.

1. Click the **Desktop Shortcut** (AnyLink.exe).

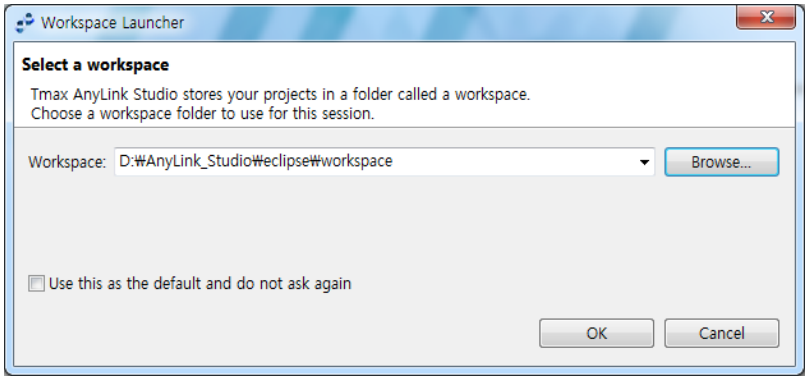
2. While Studio is loading, the AnyLink logo, copyright information, and progress are displayed.

[Figure 1.1] Studio Loading Window



3. The Workspace Launcher for configuring a workspace starts. To execute AnyLink Studio, click **[OK]**.

[Figure 1.2] Workspace Launcher



4. Click  at the top to open the **Login** page. Studio requires a login for user history management. Any user changes saved to the server are managed as user change logs and are used to detect any data conflicts between users.

   Enter the Server Name, User ID, Password, and Server IP Address and Port Number, and then click **[Log In]** to connect to the AnyLink server.

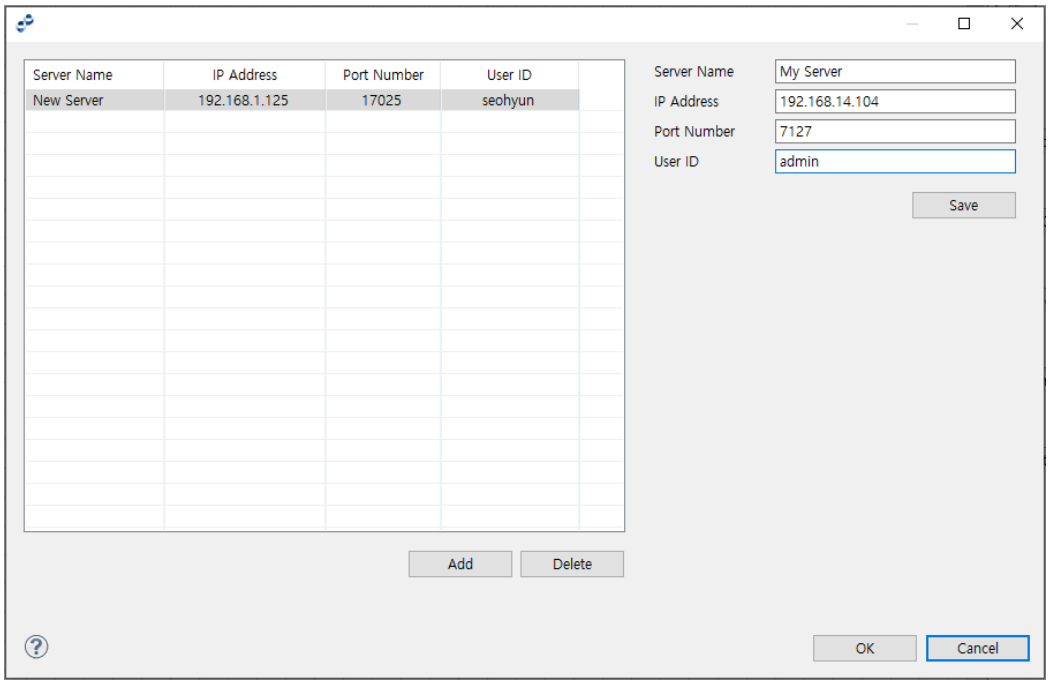
[Figure 1.3] Login Window

Item	Description
Server Name	Login info (address, port, and user ID). If server connection is not established using the login icon, Studio can be used in the local mode.

Item	Description
IP Address	DIS IP address.
Port Number	Port number (default value: 7127).
User ID	User ID created in WebAdmin.
Password	Password created in WebAdmin.

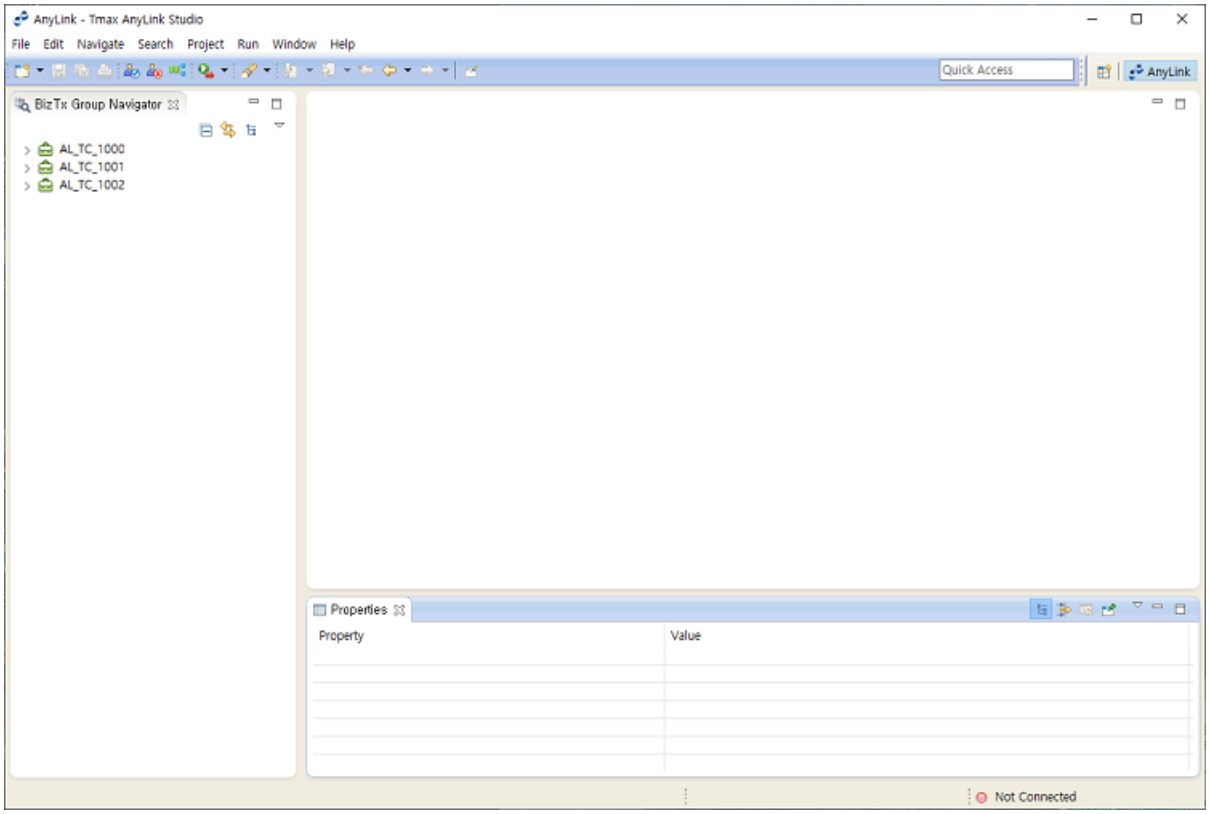
Click **[Edit]** from the **Login Window** ([Figure 1.3]) to open the window for adding, editing, or deleting the login information.

[Figure 1.4] Edit Login Dialog Box



5. Once logged in successfully, the main screen is displayed.

[Figure 1.5] Studio Main Screen



The login information is displayed at the bottom.

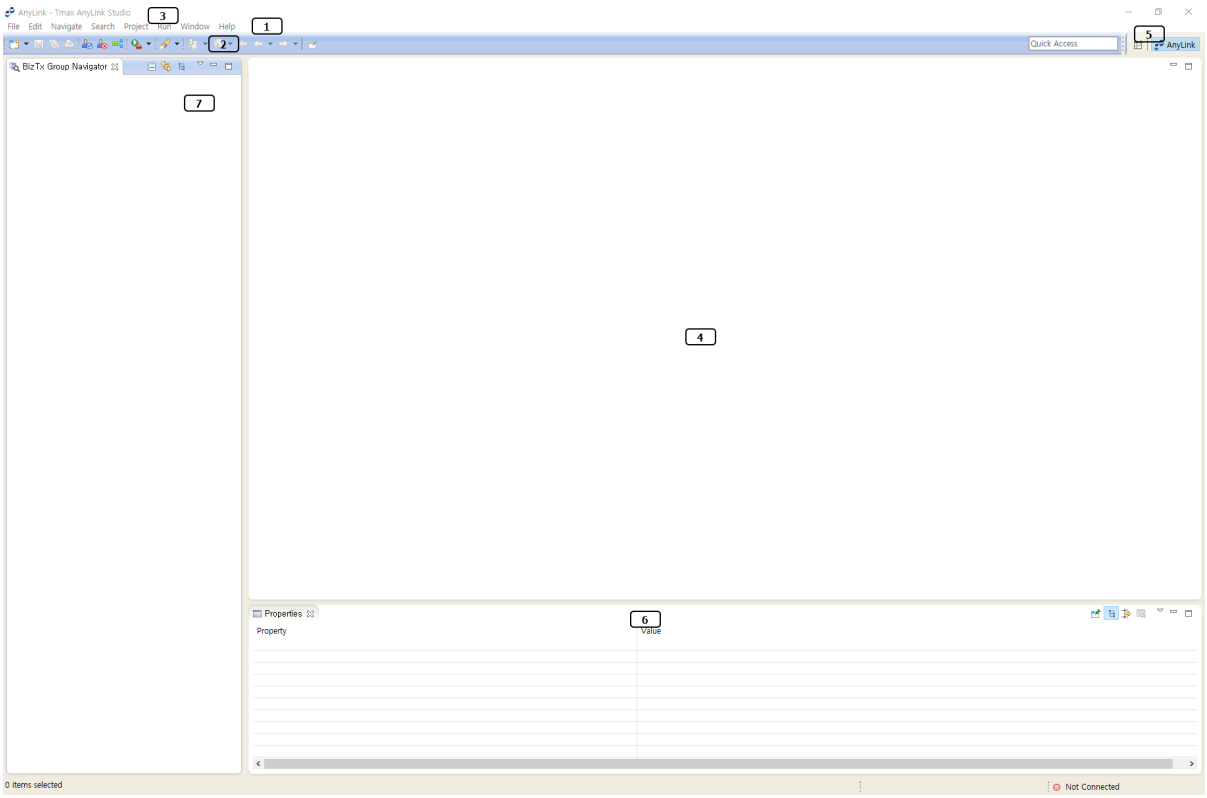
[Figure 1.6] Studio Main Screen - Login Information

1.2. Screen Layout

Studio consists of a main page called the Workbench and multiple Perspectives (layout).

The following is the initial screen of AnyLink Studio.

[Figure 1.7] Studio Initial Screen



- **(1) Menu**

  Displays Workbench menu items. For detailed description of each menu item, refer to "1.2.1. Menus".

- **(2) Tool bar**

  Displays menu items for quick menu access. Toolbar can be configured from **[Window] > [Customize Perspective]**.

  The following are the toolbar items available in AnyLink.

Icon	Description
 (Login)	Connect to DIS.
 (Logout)	Disconnect from DIS.
 (Outline)	Resource relationship view.

- **(3) Title bar**

  The Workbench title is displayed in the following format.

  ```
  {Current Perspective Name} – {Current Working File Name} – Tmax AnyLink Studio
  ```

- **(4) Editor Panel**

  Displays various editors including the Service Flow Diagram editor.

- **(5) Perspective**

  Displays Perspective shortcut buttons, and the active Perspective is highlighted. Right-click to display the context menu to customize the toolbar.

- **(6) View**

  Allows the user to search for resources and edit their properties. For detailed description of each menu item, refer to "1.2.2. Views".

- **(7)Navigator**

  Provides editor support and is used mostly for searching for Presentation or Workbench data. For detailed information, refer to "1.2.3. Navigator".

1.2.1. Menus

Workbench consists of the following menu items.

- **[File]** menu

  Create and save a project or file.

Menu	Description
[New]	Create a new project or file.
[Open File]	To open an editor in Eclipse, double-click on a file in the Workspace from the **Service Explorer**, or select a file in the **Service Explorer** and then click **[Open File]** from the context menu. Use command to open a file that is not in the Workspace.
[Close]	Close the active editor.
[Close All]	Close all open editors.
[Save]	Save the active editor file.
[Save As]	Save the active editor file under another name.

Menu	Description
[Save All]	Save the files in all open editors.
[Revert]	Revert the file in the active editor to the last saved version. Any changes made after the last save is lost.
[Move]	Drag and drop the selected file or folder to another location in the **Service Explorer**.
[Rename]	Rename the selected file in the **Service Explorer**.
[Refresh]	Refresh the Workspace data in the **Service Explorer** to sync with the file system. Projects in the Workspace reflect the current file system state. Since files in the Workbench can be modified externally, use the **Refresh** command to sync the files with those in the file system.
[Convert Line Delimiters To]	Convert the line delimiter of the active editor to that of Windows, Unix, or Mac OS 9. Windows is the default value.
[Print]	Print the contents from the active editor.
[Switch Workspace]	Switch to a new or an existing Workspace.
[Restart]	Restart Studio.
[Import]	Use the **Wizard** to import a project or file into the Workspace.
[Export]	Use the **Wizard** to export a project or file from the Workspace. **Wizard** supports various export formats.
[Properties]	Display the properties dialog box for the selected resource in the **Service Explorer**.
[Recent Files]	Display a list of recently used files in the Workbench. Set the maximum number of files to displays under **[Recent Files]** in the **'Size of recently opened files list'** from the **[Window] > [Preferences] > [General] > [Editors]** menu.
[Exit]	Close the Workbench. All open files and Perspective states are automatically saved when the Workbench closes so that it can be reopened preserving the previous state.

- **[Edit]** menu

  Edit a file in the editor.

Menu	Description
[Undo]	Undo the last change.

Menu	Description
[Redo]	Redo the last undo.
[Cut]	Copy the selected content to the clipboard and delete the original.
[Copy]	Copy the selected content to the clipboard.
[Paste]	Paste the text or object from the clipboard to the cursor location in the active editor or view.
[Delete]	Delete selected content.
[Select All]	Select all text or objects in the active editor.
[Find/Replace]	Find and replace an expression with another expression in the active editor.
[Add Bookmark]	Add a bookmark for the row at the cursor location. To see the bookmark list, go to **[Window] > [Show View] > [Other]** to open the **Show View** dialog box and then select **[General] > [Bookmarks]**.
[Add Task]	Add a task to the row at the cursor location of the active editor. To see the bookmark list, go to **[Window] > [Show View] > [Other]** to open the **Show View** dialog box and then select **[General] > [Tasks]**.

- **[Navigate]** menu

  Search and find resources and items in the Workbench.

Menu	Description
[Go Into]	If the Tree View is active, the selected item becomes the root node. The root node is an Invisible Root that is not displayed.
[Go To]	– Back: Similar to HTML browser's **[Back]** button. Returns the active view to the state before executing the **Go Into** command. – Forward: Similar to HTML browser's **[Forward]** button. Returns the active view to the state before executing **Go To** > **Back** command. – Up One Level: Makes the current node's root as the parent root to navigate to a higher level in the tree view.
[Open from Clipboard]	Open an item from the clipboard.
[Show In]	Allow the contents in the active view to be seen from another view.
[Next]	Navigate to the next item of a view or list from the active view. For example, in a Search view, navigate to the next search item.

Menu	Description
[Previous]	Navigate to the previous item of a view or list from the active view. For example, in a Search view, navigate to the previous search item.
[Last Edit Location]	Navigate to the location of the last change.
[Back]	Move to the previous position in the editor.
[Forward]	Cancel the **Back** command. Move to the last position before executing the **Back** command.

- **[Search]** menu

  Search for a file or text in the Workbench.

Menu	Description
[Search]	Search within files and Java sources.
[File]	Search within files and Java sources.
[Text]	Search for the selected text within the Workspace.

- **[Project]** menu

  Project related functions in the Workspace.

Menu	Description
[Open Project]	Open the selected project. To execute this command, the target project must be closed.
[Close Project]	Close the selected project. To execute this command, the target project must be open.
[Build All]	Execute incremental build on all projects in the Workbench. This will build all resources changed since the last incremental build. This command can only be used if **[Project] > [Build Automatically]** is not checked.
[Build Project]	Execute incremental build on the selected project. This will build all resources changed since the last incremental build. This command can only be used if **[Project] > [Build Automatically]** is not checked.

Menu	Description
[Build Working Set]	Execute incremental build on the user specified working set. This will build all resources that are affected by the changes made since the last build. This command can only be used if **[Project] > [Build Automatically]** is not checked.
[Clean]	Delete all previous build results. If **[Project] > [Build Automatically]** is checked, this will rebuild the current project.
[Build Automatically]	If this menu is checked, changed resources in the Workbench that need to be saved will be automatically built (incremental).
[Properties]	Open the Properties dialog box for the selected project or resource.

- **[Run]** menu

  Set tools for compilation.

Menu	Description
[Run As]	Compile using the selected tool.
[External Tools Configurations]	Set external tools for compilation.
[Organize Favorites]	Add a configured external tool as a favorite and access it from the **[Run As]** menu.

- **[Window]** menu

  Open, close, or configure a perspective, view, or other items in the Workbench page.

Menu	Description
[New Window]	Open a duplicate Perspective in a new Workbench page.
[New Editor]	Open the selected file in a new editor.
[Hide Toolbar]	Hide the toolbar.
[Open Perspective]	Select and open a Perspective.
[Show View]	Select and display a view. Only essential views can be accessed from the submenu in Studio. To display other views, click **[Other]** and select a view.
[Customize Perspective]	Set items, such as menu or toolbar, for the current Perspective.

Menu	Description
[Save Perspective As]	Reconfigure a Perspective, or save the current Perspective under another name.
[Reset Perspective]	Reset the layout of the current Perspective.
[Close Perspective]	Close the active Perspective.
[Close All Perspectives]	Close all open Perspectives in the Workbench.
[Navigation]	Navigate through views, editors, and Perspectives in the Workbench. Use the shortcuts for this menu to navigate using the keyboard. The following are the submenu items: – **[Show System Menu]**: Show menu to adjust or close the current view or editor. – **[Show View Menu]**: Show list menu for the active view. – **[Quick Access]**: Search for a resource quickly by entering a string. – **[Maximize / Minimize Active View or Editor]**: Maximize or minimize the current view or editor. – **[Activate Editor]**: Activate the current editor. – **[Next Editor]**: Move to the next editor in the list of recently used editors. – **[Previous Editor]**: Move to the previous editor in the list of recently used editors. – **[Switch to Editor]**: Show a list of currently open editors. Select an editor to activate it. – **[Next View]**: Move to the next view in the list of recently used views. – **[Previous View]**: Move to the previous view in the list of recently used views. – **[Next Perspective]**: Move to the next perspective in the list of recently used perspectives. – **[Previous Perspective]**: Move to the previous perspective in the list of recently used perspectives.
[Preferences]	Show the preferences window for configuring the Workbench or installed plugins. Since Studio resources are created in XML format, set **[General] > [Workspace] > [Text file encoding]** to **UTF-8**.

- **[Help]** menu

  Display help and product information.

Menu	Description
[Help Contents]	Display help TOC and contents.
[Search]	Search help contents.
[Dynamic Help]	Search help topics.
[Key Assist]	Display shortcut keys for each menu item.
[Cheat Sheets]	Select cheat sheets for installed software.
[Check for Updates]	Display a list of installed software and updates.
[Install New Software]	Search for available software and install desired software.
[About Tmax AnyLink Studio]	Display AnyLink Studio product information.

1.2.2. Views

A view provides editor support by displaying Presentation or Workbench data. A view can be displayed in a separate window or in a tabbed notebook. Workbench allows tabs to be configured at the top or bottom of the page.

Click on a tab to activate the tabbed notebook.

The following describes each tab.

- **[Project Explorer]**

  Display a list of resources in the Workspace. Select a resource to open it in the editor.

- **[Console]**

  Search for various information, such as deploy process and error messages, for using Studio.

- **[Error Log]**

  Display error logs that occurred while using the Workbench.

- **[Outline]**

  Display outline of the contents of the active editor.

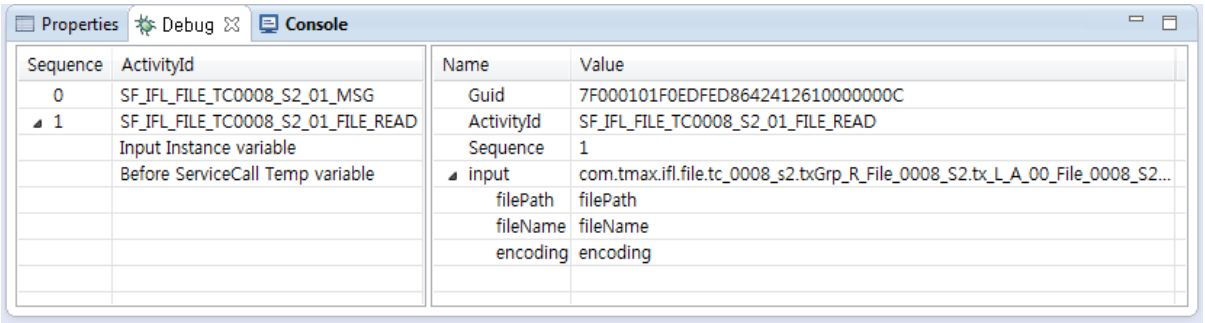
- **[Properties]**

  Display and edit the properties of the selected object in the Workbench.

- **[Debug]**

  Display Flow Debugger results.

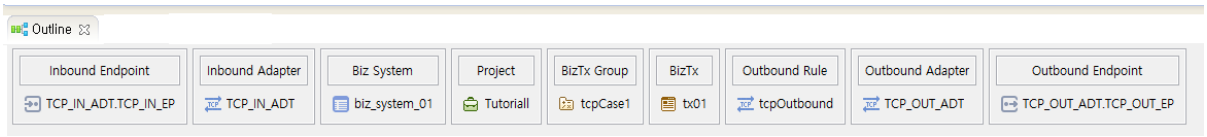
  [Figure 1.8] [Debug] Tab



- **[Relationships]**

  Display resource relationships.

  [Figure 1.9] [Relationships] Tab

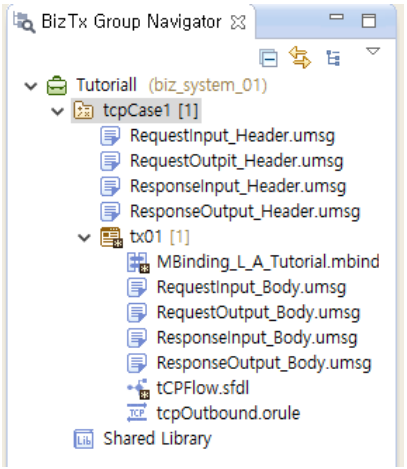


1.2.3. Navigator

Navigator can be shown in a separate window or in a tab with other Navigator tabs. Click on a Navigator's tab to activate it. Workbench allows the tabs to be configured at the top or bottom of the page.

The following is the initial AnyLink Navigator window.

[Figure 1.10] AnyLink Navigator

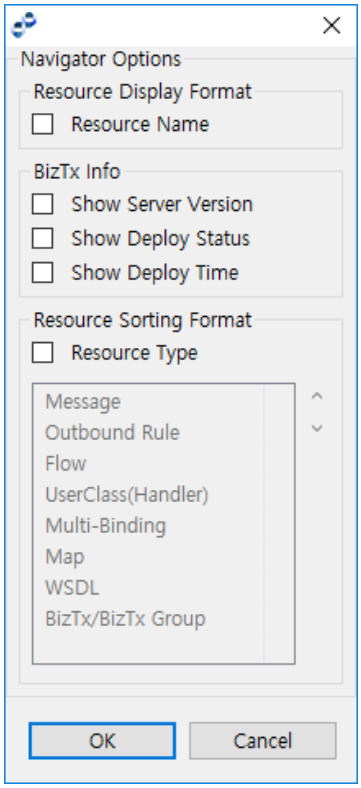


The following describes the Navigator context menu items.

Menu	Description
[New]	Create a new service (BizTx, Flow, Outbound Rule, etc.).
[Download]	Download resources from DIS. Requires connection to the server.
[Template]	Create a template or upload it to DIS. A template can be used to quickly create multiple resources that have common settings.
[Assign Biz System]	Deploy active project's resources to RTE (Runtime Engine Server) when deploying the project. In the local mode, the **Log In** window is displayed.
[Unassign Biz System]	Undeploy resources from RTE. In the local mode, the **Log In** window is displayed.
[Import Configuration]	Import a configuration file registered in DIS and register configuration data in Studio. (Example: BizTx type)
[Refresh]	Refresh the Workspace in the **Service Explorer** to sync with the file system. A project in the Workspace reflects the current file system state. Since files in the Workbench can be modified externally, use the **Refresh** command to sync the files with those in the file system.
[Paste]	Paste a project or file.
[Delete]	Delete local project or file.
[Import]	Import a project or file into the Navigator.
[Export]	Export a project or file from the Navigator.
[Validate]	Validate source.
[Team]	Manage team sources by integrating with a version control system such as SVN.
[Compare With]	Compare resources.
[Replace With]	Replace a resource by using history.

Click  to open the Navigator Options dialog box.

[Figure 1.11] Navigator Options - Resource Display Format



The following describes how to modify resource display format in the Navigator.

Option	Description
Resource Name	Option to show the resource name. If not set, the Resource ID is displayed.
Show Server Version	Option to show the BizTx (Group) resource version currently deployed on DIS while logged in.
Show Deploy Status	Option to show the deploy/undeploy status of the BizTx (Group) resource while logged in.
Show Deploy Time	Option to show the last deployed time of the BizTx (Group) resource while logged in.
Resource Sorting Format	Option to sort resources by type. Select the **'Resource Type'** check box to sort resources by type. Otherwise, resources are sorted by name regardless of their type.

1.3. Resources

Click **[New]** from the **Navigator**'s context menu to create a new resource. For detailed information, refer to each resource guide.

The following describes each resource supported in AnyLink.

Project	Description
Project	To create a service flow diagram, first create a project in the workspace. AnyLink project is created for managing AnyLink resources. The resources in the project can reference one another and form relationships, and are grouped into a package.
BizTx/BizTx Group	BizTx is a business service provided by AnyLink. It includes a series of tasks that involve receiving a client request, forwarding it to a backend service, and returning the result in a response message. A BizTx/BizTx Group resource can be used to create or edit items like a BizTx Group, tree, and message.
Multi-Binding	Multi-Binding router rule is an AnyLink service that calls a multi-binding rule through the adapter's parsing rule. Multiple services are combined into a single group when different routing method is used for each internal service type in AnyLink.
Outbound Rule	AnyLink calls an external system through an internal service, and receives the result. If an outbound rule service is called from an external component, a message is sent to the external system by using the adapter and endpoint configured in the rule.
WSDL	Web Services Description Language (hereafter WSDL) is an XML-based interface definition language that is used to define network service interface using endpoints for exchanging messages. An operation and message are defined abstractly, and an endpoint interface is described by binding a network protocol with message format. WSDL is not restricted to a specific message format and network protocol. It can be used to describe various types of network interfaces.
External Mapping	Defines data object mapping between messages.
Flow	Service flow expresses process flow through a diagram that is derived from the standard BPMN (Business Process Modeling Notation). A diagram consists of activities and events, and is defined with process flows, variables, expressions, mappings, etc. A service flow shows the service execution process in a graphical form.
User Class/Handler	Flow defines a condition to implement a control flow. A condition is defined as a single conditional expression using process variables. Define a user

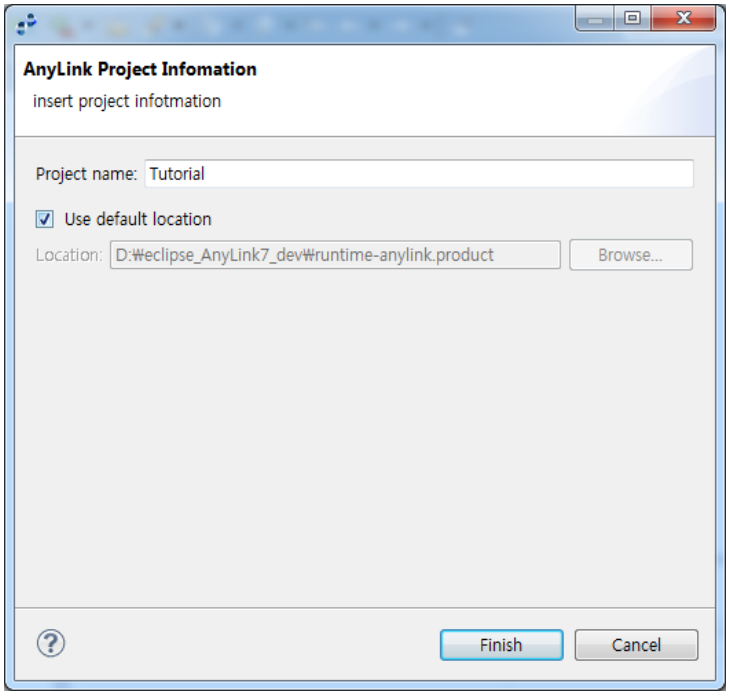
Project	Description
	class activity or handler for a more complex logic that cannot be defined with a simple expression.
Message	Message is a basic abstract data unit that is exchanged between services.
Common BizTx/BizTx Group Project	A set of resources that can be used in projects assigned to the same business system.

1.4. Creating a Project

Create a project before creating resources in AnyLink.

Click **[New] > [Project]** from the context menu in the **Navigator**. Enter the project name in the **Create Project** page, and then click **[Finish]**.

[Figure 1.12] Creating a Project



Item	Description
Project name	Project name. Special characters (?, <, >, ", *, |) are not allowed.

Chapter 2. BizTx/BizTx Group Editor

This chapter describes the functions and uses of the BizTx/BizTx Group editor.

2.1. Creating a BizTx/BizTx Group

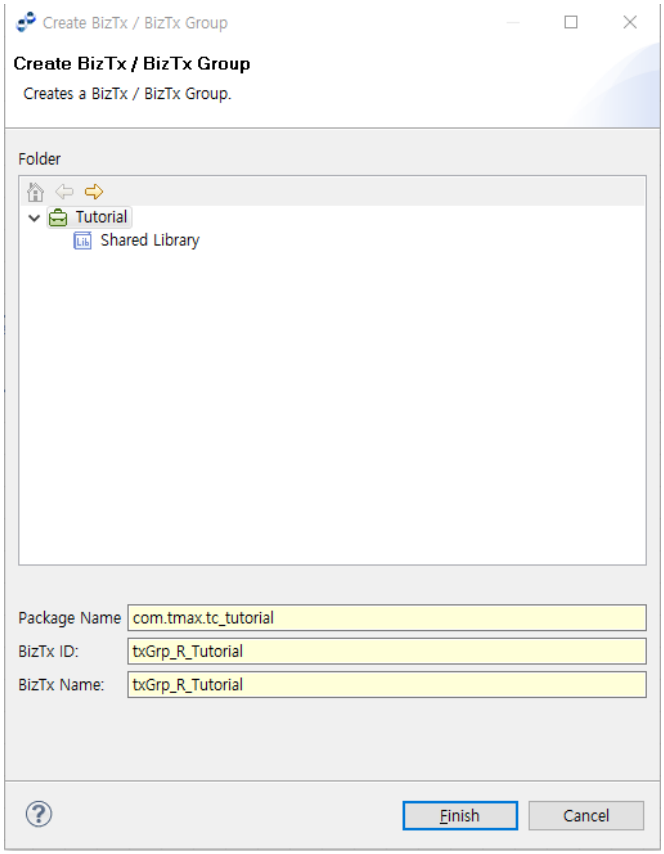
A BizTx/BizTx Group is a business resource that is needed to run the AnyLink Runtime Engine Server (hereafter RTE). BizTx and BizTx Group resources are represented in a tree structure. BizTx and BizTx structure consist of request, response, and abnormal response messages. The main purpose of the BizTx tree is to parse the request message and call the appropriate service from the AnyLink engine according to the content of the parsed message.

A BizTx/BizTx Group consists of three node types, the root BizTx Group, intermediate BizTx Group, and leaf BizTx.

Creating the Root BizTx Group

To create the root BizTx Group, select **[New] > [BizTx/BizTx Group]** from the **Project** resource's context menu in the Navigator. Enter the required items in the **Create BizTx/BizTx Group** window, and then click **[Finish]**.

[Figure 2.1] Create BizTx/BizTx Group Window - Root BizTx Group

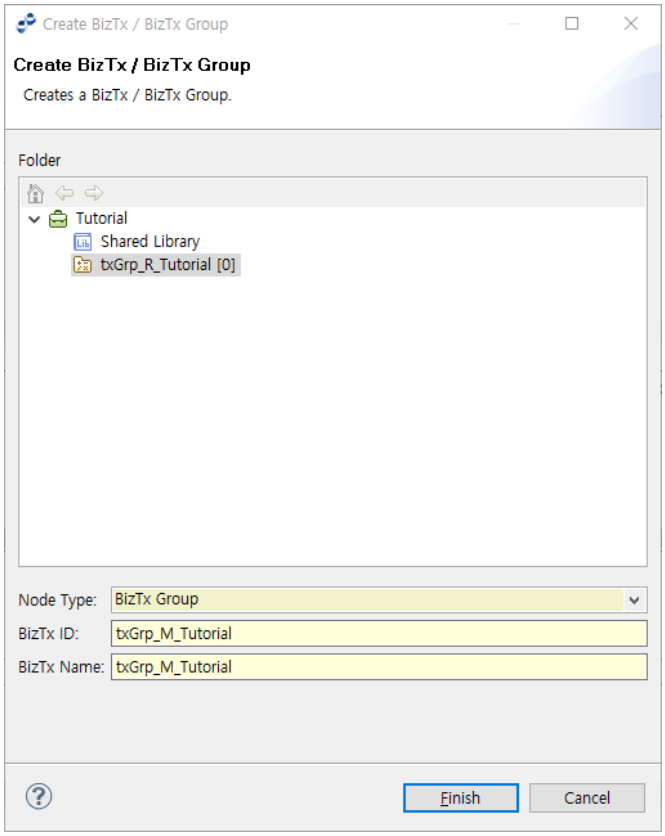


Item	Description
Package Name	Package name for the BizTx. A directory with the package name is automatically created. (Example: com.tmax.tc_tutorial)
BizTx ID	BizTx resource ID. Only alphanumeric and special (_) characters are allowed, and the first character must be in lowercase.
BizTx Name	BizTx name. Only Korean, alphanumeric, and special characters are allowed. Must follow the XML naming convention.

Creating a BizTx/BizTx Group

To create a BizTx Group, select **[New] > [BizTx/BizTx Group]** from the **Project** resource's context menu in the Navigator. Enter the required items in the **Create BizTx/BizTx Group** window, and then click **[Finish]**.

[Figure 2.2] Create BizTx/BizTx Group Window - BizTx Group (Interim) & BizTx (Leaf)



Item	Description
Node Type	Node type (BizTx Group or BizTx).
BizTx ID	BizTx resource ID. Only alphanumeric and special (_) characters are allowed, and the first character must be in lowercase.
BizTx Name	BizTx name. Only Korean, alphanumeric, and special characters are allowed. Must follow the XML naming convention.

2.2. BizTx/BizTx Group Editor

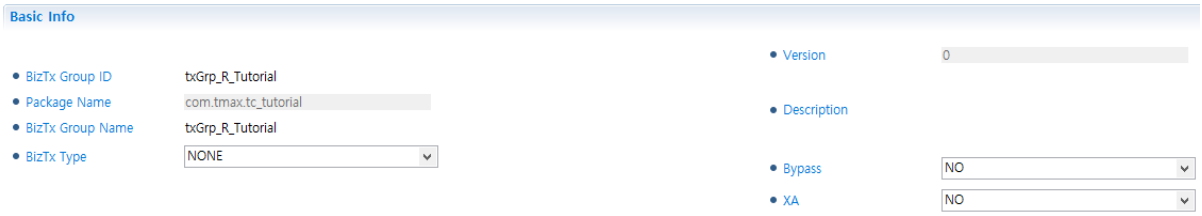
BizTx/BizTx Group editor consists of **BizTx (Group) Info**, **BizTx (Group) Options**, **Parsing Info**, **Parsing Options**, **XML Viewer** tabs. To access each tab, click on the desired tab at the bottom of the resource view.

2.2.1. BizTx/BizTx Group Info

The **[BizTx/BizTx Group Info]** tab consists of **Basic Info**, **Request Message**, **Normal Response Message**, **Business Error Message**, **Error Response Processing**, and **Invocation Service** sections. Invocation Service is only applicable to a BizTx.

Basic Info

[Figure 2.3] [BizTx (Group) Info] - [Basic Info]



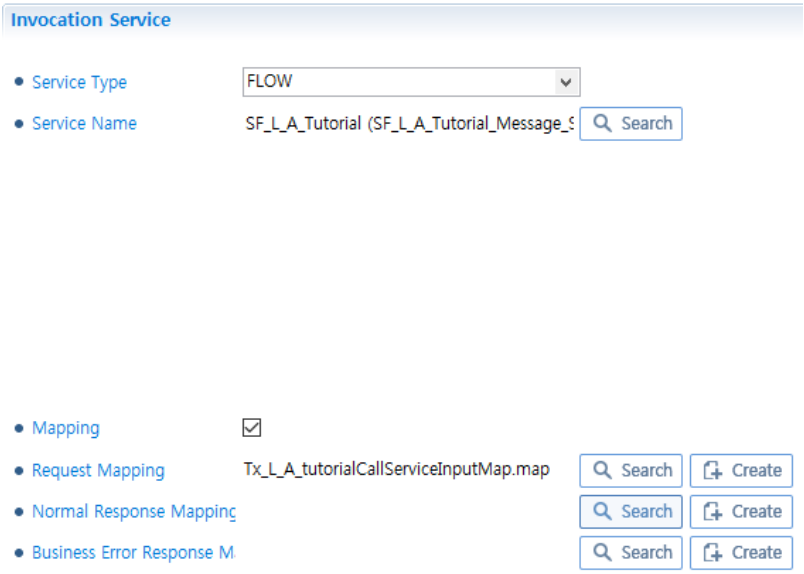
Item	Description
BizTx Group ID	BizTx Group ID.
Package Name	Package name.
BizTx Group Name	BizTx Group name.
BizTx Type	BizTx type used to classify BizTx for monitoring. A BizTx type can be registered through the **[Import Configuration]** menu.
Version	Automatically assigned at deploy time.
Description	Description of the BizTx Group (optional).
Bypass	Deliver the message without transformation when BizTx request message and service request message are the same for a single backend service call. Can only be used for an Outbound Rule with Bypass enabled Invocation service. – PARENT: Inherit the parent BizTx Group setting. – YES: Enable Bypass. – NO: Disable Bypass.
XA	Global transaction setting. Process multiple messages in a single global transaction using the 2PC protocol (prepare, commit, and rollback) through adapters, such as Tmax and Web service, that support transactions. – PARENT: Inherit the parent BizTx Group setting. – YES: Enable XA. – NO: Disable XA.

BizTx Invocation Service (BizTx)

The following describes the **BizTx Invocation Service** section of the **[BizTx Info]** tab for a BizTx. The invocation service can call a Flow, Outbound Rule, or Multi-binding service.

- **Flow**

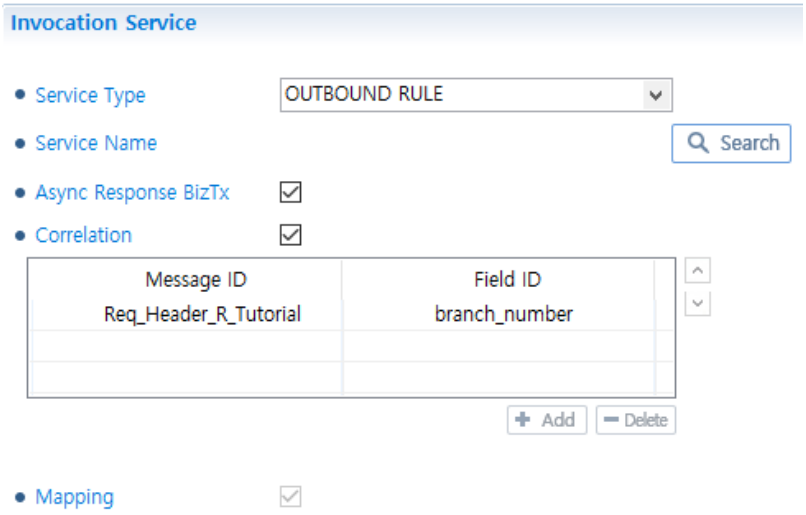
[Figure 2.4] [BizTx (Group) Info] - [Invocation Service (Flow)]



Item	Description
Service Type	Set to 'Flow'.
Service Name	Flow service name. Use **[Search]**.
Mapping	Option to map BizTx's Request/Normal Response/Business Error Response Message to Input/Output/Error parameter of the Flow's start event. – Request Mapping: Click **[Search]** to select an existing mapping file or click **[Create]** to create mapping between the Flow's start event INPUT parameter and request message. – Response Mapping: Click **[Search]** to select an existing mapping file or click **[Create]** to create mapping between the Flow's start event OUTPUT parameter and response message. – Error Response Mapping: Click **[Search]** to select an existing mapping file or click **[Create]** to create mapping between Flow's start event ERROR parameter variable and error response message.

- **Outbound Rule**

[Figure 2.5] [BizTx (Group) Info] - [Invocation Service (Outbound Rule)]

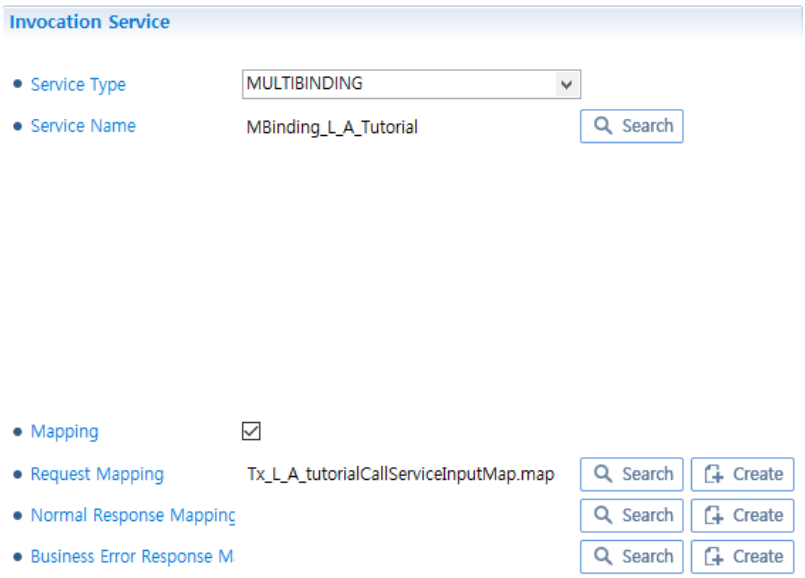


Item	Description
Service Type	Set to 'OUTBOUNDRULE'. Must be set to 'OUTBOUNDRULE' for Bypass enabled BizTx.
Service Name	Outbound Rule service name. Use **[Search]**. If Bypass is set for the BizTx, only Bypass enabled service can be selected.
Async Response BizTx	To use an Async Response BizTx, set **'BizTx Identifier Code'** to either 'RESPONSE' or 'ERROR_RESPONSE' in **[Parsing Info] > [Parsing Definition]**. The following describes the **'BizTx Identifier Code'**. – When BizTx is set to 'RESPONSE' or 'ERROR_RESPONSE': BizTx can execute the service regardless of the Response Option value. However, the service type must be set to TCP Outbound Rule with async response mode. Otherwise, the Response Option is automatically changed to ASYNC type when Async Response BizTx is selected. – When set to 'RESPONSE': BizTx's request message is added to the Normal Response Message of the TCP Outbound Rule. – When set to 'ERROR_RESPONSE': BizTx's request message is added to the Business Error Response Message of the TCP Outbound Rule. If this item is selected, the Mapping item is unavailable.
Correlation	Message field used as correlation field of the message for Async Response BizTx. If the **'BizTx Identifier Code'** is set to 'RESPONSE', the message

Item	Description
	is a Normal Response Message. If it is set to 'ERROR_RESPONSE', the message is a Business Error Response Message Correlation cannot be used for a Bypass enabled BizTx.
Mapping	Option to map BizTx's Request/Normal Response/Business Error Response Message to Outbound Rule's Request/Response/ Business Error Response Message. – Request Mapping: Click **[Search]** to select an existing mapping file or click **[Create]** to create mapping between Request Messages of BizTx and Outbound Rule. – Response Mapping: Click **[Search]** to select an existing mapping file or click **[Create]** to create mapping between the Response Messages of Outbound Rule and BizTx. – Error Response Mapping: Click **[Search]** to select an existing mapping file or click **[Create]** to create mapping between the Business Error Response Messages of Outbound Rule and BizTx. Mapping cannot be used for a Bypass enabled BizTx.

- **Multi-binding**

[Figure 2.6] [BizTx (Group) Info] - [Invocation Service (Multi-binding)]

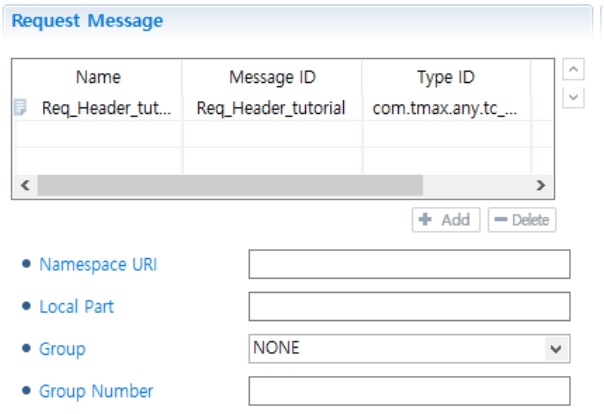


Item	Description
Service Type	Set to 'MULTIBINDING'.
Service Name	Multi-binding service name. Use **[Search]**.
Mapping	Option to map BizTx's Request/Normal Response/Business Error Response Message to Multi-binding's Request/Response/ Business Error Response Message. – Request Mapping: Click **[Search]** to select an existing mapping file or click **[Create]** to create mapping between Request Messages of BizTx and Multi-binding. – Response Mapping: Click **[Search]** to select an existing mapping file or click **[Create]** to create mapping between the Response Messages of Multi-binding and BizTx. – Error Response Mapping: Click **[Search]** to select an existing mapping file or click **[Create]** to create mapping between the Business Error Response Messages of Multi-binding and BizTx.

Request/Normal Response/Business Error Response Message

A BizTx/BizTx Group consists of messages. When a message is added to a **BizTx Group**, the message definition is inherited by its child nodes.

[Figure 2.7] [BizTx (Group) Info] - [Request Message]



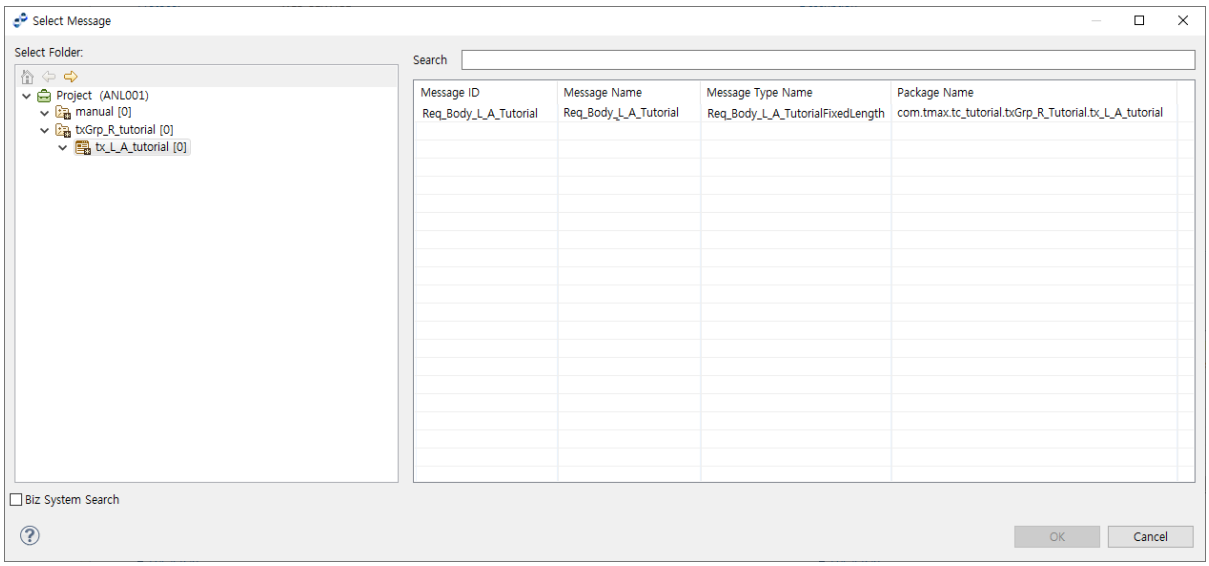
Item	Description
Request Message	Request message for BizTx.

Item	Description
Normal Response Message	Normal response message for BizTx.
Business Error Response Message	Business error response message for BizTx. A business error is any error that is not a system error. System errors are handled according to the description in the "Error Response Handling" section.

After creating a message in the **Message Editor**, click **[+Add]** to open the dialog box for selecting and adding a message. The configuration items and method are same for **Request**, **Normal Response**, and **Business Error Response Messages**.

The following is the window for adding a request message.

[Figure 2.8] Select Request Message Dialog Box



Select a **Project** or **BizTx (Group)** from the Navigator on the left to display its messages in the table on the right. If a message is selected, the message is displayed in the table. The message type can be edited from the table. Select a message to set it as a Request Message.

Note

Since AnyLink resources form the BizTx nodes of a hierarchy, the resources of a parent BizTx node cannot reference the resources of a child BizTx node.

Select Search Biz System to search for messages in all projects of the Biz System. Entering a search word in Search searches for messages with ID and name that include the word.

If another project's message is selected, the BizTx node of the message is created as a symbolic link. Before deploying the BizTx, the BizTx node created with the symbolic link must be deployed first.

The following are the common configuration items for **Request**, **Normal Response**, and **Business Error Response Messages**.

Item	Description
Namespace URI	Used to form the message's namespace together with the Local Part when a Request/Normal Response/Business Error Response Message is an XML type.
Local Part	Used to form the message's namespace together with the Namespace URI when a Request/Normal Response/Business Error Response Message is an XML type.
Group	Used in the BizTx of SOAP header. Created as an XML tag.
Group Number	Group order.

Error Response Handling

This section is used to configure error response handling.

- **NONE**

  No processing.

- **PARENT**

  Inherit parent BizTx node's setting.

- **FLOW**

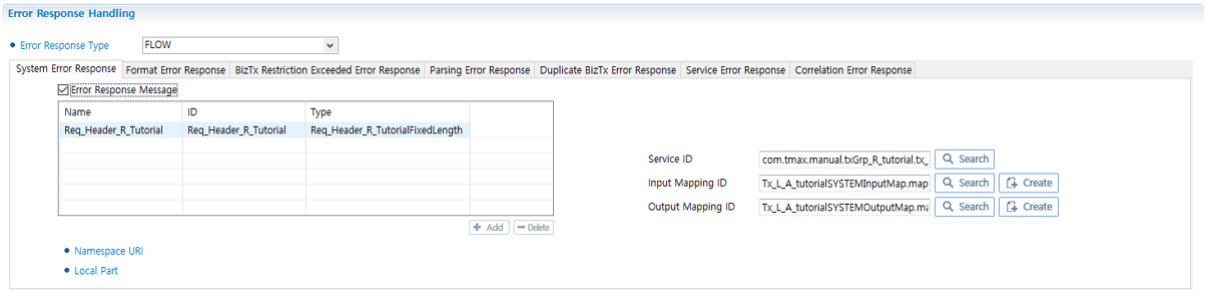
  The following tabs are displayed when FLOW is selected. Each tab represents each error response type.

Tab	Description
[System Error Response]	Handling for system errors.
[Format Error Response]	Handling for data type mismatch error when converting a message into a data object.
[BizTx Restriction Exceeded Error Response]	Handling for when a BizTx restriction set in WebAdmin is exceeded.

Tab	Description
[Parsing Error Response]	Handling for when a child node is not found.
[Duplicate BizTx Error Response]	Handling for duplicate BizTx with a value added in Check Duplicate BizTx Field when Check Duplicate BizTx is set to Yes.
[Service Error Response]	Handling for errors thrown by a flow or outbound rule.
[Correlation Error Response]	Handling for flow correlation errors.

Select the **'Error Response Message'** checkbox for the desired message type to set the error response message. If the Error Response Handling is set to 'FLOW', the service flow with the specified **'Service ID'** is called when a matching error type occurs.

[Figure 2.9] [BizTx (Group) Info] - [Error Response Mode (FLOW)]



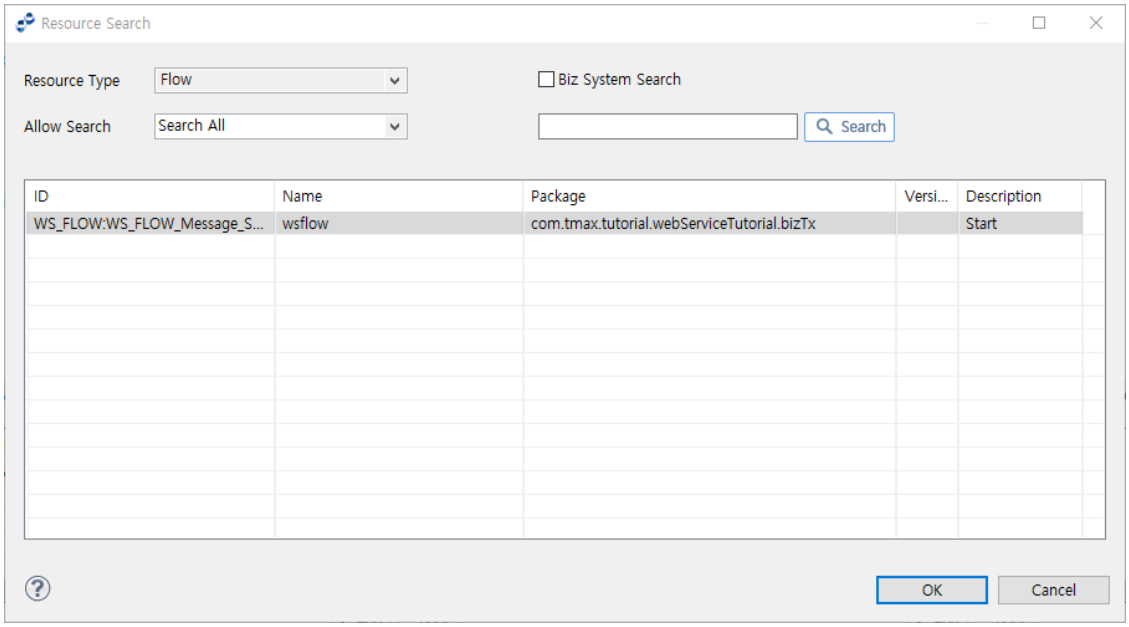
The following describes each item.

Item	Description
Error Response Message	– Error Response Message: Checkbox to enable error response message configuration. – **[Add]**: Select a message to map to the service flow's output message in [Figure 2.7]. – **[Delete]**: Delete the selected message. – Namespace URI: Used to form the message's namespace together with the Local Part when the message is an XML type. – Local Part: Used to form the message's namespace together with the Namespace URI when the message is an XML type.
Service ID	Click **[Search]** to select the Service ID of the service flow to call when a matching error type occurs.

Item	Description
Input Mapping ID	Search for and create the message to input into the service flow with the **'Service ID'**. – **[Search]**: Select a message from the **Resource Search** dialog box ([Figure 2.10]) that displays the map files. – **[Create]**: Create mapping results in a map file. This menu can also be accessed in the Navigator with BizTx/BizTx Group resource's context menu, **[New] > [External Mapping]**. Click **[Create]** to open the Mapping dialog box to map a **Request Message** with the message set as the input parameter of the start message event for the flow with the Service ID. In the **Source** section, click the context menu, **[Add Source]**, to add the **Request Message**. The message set as the flow start message event's input parameter is displayed in the **Target** section. Since the BizTx node's data object is not created for a system, format, or parsing error response message, only parent node's messages are displayed when **[Add Source]** is clicked. Mapping Code can be manually created using **[Add Mapping Code]**. Mapping dialog box cannot be accessed without specifying the **'Service ID'**.
Output Mapping ID	Search for and create the message to output after executing the service flow with the **'Service ID'**. – **[Search]**: Select a message from the **Resource Search** dialog box ([Figure 2.10]) that displays the map files. – **[Create]**: Open the Mapping dialog box to map the message set as the input parameter of the start message event for the flow with the Service ID to an **Error Response Message**. In the **Source** section, click the context menu, **[Add Source]**, to add the message set as the input parameter of the start message event. The **Error Response Message** is displayed in the **Target** section. Mapping dialog box cannot be accessed without specifying the **'Service ID'**.

Check map files to add and then click **[OK]**.

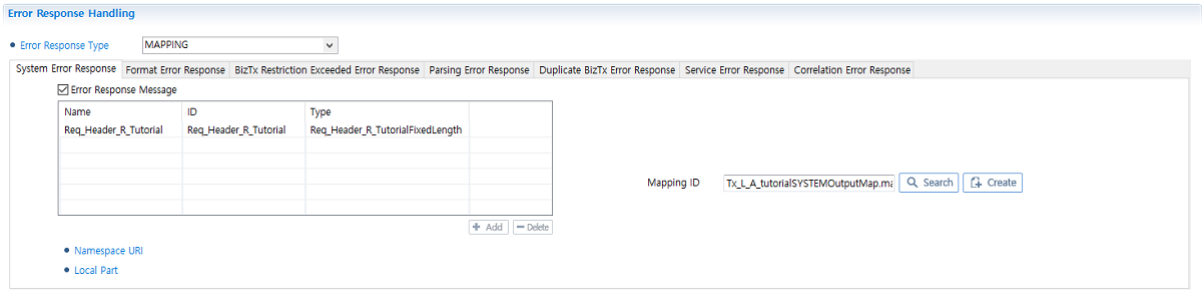
[Figure 2.10] Search Mapping ID (Search Resource)



- **MAPPING**

  Maps Error Response Messages. Defines mapping for each error response message type.

[Figure 2.11] [BizTx (Group) Info] - [Error Response Mode (MAPPING)]



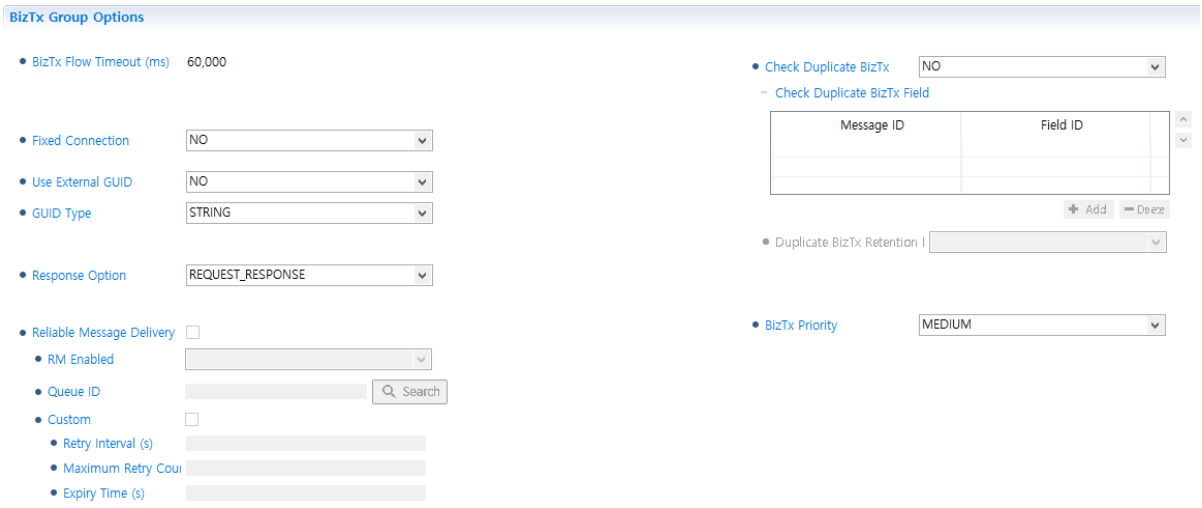
Item	Description
Error Response Message	Same as "Flow Error Response Mode".
Mapping ID	– **[Search]**: Select a message from the **Resource Search** dialog box ([Figure 2.10]) that displays the map files. – **[Create]**: Open the Mapping dialog box to map a **Request Message** with an error response message. In the **Source** section, click the context menu, **[Add Source]**, to add the **Request Message**. The error response message is displayed in the **Target** section. Since the BizTx node's data object is not created for a system, format, or parsing error response message, only parent node's messages are

Item	Description
	displayed when **[Add Source]** is clicked. Mapping Code can be manually created using **[Add Mapping Code]**.

2.2.2. BizTx/BizTx Group Options

The **[BizTx (Group) Options]** tab is used to configure Timeout, Fixed Connection, GUID, Response Option, Reliable Message Delivery, and BizTx Priority settings for a BizTx/BizTx Group.

[Figure 2.12] [BizTx (Group) Info] - [BizTx (Group) Options]



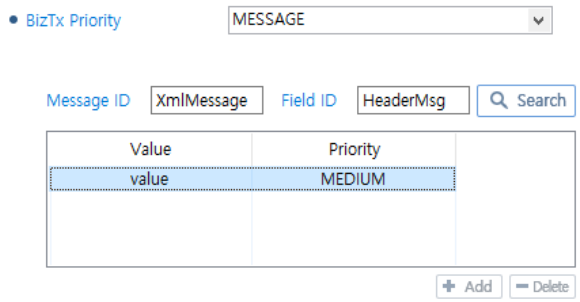
Item	Description
BizTx Flow Timeout (ms)	BizTx timeout. Handled as an error if this time is exceeded. – Not the root BizTx Group: if **'No Timeout'** is set, inherits the timeout setting of the parent BizTx Group. – Root BizTx Group: default value is 1 minute (60000 ms).
Fixed Connection	Option to send a message to another MS if a flow correlation error occurs. – PARENT: Inherit the parent BizTx Group setting. – YES: Do not send the correlation message to another MS, and close the session after handling the error response if Error Response is set. – NO: Send the correlation message to another MS in the cluster.
Use External GUID	Option to use external GUID as AnyLink GUID. – PARENT: Inherit the parent BizTx Group setting.

Item	Description
	– YES: Use external GUID when there is GUID in a message field. – NO: AnyLink generates and uses its own GUID.
GUID Type	GUID type. – PARENT: Inherit the parent BizTx Group setting. – LONG: Long data that can be up to 19 characters. – STRING: String data that can be up to 32 characters.
Response Option	– REQUEST_RESPONSE: Response Ack message is added in addition to the request and response messages. ACK or NAK message is sent to notify whether the message was processed successfully or not. – PARENT: Inherit the parent BizTx Group setting. – ONEWAY: Send the Request Message without waiting for the response. This option can only be used for a Bypass enabled BizTx. Otherwise, it is ignored.
RM Enabled	Option to guarantee message delivery. This is available only when Response Option is ONEWAY. – PARENT: Inherit the parent BizTx Group setting. – YES: Guarantee message delivery. – NO: Do not guarantee message delivery.
Queue ID	Queue used to guarantee message delivery. The queue to use must be created in advance by selecting **[System] > [MQs]** in WebAdmin. To open a dialog box where queue IDs can be searched, click **[Search]**.
Custom	Sets the BizTx queue. – Retry Interval (s): Interval to resend the message. – Maximum Retry Count: Maximum retry count to send the message. Setting to -1 means infinite retries. – Expiry Time (s): Message expiration time. Setting to -1 means that the message is not expired.

Item	Description
Check Duplicate BizTx	– PARENT: Inherit the parent BizTx Group setting. – YES: Do not allow duplicate BizTx. Duplicate BizTx is processed according to the duplicate BizTx error response routine. – NO: Allow duplicate BizTx.
Check Duplicate BizTx Field	– Message ID: Message ID. It is checked whether the message is duplicated. – Field ID: Field ID. It is checked whether the field is duplicated. To add messages and fields, click **[+ Add]**. To delete them from the table, click **[- Delete]**.
Duplicate BizTx Retention Period	Time period to retain fields extracted for duplication check. – Daily: Delete the fields every day. – Hourly: Delete the fields every hour.
BizTx Priority	– PARENT: Inherit its parent BizTx's priority unless it is the root node of the BizTx Group. – HIGH: Set the BizTx's priority to the highest level within the BizTx Group. It is allocated to the VIP thread pool set in WebAdmin. – MEDIUM: Most BizTx's have MEDIUM priority within the BizTx Group. It is allocated to the regular thread pool. – LOW: Low priority BizTx. It is allocated to the regular thread pool or can be discarded if the pool is full. – MESSAGE: Priority is set according to the specified message's value. The priority levels are as listed above.

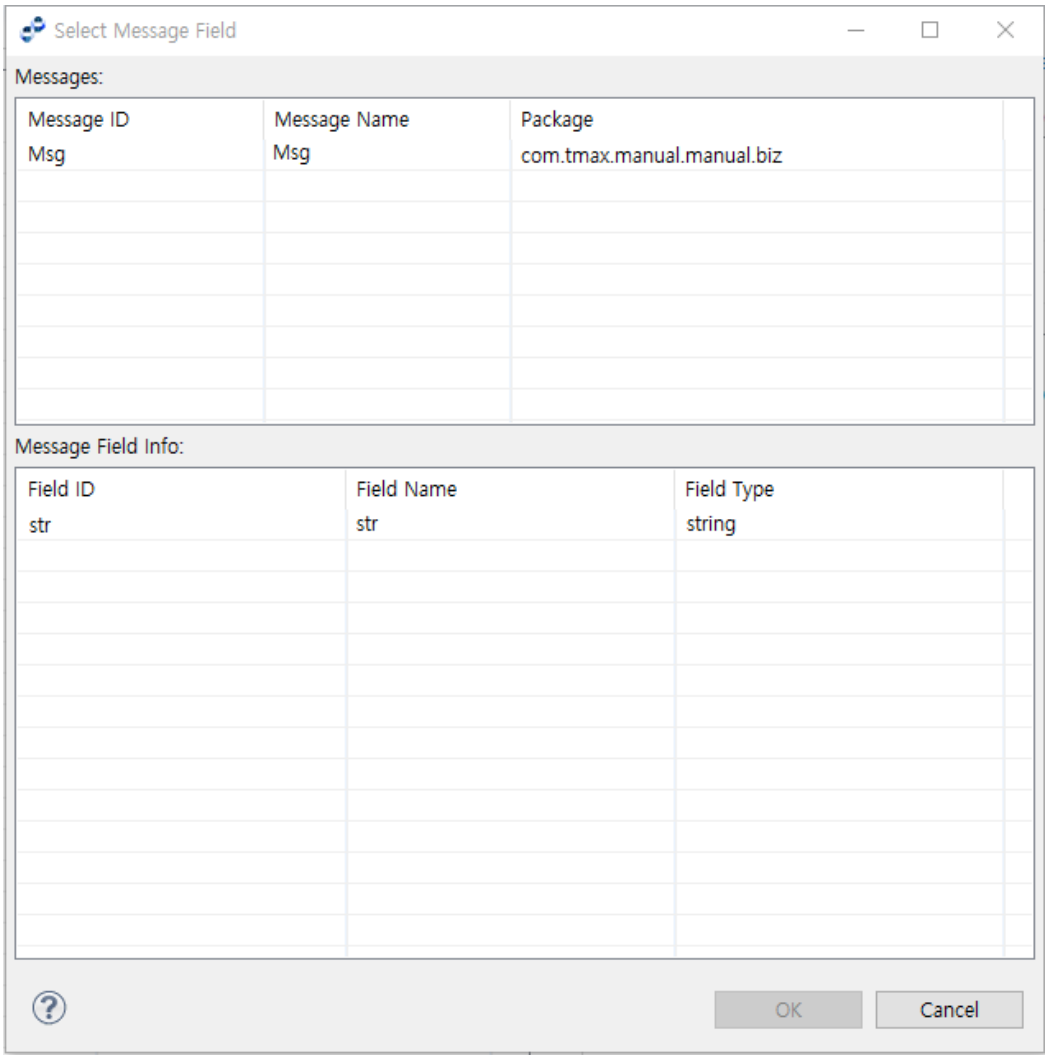
Select 'MESSAGE' for the **'BizTx Priority'** to display the following section.

[Figure 2.13] [BizTx (Group) Options] - [BizTx Priority] - Message Options



Click **[Search]** to open the Select Message Field dialog box.

[Figure 2.14] Select Message Field Dialog Box



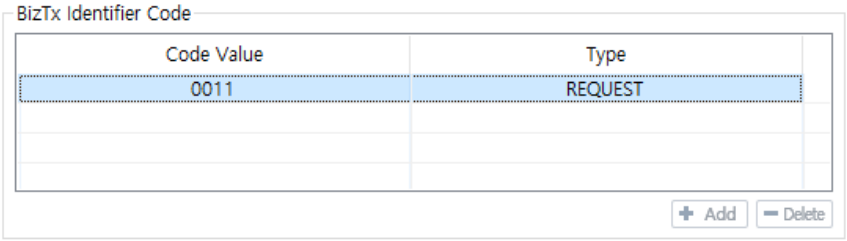
2.2.3. Parsing Info

The Parsing Info tab is used to define the standard for identifying the child BizTx to process input message for.

- **BizTx Identifier Code**

  This is required when the 'Async Response BizTx' is checked in the BizTx Info section.

  [Figure 2.15] [Parsing Info] - [Parsing Definition] - BizTx Identifier Code

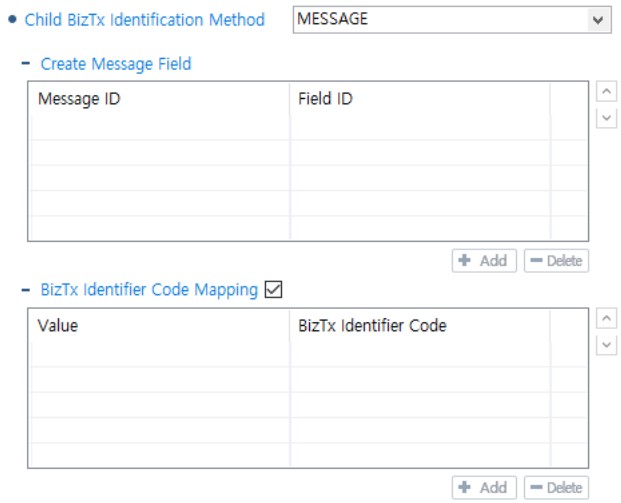


Item	Description
Code Value	Identifier code for BizTx.
Type	– RESPONSE: Normal response handling – REQUEST: Request handling – ERROR_RESPONSE: Error response handling To use the **Async Response BizTx** setting of the **Invocation Service**, this must be set to 'RESPONSE' or 'ERROR_RESPONSE'.

- **Child BizTx Identification**

  Child BizTx Identification Method can only be set for the BizTx Group. Select a method, click **[+Add]**, and then enter the value and BizTx.

[Figure 2.16] [Parsing Info] - [Parsing Definition] - Child BizTx Identification Method



Item	Description
NONE	No BizTx identification.
MESSAGE	Select the BizTx with BizTx Identifier Code that matches the selected message's field value. The message's field value must match the **Value** of the BizTx's **'BizTx Identifier Code'**.
HANDLER	Mostly used to implement parsing handler. Select the BizTx with BizTx Identifier Code returned by the parsing handler. The return value of the user class must match the **Value** of the BizTx's **'BizTx Identifier Code'**.
X_PATH	Only for an XML message. Select the BizTx with BizTx Identifier Code that matches the X_PATH value. The X_PATH value must match the **Value** of the BizTx's **'BizTx Identifier Code'**.
OFFESET_LENGTH	The Offset and Length text boxes are displayed when this option is selected. Select the BizTx with BizTx Identifier Code that matches the OFFSET_LENGTH value. The OFFSET_LENGTH value must match the **Value** of the BizTx's **'BizTx Identifier Code'**.
SOAP_ACTION	Select the BizTx with BizTx Identifier Code that matches the SOAP_ACTION value. The SOAP_ACTION value must match the **Value** of the BizTx's **'BizTx Identifier Code'**.

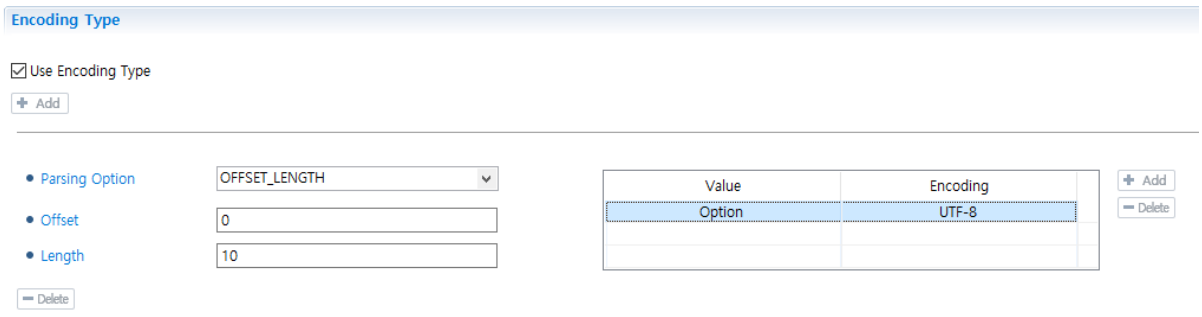
Item	Description
JSON_POINTER	Select the BizTx with BizTx Identifier Code that matches the JSON POINTER value. The JSON POINTER value must match the **Value** of the BizTx's **'BizTx Identifier Code'**.
CONDITION_MESSAGE	If the condition is true, it operates same as MESSAGE according to child MESSAGE settings. If not, the next condition tab is checked. The condition is true when the selected message's field is the entered value or there is no condition.
CONDITION_OFFSET_LENGTH	If the condition is true, it operates same as OFFSET_LENGTH according to child OFFSET_LENGTH settings. If not, the next condition tab is checked. The condition is true when the selected message's field is the entered value or there is no condition.
HTTP_METHOD	Select the BizTx with BizTx Identifier Code that matches the requested HTTP method. Available only for BizTx requested through an HTTP inbound endpoint.

2.2.4. Parsing Options

The Parsing Options tab is used to configure the encoding and message types.

- **Encoding Type**

[Figure 2.17] Encoding Type

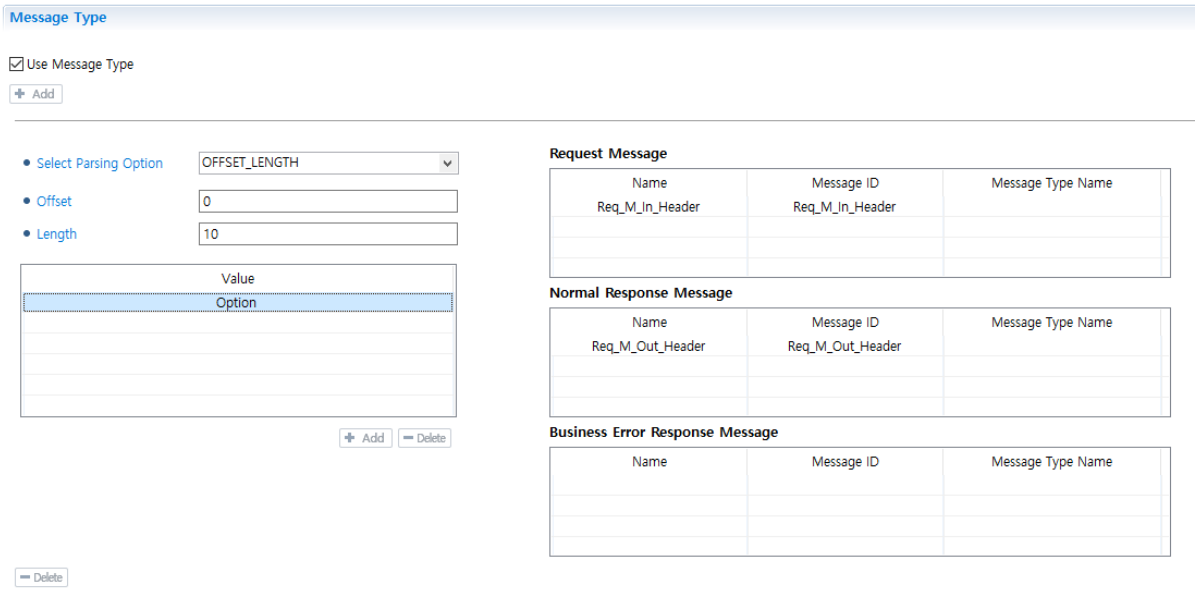


Item	Description
Use Encoding Type	Option to enable adding or deleting parsing options.
Parsing Option	– All: Use the **'Encoding Value'** set for all messages in the BizTx.

Item	Description
	– ENDPOINT: Use the **'Encoding Value'** if the ENDPOINT value is **'Option Value'**. – URL: Use the **'Encoding Value'** if the URL value is **'Option Value'**. – OFFSET_LENGTH: Use the **'Encoding Value'** if the OFFSET_LENGTH is **'Option Value'**. • Offset: Starting offset to begin parsing • Length: Number of bytes to parse
Encoding	Encoding to use.

- **Message Type**

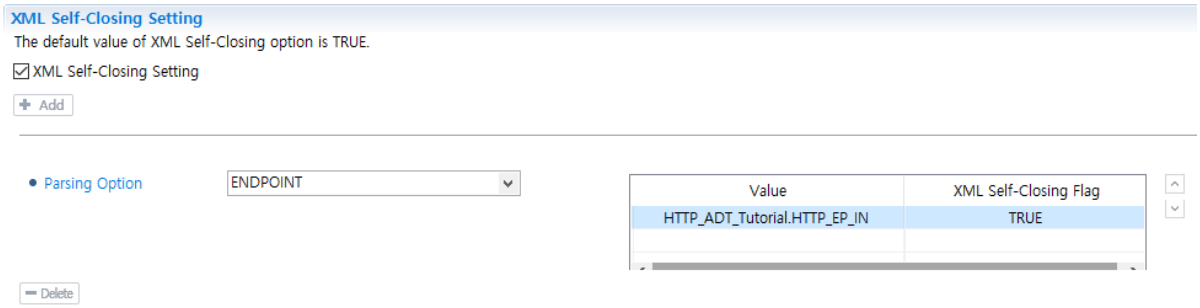
[Figure 2.18] Message Type



Item	Description
Select Parsing Option	Same as the **'Parsing Option'** field in "Encoding Type". Convert Request, Normal Response, and Business Error Response Messages to the selected message type according to each option value.
Value	User input value for the selected Parsing Option.

- **XML Self-Closing Setting**

[Figure 2.19] XML Self-Closing Setting



Item	Description
Parsing Option	Same as the **'Parsing Option'** field in "Encoding Type". Convert XML Self-Closing Flag according to each option value.
Value	User input value for the selected Parsing Option.
XML Self-Closing Flag	Option to use XML self closing.

2.2.5. XML Viewer

BizTx/BizTx Group is created in the XML format and saved in a file with .biztx extension. BizTx/BizTx Group settings can be checked by reviewing the file's XML structure. The XML Viewer is read-only.

Chapter 3. Message Editor

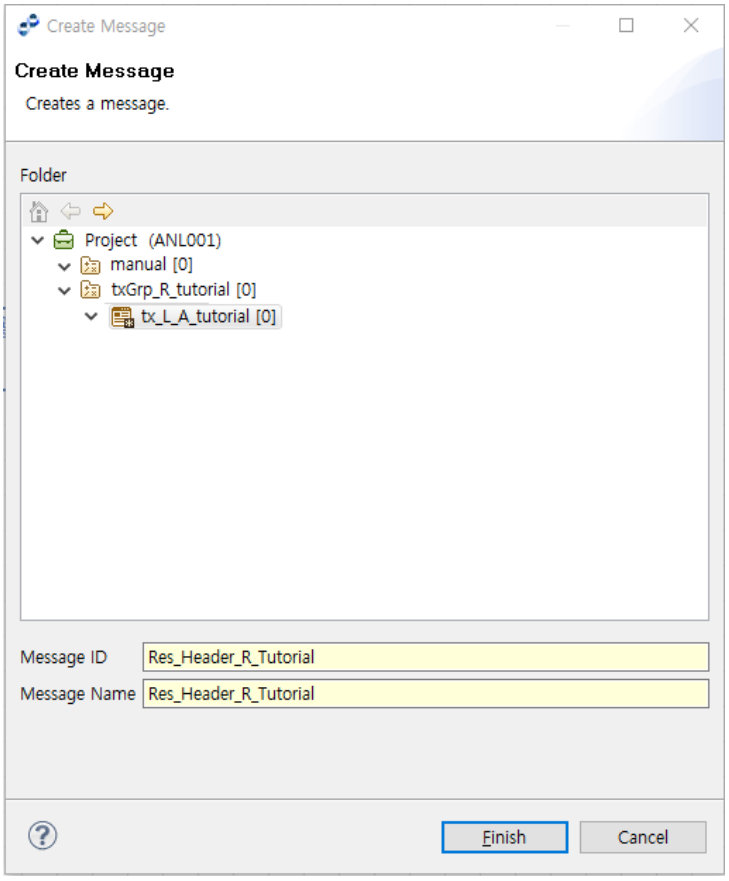
This chapter describes the functions and uses of the Message editor.

3.1. Creating a Message

A Message object is an abstraction of a message. AnyLink adapter uses a Message object to create a byte array to transmit a DataObject to the network. The byte array is converted into a DataObject in the network, and then passed to the flow. Converting a DataObject into a byte array is called marshaling, and the reverse is called unmarshalling.

To create a message, select **[New] > [Message]** from the context menu of BizTx (Group) resource in the **Navigator** on the left. Select a **[BizTx/BizTx Group]** and enter the required items, and then click **[Finish]**.

[Figure 3.1] Create Message Dialog Box

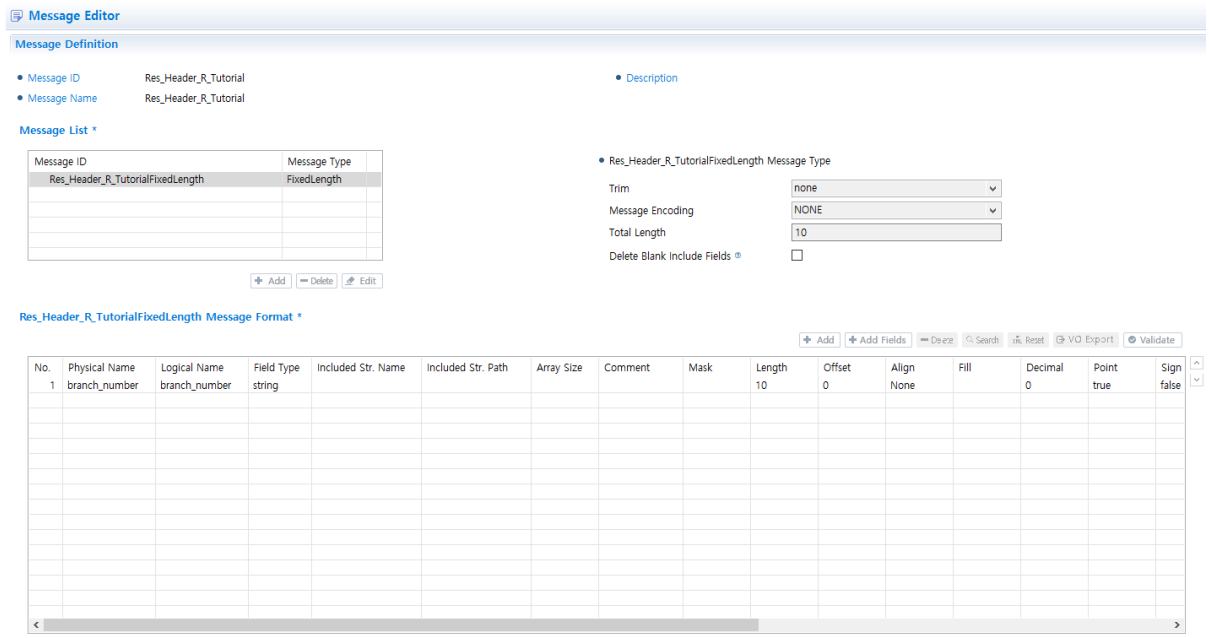


Item	Description
Message ID	Message ID. Only alphanumeric and special (_) characters are allowed, and the first character must be upper case.
Message Name	Message name. Only Korean, alphanumeric, and special characters are allowed. Must follow the XML Naming Convention.

3.2. Message Editor

After creating a Message resource, navigate to the **Message Editor** to update its information.

[Figure 3.2] Message Editor



- **Message Definition**

Item	Description
Message ID	Message ID.
Message Name	Message name.
Description	Description of the message (optional).

- **Message Configuration**

  Consists of **Message List** and **Message Type**. For information about how to configure messages by type, refer to "3.2.1. Message Configuration".

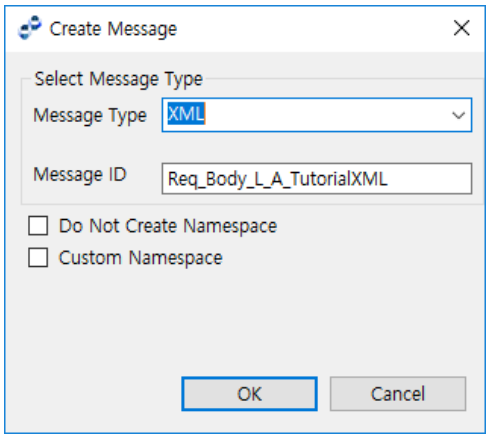
- **Message Format**

Configures a message added in **Message Configuration**. Configuration items vary depending on the message type. For more information, refer to "3.2.2. Message Format".

3.2.1. Message Configuration

The **Message Configuration** section in **Message Editor** ([Figure 3.2]) consists of **Message List** and **Message Type**. To create a message, click **[+ Add]** below **Message List**. Enter required items in the **Create Message** dialog box and then click **[OK]**.

[Figure 3.3] Create Message Dialog Box



The following describes each item.

Item	Description
Message Type	Message type. **Message Type** items vary according to the message type. – FixedLength – Bitmap – Delimeter – CSV – XML – JSON – ValueObject – KeyValue – FDL – FML – ISO8583

Item	Description
Message ID	Auto-filled with the Message ID + Message type. (Editable)
Do Not Create Namespace	If selected, XML namesapce is not created. Only available when **'Message Type'** is 'XML'.
Custom Namespace	If selected, XML namesapce is not automatically created and must be manually entered. Only available when **'Message Type'** is 'XML'.

Note

To complete creating a message, configure each field in **Message Format** and then add it to **Message List**.

After creating a message, configure it.

[Figure 3.4] Message Editor - Message Configuration

The following describes configuration items by message type.

- **Common Items**

  The following describes each item that can be configured for all Message types.

Item	Description
Message Encoding	Message encoding type. AnyLink supports: – ASCII – EBCDIC – UTF-8 – UTF-16 – UTF-16BE

Item	Description
	– UTF-16LE – EUC-KR

- **FixedLength**

Item	Description
Trim	Option to trim the Message. – none: do not trim the Message. – rtrim: trim the right of the Message. – ltrim: trim the left of the Message. – lrtrim: trim both sides of the Message.
Total Length	Total length of message fields. If the total length is undetermined (due settings such as Include or Array), set this to Unknown Length.
Delete Blank Include Fields	If selected, the field is not created as an object if array data is space (0x20) or null (0x00).

- **Bitmap**

  No additional items need to be configured for a Bitmap Message.

- **Delimeter**

Item	Description
Field Delimiter	Message field delimiter.

- **CSV**

  No additional items need to be configured for a CSV Message.

- **XML**

Item	Description
Root Element Name	Root Element name of the XML Message.
Content Type	Content Type name of the XML Message.
Complex Type Name	Complex Type name of the XML Message.
Simple Type Name	Simple Type name of the XML Message.

Item	Description
Ignore UnmarshallException	Option to ignore UnmarshallExceptions and continue processing the BizTx.
Namespace - Prefix	Namespace and prefix.

- **JSON**

Item	Description
Null Processing Method	Method for processing a null field value. – NULL: keep the null value – EMPTY_STRING: use an empty string – SKIP: remove the tag
Ignore UnmarshallException	Option to ignore UnmarshallExceptions and continue processing the BizTx.
Object Name	Object name. Json object's Root node name. The Root node is created with the name that is the same as the object name, and its sub-nodes are created with fields in the message.

- **ValueObject**

  No additional items need to be configured for a Value Object Message.

- **KeyValue**

Item	Description
Delimiter	Value to use as a delimiter.
Remove Start Delimiter	For key-value messages, a delimiter is appended to the front of the messages. For non-key-value messages, a delimiter is not appended.
Separator	Value to use as a separator.
Trim	Option to trim the Message. – none: do not trim the Message. – rtrim: trim the right of the Message. – ltrim: trim the left of the Message. – lrtrim: trim both sides of the Message.

- **FDL (Fixed Definition Language)**

  No additional items need to be configured for a FDL Message.

- **FML (Fixed Manipulation Language)**

  No additional items need to be configured for a FML Message.

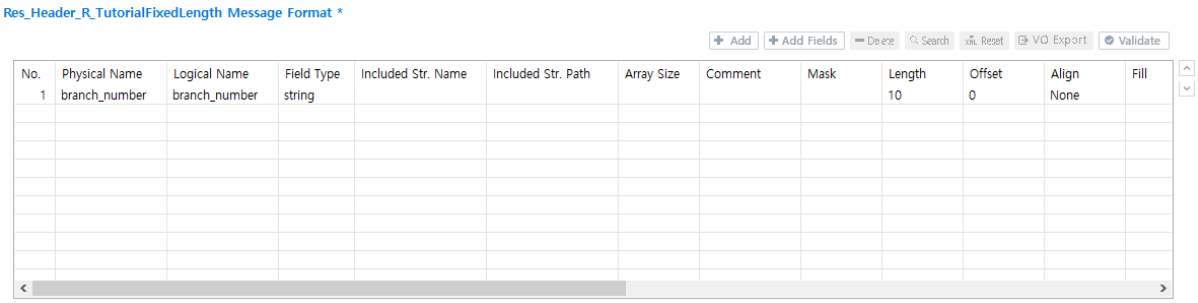
- **ISO8583**

  No additional items need to be configured for an ISO8583 Message.

3.2.2. Message Format

The **Message Format** section in **Message Editor** ([Figure 3.2]) has configuration items that vary depending on the message type.

[Figure 3.5] Message Editor - Message Format



- **Common Items**

  The following describes each item that can be configured for all Message types. For items for each type, refer to "Configuration Items for Each Message Type".

Item	Description
Physical Name	Physical name of the field (use alphabet characters without spaces). Must follow Java variable naming rules.
Logical Name	Logical name of the field. Not restricted by Java variable naming rules (Korean characters are allowed).
Field Type	Field type. – binary: Base64 encoded binary data type – BigDecimal: java.math.BigDecimal class type

Item	Description
	– BigInteger: java.math.BigInteger class type – bit: Binary data of either 0 or 1 – bitmap: Either 0 or 1 that expresses a mapping – boolean: JAVA boolean type – byte: JAVA Byte type – char: JAVA char type – Calendar: java.util.Calendar class type – date: java.util.Date class type – double: JAVA double type – float: JAVA float type – int: JAVA int type – integer: java.lang.Integer class type – include: type used to include another Message – long: JAVA long type – number: java.lang.Number class type – short: JAVA short type – string: java.lang.String class type
* Included Str. Name Included Str. Path	Name and path of the included Message when **'Field Type'** is 'Include'. Manual input is not allowed.
Array Size	Array size when multiple Message fields are placed in an array. If not set, it is assumed that the message is not an array. Either enter a number, or a field name whose value is the array size (variable-length array). The field must be a numeric type. Set to 'unbounded' to use the array without specifying its size.
Comment	Field description (optional).
Mask	Used to mask the original value of the field input to hide security sensitive data. – Replace each character of the original message with the mask string if the mask string is a single character.

Item	Description

	Example)
	
	`Mask String: 'x'`
	`Original Message: 'abcd'`
	`----------`
	`Result: 'xxxx'`
	
	– Use '?' to keep the original character in the message.
	
	Example)
	
	`Mask String: '???-xxxx-????'`
	`Original Message: '010-1111-2222'`
	`----------`
	`Result: '010-xxxx-2222'`
	
	– Append '\' to escape the special character '?'.
	
	Example)
	
	`Mask String: '?\??-xxxx-?\???'`
	`Original Message: '010-1111-2222'`
	`----------`
	`Result: '0?0-xxxx-2?22'`
	
	[Note]
	
	Since '\' is recognized as an escape character in JAVA, it must be preceded by '\' ('\\').

Corr. Field	Option to use a field as a correlation field. If set to 'true', the value can be called from MappingHandler through API. Only available when the message field type is set.

- **Buttons**

The following describes each buttons.

Item	Description
[Add]	Adds a field.
[Add Fields]	Adds multiple fields.
[Delete]	Deletes a selected field. Enabled when a field is selected.

Item	Description
[Search]	Opens a dialog box for selecting the Message to include. Enabled for an Include type field.
[XML Reset]	Resets XML messages.
[VO Export]	Exports .java and .class files of the Message to the specified path. Each file name is the same as the Message ID.
[Validate]	Validates each field by calling the validation handler configured on the server. For information about creating the handler, refer to "7.2. Handler".

Configuration Items for Each Message Type

The following describes items that can be configured for each Message types.

- **FixedLength**

Item	Description
Length	Message length. (Default value: 10) A numeric field name can be specified for variable length field (same as for variable arrays).
Offset	Offset that expresses the start of a field. Since it is calculated automatically, it cannot be set. The offset is the value that the lengths of all previous fields are added up. If one or more previous fields have a variable length, the offset is expressed as Unknown.
Align, Fill	Character used to fill the empty spaces when marshaling DataObject field into a byte array after aligning the values to either left or right when the length of the value is shorter than that of the DataObject field.
Decimal	Number of decimal places to keep for numeric types. – If set to 1 for a dto numeric field whose value is 123.4321xxxx, the field will be marshaled to 123.4. If set to 4, it will be marshaled to 123.4321. – If set to 4 for the previous field (123.4321xxxx) set with a fixed-length of 6, it will be marshaled to 3.4321 giving precedence to the decimal setting and then the length setting of 6, which includes the decimal point. – If set to -1, the whole numbers are preserved before the decimal places. In this case, the marshaled value will be 123.43 for the previous example.
Point	Used together with the Decimal item to support assumed decimal point or fixed point used in Mainframe.

Item	Description
	For example, set Decimal to 3 and Point to false if the fixed point position is the third from the right end of the number.
Sign	Used for fields that use comp3 or unpacked decimal encoding that uses the assumed decimal point used in Mainframe. C, D, or F value is appended at the end according to the encoded value. – If Sign = true, a sign is always appended. – If Sign = false, a sign is appended only when the dto field value is negative.
Encode (Fixed Length)	Message field encoding type. Encoding Types: – Char – KoreanSingleByte – KoreanDoubleByte – KoreanSpace – CharSOSI – CharSingleByte – AsciiSOSIInsert – NoConversion – COMP – COMP-2 – COMP-3 – UnpackedDecimal – BCD – IEEE754SinglePrecision – IEEE754DoublePrecision – Two's_Complement
Trailer	Trailer value used to fill when lower 4 bit padding is needed for a string or numeric Message field type when the Encode option is BCD.

Item	Description
Date Pattern (yyyy-MM-dd)	Date format. Only enabled for a Date or Calendar field. (Example: yyyyMMdd, yyyy-MM-dd, yy-MM-dd)
Endian	Record byte order when saving to memory. – Big Endian: keeps the byte order of the record. – Little Endian: reverses the byte order of the record.
EOD	Set to true only for the last field. Otherwise, false. If set to true, the field value is not trimmed according to the Length setting. The field is created as a Byte array.
Unsigned	Set to true for an unsigned field. Otherwise, false.

- **Bitmap**

Item	Description
Length	Message length. (Default value: 10) A numeric field name can be specified for variable length field (same as for variable arrays).
Encode (Bitmap)	Message encoding.
Bitmap Number	Bitmap number of each field. Used as an index to select a field when there are multiple fields.
Number of Fields in Bitmap	Number of fields to use in the bitmap. Only enabled for Bitmap types.

- **Delimiter**

Item	Description
Delimiter	Delimiter for when Message fields are set to an Array. Use a character, such as 'a', 'b', or 'c', or a special character such as '\n' or '\t'. An unprintable character must be entered in Unicode such as '\u0000' or '\u0020' (empty space).
EOD	Set to true only for the last field. Otherwise, false.

- **CSV**

  No additional items need to be configured for a CSV Message.

- **XML**

Item	Description
Is Element	Specifies whether the field is an element. – true: element. – false: attribute.
Attribute Name	Attribute name. (Indicates that the field is an attribute) Only available when **'Is Element'** is set to false.
Attr Use	Specifies whether the field is required. Enabled only if the field is an attribute. Similar to the "use" attribute of XSD. – optional: can be set optionally. – required: must be set. – prohibited: cannot be set.
Element Name	Element name. (Indicates that the field is an element) Only available when **'Is Element'** is set to true.
XML Type	XML type in QName format for the selected field type. (Example: {http://www.tmax.co.kr/test}tesType)
Min Occurs Max Occurs	Same as 'minOccurs' and 'maxOccurs' attributes in XSD. Max Occurs can be set to unbounded.
Nillable	Option to allow using null for the field value.
Date Pattern(yyyy-MM-dd)	Date format. Only enabled for a Date or Calendar field.

- **JSON**

Item	Description
EnCode(JSON)	Message field encoding.
Date Pattern(yyyy-mm-dd)	Date format. Only enabled for a Date or Calendar field.
JsonKey	JsonKey to parse. If not set, Physical Name is parsed.

- **ValueObject**

  No additional items need to be configured for a Value Object Message.

- **KeyValue**

Item	Description
Date Pattern(yyyy-mm-dd)	Date format. Only enabled for a Date or Calendar field.
Trim	Option to trim the Message. – none: do not trim the Message. – rtrim: trim the right of the Message. – ltrim: trim the left of the Message. – lrtrim: trim both sides of the Message.
Null Value	If an entered value and an actually received value are the same, a Message is created without including the matched key-value during marshalling. If the values are not the same or a value that matches the key-value is not set, a Message is created with including the key-value.

- **FDL (Fixed Definition Language)**

Item	Description
Field Key	Basic field info and field key. Must use the field key configured on the Tmax (for FDL) that AnyLink will connect to.

- **FML (Fixed Manipulation Language)**

Item	Description
Field Key	Basic field info and field key. Must use the field key configured on the Tmax (for FDL) that AnyLink will connect to.

- **ISO8583**

Item	Description
Length	Message length. (Default value: 10) A numeric field name can be specified for variable length field (same as for variable arrays).

Chapter 4. Outbound Rule Editor

This chapter describes the functions and uses of the Outbound Rule editor.

4.1. Overview

An Outbound Rule is a service that sends data from RTE to Outbound Adapter. An Outbound Rule stores the message format that was received from the Inbound Adapter, and it can be used to define the format or content of the message for a successful or failed transfer.

The default Outbound Rule is created if the user does not create one. If an Outbound Rule is created, it automatically processes any messages with an input data format that matches the format defined in the rule.

An Outbound Rule can be added to the BizTx, but this may require data mapping between the Outbound Rule message and input message.

Note

1. This chapter only covers Custom Log and Batch Outbound Rules. For detailed information about creating other Outbound Rules, refer to each Adapter guide.

2. A project and BizTx/BizTx Group must be already created. For more information, refer to "1.4. Creating a Project" and "2.1. Creating a BizTx/BizTx Group".

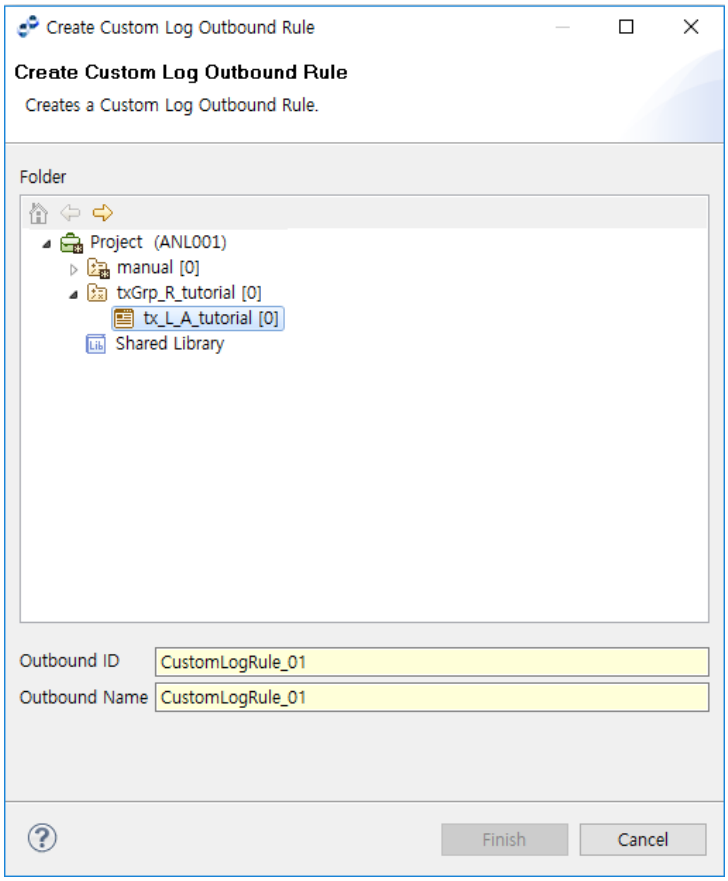
4.2. Custom Log Outbound Rule

A Custom Log Outbound Rule can be configured through Service Flow Rule Activity, Response Message Activity's Request, Response part and Start Message event mappings. This section describes how to create and configure a Custom Log Outbound Rule.

4.2.1. Creating a Custom Log Outbound Rule

To create an Outbound Rule, select **[New] > [Outbound Rule] > [Custom Log Outbound Rule]** from the context menu of the **Project Navigator**. Enter the required items in the **Create Outbound Rule** window, and then click **[Finish]**.

[Figure 4.1] Create Custom Log Outbound Rule Window



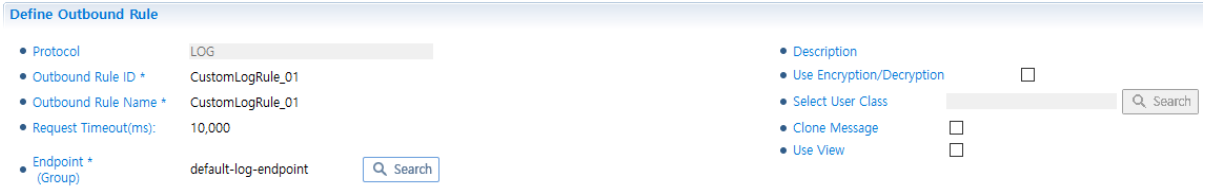
Item	Description
Outbound ID	Outbound Rule ID. Only alphanumeric and special (_) characters are allowed, and the first character must be capitalized.
Outbound Name	Outbound Rule name. Only Korean, alphanumeric, and special (_) characters are allowed. Must follow the XML Naming Convention.

4.2.2. Configuring a Custom Log Outbound Rule

The following describes the configuration items for a Custom Log Outbound Rule.

- **Define Outbound Rule**

[Figure 4.2] Define Outbound Rule

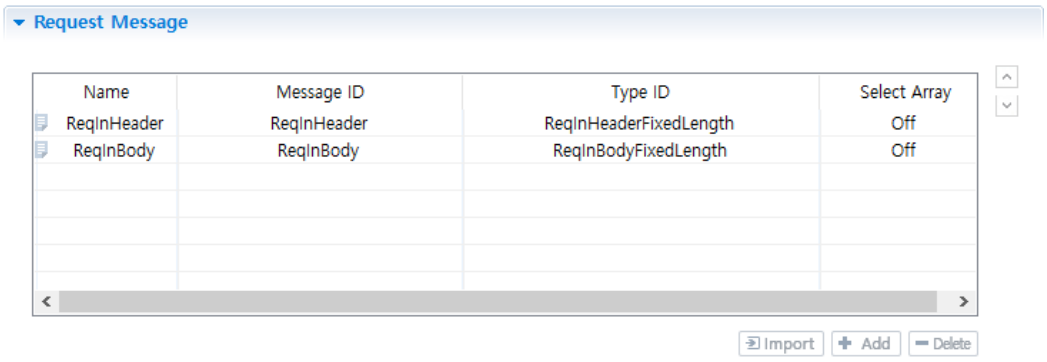


Item	Description
Protocol	Protocol of the Outbound Rule. The protocol is selected when an Outbound Rule is initially created.
Outbound Rule ID	Outbound Rule ID. Duplicate value is not allowed. (Required)
Outbound Rule Name	Outbound Rule name. (Required)
Request Processing Timeout (ms)	Timeout for processing outbound data transfer.
Endpoint (Group)	Endpoint or endpoint group connected to the external system to send data to. Click **[Search]** to open the **Resource Search** window to select an endpoint or endpoint group. (Required)
Description	Description about the Outbound Rule for easy identification.
Use Encryption/Decryption	Option to encrypt/decrypt log data.
Select User Class	Encryption user class. Click **[Search]** to open the **Resource Search** window to select a user class.
Clone Message	Option to clone a variable to be able to log its value regardless of BizTx execution time (using this option may reduce memory efficiency).
Use View	Option to use database view. If the view is used, request messages, log table mapping, and Insert SQL statements are not editable.

- **Request Message**

Input message type that is used to match the format of the input received from the Outbound endpoint.

[Figure 4.3] Request Message



- **Log Table**
  - **Log Table Name**

Select a log table included in the endpoint. To set a log table name, Endpoint (Group) must be set. Click **[Mapping]** to open the **Mapping** dialog box ([Figure 4.5]) to map the request message to a log table column.

[Figure 4.4] [Log Table] - Log Table Name

If the database account of the database set in Endpoint (Group) only has DML privilege, the table list cannot be retrieved. To retrieve the list, set the following jvm option of JEUS Domain Admin Server (DAS).

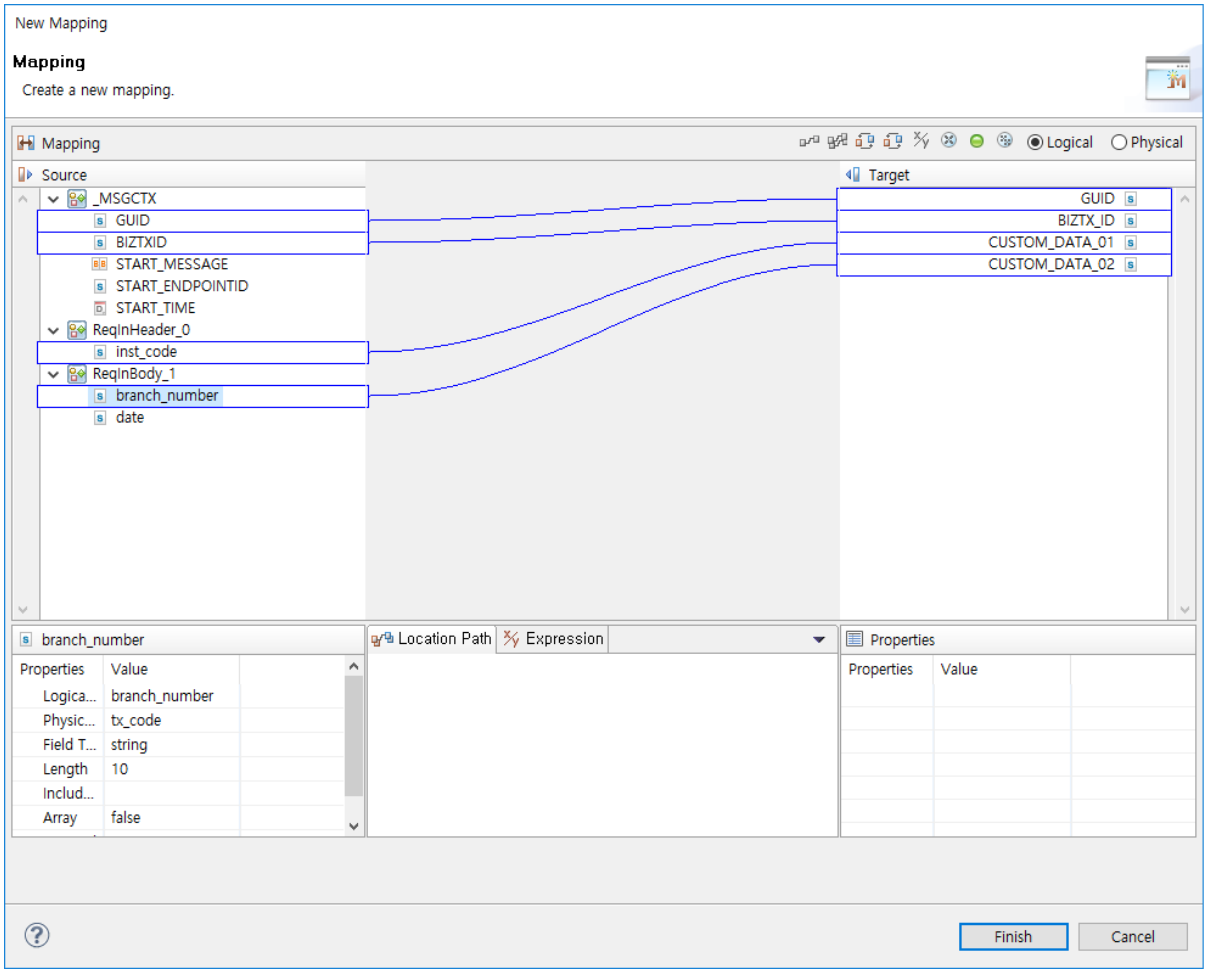
```
-Danylink.dis.db.noperm=true
```

– **Mapping Dialog Box**

AnyLink BizTx's transaction or trace information or messages set as request messages are mapped to custom log table columns. The mapped information is saved in a database table at logging.

The Source section displays **_MSGCTX** messages that can be mapped to AnyLink transaction or trace information and request messages set in the custom log rule. The Target section displays columns of the database table set in Log Table Name.

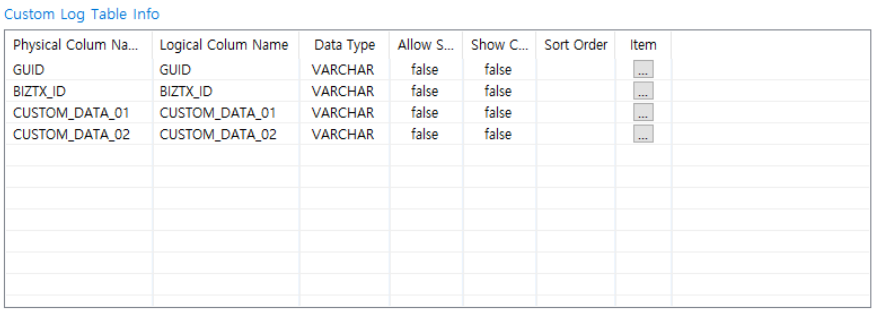
[Figure 4.5] [Log Table] - [Custom Log Table Info] - Mapping Dialog Box



– **Custom Log Table Info**

Displays information about the selected Custom Log Table. Use **[Add]** or **[Delete]** to modify the table.

[Figure 4.6] [Log Table] - Custom Log Table Info

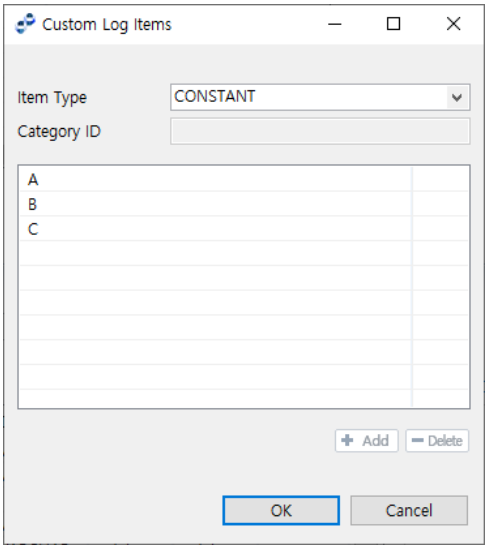


Item	Description
Physical Colum Name	Physical column name.
Logical Colum Name	Logical column name.

Item	Description
Data Type	Data type of the column (editable). If set to BLOB or CLOB type, Allow Search and Show Column are both set to false and cannot be edited.
Allow Search	Option to enable the column to be searchable in WebAdmin. Unless it is set to true, the column is not displayed as a custom log search condition.
Show Column	Option to show the column in WebAdmin.
Sort Order	Sort order for search results in WebAdmin. – Ascending – Descending
Item	Item that can also be configured from a combo box in WebAdmin. Clicking the button opens the **Custom Log Items** ([Figure 4.7]) dialog box where items can be configured.

Configure an item in the following dialog box and then click **[OK]**.

[Figure 4.7] [Log Table] - Custom Log Items Dialog Box



Item	Description
Item Type	– CONSTANT: Enter a fixed value to the table below. – USER_META: Use meta information of WebAdmin.

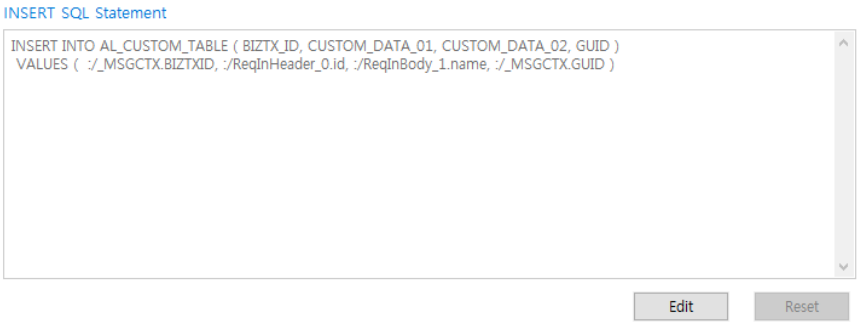
Item	Description
Category ID	Category ID of meta information to use for the item list. Enabled when Item Type is set to USER_META.
CONSTANT Item Table	Enabled when Item Type is set to CONSTANT. To add an item, click **[+ Add]**. To delete an item, click **[- Delete]**.

– **INSERT SQL Statement**

SQL created by mapping a Request Message to a log table column name.

Click **[Edit]** to edit the SQL, and **[Reset]** to undo changes and display the original SQL.

[Figure 4.8] [Log Table] - INSERT SQL Statement



4.2.3. Custom Log Example

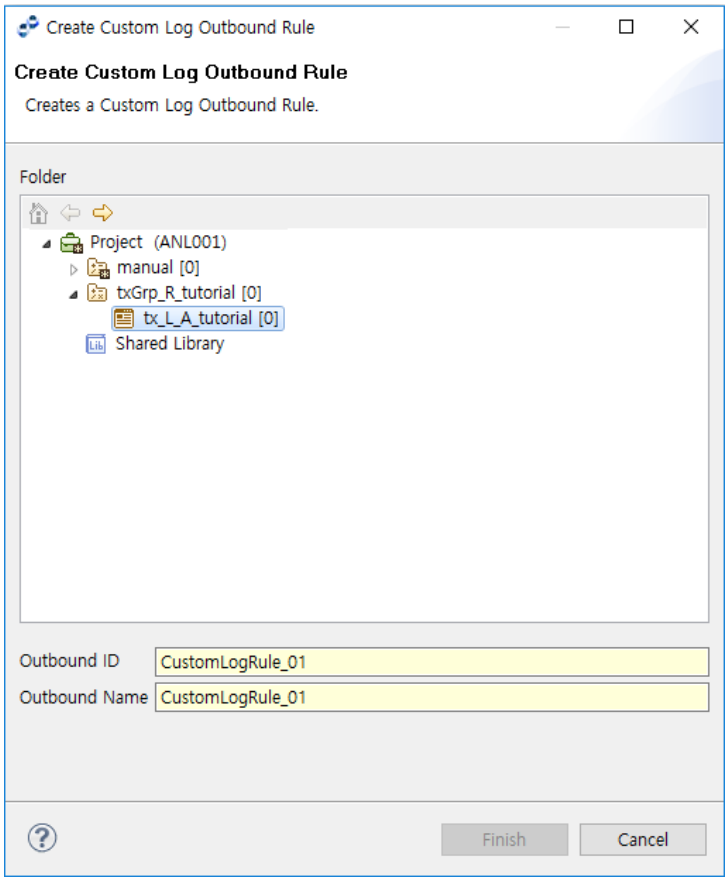
This section describes an example of performing custom logging while executing BizTx.

> **Note**
>
> For information about creating and deploying resources and executing BizTx, refer to *AnyLink TCP Adapter User Guide*.

Creating a Custom Log Outbound Rule

Select **[New] > [Outbound Rule] > [Custom Log Outbound Rule]** from the context menu of the **Project Navigator**.

[Figure 4.9] Create Custom Log Outbound Rule Window

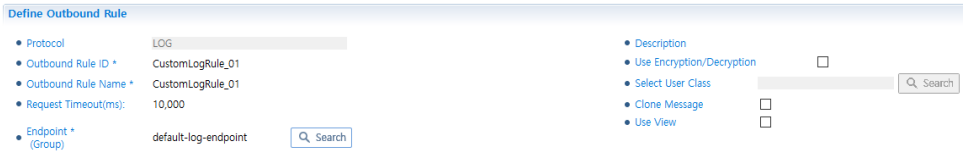


Item	Description
Outbound ID	CustomLogRule_01
Outbound Name	CustomLogRule_01

Configuring a Custom Log Outbound Rule

- **Outbound Rule Definition**

[Figure 4.10] Outbound Rule Definition

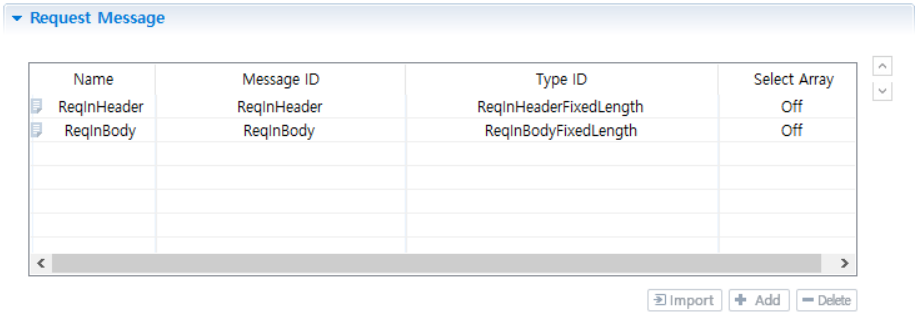


Item	Description
Protocol	LOG
Outbound Rule ID	CustomLogRule_01

Item	Description
Outbound Rule Name	CustomLogRule_01
Request Processing Timeout (ms)	10,000
Endpoint (Group)	default-log-endpoint

- **Request Message**

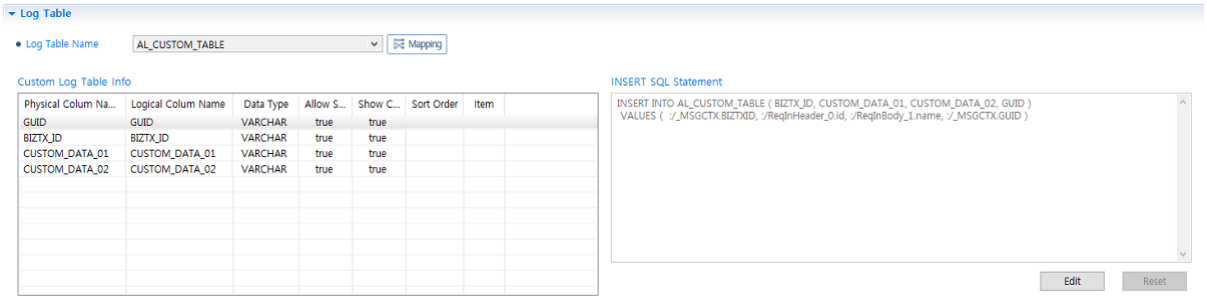
[Figure 4.11] Request Message



Item	Description
Request Message	Request input header (ReqInHeader) Request input body (ReqInBody)

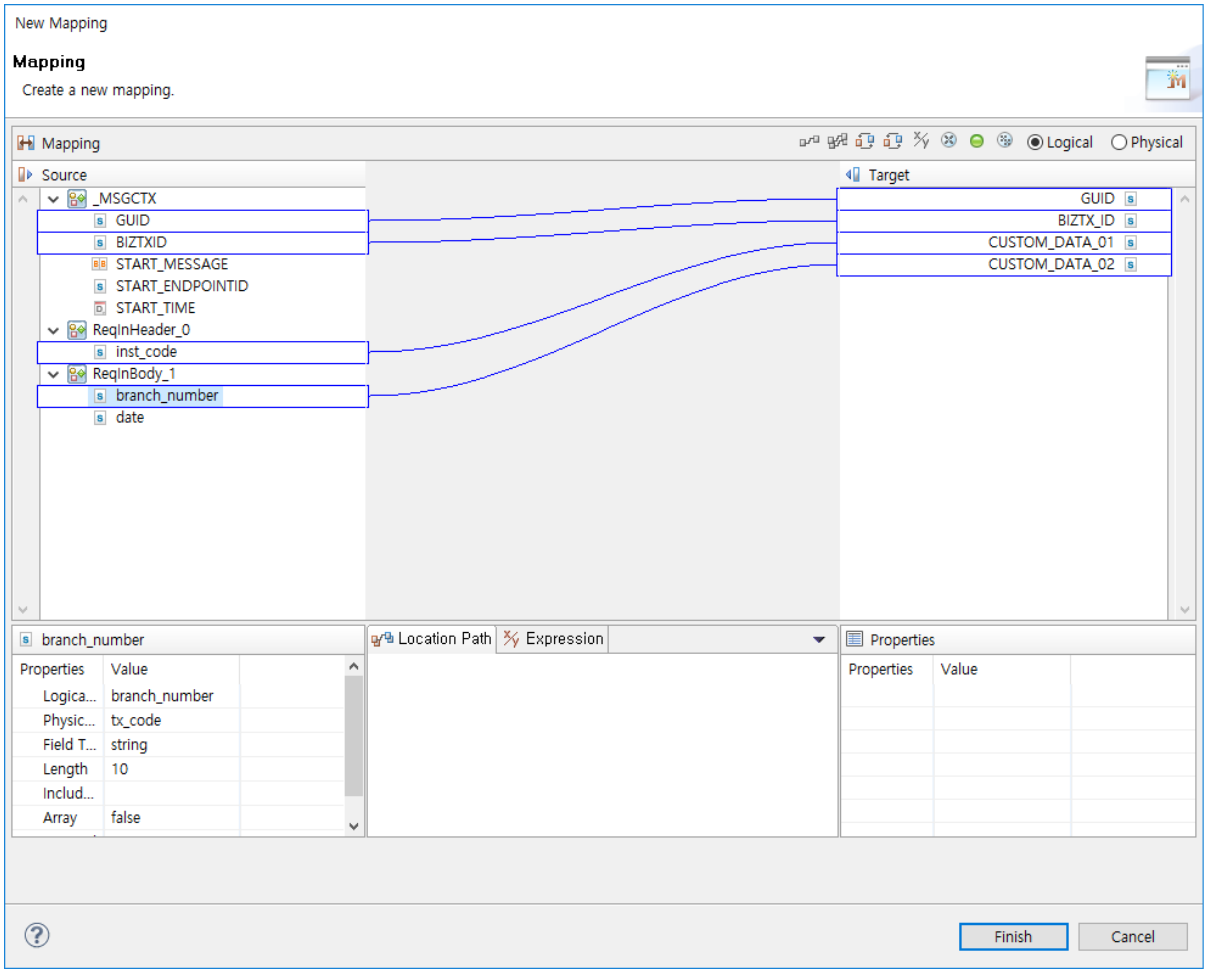
- **Log Table**

[Figure 4.12] Log Table



Item	Description
Log Table Name	AL_CUSTOM_TABLE
Custom Log Table Info	Allow Search: true Show Column: true

[Figure 4.13] Log Table Mapping

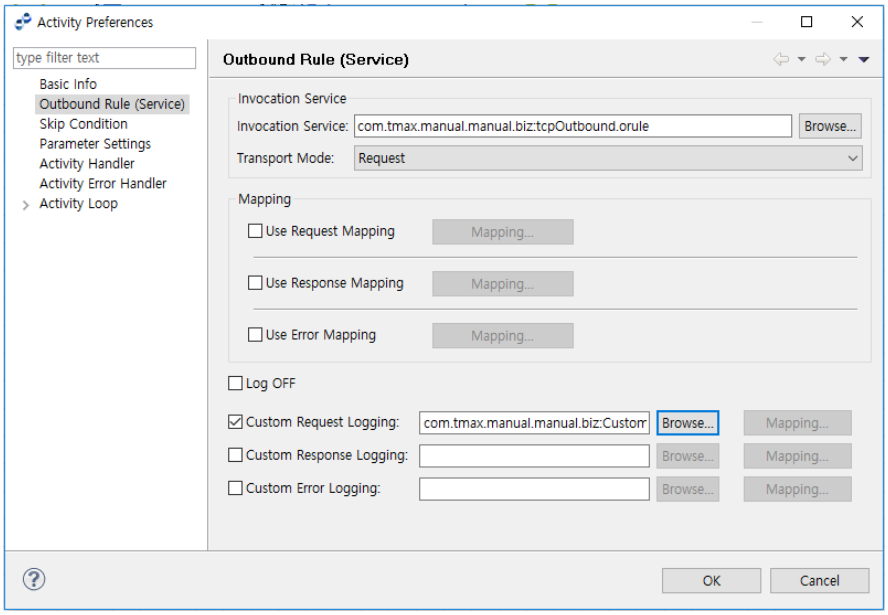


Configuring a Flow

This example shows how to configure a Request Custom Logging for the input messages of TCP Outbound Rule.

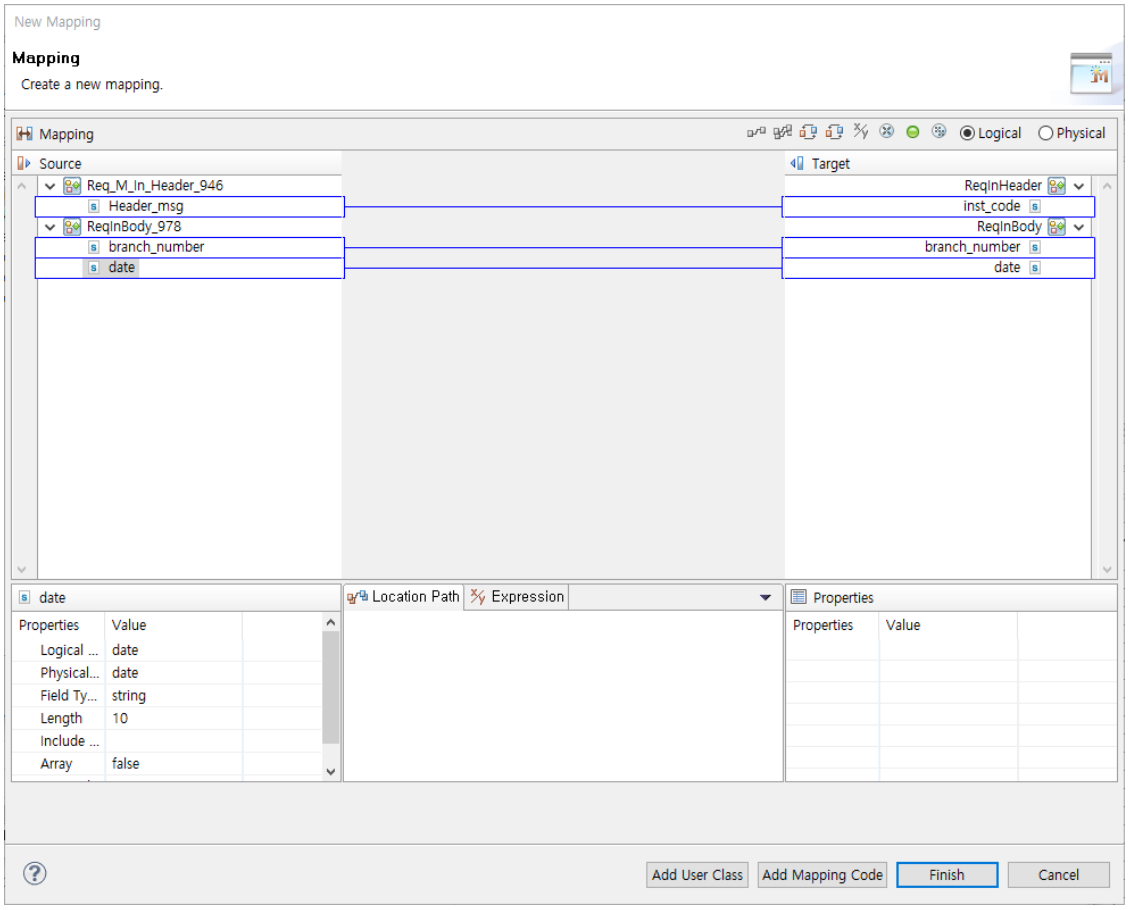
The following figure is the Outbound Rule (Service) configuration section in the TCP Activity Preferences window. Click **[Browse...]** next to the Request Custom Logging, and select a Custom Log Outbound Rule file (CustomLogRule_01.orule).

[Figure 4.14] Outbound Call - Outbound Rule (Service)



The following is the Request Custom Log Mapping window. Add the Activity's input parameters in the **Source** side, and Custom Log Outbound Rule's request messages in the **Target** side.

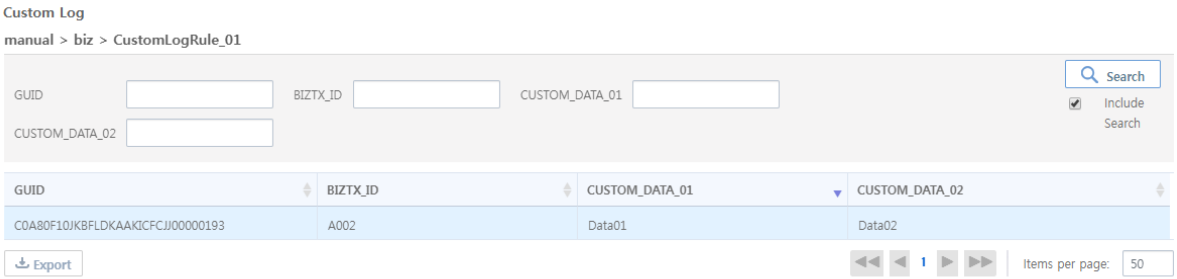
[Figure 4.15] Request Custom Log Mapping



BizTx Test Run Result

The custom logging result can be check by selecting **[Monitoring]** > **[Custom Log]** in WebAdmin.

[Figure 4.16] Custom Log



Item	Description
GUID	BizTx GUID.
BIZTX_ID	**BizTx Code** field value of the ReqInHeader (TCP Activity's Input Parameter) message.
CUSTOM_DATA_01	**ID** field value of the ReqInHeader (TCP Activity's Input Parameter) message.
CUSTOM_DATA_02	**Name** field value of the ReqInHeader (TCP Activity's Input Parameter) message.

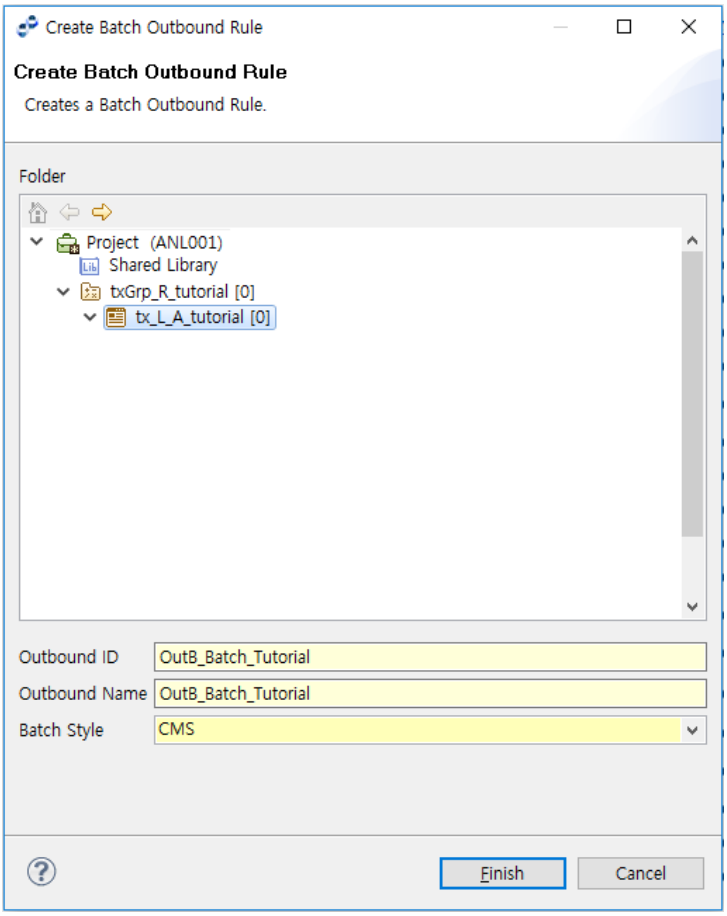
4.3. Batch Outbound Rule

This section describes how to create and configure a Batch Outbound Rule. Configuration items vary depending on the batch style.

4.3.1. Creating a Batch Outbound Rule

To create a Batch Outbound Rule, select **[New] > [Outbound Rule] > [Batch Outbound Rule]** from the context menu of the **Project Navigator**. Enter the required items in the **Create Outbound Rule** window, and then click **[Finish]**.

[Figure 4.17] Create Batch Outbound Rule Window



Item	Description
Outbound Rule ID	Outbound Rule ID. Only alphanumeric and special (_) characters are allowed, and the first character must be capitalized.
Outbound Rule Name	Outbound Rule name. Only Korean, alphanumeric, and special (_) characters are allowed. Must follow the XML Naming Convention.
Batch Style	Batch style. Select one of: – CMS: Korea Fair Trade Commission (KFTC) CMS (0600 line). – DACOM_EDI: Batch transfer style used to connect to the EDI service provided by DACOM MagicLink as a host. – KIDI: Korea Insurance Development Institute style used for file transfer between the compulsory insurance subscription management network and insurance company or mutual aid association systems (0600 line). – SSNW: Samsung Networks style used to communicate with its Alien gateway.

Item	Description
	– KLIA: Korea Life Insurance Association style similar to KFB. However, it does not send a request of ending a job after sending a file through a request of starting receiving, but it waits for a request of ending a job from the organization that requested starting receiving. – SONBO: Bancassurance Non-life Insurance (used to exchange files between Bancassurance system and banks or insurance companies). – KNIA: General Insurance Association of Korea style used to connect to the association's network. It is different from SONBO in the way that it does not know the full length of a data message to send or receive. – KFB: Credit Information style of Korea Federation of Banks (KFB). Batch transfer protocol used to exchange credit information files between KFB and joining organizations (banks). – FTP: File Transfer Protocol (FTP) used to exchange files between client and server across a TCP/IP network. – KCREDIT: Batch transfer protocol (proposed by Korea Credit Information Services) used when a participating organization transmits or receives a credit information file to or from KCREDIT. – KSD: Korea Security Depository (KSD) style used to exchange data and files between KSD systems and joining organizations' systems. – KOTA: Korea Finance Corporation style used to communicate with Korea Finance Corporation. Similar to CMS.

Note

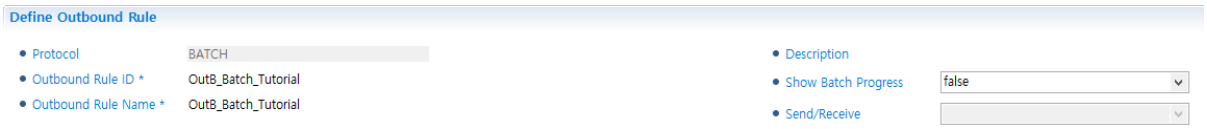
0600 line is used for business transactions related to KFTC batch transfer, Federation of Banks, CMS, bank sheet, KIS rating style, etc. In this batch style, the start message is always 0600/001 and missing message processing is identical. If a 0600/001 message comes in as the start message, the message header or transfer instruction message is exchanged and data is sent as a 0320 message. If 100 messages (0320) are sent and then a 0620 message is sent for a missing message, a 0300 is received as the response and 0310 is used to transfer missing data. It there are no missing messages, 0320 messages are continuously sent.

4.3.2. Configuring a Batch Outbound Rule

The following describes the configuration items for a Batch Outbound Rule.

- **Outbound Rule Definition**

[Figure 4.18] Outbound Rule Definition

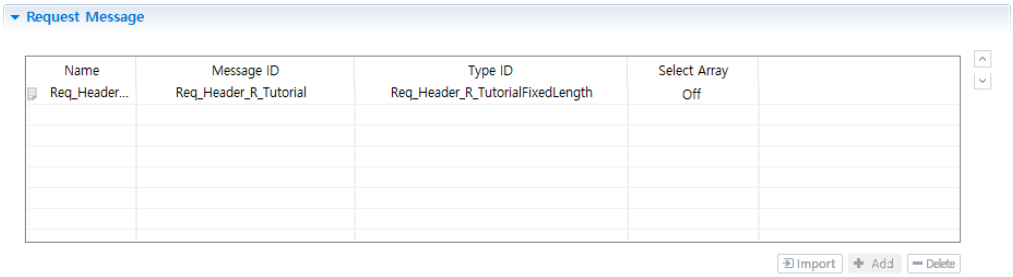


Item	Description
Protocol	Protocol of the Outbound Rule. The protocol is selected when an Outbound Rule is initially created.
Outbound Rule ID	Outbound Rule ID. Duplicate value is not allowed. (Required)
Outbound Rule Name	Outbound Rule name. (Required)
Description	Description about the Outbound Rule for identification.
Show Batch Progress	Option to use the menu for showing batch progress in WebAdmin. (Default value: false)
Send/Receive	Sets whether the batch is sending or receiving. Only available when Show Batch Progress is set to true.

- **Request Message**

Input message type. Messages including data (such as file code, txDate, and filler) that can be used to search for batch style are configured.

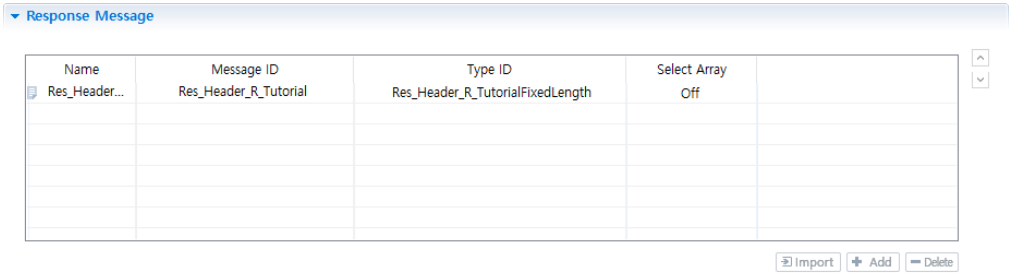
[Figure 4.19] Request Message



- **Response Message**

Message to map batch style search result. Click **[Add]** to set the response message.

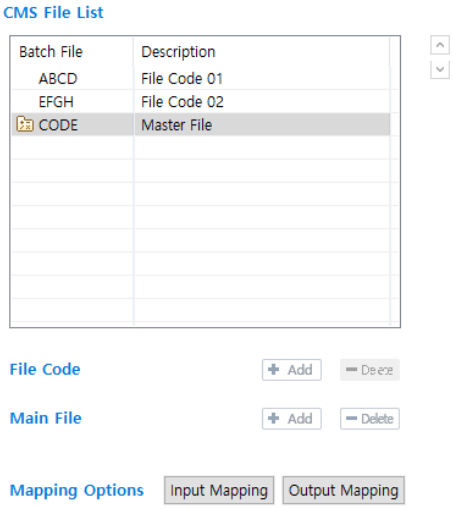
[Figure 4.20] Response Message



- **File List**

Click **[Input Mapping]** to map a file code to the request message, and click **[Output Mapping]** to map the response message to the Batch Style setting.

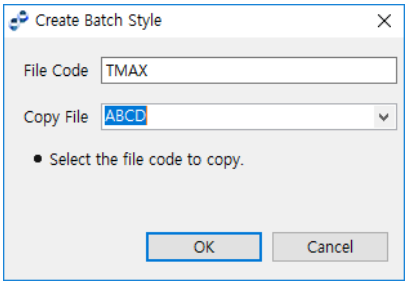
[Figure 4.21] File List



– **File Code**

Click **[Add]** to open the **Create Batch Style** dialog box to add a file code and set a batch style for it. If a file with an existing file code is selected in **Copy File**, a batch style of the selected file code is created. Click **[Delete]** to delete the file code.

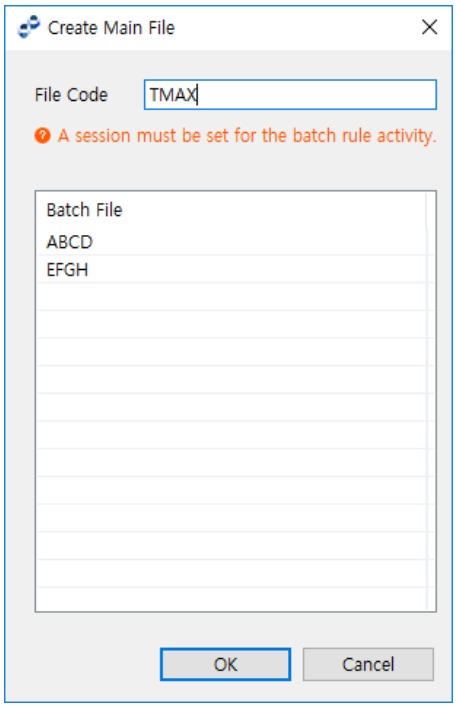
[Figure 4.22] Create Batch Style Dialog Box



– **Main File**

Main file represents a group of file codes. Click **[Add]** to open the **Create Main File** dialog box to add a main file and select file codes to include. Click **[Delete]** to delete the main file.

[Figure 4.23] Create Main File Dialog Box

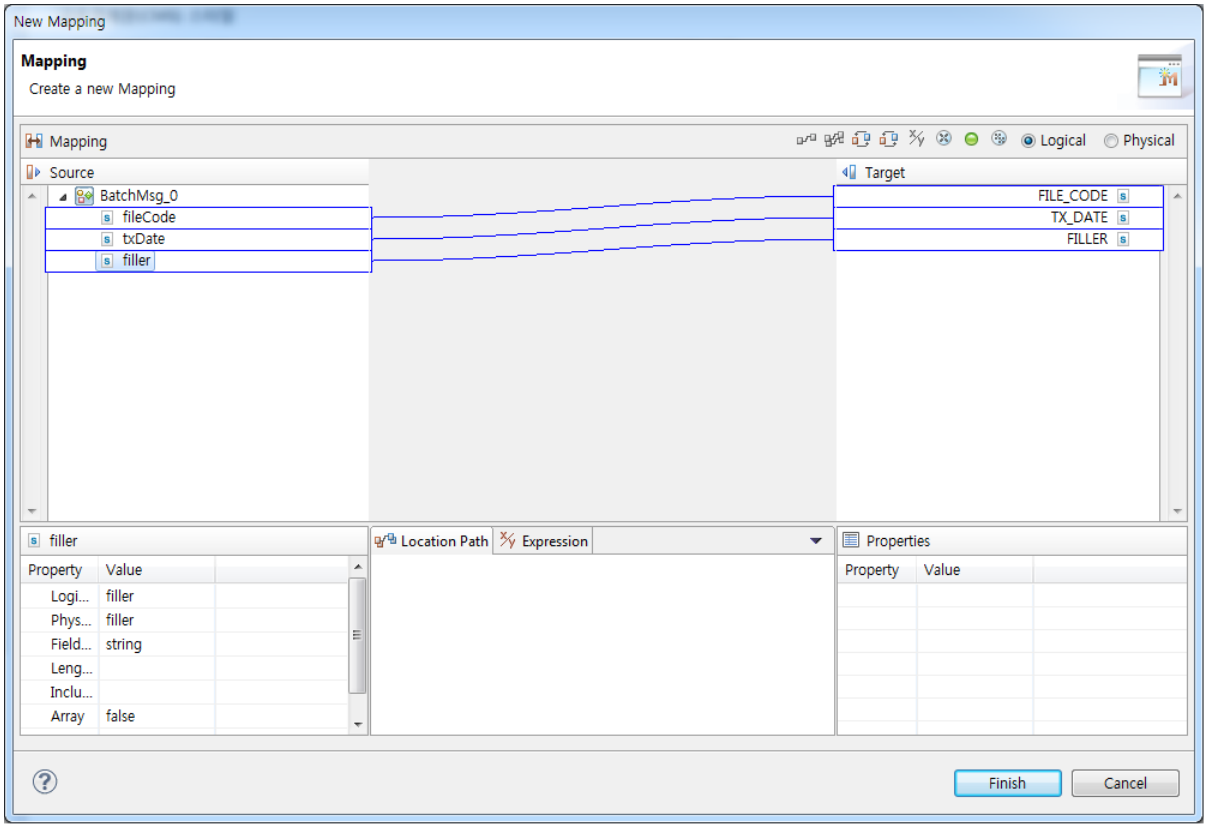


– **Input Mapping**

Map data such as file code, txDate, and filler to get information set in the batch style.

The Source section displays request messages set in the batch rule. The Target section displays FILE_CODE, TX_DATE, and FILLER.

[Figure 4.24] Input Mapping Dialog Box

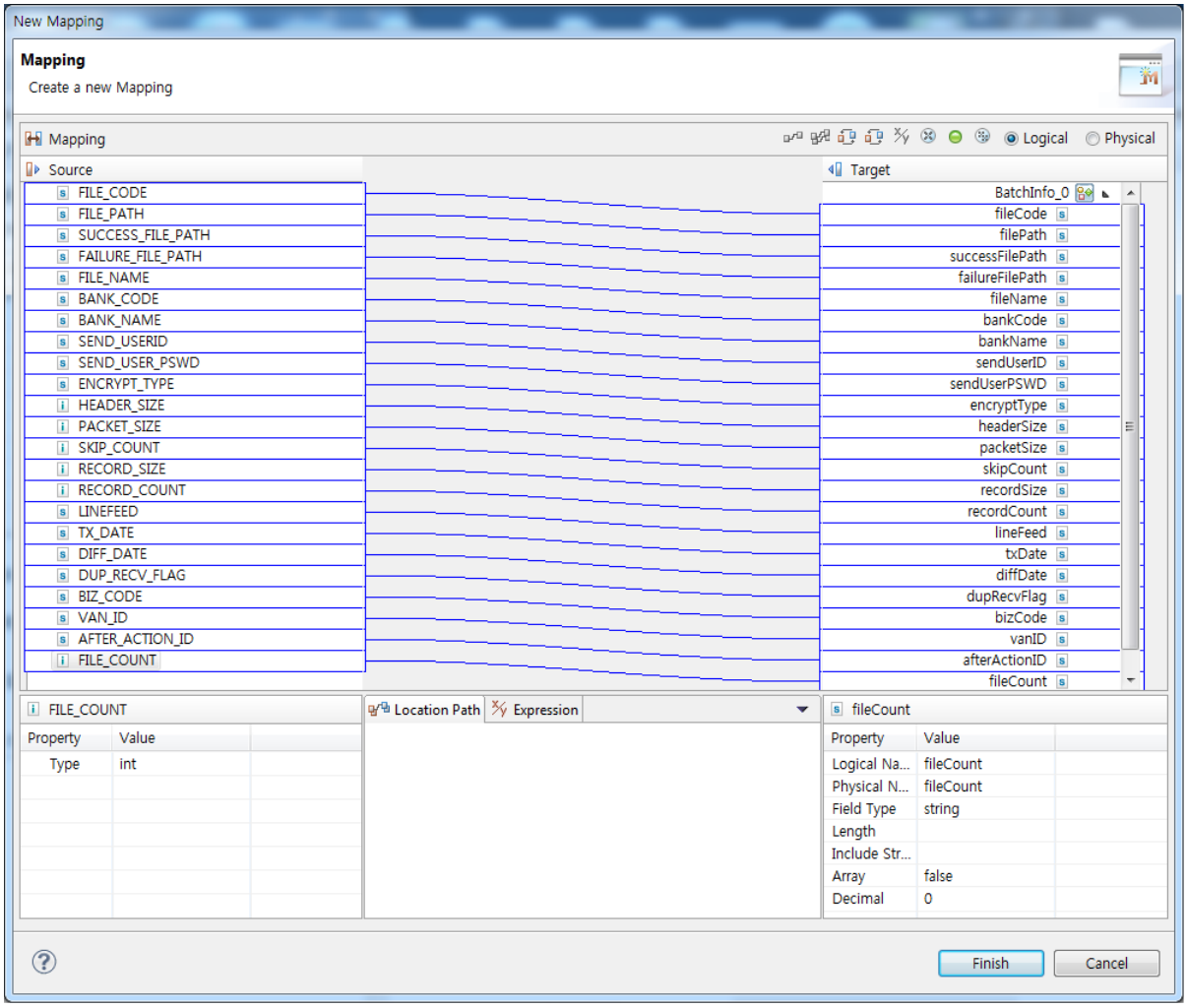


– **Output Mapping**

Map batch style information to a response message. Style information about the batch requested through Input Mapping can be mapped.

The Source section displays values configured for the style, AFTER_ACTION_ID, and FILE_COUNT.

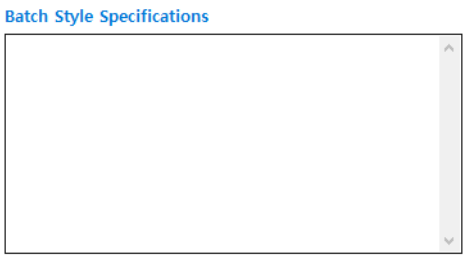
[Figure 4.25] Output Mapping Dialog Box



– **Batch Style Specifications**

Batch style specifications.

[Figure 4.26] Batch Style Specifications

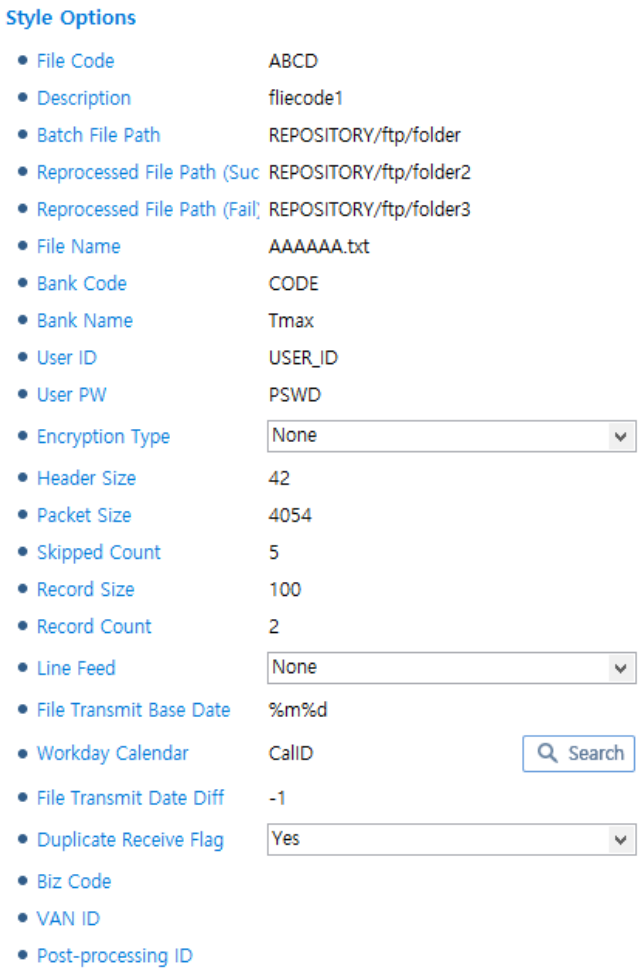


● **Style Options**

For easy creation of a batch job, each batch style has common items used in flows for each file code. They may not be the same as actual specifications.

– **CMS**

[Figure 4.27] CMS Style



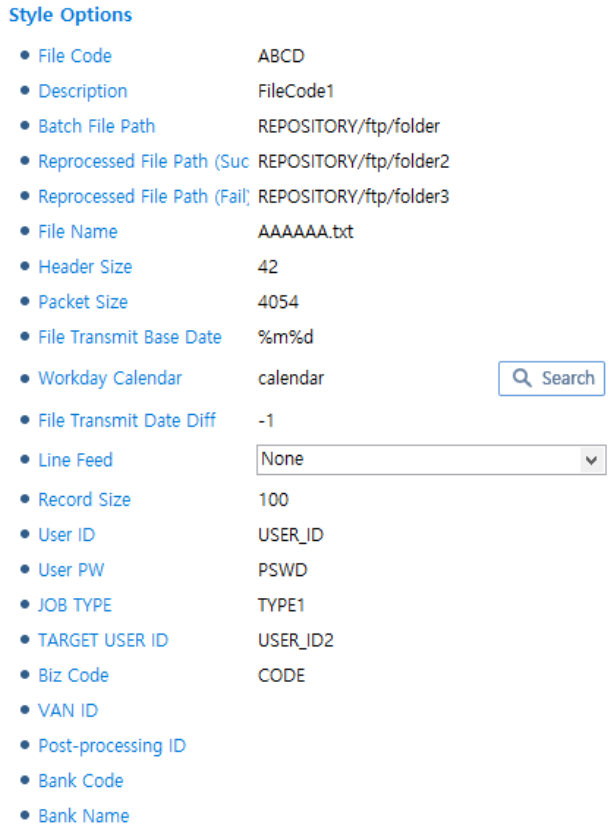
Item	Description
File Code	File code. (Required)
Description	File code description.
Batch File Path	Batch file path in the node.
Reprocessed File Path (Success)	Reprocessed file path in the node for success.
Reprocessed File Path (Fail)	Reprocessed file path in the node for failure.
File Name	File name.
Bank Code	Bank code.

Item	Description
Bank Name	Bank name.
User ID	User ID.
User PW	User password.
Encryption Type	Encryption type. Select one of: – None – KFTC CMS Algorithm – KFTC CMS Algorithm (2009) – KFTC CMS Algorithm (2015)
Header Size	Header size. Set to the Byte item when sending a file info message.
Packet Size	Max message size without the common **'Header Size'**.
Skipped Count	Skipped count.
Record Size	Data record size defined in the file.
Record Count	Number of records to include in a sequence.
Line Feed	Line feed character.
File Transmit Base Date	Base date used to set the file transmit date in the scheduler.
Workday Calendar	Workday calendar created in WebAdmin. The calendar is used to calculate File Transmit Date Diff by excluding holidays.
File Transmit Date Diff	Days to add or subtract from the base date to set the transfer date when scheduling a file transfer. – Negative (-): Before the base date. – Positive (+): After the base date. – Default value: File Transmit Base Date.
Duplicate Receive Flag	Option to indicate a duplicate receive.
Biz Code	Biz Code.
VAN ID	VAN ID.
Post-processing ID	Post-processing ID.

– **DACOM_EDI**

For information about items that are not described in the following, refer to CMS ([Figure 4.27]).

[Figure 4.28] DACOM_EDI Style



Item	Description
JOB TYPE	Job type.
TRGET USER ID	Target user ID.

– **KIDI**

For information about items that are not described in the following, refer to CMS ([Figure 4.27]).

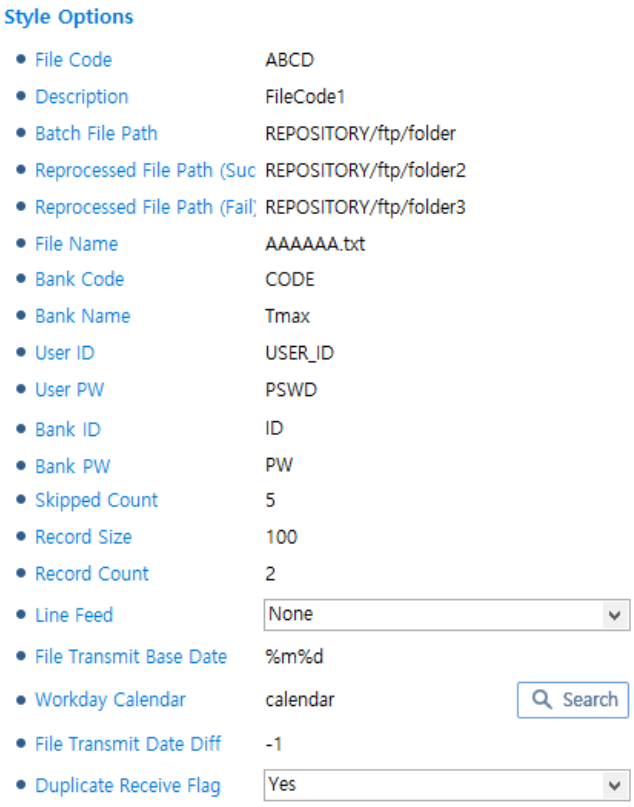
[Figure 4.29] KIDI Style

Item	Description
SYSTEM ID	System ID.

– **SSNW**

For information about items that are not described in the following, refer to CMS ([Figure 4.27]).

[Figure 4.30] SSNW Style

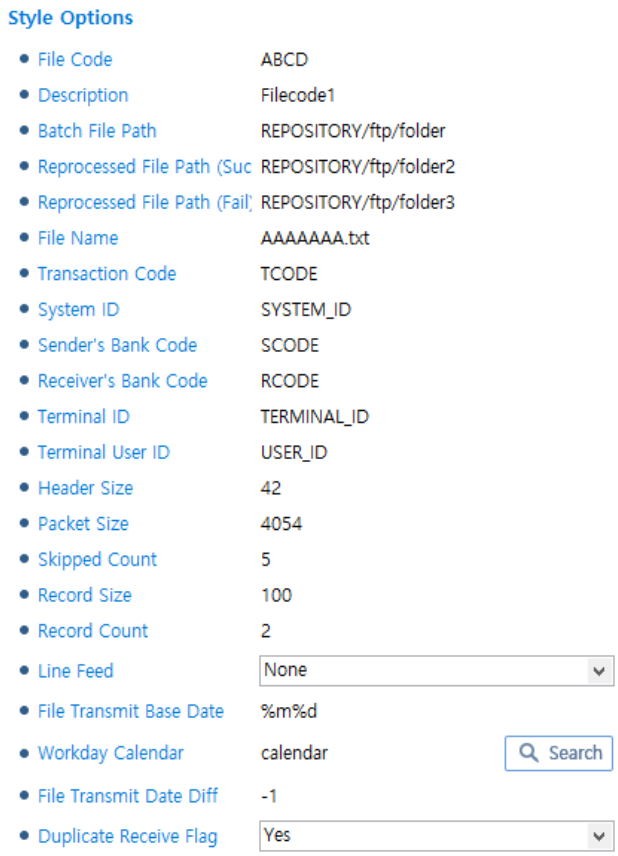


Item	Description
Bank ID	Organization ID.
Bank PW	Password for the organization ID.

– **KLIA**

For information about items that are not described in the following, refer to CMS ([Figure 4.27]).

[Figure 4.31] KLIA Style

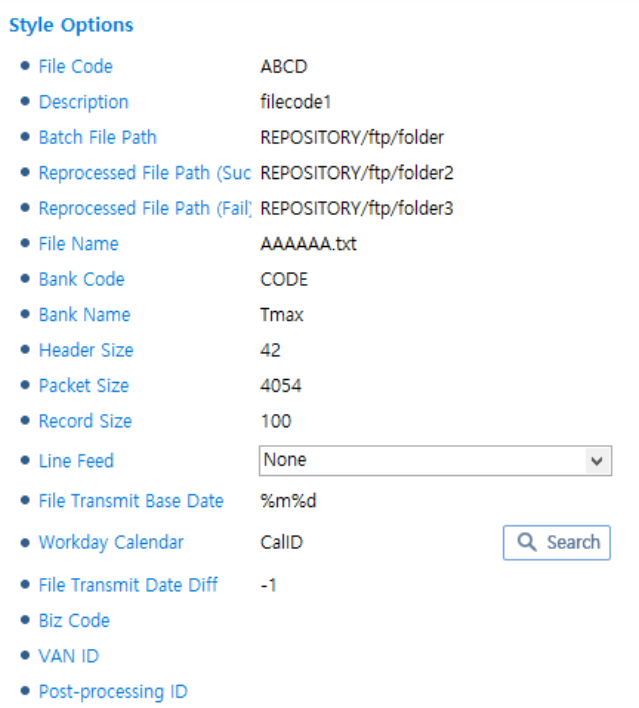


Item	Description
Transaction Code	Transaction code.
System ID	System ID.
Sender's Bank Code	Sender's organization code.
Receiver's Bank Code	Receiver's organization code.
Terminal ID	Terminal ID.
Terminal User ID	Terminal user ID.

– **SONBO**

For information about items, refer to CMS ([Figure 4.27]).

[Figure 4.32] SONBO Style

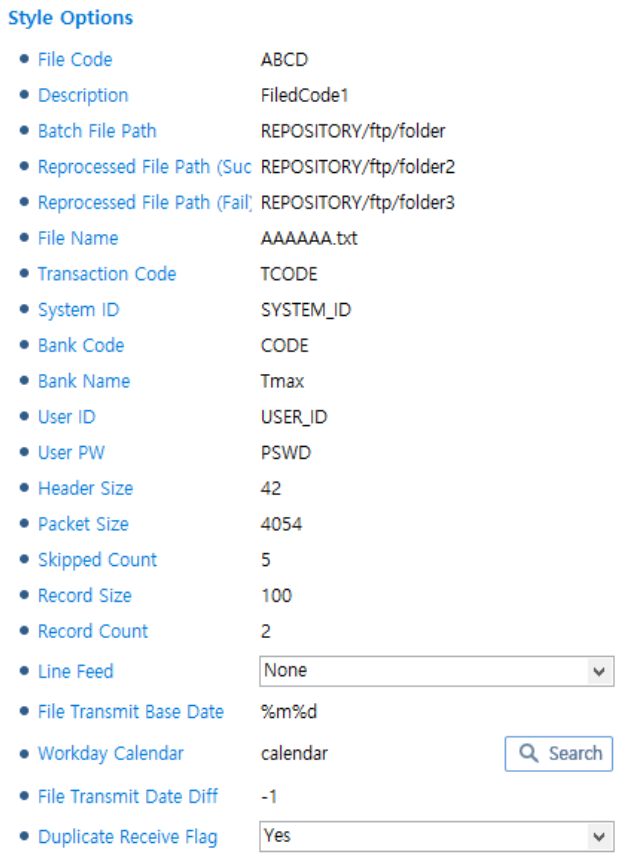


– **KNIA**

For information about items that are not described in the following, refer to CMS ([Figure 4.27]).

Since this style does not know the full length of a data message to send or receive, select Use Length Info Key and enter the length information by using an expression in the Message/Error Handling tab of the TCP Endpoint Details page in WebAdmin. You can go to the page by selecting [System] > [Adapter], clicking Endpoint List, and then clicking the endpoint.

[Figure 4.33] KNIA Style

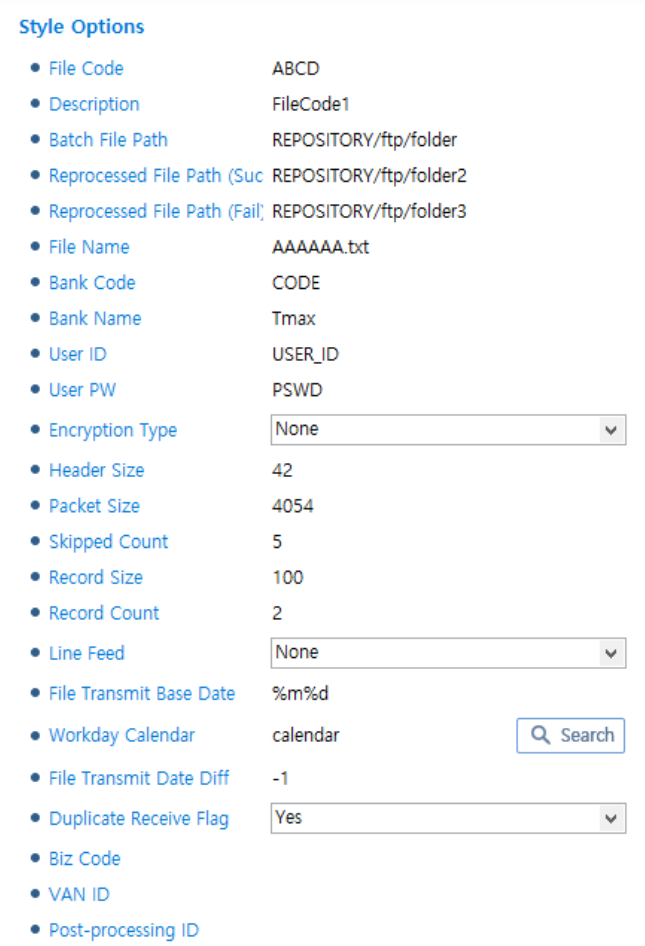


Item	Description
Transaction Code	Transaction code.
System ID	System ID.

– **KFB**

For information about items, refer to CMS ([Figure 4.27]).

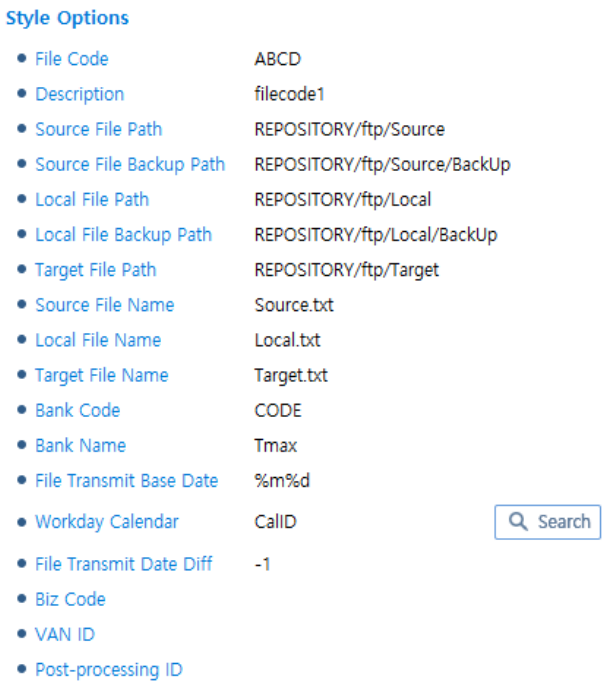
[Figure 4.34] KFB Style



– **FTP**

For information about items that are not described in the following, refer to CMS ([Figure 4.27]).

[Figure 4.35] FTP Style

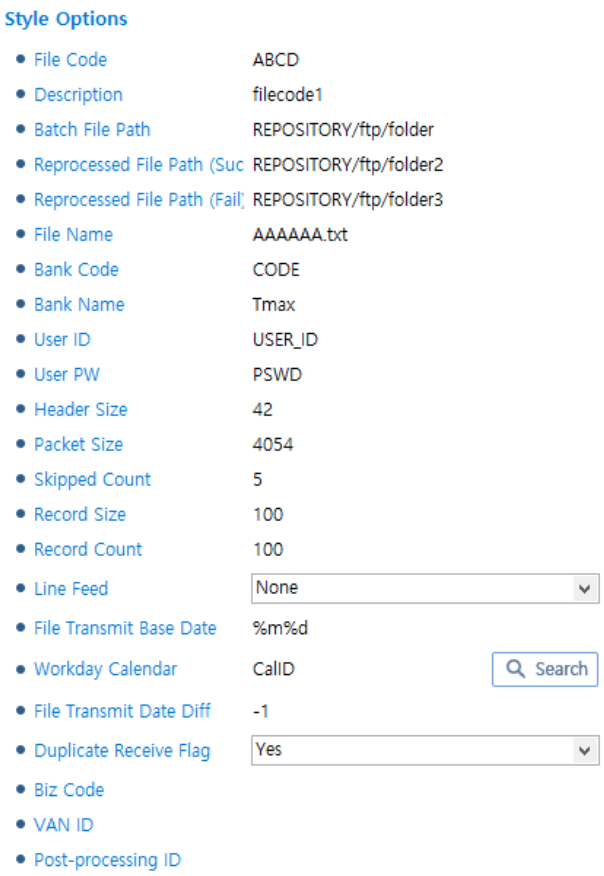


Item	Description
Source File Path	Source file path.
Source File Backup Path	Source file backup path.
Local File Path	Local file path.
Local File Backup Path	Local file backup path.
Target File Path	Target file path.
Source File Name	Source file name.
Local File Name	Local file name.
Target File Name	Target file name.

– **KCREDIT**

For information about items, refer to CMS ([Figure 4.27]).

[Figure 4.36] KCREDIT Style

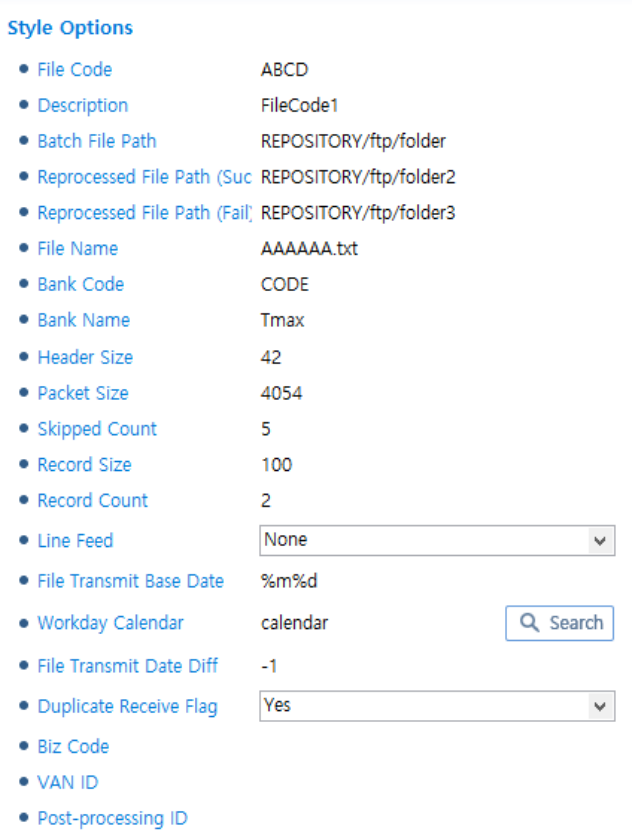


– **KSD**

For information about items, refer to CMS ([Figure 4.27]).

This style receives data by splitting it. For more information, refer to "TCP MESSAGE SPLIT ACTIVITY".

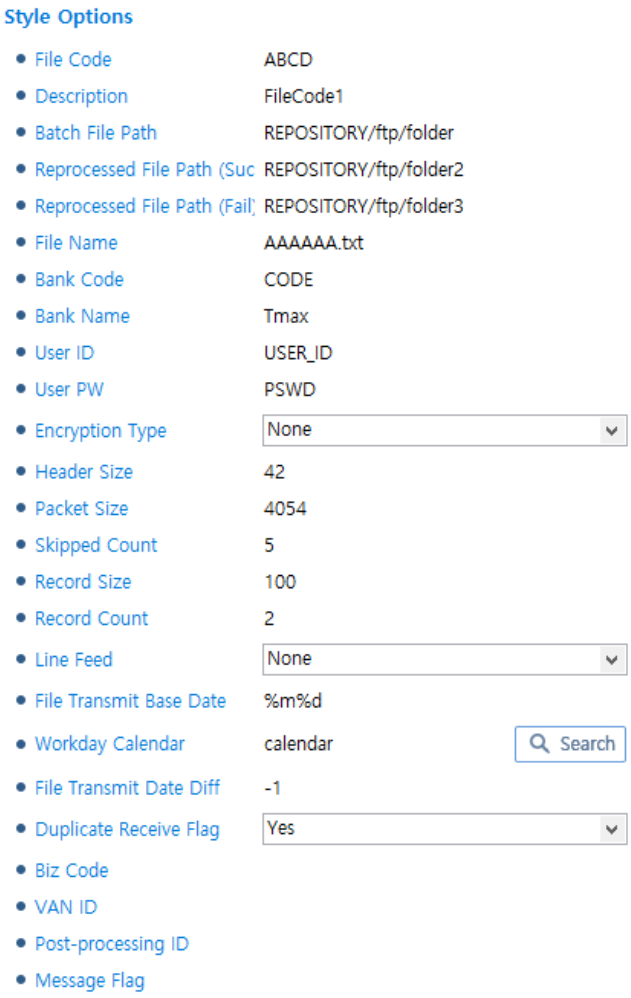
[Figure 4.37] KSD Style



– **KOTA**

For information about items that are not described in the following, refer to CMS ([Figure 4.27]).

[Figure 4.38] KOTA Style



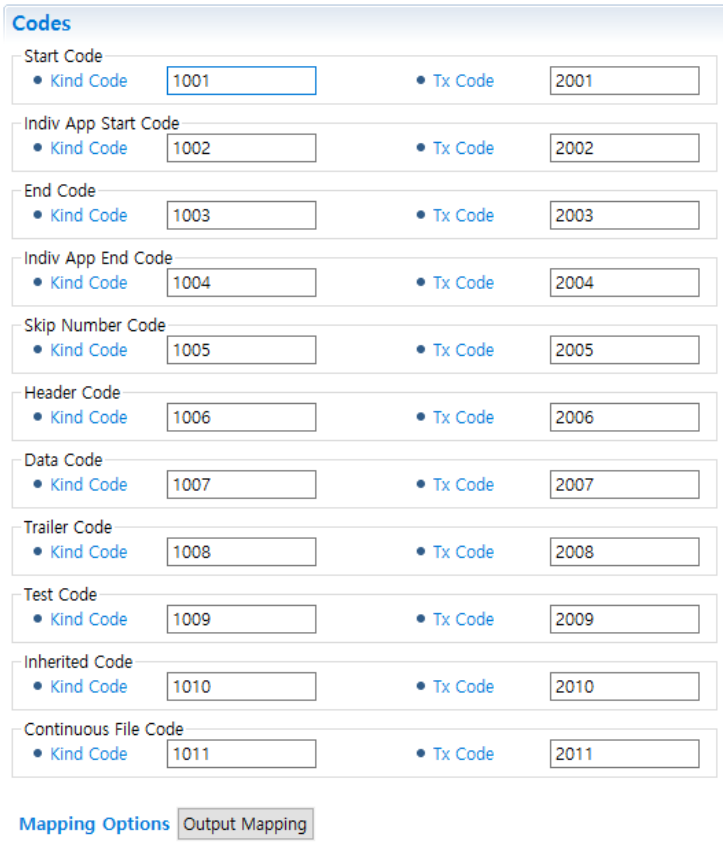
Item	Description
Message Flag	Message flag.

Code Information Configuration

- **Code Information**

  By entering codes of batch flows, you can map the codes to response messages and use them.

  [Figure 4.39] Code Information

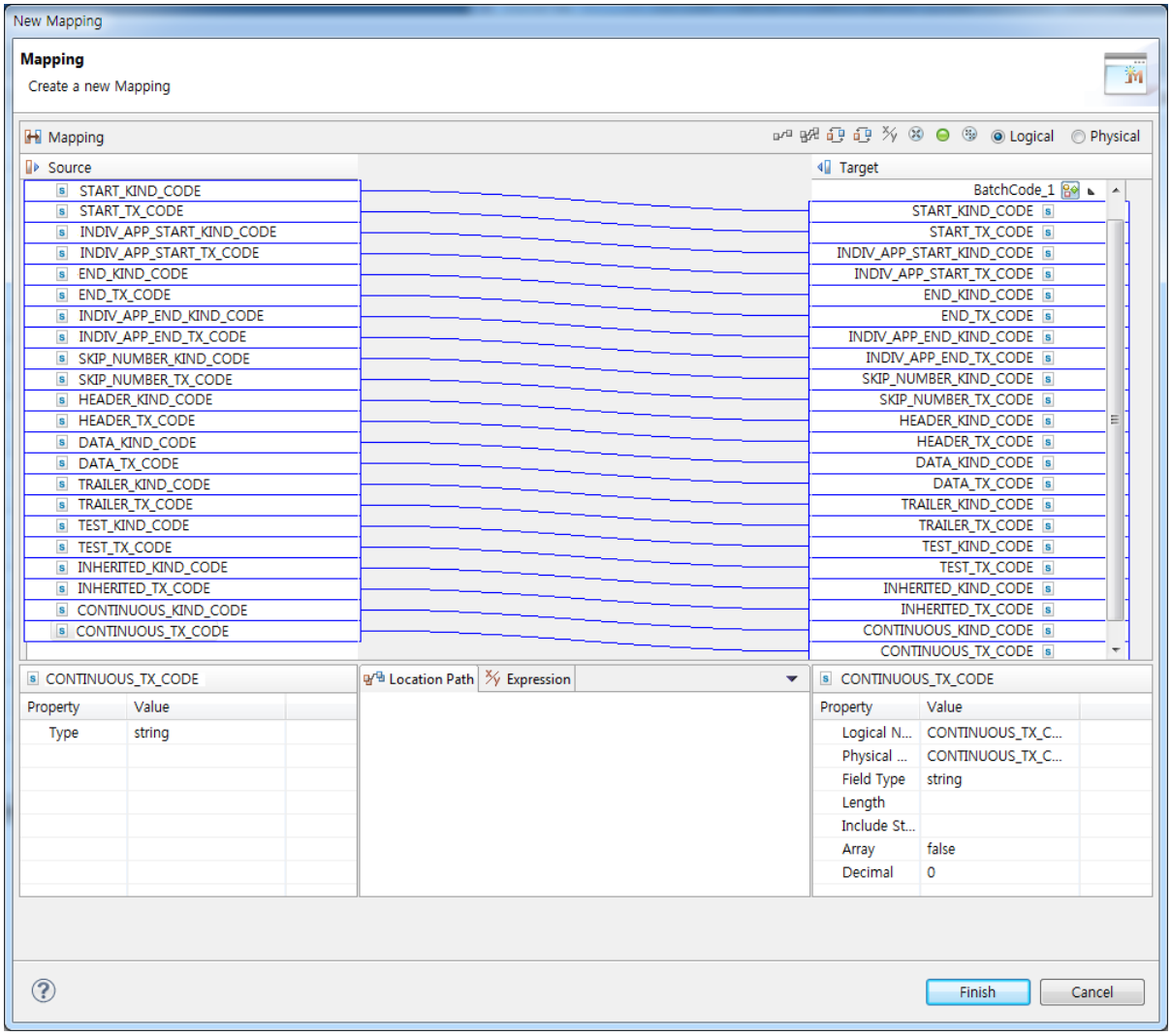


Item	Description
Kind Code	Kind code.
Tx Code	Tx code.
Output Mapping	Maps each code to a response message set in the rule.

- **Output Mapping**

  Maps each code to a response message.

[Figure 4.40] Output Mapping



Section	Description
Source	Displays code information fields.
Target	Displays response messages.

Chapter 5. Multi-Binding Rule Editor

This chapter describes the functions and uses of the Multi-Binding Rule editor.

5.1. Multi-Binding Rule

Multi-Binding handles internal dynamic routing of messages to services through Multi-Binding router that is configured with a Multi-Binding Rule, which performs internal routing by grouping multiple services into a single service.

Multi-Binding Rule groups multiple services and depending on the execution condition splits the flow or concurrently calls multiple services via the multicasting function.

Multi-Binding represents a single 'rule' service that consists of various services including service's message event (Receive), adapter's Outbound Rule, Multi-Binding Rule, and internal service. Multi-Binding Rule can also be called from within a flow by a parsing rule of an adapter.

Splitting a flow into multiple services is similar to the concept of a split used in a "gate activity", a service flow activity. Multicasting, which makes concurrent service calls, performs correlation between the service flow and message events.

Note

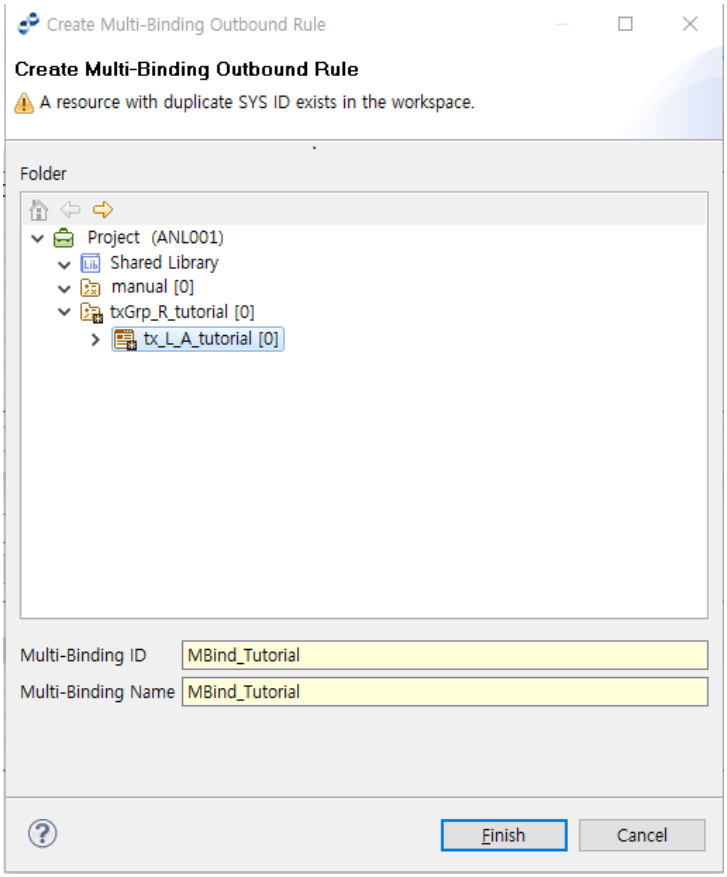
1. For more information about Multi-Binding Rule, refer to *AnyLink Runtime Engine Server Guide*.

2. A Multi-Binding Rule must be created before using Multi-Binding in the Flow Editor. For more information about the Flow Editor, refer to "Chapter 6. Service Flow Editor".

5.2. Creating a Multi-Binding Rule

To create a Multi-Binding Rule, select **[New] > [Multi-Binding]** from the context menu of the **Project Navigator**. Enter the required items in the **Create Multi-Binding Rule** window, and then click **[Finish]**.

[Figure 5.1] Create Multi-Binding Rule Window



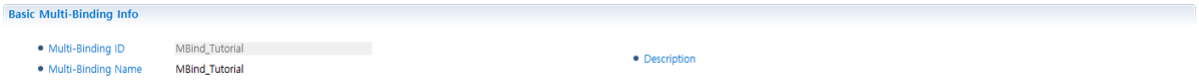
Item	Description
Multi-Binding ID	Multi-Binding ID. Only alphanumeric and special (_) characters are allowed, and the first character must be capitalized.
Multi-Binding Name	Multi-Binding name. Only Korean, alphanumeric, and special (_) characters are allowed. Must follow the XML Naming Convention.

5.3. Configuring a Multi-Binding Rule

The following describes the configuration items for a Multi-Binding Rule.

- **Basic Info**

[Figure 5.2] Multi-Binding - Basic Info



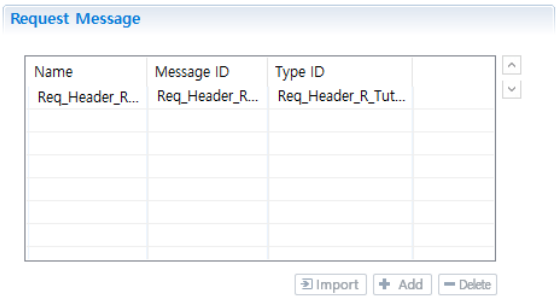
Item	Description
Multi-Binding ID	Resource ID (cannot be edited).

Item	Description
Multi-Binding Name	Multi-Binding name. Duplicate value is not allowed.
Description	Description about the Multi-Binding Rule for easy identification.

- **Request/Response/Business Error Response Message**

Click **[Add]** and add a message in the **Select Message** dialog box ([Figure A.1]), or click **[Delete]** to delete the selected message.

[Figure 5.3] Multi-Binding - Request Message

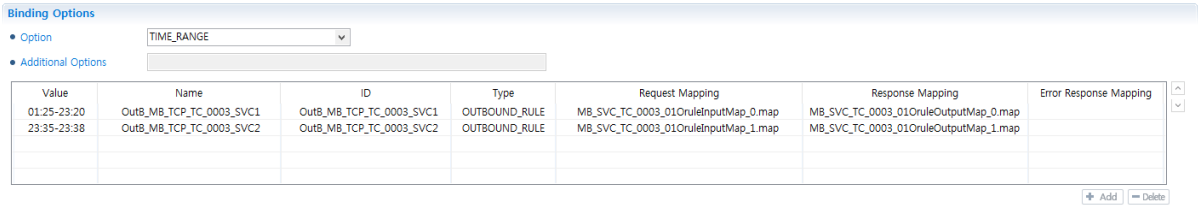


Item	Description
Request Message	Input message format for Multi-Binding Rule.
Response Message	Success response message format when outputting Multi-Binding result.
Business Error Response Message	Failure response message format when outputting Multi-Binding result.

- **Binding Options**

For more information about adding or deleting a mapping, refer to "5.4. Mapping".

[Figure 5.4] Multi-Binding - Binding Options



Item	Description
Routing Option	Routing method for Multi-Binding. – Value: Use the result of the expression set in the 'Option Details' field for routing.

Item	Description
	– RoundRobin: Route each service in round robin fashion. – WeightBased: Weight-based RoundRobin method. – Multicast: Send message to all services and messages. Only applicable to one-way services. – TimeRange: Time range for making a service call. – Flow Correlation: Service flow correlation. First routes to the service item with the matching value. The service item must be a service flow message event that supports correlation. – Handler: Uses user-defined handler to select the routing method. The handler name must be set in the 'Option Details' field. Note that updating this option value will delete all related mappings in the mapping list.
Option Details	Additional details about the selected routing method in the **'Routing Option'** item. Only applicable when the Value or Handler is selected as the 'Routing Option'. – Value: Name of the expression to compute the value that determines which service to route to. – Handler: Handler class name that determines the service to route to.
Mapping List	Mapping between Multi-binding message and service message. The target service is selected according to the specified routing method.

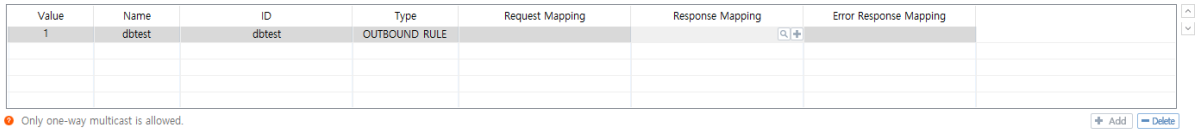
5.4. Mapping

This section describes how to add or delete a message mapping in **[Binding Options] > [Mapping List]**.

5.4.1. Adding/Deleting a Mapping

Click **[Add]** or **[Delete]** below the list to add or delete a mapping. Click **[Add]** to directly enter a new mapping in the new row of the **Mapping List**, or click **[Delete]** to delete the selected row.

[Figure 5.5] Add/Delete Mapping Tab



The following describes the mapping items.

Item	Description
Value	Routing method selected in **[Binding Options] > [Routing Option]**. For example, if the selected option is 'Value' the service with the value that matches the input value is selected for routing.
Name	Name of the service to route to. This is automatically filled when the Service ID is selected in the ID column (cannot be edited).
ID	Service ID to route to. Click **[Search]** in the **'ID'** column to open the **Search Resource** window and select a resource.
Type	Service type. This is automatically filled when the Service ID is selected in the ID column (cannot be edited).
Request Mapping	Mapping between Multi-Binding Rule input message and the target service. Click on the column to display the **[Search]** and **[Add]** buttons. – **[Search]**: Select an existing mapping file from the **Search Resource** window. – **[Add]**: Create a new message mapping from the **Mapping** window.
Response Mapping	Mapping between Normal Response message of the target service and Multi-Binding Normal Response message. Click on the column to display the **[Search]** and **[Add]** buttons (refer to description for Request Mappings).
Error Response Mapping	Mapping between Business Error message of the target service and Multi-Binding Business Error message.

Item	Description
	Click on the column to display the **[Search]** and **[Add]** buttons (refer to description for Request Mappings).

Chapter 6. Service Flow Editor

This chapter describes the functions and uses of the Service Flow editor.

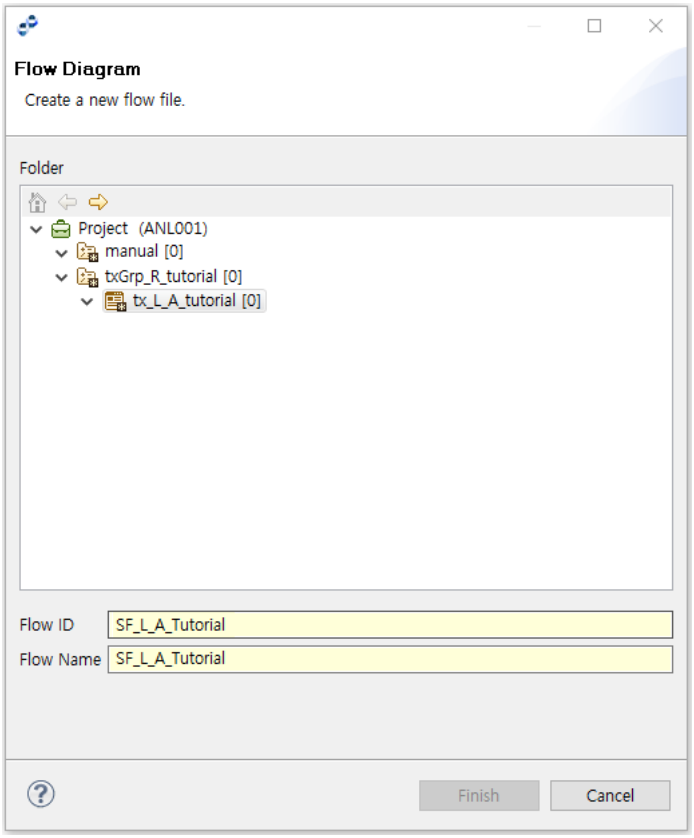
6.1. Overview

A Service Flow is a graphical representation of a sequence of steps in a business service. An AnyLink project must be created before creating a Service Flow Diagram (SFDL: Service Flow Definition Language) file. The incoming message events of the Service Flow are registered as internal services in AnyLink. Most of the services in AnyLink are message events of the Service Flow, and they call the Flow's service activities according to the defined direction of the Service Flow after each step.

6.2. Creating a Service Flow

To create a Service Flow, select **[New] > [Flow]** from the context menu of the **Project Navigator**. Enter the required items in the **Flow Diagram** window, and then click **[Finish]**.

[Figure 6.1] Flow Diagram

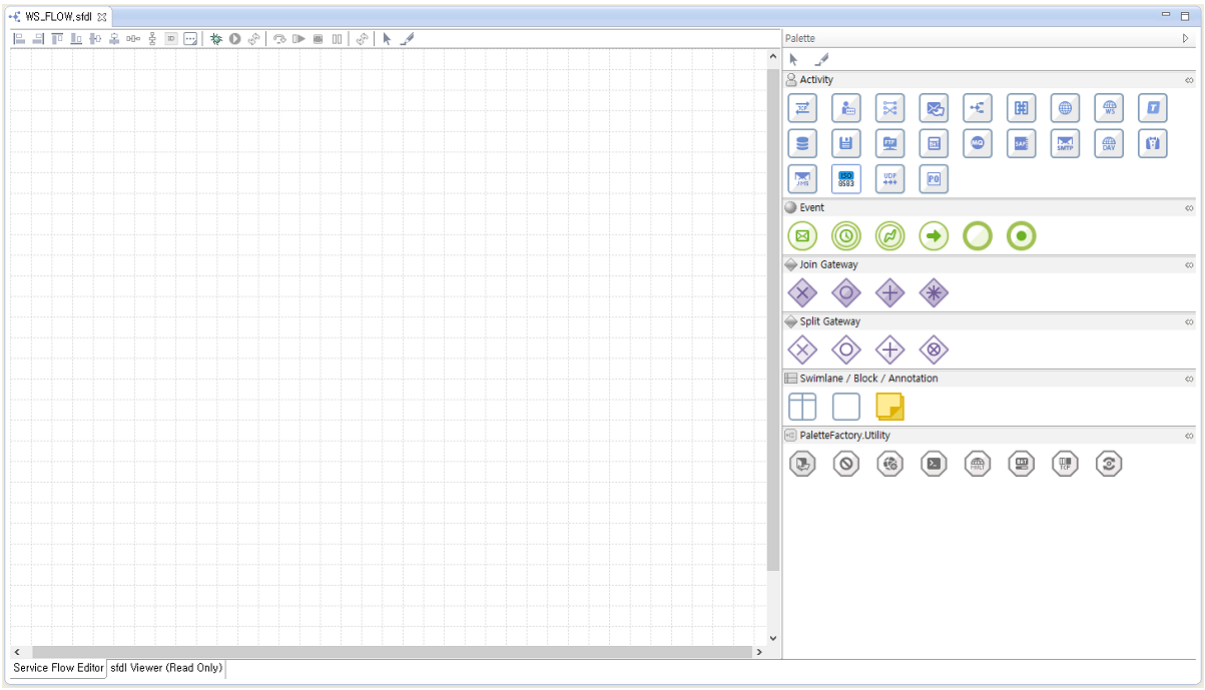


Item	Description
Flow ID	Service Flow resource ID. Only alphanumeric and special (_) characters are allowed, and the first character must be an alphabet character.
Flow Name	Service Flow name. Only Korean, alphanumeric, and special (_) characters are allowed. Must follow the XML Naming Convention.

6.3. Service Flow Diagram

When a Service Flow is created, the following page is displayed with the editor on the left and the editor palette on the right.

[Figure 6.2] Service Flow Page



- **[Toolbar]**

  The Toolbar consists of buttons for object alignment, display settings, and debugger functions.

[Figure 6.3] Service Flow Page - Toolbar

Button	Description
	Align objects to the left.
	Align objects to the right.
	Align objects to the top.

Button	Description
	Align objects to the bottom.
	Align objects to the center (horizontally).
	Align objects to the center (vertically).
	Distribute objects evenly (horizontally).
	Distribute objects evenly (vertically).
	Edit ID/Name label.
	Toggle (show/hide) transition condition.
	Run in debug mode. Opens the 'Select Server' window.
	Open the 'Message Settings" dialog box for debugging.
 (13th icon)	Refresh debugger status.
	Go to the next activity.
	Go to the next break point.
	Go to the end of the flow.
	Pause debugger at the next activity if the debugger is running.
 (18th icon)	Show debugger status.
	Change mouse pointer to 'Selection' mode.
	Change mouse pointer to 'Transition' mode.

- **[Palette]**

  The palette consists of objects that can be used to create a Service Flow diagram.

  A Service Flow, which is also called a process, consists of activities, events, and transitions. Click and move an object to the editor.

[Figure 6.4] Service Flow - Palette

Object Type	Description
Selection, Transition	– (Select): Return to select mode. – (Transition): Arrow that represents the direction and order of flow between activities or events. Use this to connect the objects. A transition connects activities, events, and gateways to define a business process. A transition can also define a condition to determine the direction of the flow after a gateway.
Activity	Also called a task. Represents actions that must be performed in the current step of the Service Flow. This is mostly used to represent an Outbound Service. For more information, refer to "6.4. Activity".
Event	Represents an event, such as a message, error, or timeout event. An event can be a Start Event without an incoming transition, an Intermediate Event with both incoming and outgoing transitions, or an End Event without an outgoing transition. For more information, refer to "6.5. Event".
Gateway	Controls the convergence and divergence of a service flow. A gateway can be a split gateway that splits a single transition into multiple transitions or a join gateway that joins multiple transitions into a single transition. For more information, refer to "6.6. Gateway".
Swimlane/Block/Annotation	– Swimlane: Dividing line between tasks in the Service Flow – Block: Used to group related Activities. – Annotation: Memo object. For detailed information, refer to "6.7. Swimlane/Block/Annotation".

Object Type	Description
Utility	Set of utilities used in the Service Flow. For detailed information, refer to "6.8. Utility".

6.4. Activity

The following describes the Activity types.

Menu	Description
(TCP Activity)	TCP outbound rule service call.
(User Class Activity)	User class activity.
(Mapping Activity)	Source and target variable mapping.
(Response Message Activity)	Service response message. Sends a normal or error response message to the request message of the message event.
(Subflow Activity)	Service Flow process call.
(Multi-Binding Activity)	Multi-binding service call.
(HTTP Activity)	HTTP outbound rule service call.
(Web Service Activity)	Import a web service.
(TMAX Activity)	Tmax (middleware) call.
(DB Activity)	DB adapter service call.
(File Activity)	File adapter service call.
(FTP Activity)	FTP adapter service call.
(Batch Activity)	Batch outbound rule call.
(MQ Activity)	MQ outbound rule call.

Menu	Description
(SAP Activity)	SAP outbound rule call.
(SMTP Activity)	SMTP outbound rule call.
(WebDav Activity)	WebDav outbound rule call.
(Tuxedo Activity)	Tuxedo outbound rule call.
(JMS Activity)	JMS outbound rule call.
(ISO8583 Activity)	ISO8583 outbound rule call.
(UDP Activity)	UDP outbound rule call.
(ProObject Activity)	ProObject outbound rule call.

An Activity has input and output parameters.

An Activity reads from the input parameter and writes to the output parameter. Both parameters are declared in the Block Activity that includes the Process or Activity.

[Figure 6.5] Input/Output of an Activity

6.4.1. Configuring an Activity

Add an Activity to the editor, and then double-click on it or place the cursor on the Activity to show the **Activity Preferences** menu to configure the Activity.

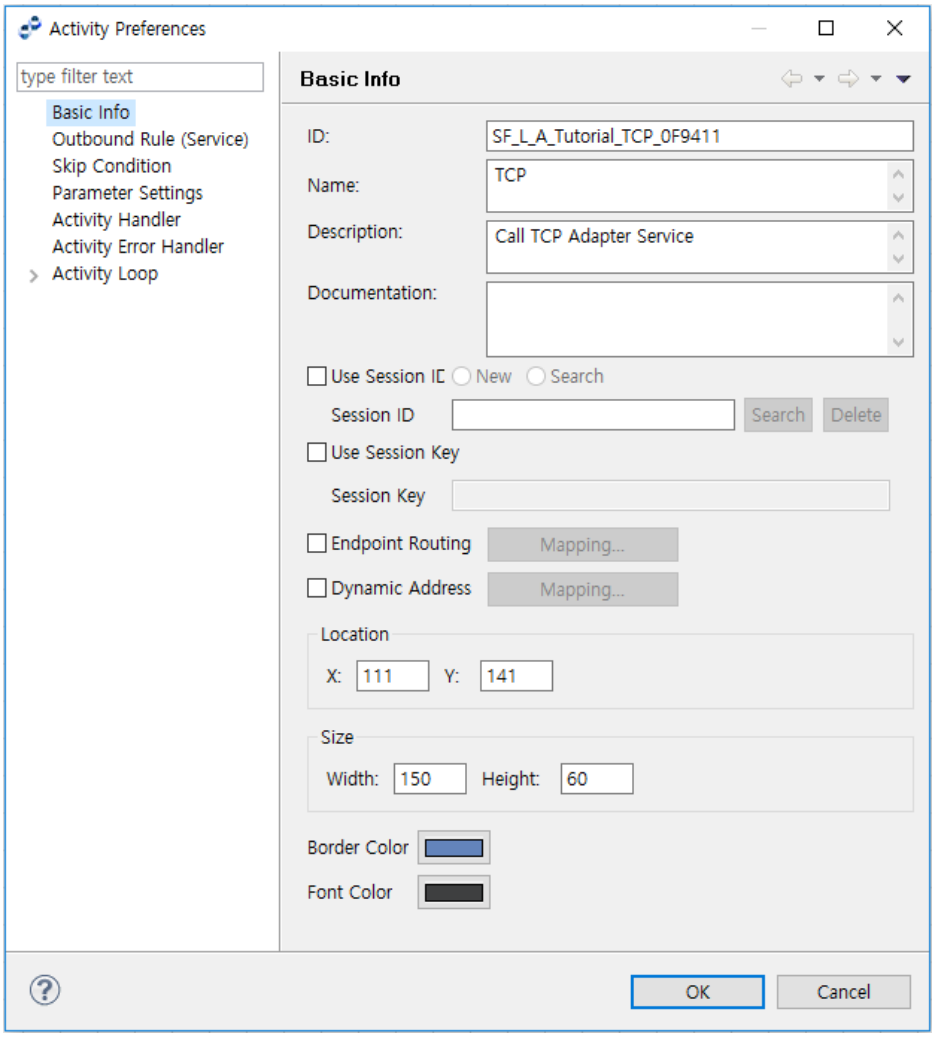
[Figure 6.6] Activity Preferences Menu

Menu	Description	Menu	Description
	Basic Info		Loop Properties
	Parameter Properties		Copy
	Handler Properties		Delete

Basic Info

The following describes the **Activity Preferences** window.

[Figure 6.7] Activity Preferences - Basic Info



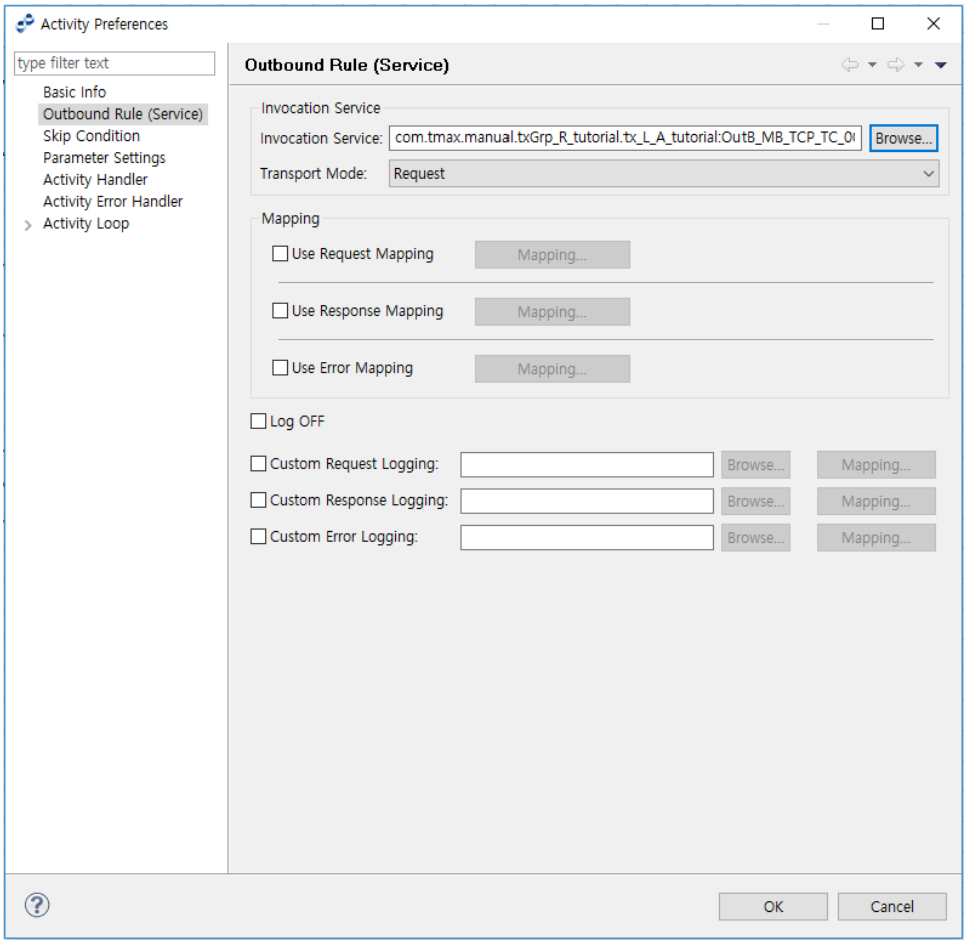
Item	Description
ID	Activity ID.
Name	Activity name.
Description	Activity role.
Documentation	Activity description.
Session ID	Session ID of the Activity. This is used to connect with other Activities or Events with the same Session ID.
Session Key	Session key. This is used to connect to a session with the key. Only available for TCP activities.
Endpoint Routing	Routes to an endpoint or endpoint group with mapped VALUE. The endpoint group set in the outbound rule must be mapped to an endpoint group in the page displayed by selecting **[System] > [Adapter]** in WebAdmin.

Item	Description
	When setting VALUE of sub endpoint or endpoint group, a request message is sent to the endpoint.
Dynamic Address Change	Dynamically changes IP address and port number of a target to request to. Only available for TCP clients.
Dynamic URL Change	Dynamically changes IP address and port number of a target to request to. Only available for HTTP outbound.
Location	Activity position (x, y). Position can be set with this option, or more easily by using the mouse.
Width	Width (can also be adjusted with the mouse).
Border Color	Border color.
Font Color	Font color.

Outbound Rule (Service)

The following describes the **Outbound Rule (Service)** page of the **Activity Preferences** window.

[Figure 6.8] Activity Preferences - Outbound Rule



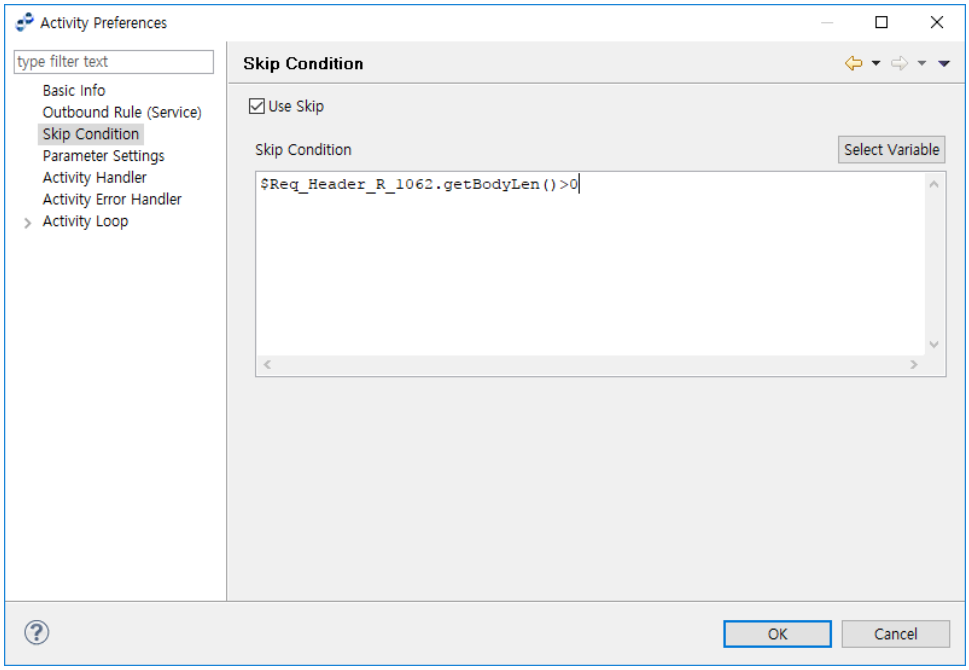
Item	Description
Invocation Service	Select an existing Outbound Rule Service that can be used for the Activity. Click **[Browse...]** to search for the service.
Transport Mode	● Request: most general messaging type that sends a request and waits for a response. ● One Way: sends a request, but does not wait for a response. ● One Way with ack: sends a request and waits for an ACK message. A delivery channel normally sends an ACK or NAK message to notify of the delivery result.
Mapping	– Use Request Mapping: define mapping between the Service Flow's INPUT parameter and request message. – Use Response Mapping: define mapping between response message and OUTPUT parameter.

Item	Description
	– Use Error Mapping: define mapping between parameter variable and error message. For information about mapping, refer to "A.2. Mapping Dialog Boxes".
Log OFF	Execute log off on the Activity.
Custom Request Logging	Option to use custom request logging. Click **[Browse...]** to select a Custom Log Outbound Rule, and then click **[Mapping...]** to create a mapping.
Custom Response Logging	Option to use custom response logging. Click **[Browse...]** to select a Custom Log Outbound Rule, and then click **[Mapping...]** to create a mapping.
Custom Error Logging	Option to use custom error logging. Click **[Browse...]** to select a Custom Log Outbound Rule, and then click **[Mapping...]** to create a mapping.

Skip Condition

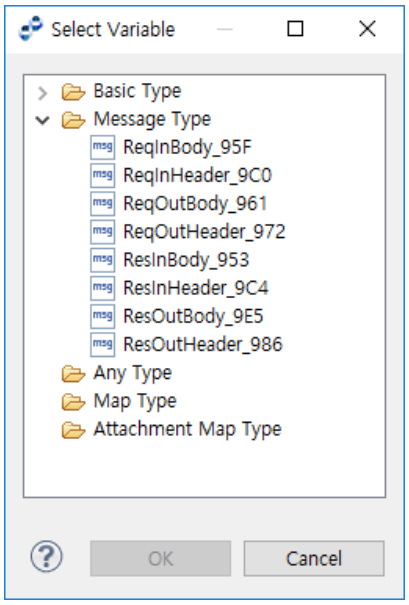
The following describes the **Skip Condition** page of the **Activity Preferences** window.

[Figure 6.9] Activity Preferences - Skip Condition



Item	Description
Use Skip	Option to use skip condition for the process variable.
Skip Condition	Condition for the selected variable. Click **[Select Variable]** to open the **Select Variable** dialog box ([Figure 6.10]) for selecting a process variable.

[Figure 6.10] Activity Preferences - Select Variable

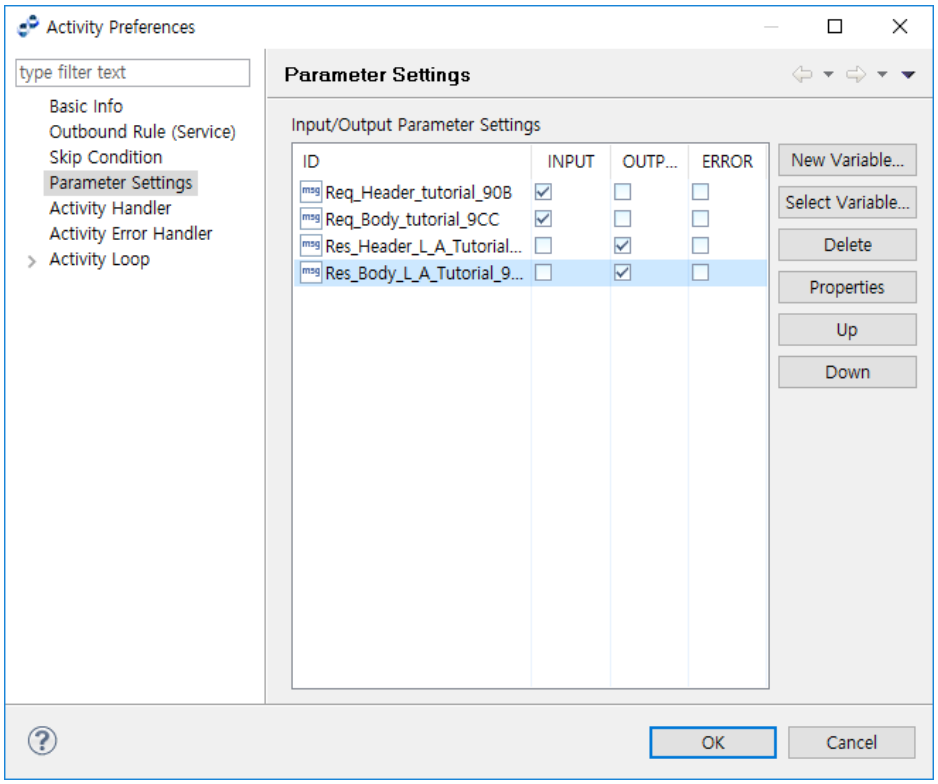


Parameter Settings

The variable set in the parameter is used in the service call. If a mapping exists, the input parameter is used as the source of the input mapping, and the output parameter as the target of the output mapping.

The following describes the **Parameter Settings** page of the **Activity Preferences** window.

[Figure 6.11] Activity Preferences - Parameter Settings



- Input/Output Parameters

Item	Description
ID	Variable ID.
INPUT	Option to set the variable type to input.
OUTPUT	Option to set the variable type to output.
ERROR	Option to set the variable type to error.

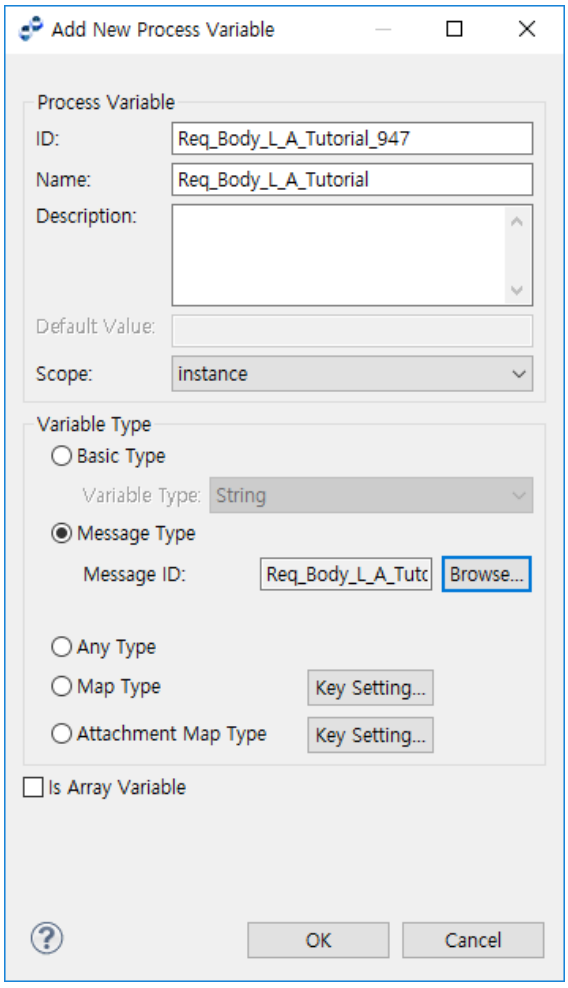
- Buttons

Button	Description
[New Variable]	Opens the **Add New Process Variable** dialog box to create a new variable ([Figure 6.12]).

Button	Description
[Select Variable]	Opens the **Select Variable** dialog box to select an existing variable ([Figure 6.10]).
[Delete]	Deletes the selected variable.
[Properties]	Shows the selected variable information.
[Up]	Moves the selected variable one step up.
[Down]	Moves the selected variable one step down.

Click **[New Variable]** to open the **Add New Process Variable** window.

[Figure 6.12] Activity Preferences - Add New Process Variable



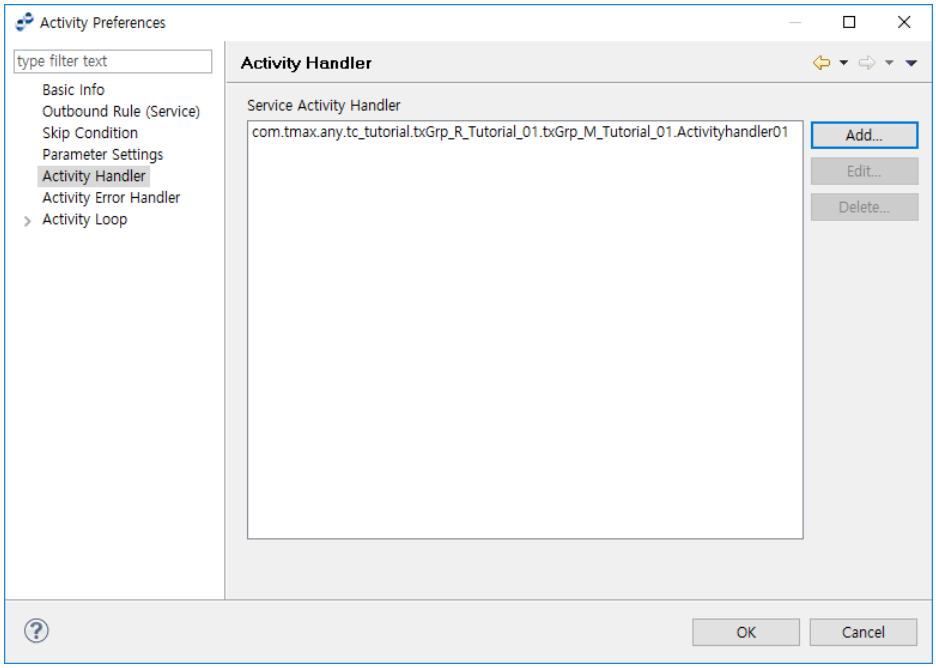
Item	Description
ID	Variable ID.
Name	Variable Name.

Item	Description
Description	Variable description.
Default Value	If the Variable Type is **'Basic Type'**, enter a default value.
Scope	Preset to 'instance' (cannot be edited).
Variable Type	– Basic Type: one of String, Float, Integer, DateTime, or Boolean type. – Message Type: a message is selected from DTO files that exist under the project's service group through the 'Select Message' window. – Any Type: select this type when using a Java Class object as the Process Variable. – Map Type: specifies how to access key value, msg, and ctx information. Click **[Key Setting...]** to open the **'Map Data Field Key Setting'** window. A header variable for a service call must be a map type specified in the format, <String, String>. – Attachment Map Type: set an attachment file. This is normally used for a file adapter attachment.
Is Array Variable	Option to use an array variable.

Activity Handler

The following describes the **Activity Handler** page of the **Activity Preferences** window.

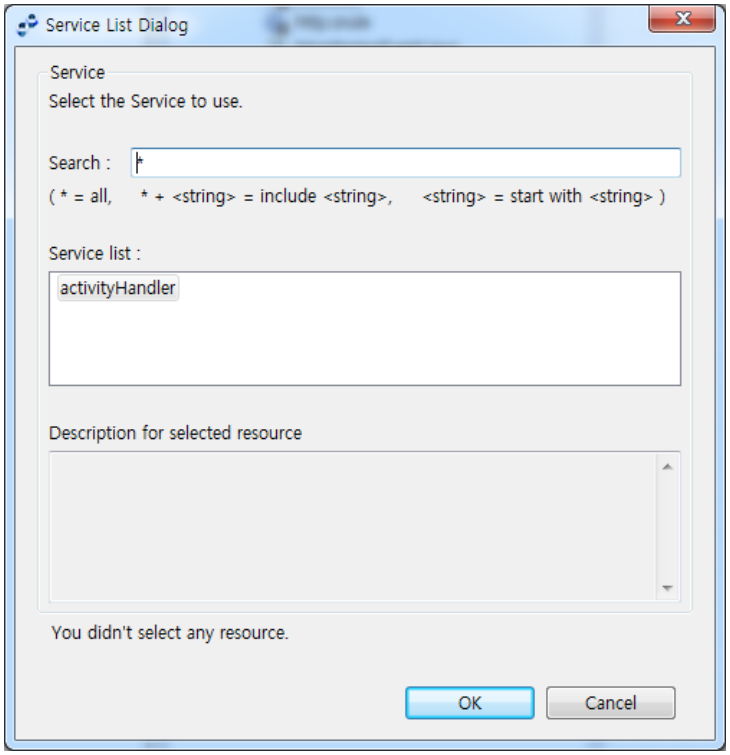
[Figure 6.13] Activity Preferences - Activity Handler



Button	Description
[Add...]	Opens the **Service List Dialog** ([Figure 6.14]) to select a user-defined service as an Activity Handler.
[Edit...]	Opens a window to edit the selected handler.
[Delete...]	Deletes the selected handler.

The following describes **Service List Dialog** where user-created activity handlers can be set.

[Figure 6.14] Service List Dialog

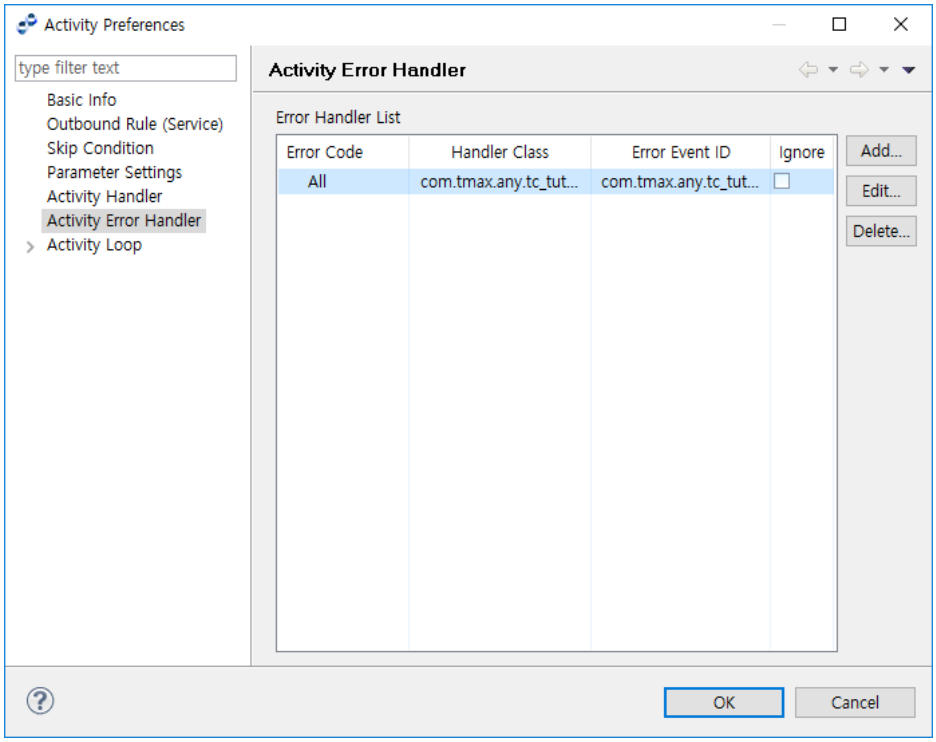


Item	Description
Search	Searches for activity handlers.
Service list	Displays handlers.

Activity Error Handler

The following describes the **Activity Error Handler** page of the **Activity Preferences** window. Buttons are same as those for the **Activity Handler** ([Figure 6.13]) page.

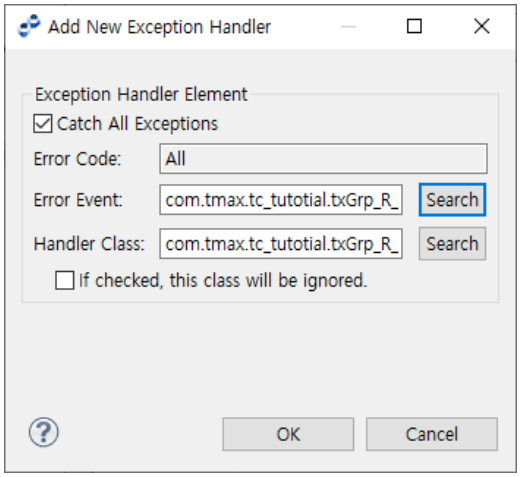
[Figure 6.15] Activity Preferences - Activity Error Handler



Item	Description
Error Code	Error code for the handler to process.
Handler Class	User class for the handler.
Error Event ID	Error event ID.
Ignore	Option to ignore the error event in the Activity.

Clicking **[Add...]** opens the following dialog box.

[Figure 6.16] Add New Exception Handler

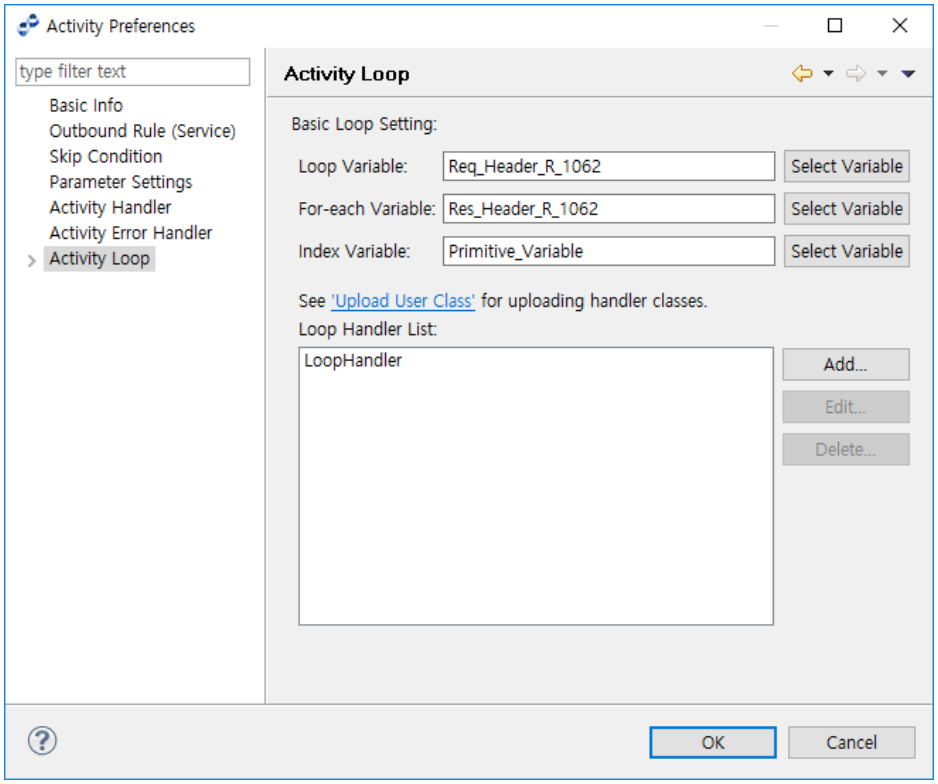


Item	Description
Catch All Exceptions	Catches all exceptions.
Error Code	Error code to catch. If Catch All Exceptions is selected, the value is set to All automatically.
Error Event	Error event. You can search for an error event from **Service List Dialog** ([Figure 6.14]) by clicking **[Search]**.
Handler Class	Handler class. You can search for a handler class from **Service List Dialog** ([Figure 6.14]) by clicking **[Search]**.

Activity Loop

The following describes the **Activity Loop** page of the **Activity Preferences** window.

[Figure 6.17] Activity Preferences - Activity Loop



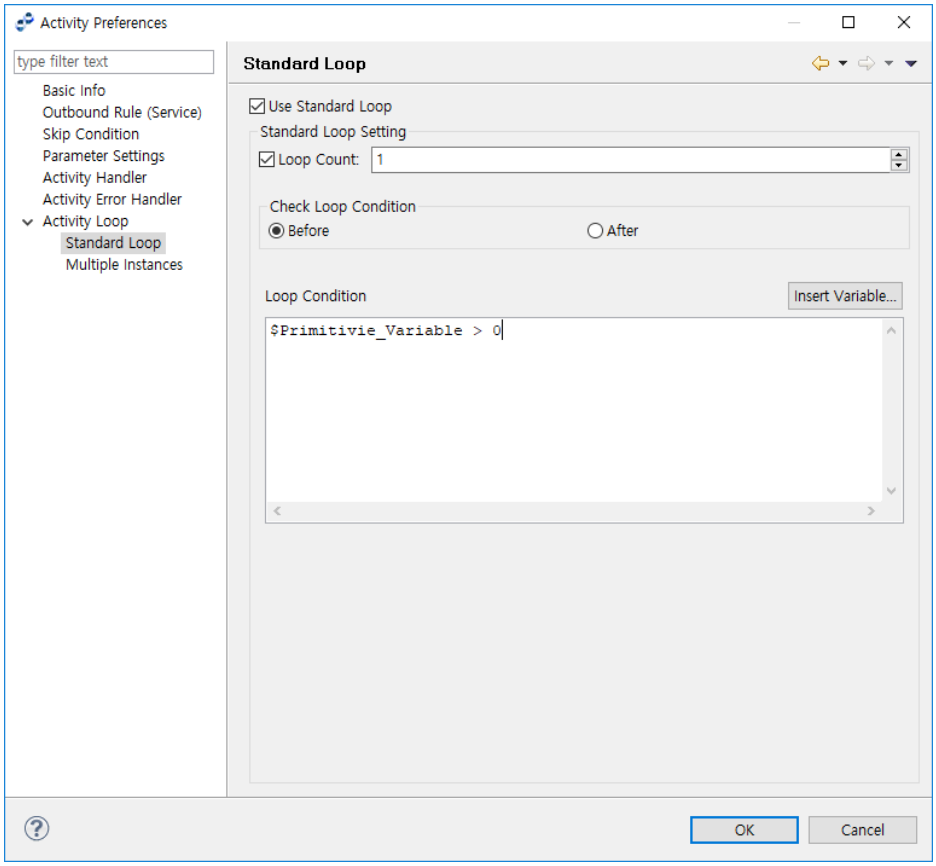
Item	Description
Loop Variable	Variable that holds an array item.

Item	Description
For-each Variable	Variable that shares the type of the Loop Variable. Value is set in the order of the array items assigned to the Loop Variable.
Index Variable	An integer variable whose value is incremented after each pass.
Loop Handler List	List of loop handler classes.

- **Standard Loop**

  Activity is repeated sequentially a specified number of times.

[Figure 6.18] Activity Preferences - Standard Loop - Standard Loop



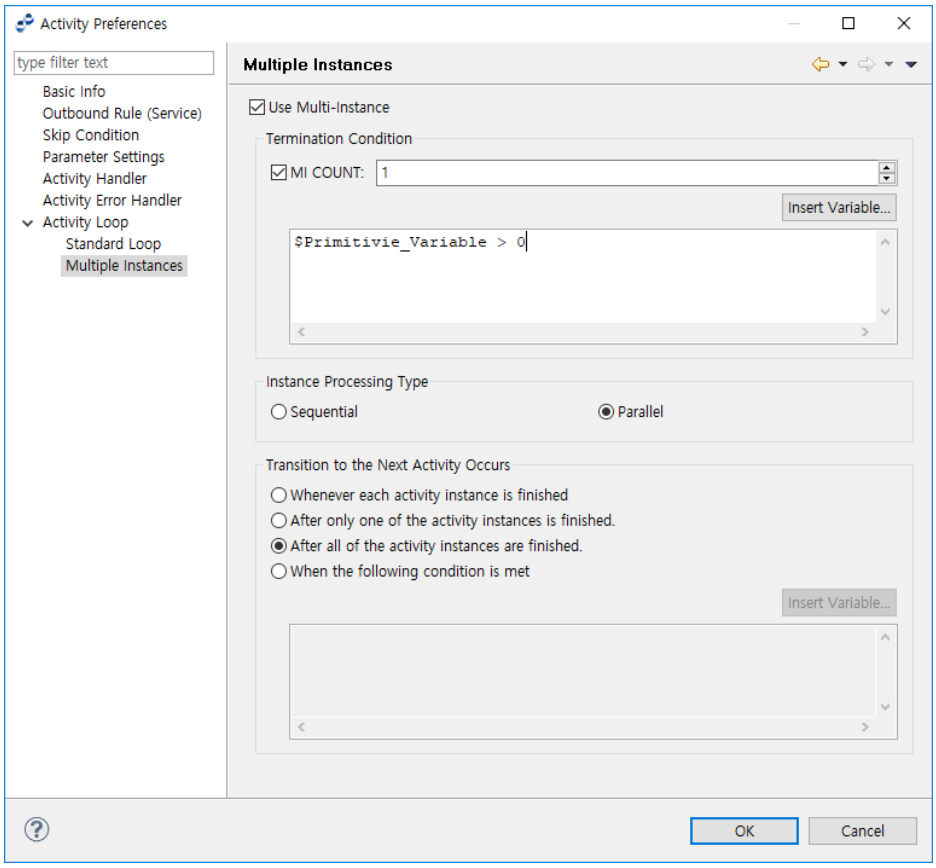
Item	Description
Loop Count	Maximum number of iterations of the loop. The loop may terminate before reaching this count if the **'Loop Condition'** is not met. If the **'Loop Count'** is set without a **'Loop Condition'**, the **'Loop Condition'** is set to true by default.
When to Check	When to check the **'Loop Condition'**. – Before: before executing the Activity (while loop).

Item	Description
	– After: after executing the Activity (until loop). The loop is executed at least once even if the **'Loop Condition'** is not met.
Loop Condition	Loop condition. The loop may terminate before reaching this count if the **'Loop Condition'** is not met. If the **'Loop Count'** is not set, the Activity repeats until the **'Loop Condition'** fails. If the **'Loop Count'** is not set, the **'Loop Condition'** must be set to an expression that does not always evaluate to true. Use <Ctrl>+Space as a shortcut to insert a variable.

- **Multiple Instances**

Activity is executed sequentially or in parallel, and the loop count or loop condition may not be fixed.

[Figure 6.19] Activity Preferences - Activity Loop - Multiple Instances



– **Termination Condition**

Termination condition for the Activity. Use **'MI Count'** to set the maximum number of iterations and the ‘Loop Condition’ to set the loop termination condition.

- **Instance Processing Type**

Item	Description
Sequential	Start instances sequentially.
Parallel	Start all instances in parallel.

- **Transition to the Next Activity Occurs**

When to pass the Process Instance's token to the next Activity after the current Activity is finished.

Item	Description
Whenever each activity instance is finished	Token is passed to the next Activity after each Activity Instance is finished. As many tokens as the number of finished instances are passed.
After only one of the activity instances is finished	Token is passed after one instance is finished. Any preceding instance's token is ignored.
After all of the activity instances are finished.	Token is passed after all Activity Instances are finished.
When the following condition is met	Token is passed when the specified condition is met.

Note

If using Multiple Instances and **'Instance Processing Type'** is set to 'Parallel', For-each Variable and Index Variable must be local variables since they may cause concurrency issues. If the **'Instance Processing Type'** is set to 'Sequential' and **'Transition to the Next Activity Occurs'** is set to 'After all of the activity instances are finished', the loop works like a Standard Loop.

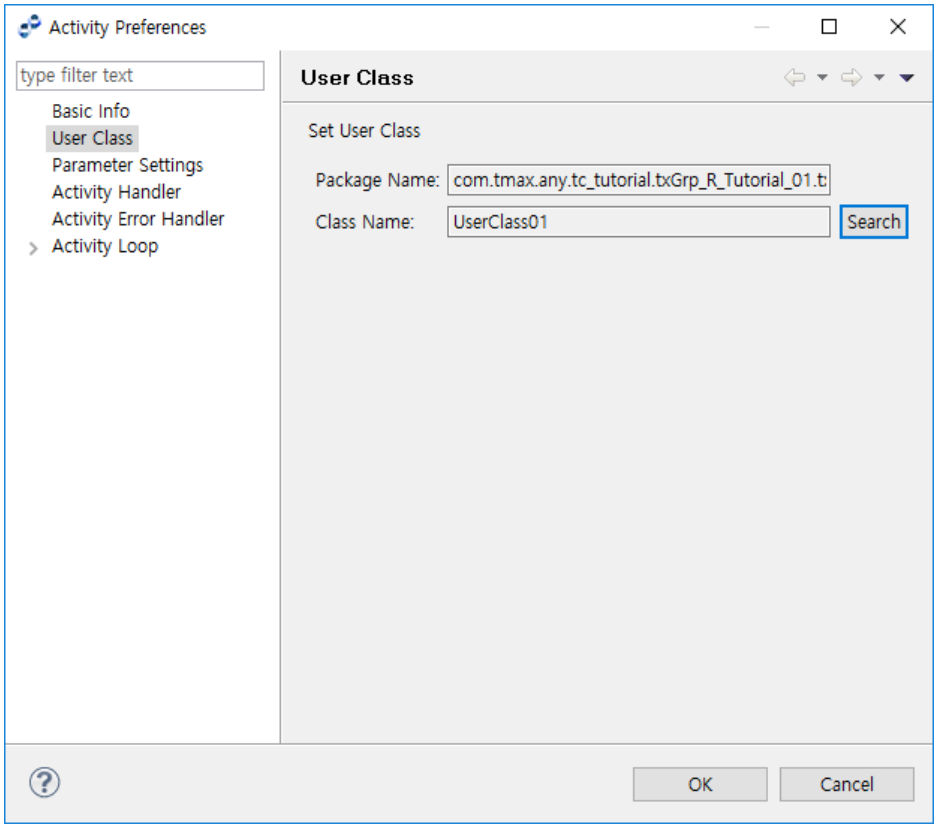
6.4.2. Configuring a Specific Activity Type

The following describes how to configure a specific Activity type.

User Class Activity

A User Class Activity can be set by selecting an existing user class, which handles complex logic, from the **Activity Preferences** window.

[Figure 6.20] Activity Preferences - User Class Activity

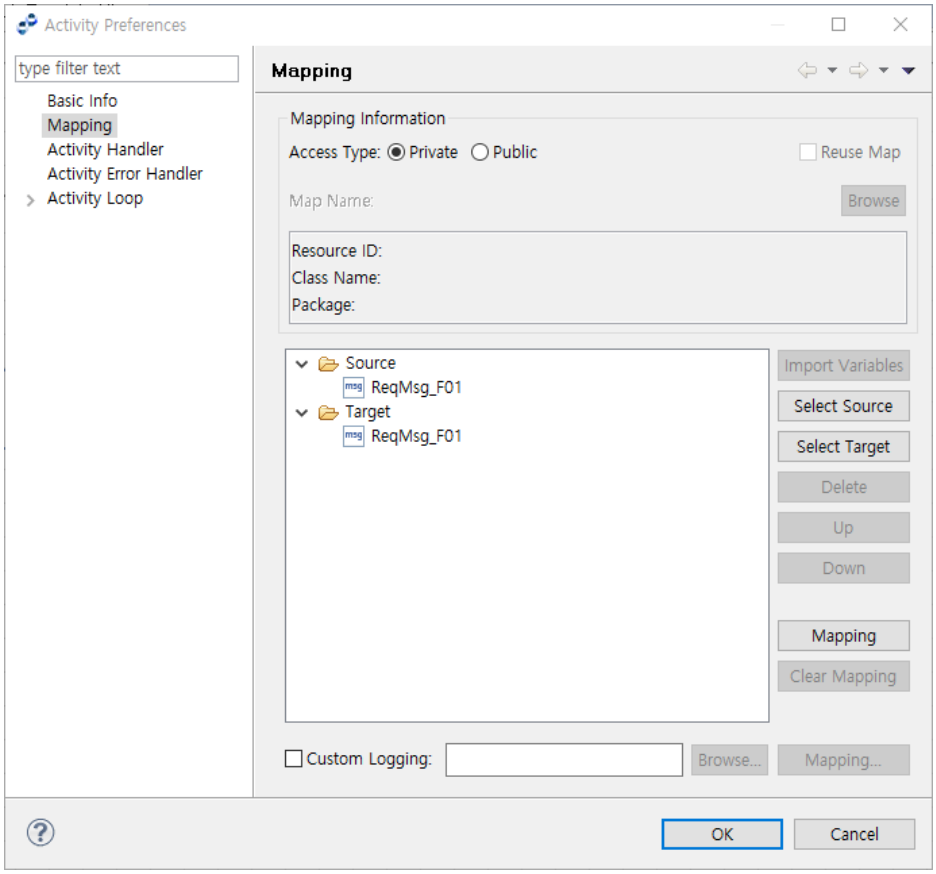


Item	Description
Package Name	User class package name.
Class Name	User class name.

Mapping Activity

Converts the variable values by using a pre-defined mapping. Mapping can be set from the **Activity Preferences** window.

[Figure 6.21] Activity Preferences - Activity Mapping



- **Mapping Information**

Item	Description
Access Type	If Private is selected, mapping information is saved in the Activity instead of a separate map file to prevent access.

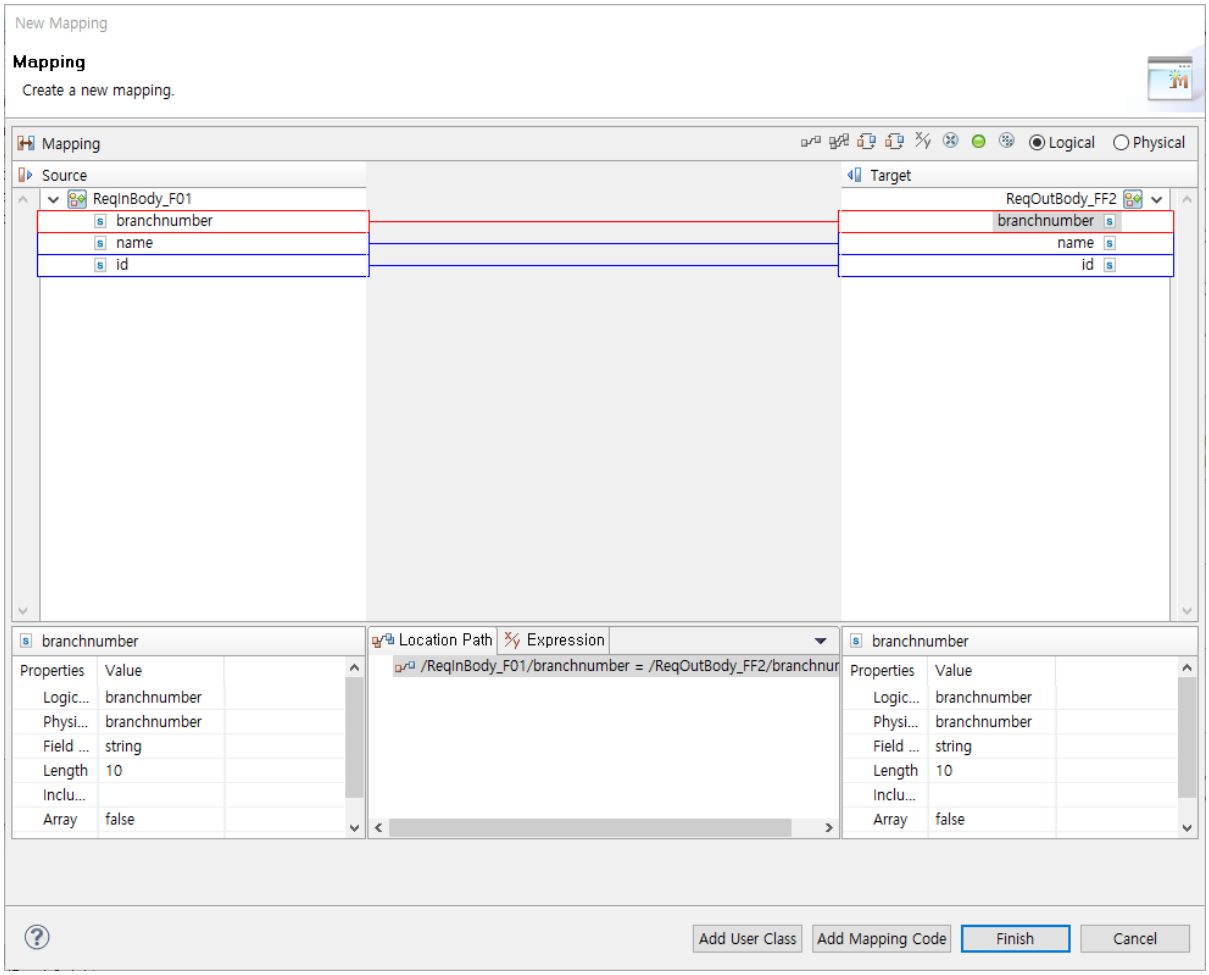
- **Buttons**

Button	Description
[Import Variables]	Imports the message set in Public Mapping.
[Select Source]	Source process variable.
[Select Target]	Target process variable.
[Delete]	Deletes the selected variable.
[Up]	Moves the selected variable one step up.
[Down]	Moves the selected variable down step up.
[Mapping]	Opens the **Mapping** dialog box ([Figure 6.22]) to map source and target process variables.

Button	Description
[Clear Mapping]	Clears the mapping information.

The following describes **Mapping** dialog box where source and target process variables can be mapped. To add a user mapping handler, click **[Add User Class]**. To enter Java code to be called during mapping, click **[Add Mapping Code]**.

[Figure 6.22] Mapping Dialog Box



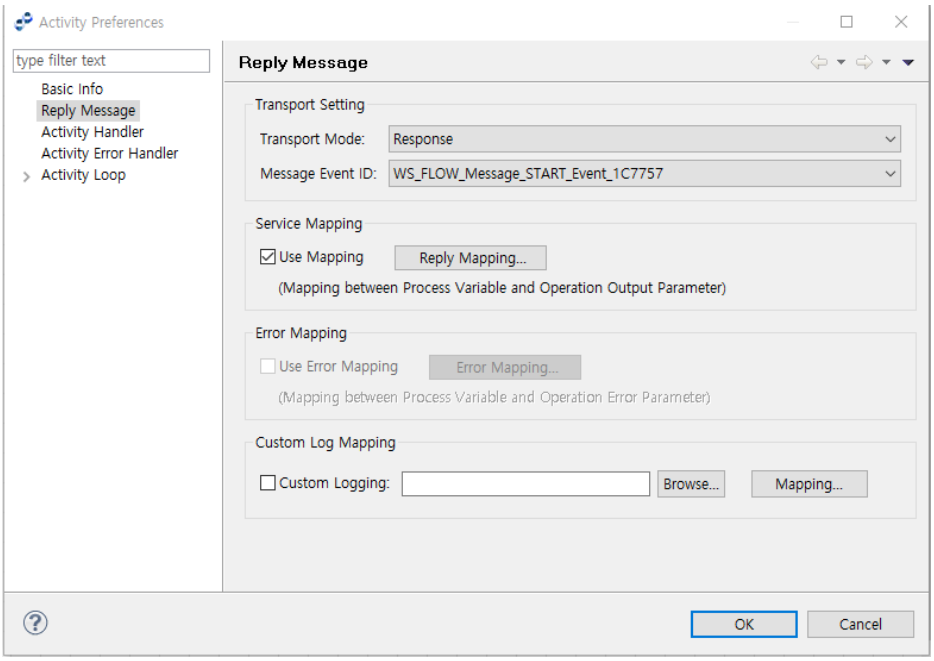
Item	Description
Source	Variables to map.
Target	Variables for which a value is defined through mapping.
Properties	Searches variables or fields selected in Source or Target. The left Properties section is for Source, and the right Properties section is for Target.
Location Path	Connection information for the mapping.
Expression	Maps values.

Item	Description
	Strings are enclosed in double quotation marks (" ").

Reply Message Activity

Sends a response to the message event of the inbound service. A Reply Message Activity can be set from the **Activity Preferences** window.

[Figure 6.23] Activity Preferences - Reply Message



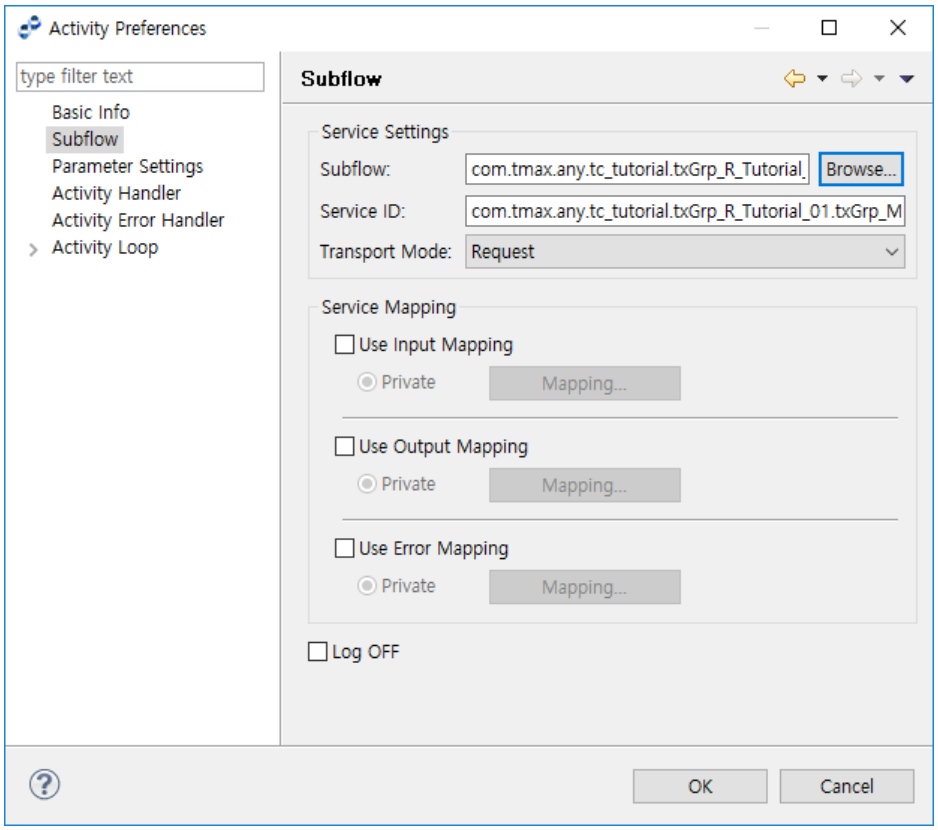
Item	Description
Trans Mode	– Response: Use the Reply Message Activity as a response. – Error: Use the Reply Message Activity as an error. – Response None: Do not send a response message.
MessageEvent ID	Message event of the Service Flow.
Use Mapping	Option to create mapping between process variable and output parameter from the **Reply Mapping** window. The **'Transport Mode'** must be set to 'Mapping'. Click **[Reply Mapping...]** to open the Mapping window.
Use Error Mapping	Option to create mapping between process variable and error parameter from the **Error Mapping** window. The **'Transport Mode'** must be set to 'Error'.

Item	Description
	Click **[Error Mapping...]** to open the Mapping window.
Custom Log Mapping	Option to create mapping between the Activity's Custom Log and Activity variable. – **[Browse...]**: Search Custom Log rules. Clicking this button opens **Service List Dialog** ([Figure 6.14]) where you can search for the rules. – **[Mapping...]**: Map Custom Log rules' request messages.

Subflow Activity

A Response Message Activity can call another flow by configuring a subflow from the **Activity Preferences** window.

[Figure 6.24] Activity Preference - Subflow



Item	Description
Subflow	Service Flow.
Service ID	Service ID of the selected Subflow.

Item	Description
Transport Mode	● Request: most general messaging type that sends a request and waits for a response. ● One Way: sends a request, but does not wait for a response. ● One Way with ack: sends a request and waits for an ACK message. A delivery channel normally sends an ACK or NAK message to notify of the delivery result.
Service Mapping	Create an in/out mapping between Outbound Service's variable and Activity parameter. Select 'private' to save the mapping in the Service Mapping section of the SFDL file so that other flows or Activities cannot access it. If not selected, a map file is created and made public so that it can be reused.

Multi-Binding Activity

Calls multiple services that are configured in a single Activity.

6.5. Event

The following describes the Event types.

Event	Description
 (Message Event)	Message event for receiving a message. Can be either a Start or Intermediate event.
 (Timer Event)	Timer event that periodically starts a process by specifying the start and end times and the interval. Must be an Intermediate event.
 (Error Event)	Error event. Can be a Start, Intermediate, or End event.
 (Link Event)	Event for synchronization or process flow control. Can be a Start, Intermediate, or End event.
 (Normal End Event)	An End event that has terminated successfully.
 (Abnormal End Event)	An End event that has been forcibly terminated. This event is displayed as an abnormal termination in WebAdmin.

6.5.1. Configuring an Event

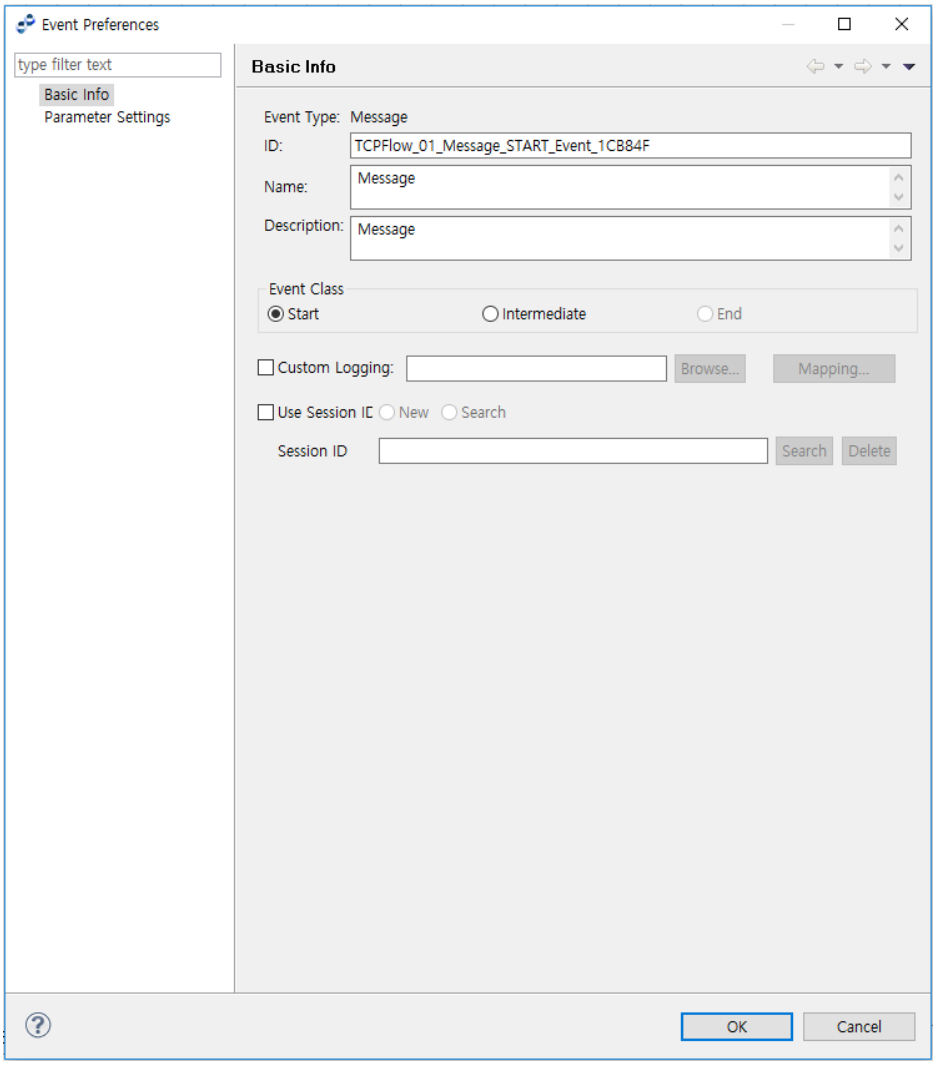
Add an Event to the editor, and then double-click on it or place the cursor on the Event to show the **Event Preferences** menu to configure the Event.

[Figure 6.25] Event Preferences Menu

The Copy and Delete buttons are same as those in **Activity Preferences** menu.

The following is the **Basic Info** page of the **Event Preferences** window. The description of the items that also appear in the **Activity Preferences** window has been omitted here.

[Figure 6.26] Event Preferences - Basic Info



Item	Description
Event Class	Select Start, Intermediate, or End type depending on the event's role and position in the Service Flow. Some events may have additional constraints.
Custom Logging	Option to use custom logging. – **[Browse...]**: Select a Custom Log Outbound Rule. – **[Mapping...]**: Create a mapping.
Session ID	Click **[New]** to enter a Session ID, or click **[Search]** to select an existing Session ID.

6.5.2. Event Configuration by Type

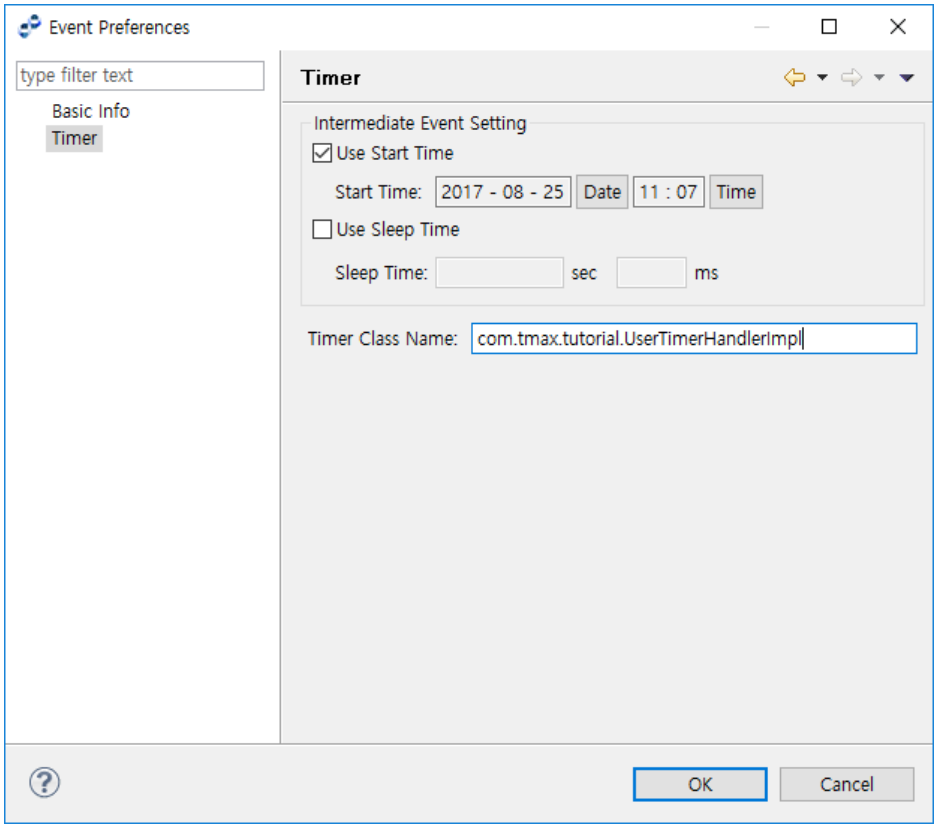
- **Message Event**

  Message Event has the **Parameter Settings** section. For more information, refer to the description about the **Activity Preferences** window ([Figure 6.11]).

- **Timer Event**

  Timer Event has the **Timer** section with the start and end times and the interval for starting a process periodically.

[Figure 6.27] Event Preferences - Timer

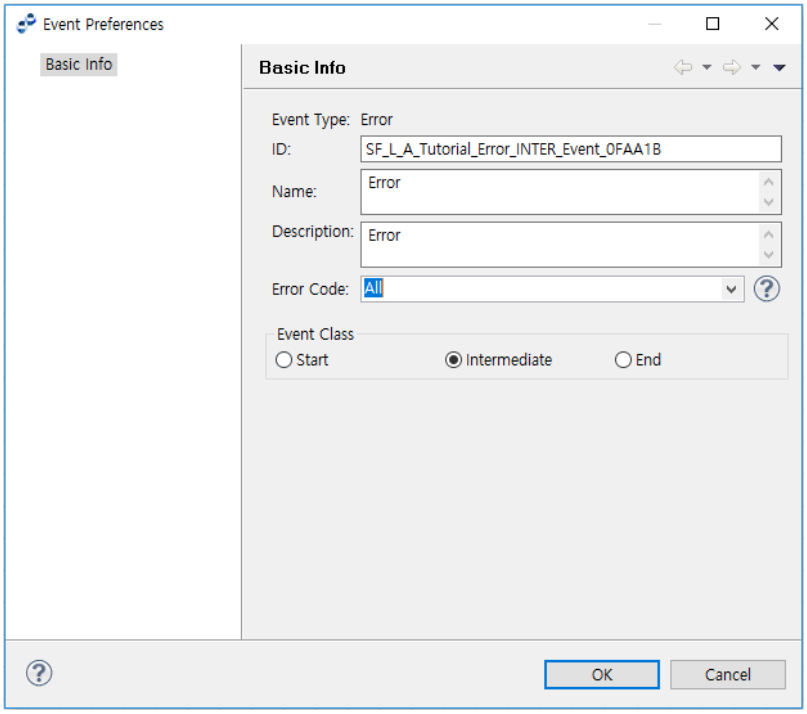


Item	Description
Start Event Setting	– If only 'Start Time' is set: executes once at the set time after server startup. – If only 'End Time' is set: executes once immediately after server startup (same as when no settings are entered). – If only 'Period' is set: executes at the set interval after server startup. – If 'Start Time', 'End Time', and 'Period' are set: executes at the set interval starting from the 'Start Time' until the 'End Time'.
Intermediate Event Setting	– If only 'Start Time' is set: sleep until the set time, or sleep indefinitely if the set time has already passed. – If only 'Sleep Time' is set: sleep during the set period. – If 'Start Time' and 'Sleep Time' are set: sleep until the 'Start Time', and sleep for start time + n*sleep time if the 'Start Time' has already passed.
Timer Class Name	User class name.

- **Error Event**

  Error Event is set with **Error Code** information. An error code is required. Error codes defined in AnyLink can be checked by clicking the **?** button at the right of the Error Code field.

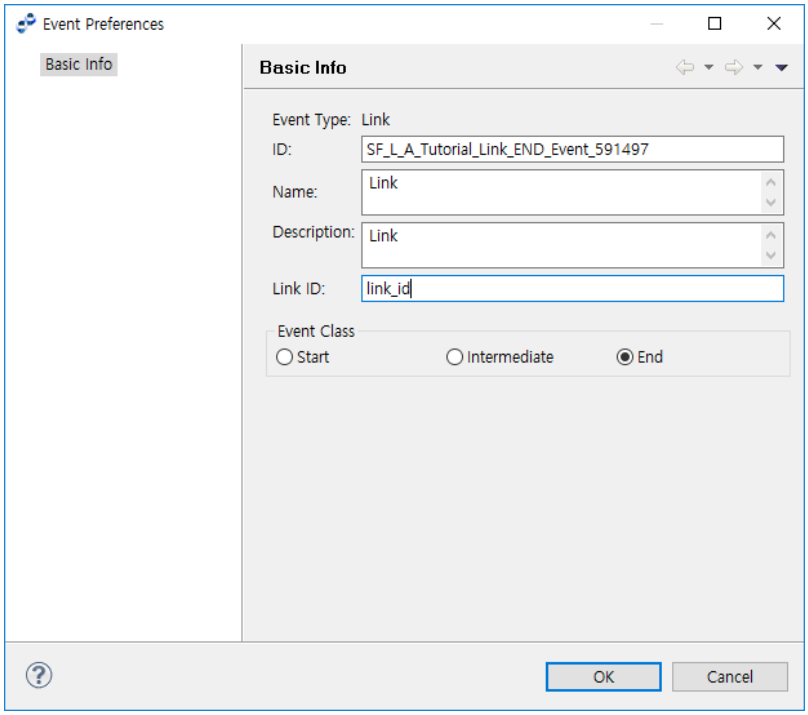
  [Figure 6.28] Error Preferences - Error Event



- **Link Event**

  Link Event is set with **Link ID** information. Enter the target ID to connect to.

[Figure 6.29] Event Preferences - Link Event



6.6. Gateway

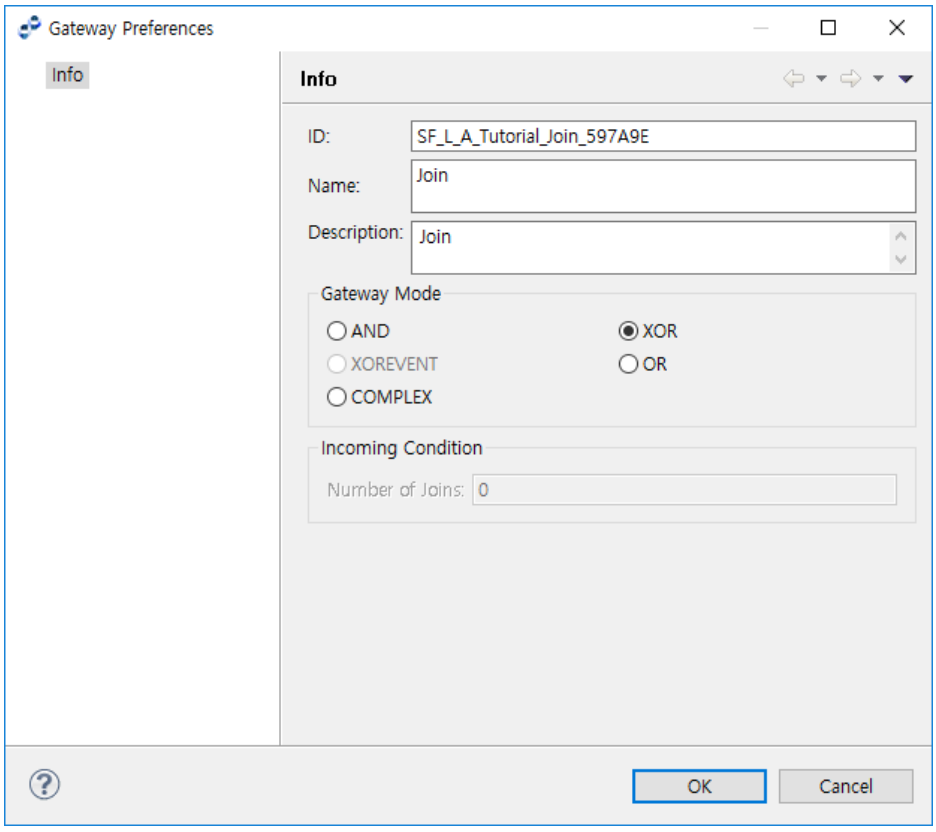
The following describes gateway types.

Item	Description
(XOR Split) (XOR Join)	XOR Split is when a process instance is routed to one of many Transitions. A process that was split using an XOR Split can be rejoined using an XOR Join. A condition must be set for each transition of the XOR Split, and conditions are evaluated in the specified order until one evaluates to 'true'. If all conditions are evaluated as 'false', an exception occurs. Hence, it is recommended to specify an 'otherwise' condition for such case.
(XOR Event Split)	XOR Event Split is similar to XOR Split, but routing is determined based on the occurrence of the next activity itself instead of on the activity transition (deferred choice pattern). XOR Event Split must be followed by a Timer Event, Message Event, or a block that starts with an Event.

Item	Description
(Or Split) (Or Join)	Or Split is when a process instance is routed to all Transitions whose condition evaluates to 'true'. Or Split must also be rejoined using Or Join. Transition's conditions are similar to those for XOR Split, except that all Transitions that evaluate to 'true' are executed even if multiple Transitions have overlapping conditions.
(And Split) (And Join)	And Split is when a process instance is routed to all Transitions unconditionally. Since And Split splits to all Transitions, no conditions are required. And Join executes the next Activity only after receiving execution results from all split Transitions.
(Complex Join)	Complex Join executes the next Activity after joining the specified number of Transitions.

The following describes the **Gateway Preferences** window.

[Figure 6.30] Gateway Preferences



Item	Description
Gateway Mode	Gateway type.
Incoming Condition	Number of transitions to join. Enabled only for a Complex Gateway.

6.7. Swimlane/Block/Annotation

The following describes the Swimlane, Block, and Annotation objects.

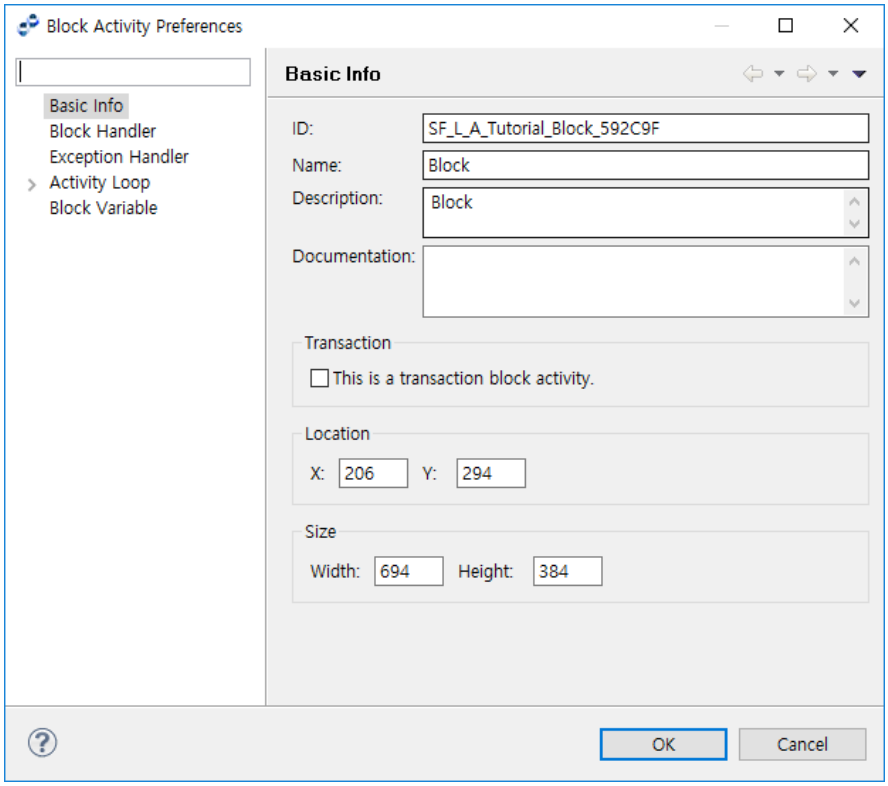
Object	Description
 (Swimlane)	Swimlane is a dividing line between tasks in the Service Flow. Swimlanes can be used to easily distinguish the sequence of tasks performed in parallel or actions. Double-click on the top label of the Swimlane to open the Swimlane Preferences window to edit the ID, Name, Size, and Border Color of the Swimlane.
 (Block Activity)	Block Activity can be set with process variables that can be used within the Block, and Message, Timer, and Error events.
 (Annotation)	Memo used to enter object description for user convenience.

6.7.1. Block Activity

A Block, similar to a Java function in concept, includes a group of Service Flow Activities and is created to handle events or exceptions that occur in the Block.

'Block Handler', **'Exception Handler'**, **'Activity Loop'**, and **'Block Variable'** can be configured for a Block as for an Activity. Click **[Block Variable]** menu to define variables that can only be used in the Block.

[Figure 6.31] Block Activity Preferences



Item	Description
This is transaction block	Option to allow the Block to support transactions. In a Transaction Block, Tx_Begin is automatically called when entering the Block, Tx_Commit is called when terminating the Block with 'None' event, and Tx_Rollback is called when terminating the Block abnormally or with 'Terminate' event.

6.8. Utility

Utility provides a set of utilities used in a Service Flow.

The following describes utility objects.

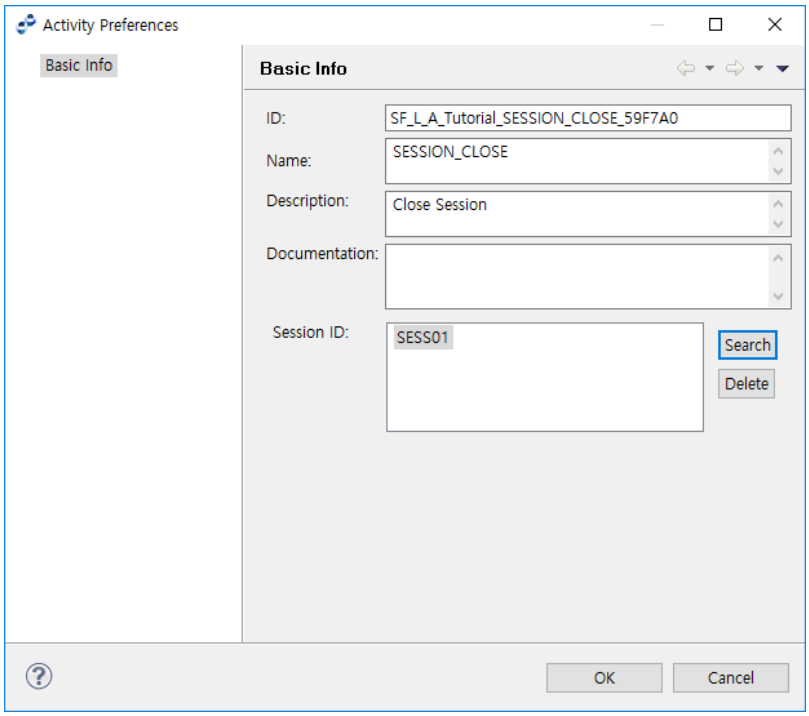
Object	Description
(SESSION CLOSE ACTIVITY)	Closes the session with the specified Session ID.
(SESSION ABORT ACTIVITY)	Forcibly ends the session with the specified Session ID.
(NETWORK MANAGEMENT ACTIVITY)	Manages network status.

Object	Description
(MANAGER APPROVAL ACTIVITY)	Manages terminal login/logout and retrieves information about terminal users and administrators.
(HTTP MULTIPART ACTIVITY)	Saves files received through HTTP multipart to a temporary path. The files are automatically deleted after the flow completes.
(BATCH PROGRESS ACTIVITY)	Manages batch progress.
(TCP MESSAGE SPLIT ACTIVITY)	The next Intermediate Event receives fragmented TCP messages.
(CONTROL SESSION ACTIVITY)	Forcibly ends a session with a specified endpoint ID or session key.

6.8.1. SESSION CLOSE ACTIVITY

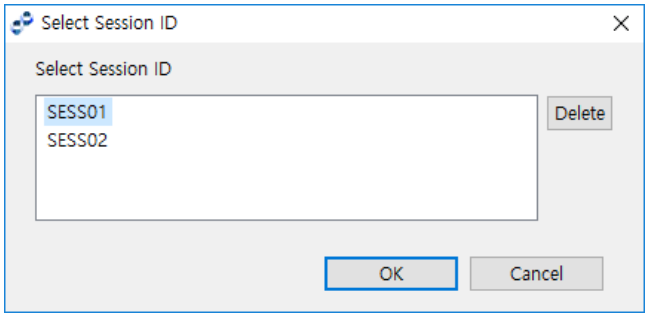
The following describes the **Session Close Preference** window for SESSION CLOSE ACTIVITY.

[Figure 6.32] Session Close Preference



Select the **'Session ID'** of the Service Flow. Click **[Search]** to open the Select Session ID dialog box.

[Figure 6.33] Select Session ID Dialog Box



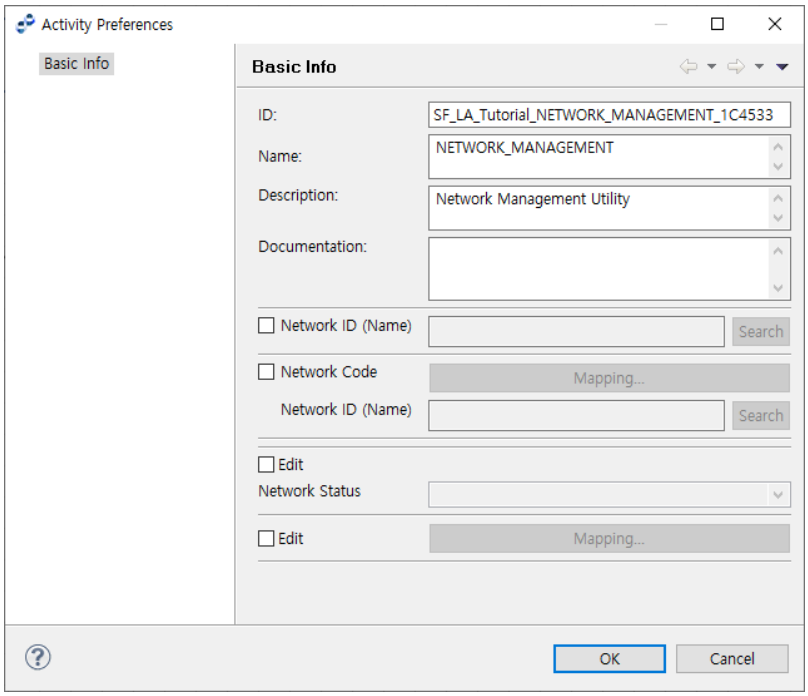
6.8.2. SESSION ABORT ACTIVITY

The usage is the same as in the SESSION CLOSE ACTIVITY. Refer to "6.8.1. SESSION CLOSE ACTIVITY".

6.8.3. NETWORK MANAGEMENT ACTIVITY

The following describes the **Network Management Preference** window for NETWORK MANAGEMENT ACTIVITY.

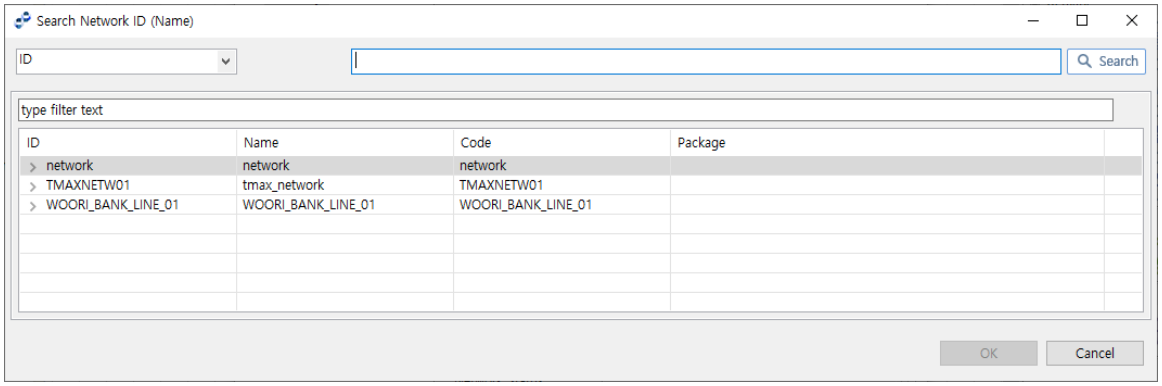
[Figure 6.34] Network Management Preference



To select a fixed network, select **'Network ID (Name)'** and click **[Search]**. This opens the **Search Network ID (Name)** dialog box where you can select a network ID added in WebAdmin.

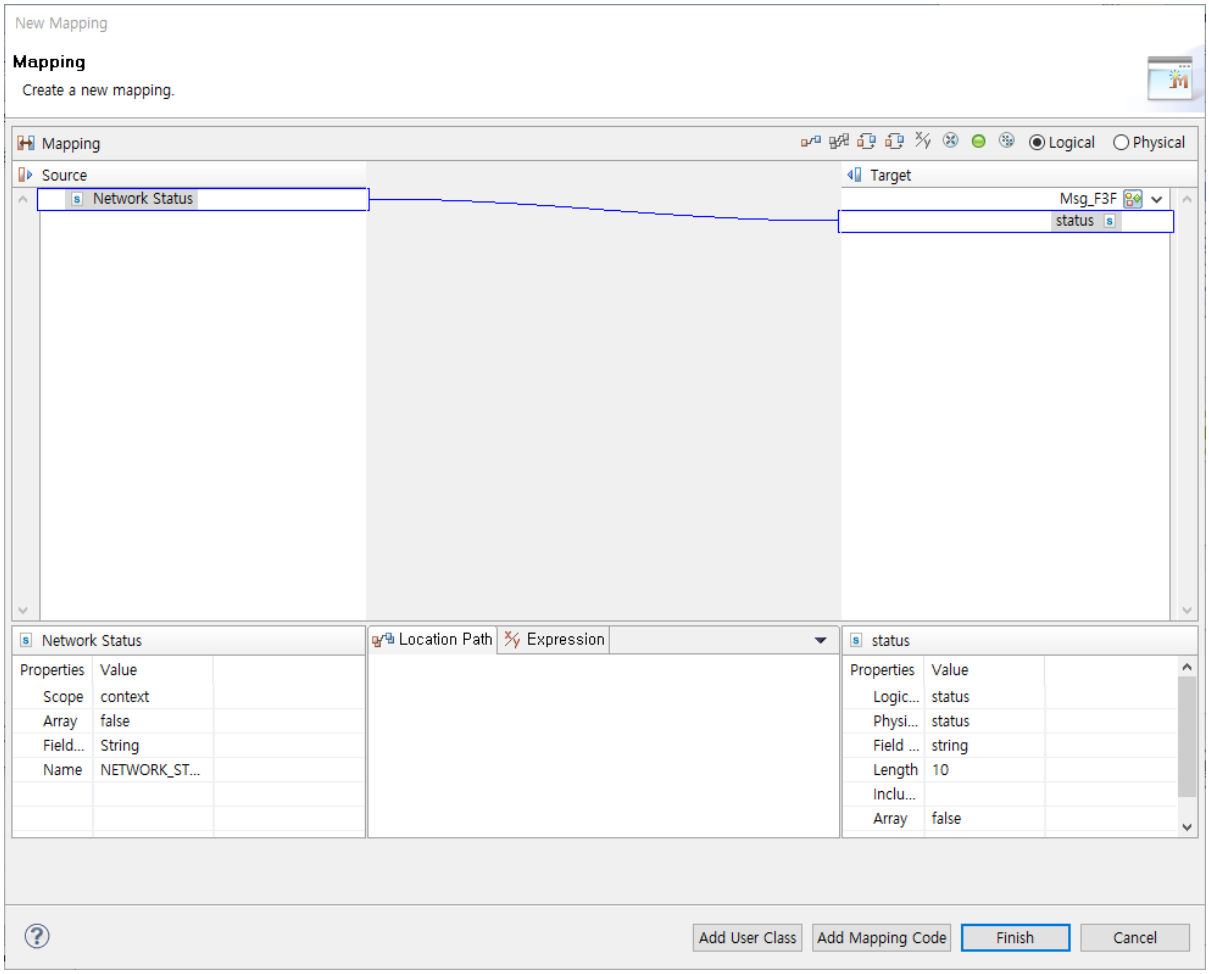
If the flow variable has a code value, select **'Network Code'** and click **[Mapping...]**. Map the code and then click **[Search]**. This opens the **Search Network ID (Name)** dialog box where you can select a network ID added in WebAdmin.

[Figure 6.35] Search Network ID (Name) Dialog Box



To change network status, select **'Edit'**. To search network status, select **'Search'** and click **[Mapping...]**. This opens network status **Mapping** where you can map network status to a flow variable.

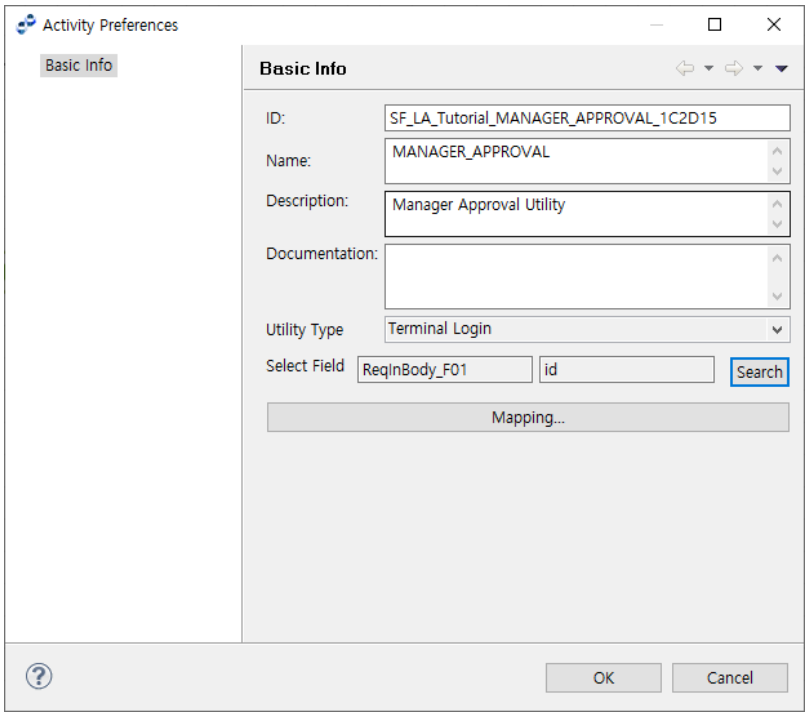
[Figure 6.36] Network Status Mapping



6.8.4. MANAGER APPROVAL ACTIVITY

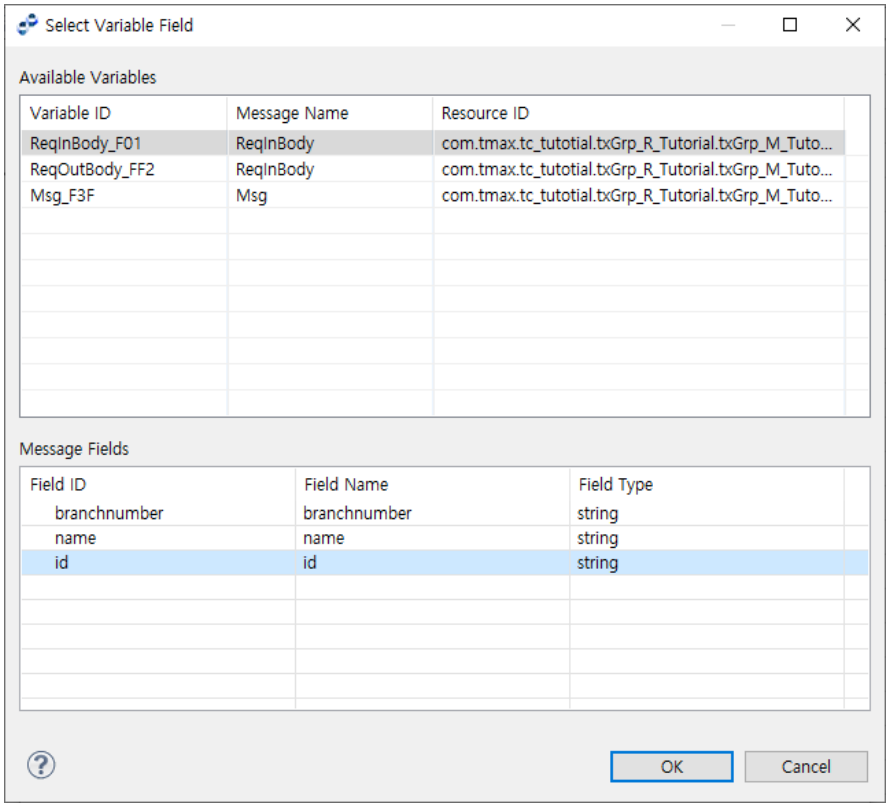
The following describes the **Manager Approval Preference** window for MANAGER APPROVAL ACTIVITY.

[Figure 6.37] Manager Approval Preference



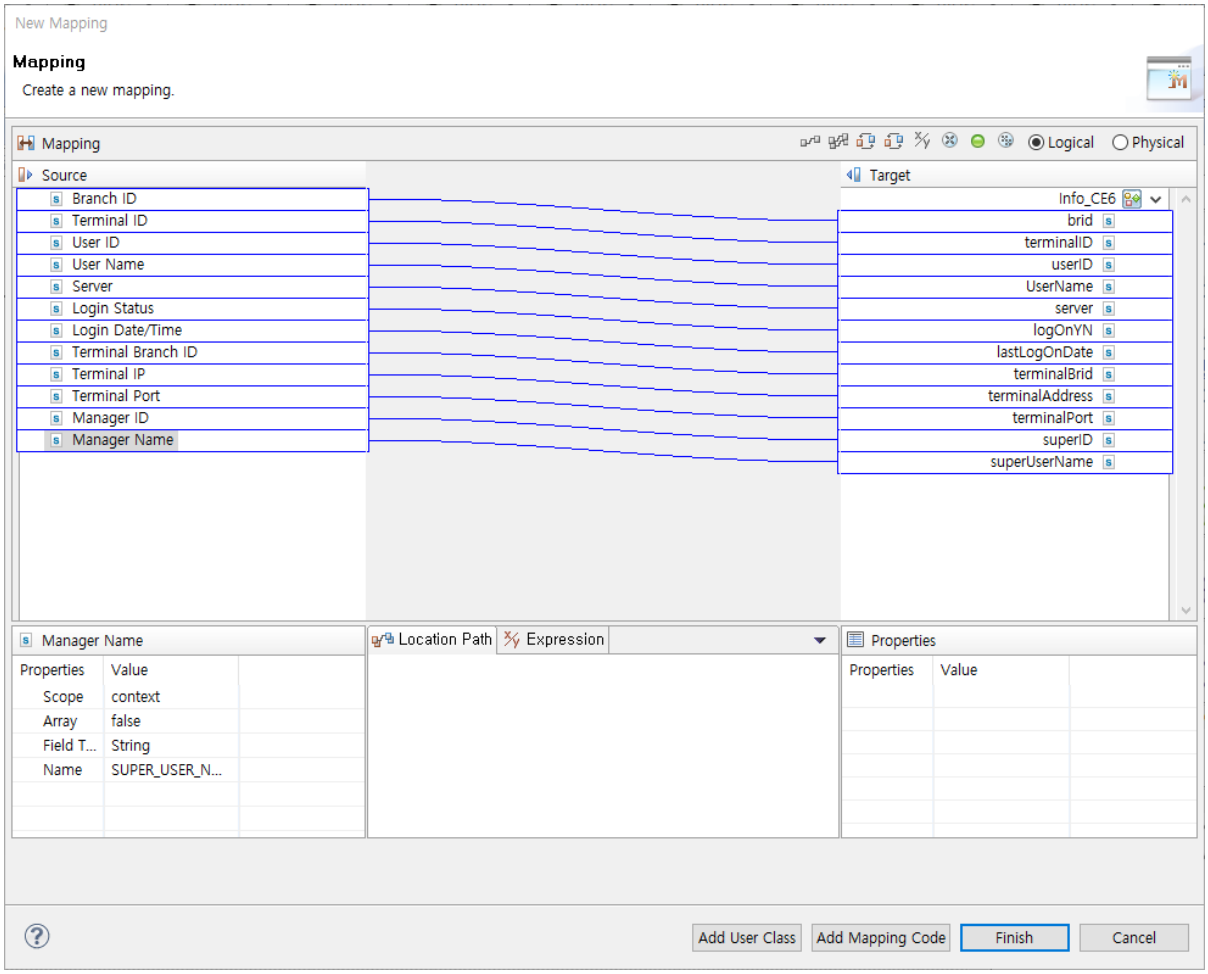
Select an action to take from Utility Type. User a terminal, or map User ID to search for from the **Select Variable Field** dialog box opened by clicking **[Search]** at the right of **Select Field**. The field value must be the same as User ID that can be searched for from WebAdmin by selecting **[Configuration] > [Terminal] > [Users]**.

[Figure 6.38] Select Variable Field



To map the results of taking an action for the utility type, click **[Mapping...]**. This opens the **Terminal/User Information Mapping** dialog box where you can map the results.

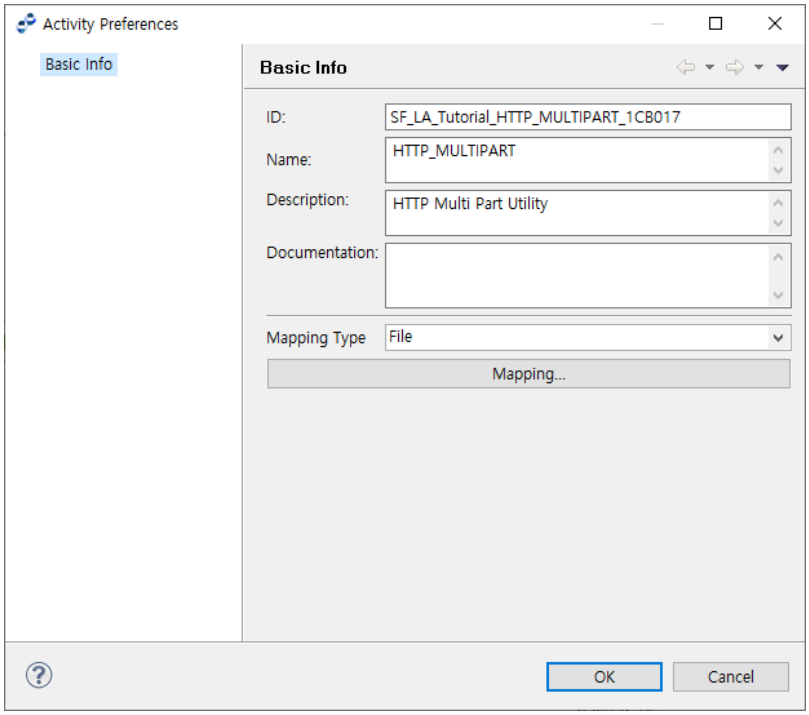
[Figure 6.39] Terminal/User Information Mapping



6.8.5. HTTP MULTIPART ACTIVITY

The following describes the **HTTP Multipart Preference** window for HTTP MULTIPART ACTIVITY.

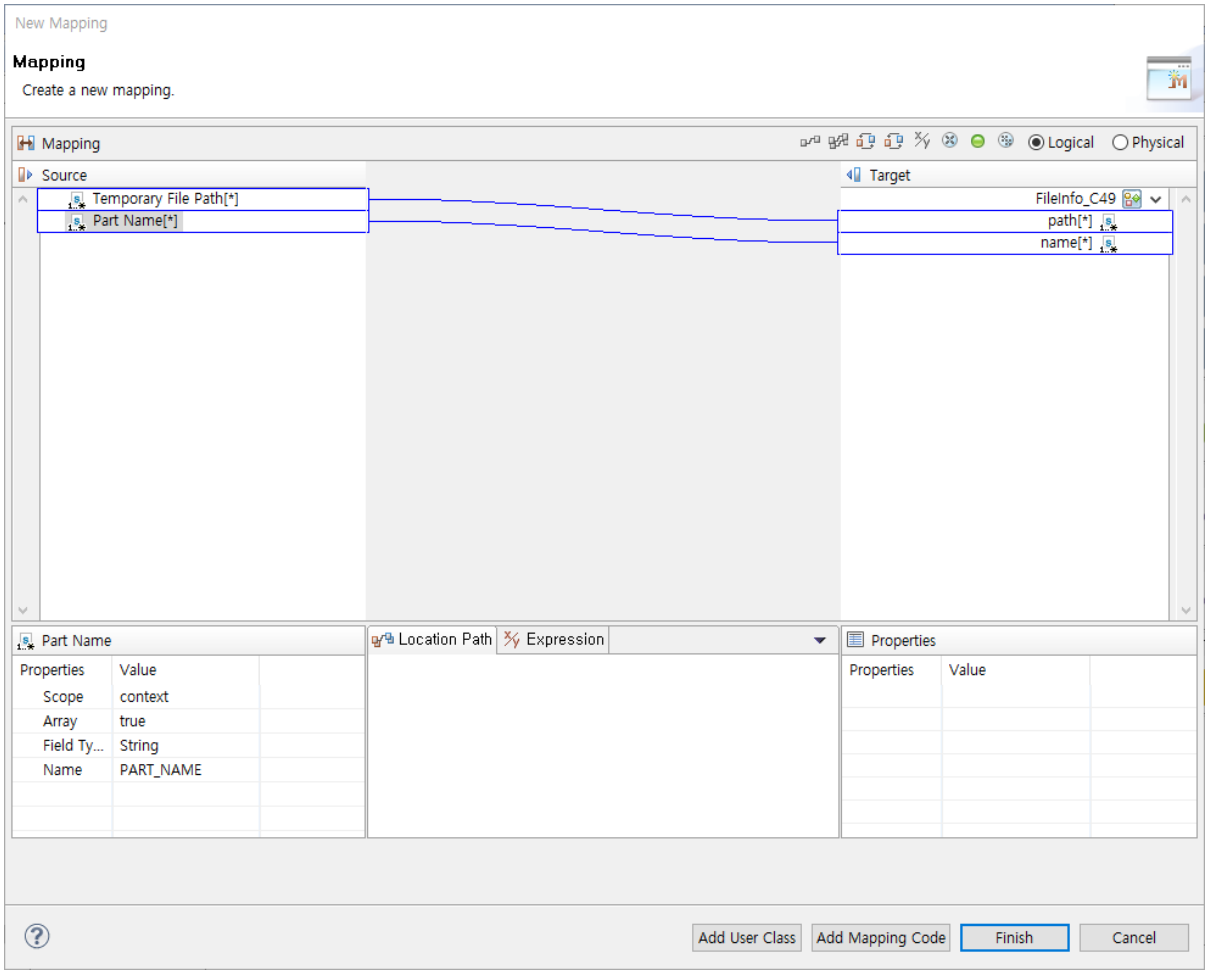
[Figure 6.40] HTTP Multipart Preference



Select a type from **Mapping Type**. The File mapping can be created through file path and name, and the Byte mapping can be created through file contents and information.

To map multipart file information, click **[Mapping...]**. This opens the **Multipart Mapping** dialog box where you can map the information.

[Figure 6.41] Multipart Mapping

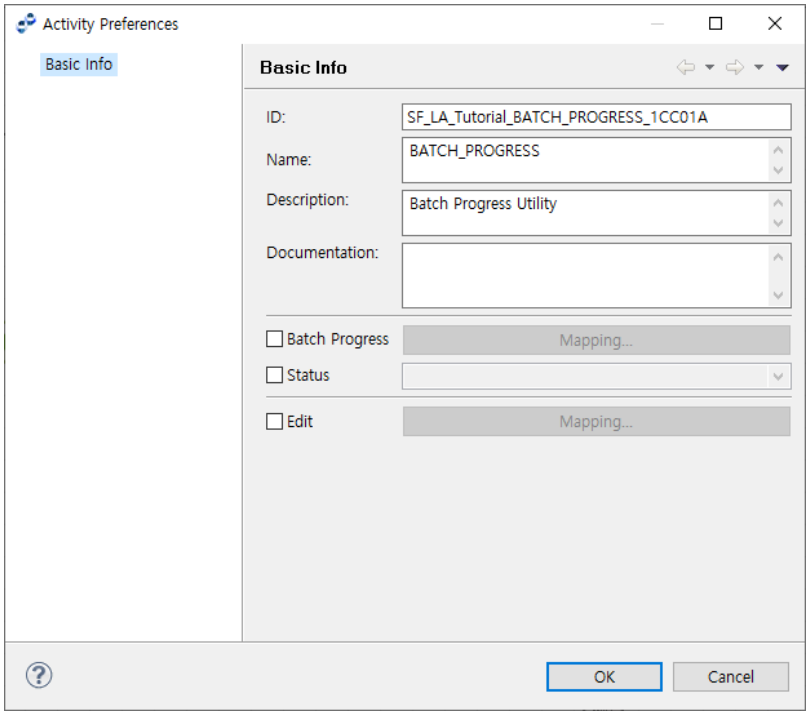


6.8.6. BATCH PROGRESS ACTIVITY

The following describes the **Batch Progress Preference** window for BATCH PROGRESS ACTIVITY.

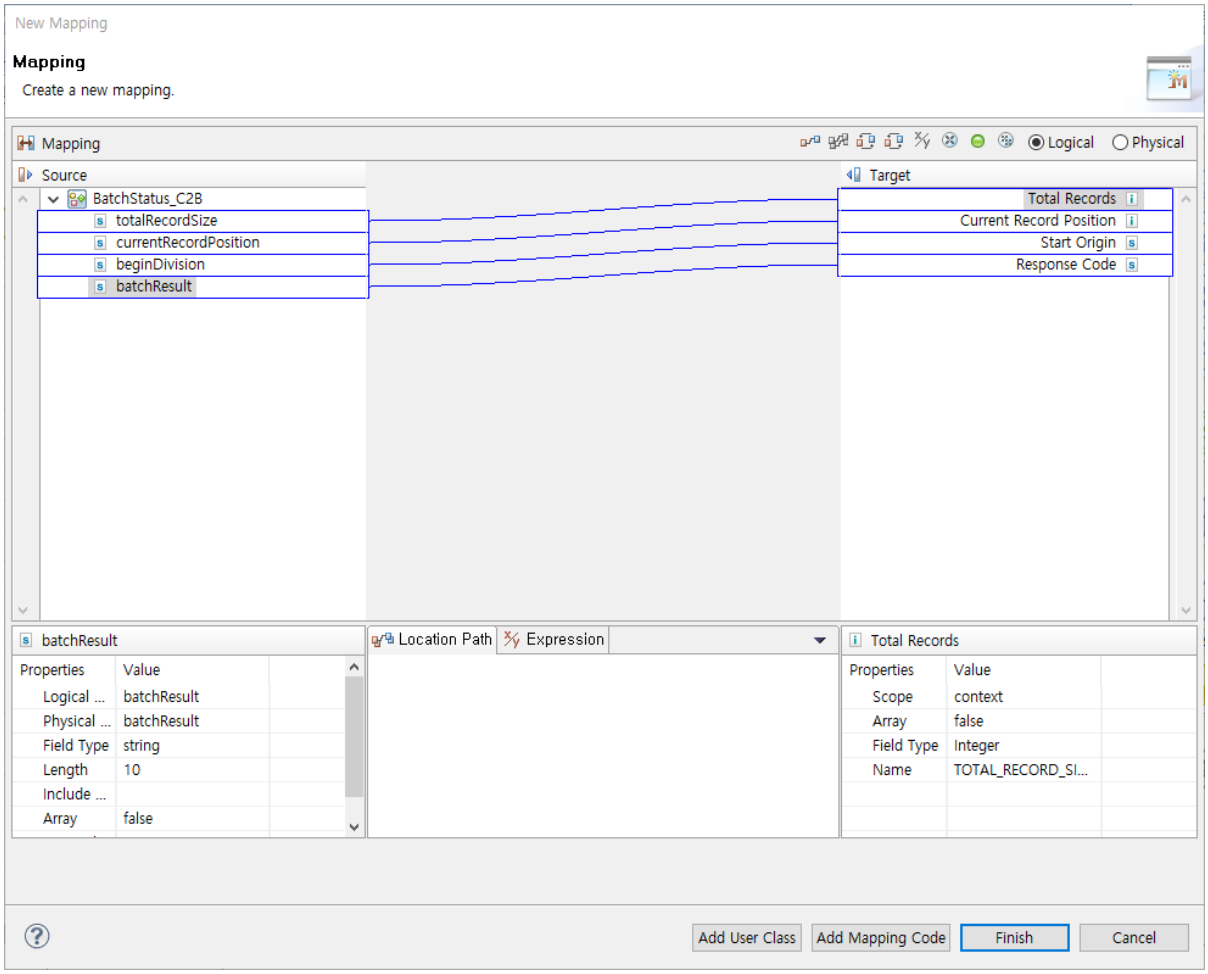
To show batch progress, set Show Batch Progress to **true** when defining the batch outbound rule used by the flow.

[Figure 6.42] Batch Progress Preference



To change batch progress information, select **'Batch Progress'** and click **[Mapping...]**. This opens the **Batch Progress Mapping** dialog box where you can map batch progress information.

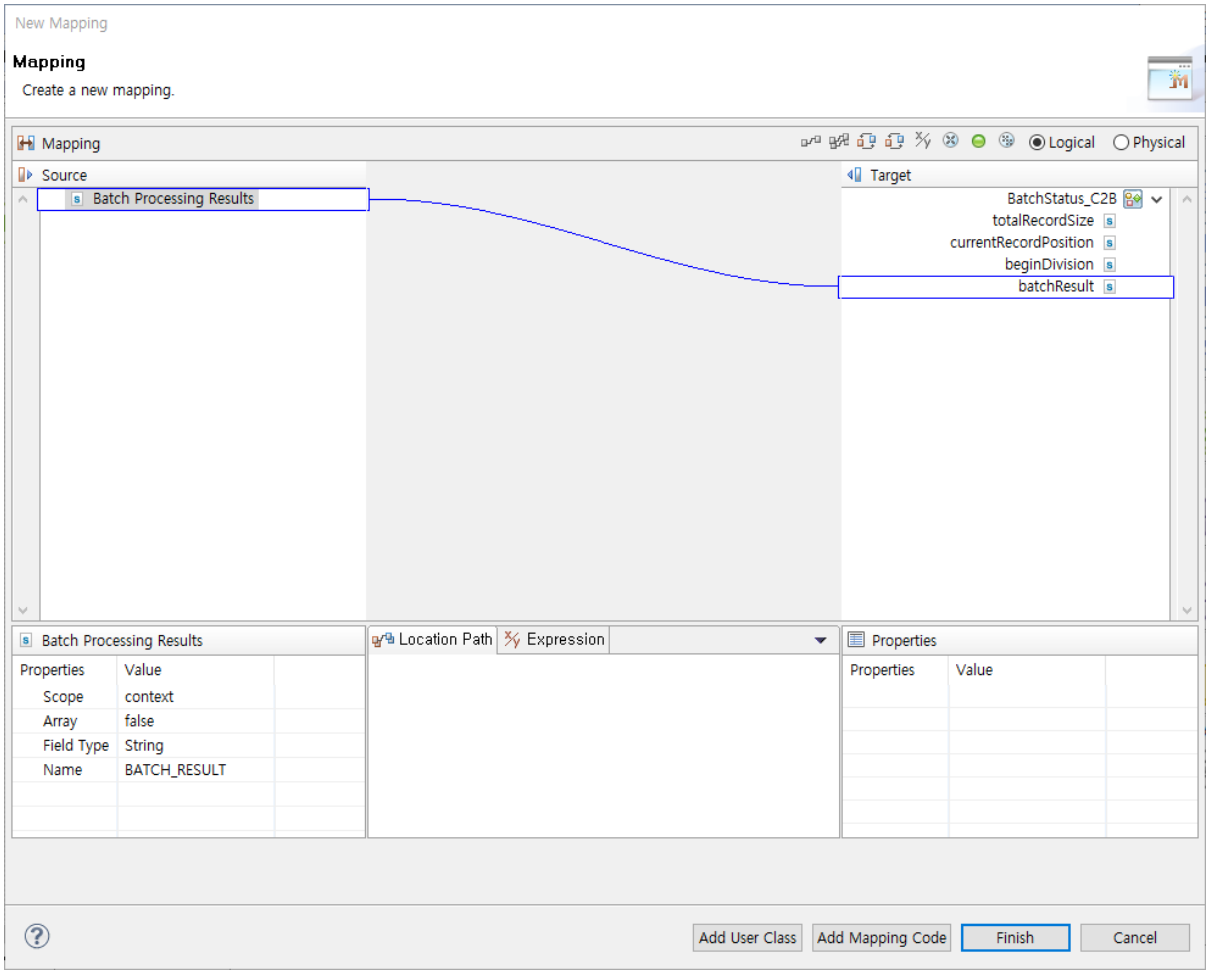
[Figure 6.43] Batch Progress Mapping



Item	Description
Batch Progress	Changes batch progress information through mapping. – Total Records: Total number of records. – Current Record Position: Current record position. – Start Origin: Start origin. – Response Code: Response code.
Status	Changes batch process status (either Complete or Failed).
Search	Searches batch processing results through mapping.

To change batch status, select **'Status'** and select the status. To search batch process results, select **'Search'** and click **[Mapping...]**. This opens the **Batch Process Result Mapping** dialog box where you can map batch process results.

[Figure 6.44] Batch Process Result Mapping

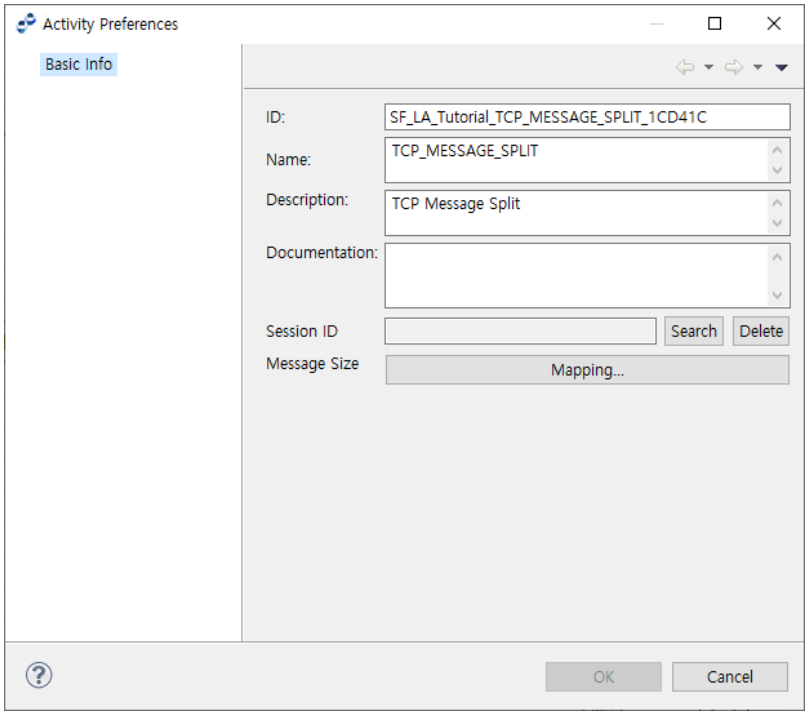


6.8.7. TCP MESSAGE SPLIT ACTIVITY

The following describes the **TCP Message Split Preference** window for TCP MESSAGE SPLIT ACTIVITY. To use this function, select Use Message Fragmentation in the TCP Endpoint Details page shown by selecting **[System] > [Adapter]** in WebAdmin.

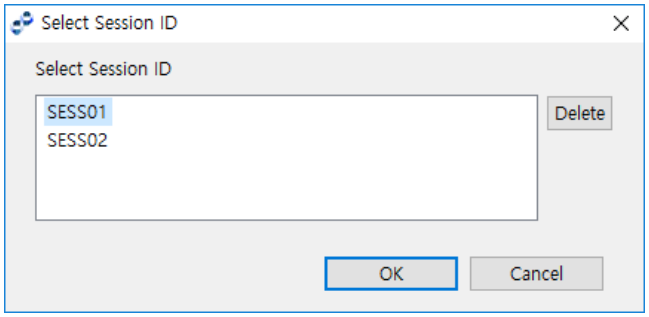
If TCP MESSAGE SPLIT ACTIVITY is set, Intermediate Event processed after the activity receives a message by using 'Total Length' and 'Split Length' set in TCP MESSAGE SPLIT ACTIVITY. 'Total Length' is the total length of a message, and 'Split Length' is the length of a message to read after splitting it.

[Figure 6.45] TCP Message Split Preference



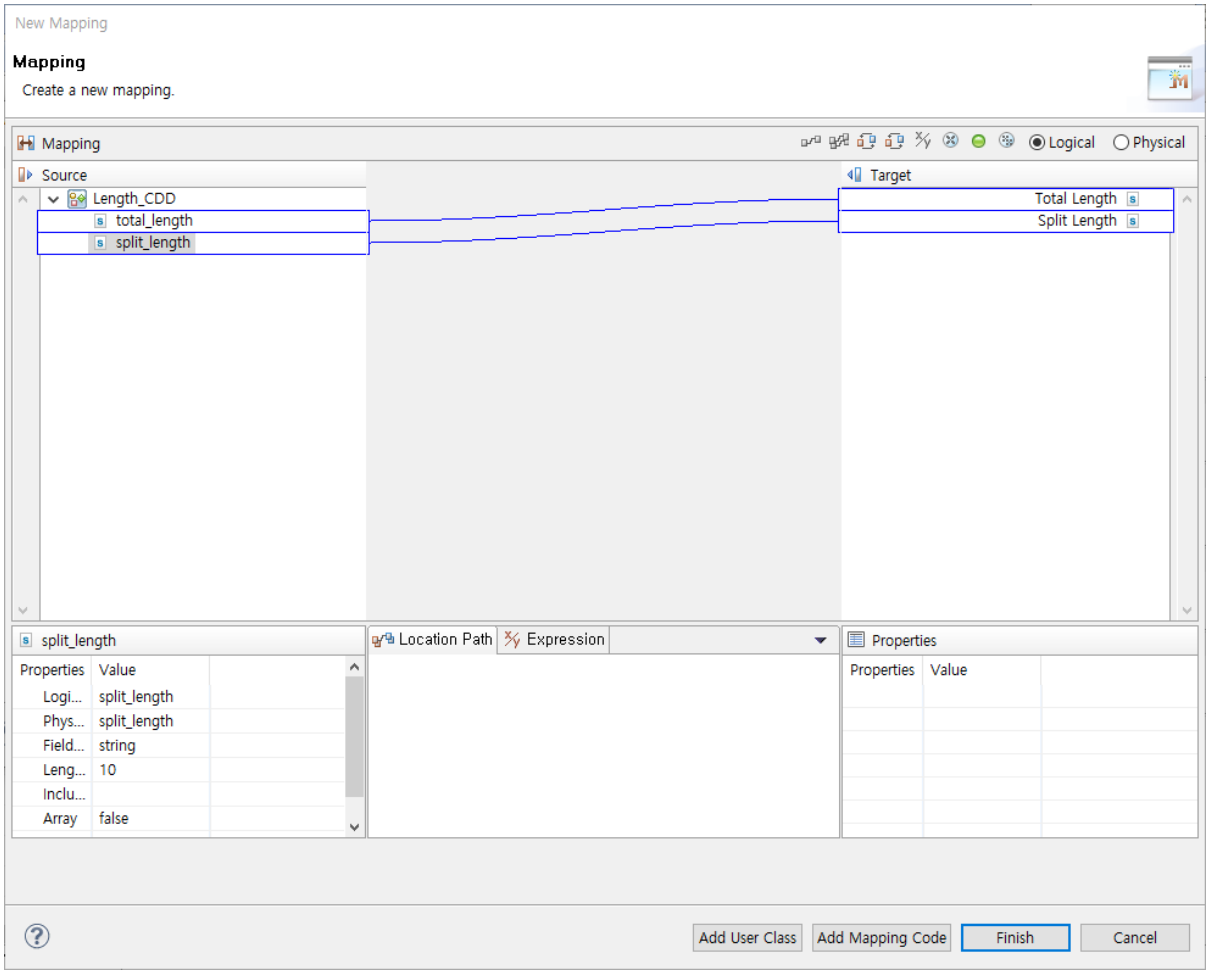
Select **'Session ID'** added in the service flow. Clicking **[Search]** opens the **Select Session ID** dialog box.

[Figure 6.46] Select Session ID



To map the length of messages to receive, click **[Mapping...]**. This opens the **Message Length Mapping** dialog box where you can map the length. **Intermediate Event** processed after this mapping receives a message by using the total mapped length and split length.

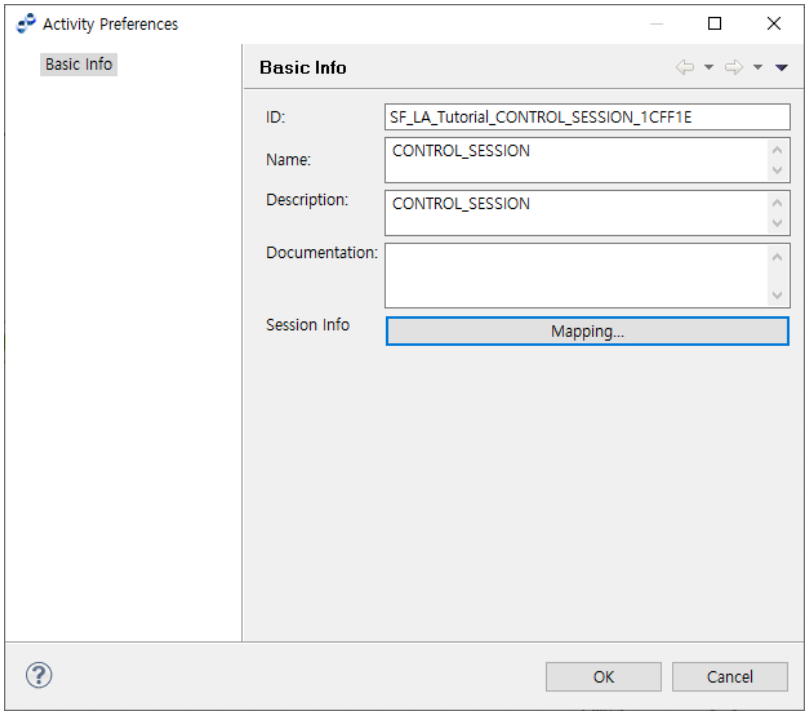
[Figure 6.47] Message Length Mapping



6.8.8. CONTROL SESSION ACTIVITY

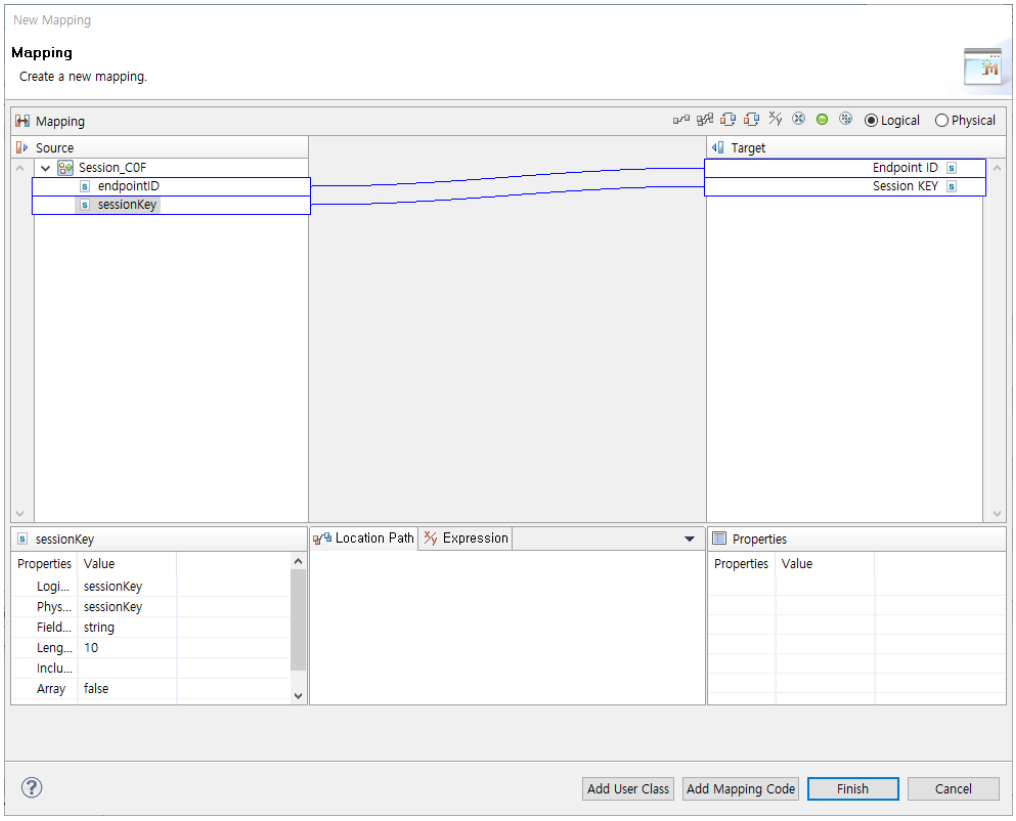
The following describes the **Control Session Preference** window for CONTROL SESSION ACTIVITY.

[Figure 6.48] Control Session Preference



To map a session to end, click **[Mapping...]**. This opens the **Session Information Mapping** dialog box where you can map the session.

[Figure 6.49] Session Information Mapping



Chapter 7. User Class/Handler Editor

This chapter describes the functions and uses of the User Class/Handler editor.

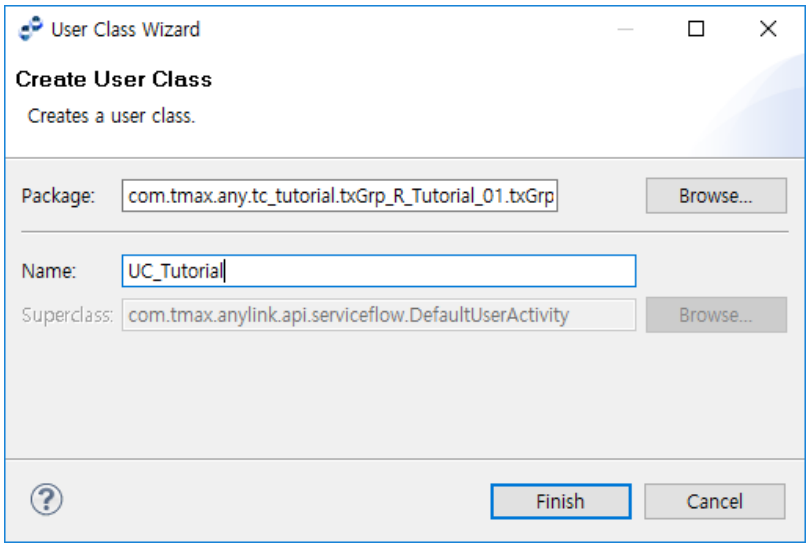
7.1. User Class

A user class is used to define logic that is too complex to be expressed as a flow condition.

7.1.1. Creating a User Class

To create a user class, select **[New] > [User Class]** from the context menu of the **Project Navigator**. Enter the required items in the **Create User Class** window, and then click **[Finish]**.

[Figure 7.1] Create User Class



Item	Description
Package	Package of the user class.
Name	User class name (follow the Java class naming convention).
Superclass	Superclass of the user class. (Default value: DefaultUserActivity)

7.1.2. User Class Implementation

Once a user class is created, enter the class implementation in the editor.

A user class is implemented like a Java class, and it inherits the DefaultUserActivity class and overrides the action method.

```
package com.tmax.tc_tutorial.txGrp_R_tutorial.tx_L_A_tutorial;

import com.tmax.anylink.api.serviceflow.ActivityContext;
import com.tmax.anylink.api.serviceflow.DefaultUserActivity;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;

public class UC_Tutorial extends DefaultUserActivity {

 private static final Logger logger = Logger.getLogger(UC_Tutorial.class.getName());


 public void action(ActivityContext arg0) throws AnyLinkException {
  // TODO Auto-generated method stub

 }

}
```

7.1.3. getEnv Function

The getEnv function is used for a user class or handler to get system and BizTx information.

The following describes keys that are available in the function and values that can be obtained with the keys.

Key	**Value Description**
install.root	Root path where AnyLink is installed.
domain.home	Domain home path.
server.home	Server home path.
server.name	Server name.
adminServer.name	adminServer name.
cluster.name	Cluster name.
bizsystem.id	Biz system ID.
hostname	hostname.
biztx.id	BizTx ID.
guid	GUID.
startendpoint.id	ID of an endpoint with started BizTx.

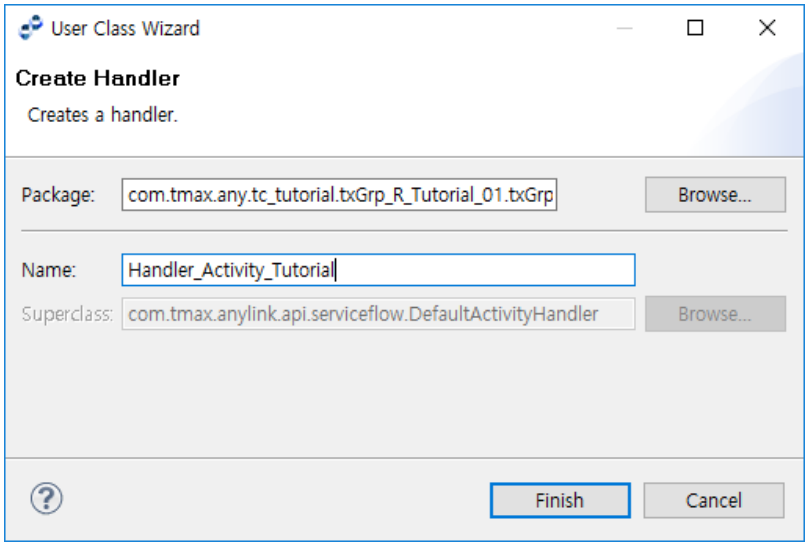
Key	Value Description
startadapter.id	ID of an adapter with started BizTx.
startendpoint.sysid	SysID of an endpoint with started BizTx.
startendpoint.connection.count	Connection count of an endpoint with started BizTx.
request.ip	IP address of a counterpart that requests BizTx, or job ID if a job schedule is used.
request.port	Port number of a counterpart that requests BizTx.
request.url	URL that requests BizTx.
request.http.cookie	Cookie set in the HTTP request through which BizTx is executed.
connection.guid	Connection GUID.
biztx.name	BizTx name.
biztx.code	BizTx code.
biztx.type	BizTx type.
correlation.value	Correlation value. Available for mapping handlers and mapping activities.
local.ip	Local IP address.
local.port	Local port number.
connect.type	Connection type.
${sys:variable_name}	System variable set in WebAdmin.

7.2. Handler

7.2.1. Creating a Handler

To create a handler, select **[New] > [Handler]** and select the desired handler type from the context menu of the **Project Navigator**. Enter the required items in the **Create Handler** window, and then click **[Finish]**.

[Figure 7.2] Create Handler - Activity Handler



Item	Description
Package	Package of the handler class.
Name	Handler class name (follow Java class naming convention).
Superclass	Superclass of the handler class.

7.2.2. Handler Class Implementation

Once a handler is created, enter the class implementation in the editor. A handler class is implemented like a Java class, and it inherits its superclass.

The following are the handlers provided in AnyLink.

Handler Class Type	Super Class	Description
Activity Handler	DefaultActivityHandler	Defines Activity's operation at a specific time, e.g., start or end time.
Activity Error Handler	DefaultActivityErrorHandler	Defines the action to perform when an error occurs in the Activity.
Process Handler	DefaultProcessHandler	Defines the operation to perform at a specified time, such as the process start or end time.
Process Error Handler	DefaultProcessErrorHandler	Defines the action to perform when a process error occurs.
Adapter Message Handler	DefaultAdapterMessageHandler	Performs preprocessing before receiving a message or sending the message to an endpoint.

Handler Class Type	Super Class	Description
Encryption Handler	DefaultEncryptionHandler	Defines encryption method for logging.
User Mapping Handler	DefaultUserMappingHandler	Defines complex mapping that is not supported in the Mapping window.
Parsing Handler	DefaultParsingHandler	Parses messages.
Timer Handler	DefaultUserTimer	Defines additional Timer event configuration.
Message Validation Handler	DefaultMessageValidityHandler	Defines message validation logic.
Routing Handler	DefaultRoutingHandler	Implements logic for multi-binding.
SSO Login Handler	DefaultSsoLoginHandler	Implements logic for SSO login.
Monitoring Event Handler	DefaultMonitoringEventHandler	Implements logic for processing events occurred during monitoring.
Message Management Handler	DefaultMsgTransferHandler	Implements logic for sharing messages.
Administrator Handler	DefaultPartnerAddressHandler	Implements logic to call by using Administrator Info set in External Contacts.

Activity Handler

Activity handler is called when an Activity is started, finished, cancelled, or finished handling an error.

Handler	Description
started	Called when Activity starts.
finished	Called when Activity is finished.
cancelled	Called when Activity is cancelled.
errorOccurred	Called when Activity is finished handling an error.

The following handler logs the Activity ID when an Activity is started, finished, cancelled, or finished handling an error.

```
package com.tmax.pkg.handlerCase1;

import com.tmax.anylink.api.serviceflow.ActivityContext;
import com.tmax.anylink.api.serviceflow.DefaultActivityHandler;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;

public class ActivityHandlerImpl extends DefaultActivityHandler {

private static Logger logger = Logger.getLogger(ActivityHandlerImpl.class.getName());
```

```
 @Override
 public void started(ActivityContext paramActivityContext)
   throws AnyLinkException {
  // TODO Auto-generated method stub
  logger.info("==== Activity Started : " + paramActivityContext.getActivityId() );
 }

 @Override
 public void finished(ActivityContext paramActivityContext)
   throws AnyLinkException {
  // TODO Auto-generated method stub
  logger.info("==== Activity Finished : " + paramActivityContext.getActivityId() );
 }

 @Override
 public void cancelled(ActivityContext paramActivityContext,
   String paramString) throws AnyLinkException {
  // TODO Auto-generated method stub
  logger.info("==== Activity Canceled : " + paramActivityContext.getActivityId() );
 }

 @Override
 public void errorOccurred(ActivityContext paramActivityContext,
   Throwable paramThrowable) throws AnyLinkException {
  // TODO Auto-generated method stub
  logger.info("==== Acitivity Error Occured : " + paramActivityContext.getActivityId()
 );
 }
}
```

After implementing the handler, click **[Add...]** from the Flow Activity's **[Activity Preferences]** > **Activity Handler** page to register the handler.

Activity Error Handler

Activity Error Handler is called when an error occurs while executing an Activity.

The following handler logs the Activity ID and error message when an error occurs while executing an Activity.

```
package com.tmax.pkg.handlerCase1;

import com.tmax.anylink.api.serviceflow.ActivityContext;
import com.tmax.anylink.api.serviceflow.DefaultActivityErrorHandler;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;
```

```
public class ActivityErrorHandlerImpl extends DefaultActivityErrorHandler {

 private static Logger logger =
Logger.getLogger(ActivityErrorHandlerImpl.class.getName());
 @Override
 public void handle(ActivityContext paramActivityContext,
   Throwable paramThrowable) throws AnyLinkException {
  // TODO Auto-generated method stub
  logger.info("==== Activity ID : "+paramActivityContext.getActivityId());
  logger.info("==== Error Message : "+paramThrowable.getMessage());
 }
```

After implementing the handler, click **[Add...]** from the Flow Activity's **[Activity Preferences]** > **Activity Handler** page to register the handler with the error code so that the handler is invoked when an error with the specified error code occurs.

Process Handler

Process Handler is called when a Service Flow starts or ends.

The following handler logs the Flow ID when the flow starts or ends.

```
package com.tmax.pkg.handlerCase1;

import com.tmax.anylink.api.serviceflow.DefaultProcessHandler;
import com.tmax.anylink.api.serviceflow.ProcessContext;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;

public class ProcessHandlerImpl extends DefaultProcessHandler {

 private Logger logger = Logger.getLogger(ProcessHandlerImpl.class.getName());
 @Override
 public void finished(ProcessContext ctx) throws AnyLinkException {
  // TODO Auto-generated method stub
  super.finished(ctx);
  logger.info("==== Flow Finished : " + ctx.getProcessId());
 }

 @Override
 public void started(ProcessContext ctx) throws AnyLinkException {
  // TODO Auto-generated method stub
  super.started(ctx);
  logger.info("==== Flow Started : " + ctx.getProcessId());
```

```
 }
}
```

After implementing the handler, click **[Properties]** from the context menu of the editor and then click **[Add...]** from the **[Process Handler]** tab to register the handler.

Process Error Handler

Process Error Handler is called when an error occurs while executing a Flow.

The following handler logs the Flow ID and error message when an error occurs while executing a Flow.

```
package com.tmax.pkg.handlerCase1;

import com.tmax.anylink.api.serviceflow.DefaultProcessErrorHandler;
import com.tmax.anylink.api.serviceflow.ProcessContext;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;

public class ProcessErrorHandlerImpl extends DefaultProcessErrorHandler {

 private static Logger logger = Logger.getLogger(ProcessHandlerImpl.class.getName());

 @Override
 public void handle(ProcessContext context, Throwable error)
   throws AnyLinkException {
  // TODO Auto-generated method stub
  logger.info("==== Flow ID : " + context.getProcessId());
  logger.info("==== Error Message : " + error.getMessage());
 }
```

After implementing the handler, click **[Properties]** from the context menu of the editor and then click **[Add...]** from the **[Process Handler]** tab to register the handler with the error code so that the handler is invoked when an error with the specified error code occurs.

Adapter Message Handler

Adapter Message Handler transforms messages that are received and sent through an endpoint. The **receive** method transforms incoming messages and the **send** method transforms outgoing messages.

The following handler changes the input message to all lower cases and the output message to all upper cases.

```
package com.tmax.pkg.adapterMSGCase1.tx1;
```

```
import com.tmax.anylink.api.adapter.DefaultAdapterMessageHandler;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;

public class AdapterMessageHandlerImpl extends DefaultAdapterMessageHandler {

 private static Logger logger =
Logger.getLogger(AdapterMessageHandlerImpl.class.getName());

 @Override
 public Object receive(Object arg0) throws AnyLinkException {
  // TODO Auto-generated method stub
  String receiveMsg = new String((byte[]) arg0);
  logger.info("==== RECEIVE MSG : " + receiveMsg );

  return receiveMsg.toLowerCase().getBytes();
 }

 @Override
 public Object send(Object arg0) throws AnyLinkException {
  // TODO Auto-generated method stub
  String sendMsg = new String((byte[]) arg0);
  logger.info("==== SEND MSG : " + sendMsg );

  return sendMsg.toUpperCase().getBytes();
 }

}
```

After implementing the handler, deploy the BizTx that includes the handler. In WebAdmin, register the handler as the **'Message Handler'** in the **[Advanced Settings]** tab of the Endpoint settings page.

Encryption Handler

Encryption Handler encrypts or decrypts a message when saving or retrieving a trace or custom log. The **decrypt** method decrypts and the **encrypt** method encrypts a message.

The following handler uses BASE64 encoding method for encrypting or decrypting a message.

```
package com.tmax.pkg.encryptionCase1.tx1;

import com.tmax.anylink.api.log.DefaultEncryptionHandler;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.common.util.Base64;
import com.tmax.anylink.logging.Logger;
```

```
public class EncryptionHandlerImpl extends DefaultEncryptionHandler {

 private static Logger logger =
Logger.getLogger(EncryptionHandlerImpl.class.getName());

 @Override
 public byte[] decrypt(byte[] arg0) throws AnyLinkException {
  // TODO Auto-generated method stub

  logger.info("==== Encrypted MSG : " + new String(arg0) );

  Base64 decoder = new Base64();
  byte[] decryptedMessage = decoder.decode(new String(arg0));

  return decryptedMessage;
 }

 @Override
 public byte[] encrypt(byte[] arg0) throws AnyLinkException {
  // TODO Auto-generated method stub

  logger.info("==== Original MSG : " + new String(arg0) );

  Base64 eecoder = new Base64();
  byte[] encryptedMessage = encrypt(arg0);

  return encryptedMessage;
 }
}
```

After implementing the handler, select the **'Use Encryption/Decryption'** option in **Custom Log Outbound Rule** and select the handler in **'Select User Class'** option. After deploying the BizTx that includes the handler, go to WebAdmin and register the handler with the package name as the **'Encryption User Class Name'** in the **[Trace Log]** tab of the Log Adapter settings page.

User Mapping Handler

User Mapping Handler is used to implement custom mapping that is usually created from the Mapping dialog box. The Source is passed in as **arg1** and the Target is returned as **Object[]**.

The following handler performs one-to-one mapping of input and output sequentially.

```
package com.tmax.pkg.handlerCase1;

import com.tmax.anylink.api.serviceflow.ActivityContext;
import com.tmax.anylink.api.serviceflow.DefaultUserMapping;
```

```
import com.tmax.anylink.common.AnyLinkException;

public class UserMappingHandlerImpl extends DefaultUserMapping {

 @Override
 public Object[] mapping(ActivityContext arg0, Object[] arg1)
   throws AnyLinkException {
  // TODO Auto-generated method stub
  Object[] mappedTarget = new Object[arg1.length];
  for(int i = 0; i < arg1.length; i++){
   mappedTarget[i] = arg1[i];
  }
  return mappedTarget;
 }
}
```

After implementing the handler, click **[Add User Class]** from the Mapping dialog box to register the handler.

Parsing Handler

Parsing Handler can be used when the **'Child BizTx Identification Method'** under **Parsing Info** for a **[BizTx Group]** is set to 'HANDLER'. The handler's return value, BizTx Identifier Code, is used to find the BizTx for parsing.

The following handler that returns the first four characters of the request message as the BizTx Identifier Code.

```
package com.tmax.pkg.handlerCase1;

import com.tmax.anylink.api.adapter.DefaultParsingHandler;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;

public class ParsingHandlerImpl extends DefaultParsingHandler {

 private static Logger logger = Logger.getLogger(ParsingHandlerImpl.class.getName());


 @Override
 public String parsing(Object arg0) throws AnyLinkException {
  // TODO Auto-generated method stub
  String inputMsg = new String((byte[]) arg0);
  logger.info("====Input Message : " + inputMsg);
  String parsingMsg = inputMsg.substring(0, 3);
  logger.info("====Parsing Message : " + parsingMsg);
```

```
  return parsingMsg;
 }
}
```

After implementing the handler, set the **'Child BizTx Identification Method'** under **Parsing Info** in the **BizTx Group Editor** to 'HANDLER' and then register the handler.

Timer Handler

Timer Handler fires timer events. The **nextTime** method returns the next time to fire the timer event, and **shouldFireTimerEvent** method returns whether or not to fire the timer event.

The following handler fires the timer event at 500 ms interval unconditionally.

```
package com.tmax.pkg.handlerCase1;

import java.util.logging.Logger;

import com.tmax.anylink.api.serviceflow.ActivityContext;
import com.tmax.anylink.api.serviceflow.DefaultUserTimer;

public class TimerHandlerImpl extends DefaultUserTimer {

 private static Logger logger = Logger.getLogger(TimerHandlerImpl.class.getName());
 private long startTime = -1;
 private final static long PERIOD = 20;

 @Override
 public long nextTime(ActivityContext arg0) {
  // TODO Auto-generated method stub
  if(startTime < 0) {
   startTime = System.currentTimeMillis() + PERIOD ;
  } else {
   startTime += PERIOD ;
  }

  logger.info("####### TimerHandlerImpl startTime : " + startTime);

  return startTime;
 }

 @Override
 public boolean shouldFireTimerEvent(ActivityContext arg0, long arg1) {
  // TODO Auto-generated method stub
  return true;
```

```
 }
}
```

After implementing the handler, set the **'Timer Class Name'** to the handler name in the **[Timer]** tab of the **[Timer Event] > [Event Preference]** page.

Message Validation Handler

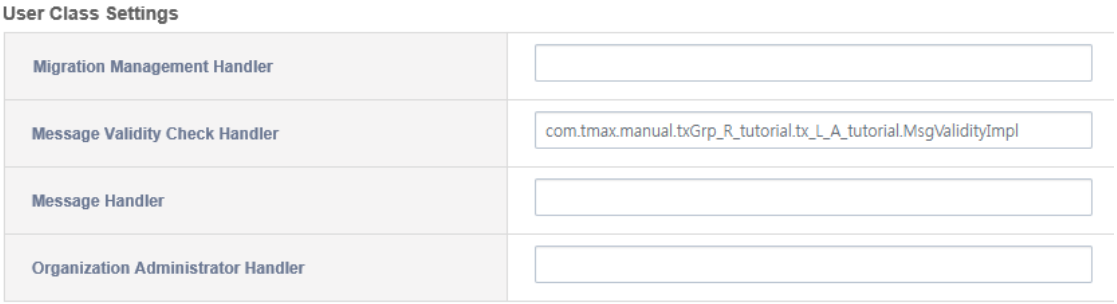
Message Validation Handler performs message validation (including meta data).

The following handler checks the message's **Physical Name** and returns false if it contains the string "ab".

```
package com.tmax.pkg.msgValidityCase1;

import java.util.HashMap;
import java.util.Map;

import com.tmax.anylink.api.validation.DefaultMessageValidityHandler;
import com.tmax.anylink.common.AnyLinkException;
import com.tmaxsoft.promapper.structure.StructureFieldType;

public class MsgValidityImpl extends DefaultMessageValidityHandler {

 @Override
 public Map<StructureFieldType, Boolean> messageMetaCheck(
  StructureFieldType[] arg0) throws AnyLinkException {
  // TODO Auto-generated method stub
  Map<StructureFieldType, Boolean> ret = new HashMap<StructureFieldType, Boolean>();


  for(StructureFieldType type : arg0){
   if(type.getPhysicalName().contains("ab")){ // invalid message field
    ret.put(type, true); // save the message field as'true' in the map
   }else{  // valid message field
    ret.put(type, false); // save the message field as'true' in the map
   }
  }
  return ret;
 }
}
```

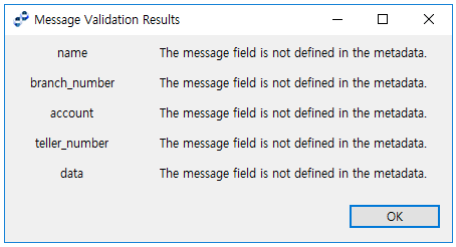
After implementing the handler, click **[Upload Handler]** from the context menu of the handler in the Navigator to upload the handler to DIS. In WebAdmin, enter the full package handler name in **[Admin] > [DIS Configuration] > [User Class Setting] > [message-validity-class-name]**.

[Figure 7.3] Adding Message Validation Handler (WebAdmin)

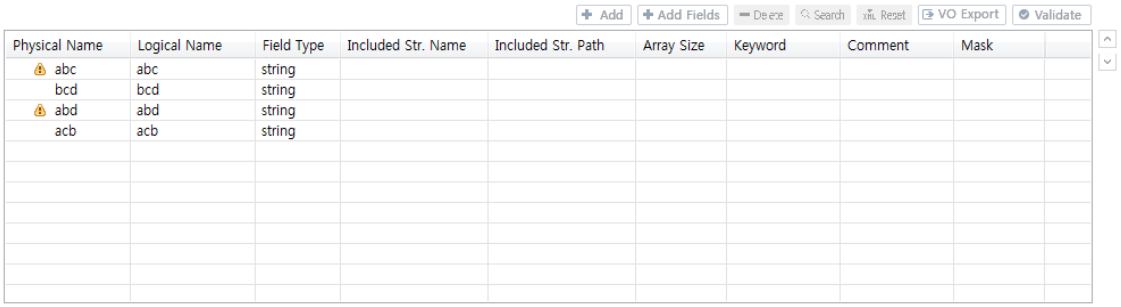


Click **[Validate]** from the Message Editor to validate the message.

[Figure 7.4] Message Validation Result - Dialog Box



[Figure 7.5] Message Validation Result - Editor



Note

To use the message validation handler, the compiler version must match the Java version of the DIS server. (**[Window] > [Preferences] > [Java] > [Compiler] > [Compiler compliance level] : 1.6**)

Routing Handler

Routing Handler is used for multi-binding. The **route** method is executed when multi-binding is called. A service that has the same value as the return value is called.

The following handler performs routing by using message contents as a value.

```
package com.tmax.pkg.handlerCase1;

import com.tmax.anylink.api.MessageContext;
import com.tmax.anylink.api.multibinding.DefaultRoutingHandler;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;

public class routing extends DefaultRoutingHandler {

 private static final Logger logger = Logger.getLogger(routing.class.getName());

 @Override
 public String route(MessageContext arg0) throws AnyLinkException {
  return (String)new String((byte[])arg0.getContent());
 }
}
```

After implementing the handler, set the multi-binding binding option to **'HANDLER'** and enter the handler to additional information of the setting.

SSO Login Handler

SSO Login Handler is used for SSO login. The **login** method, called when a user tries to log in, determines whether the login is successful or not by using the return value, **SsoHandlerResult**. The method has two parameters: user ID and user password.

The following handler outputs the ID and password of a user who tries to log in as log and makes the user succeed in login.

```
package com.tmax.pkg.handlerCase1;

import com.tmax.anylink.api.SsoLogin.DefaultSsoLoginHandler;
import com.tmax.anylink.api.SsoLogin.SsoHandlerResult;
import com.tmax.anylink.api.SsoLogin.SsoHandlerResult.HandlerResultType;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;

public class ssologin extends DefaultSsoLoginHandler {

 private static final Logger logger = Logger.getLogger(ssologin.class.getName());

 @Override
 public SsoHandlerResult login(String arg0, String arg1)
   throws AnyLinkException {
  logger.info("SSO ID : " + arg0 + ", PW : " + arg1);
  SsoHandlerResult result = new SsoHandlerResult();
```

```
  result.setResult(HandlerResultType.SUCCESS);
  return result;
 }
}
```

After implementing the handler, set the **'SSO Method'** item to 'UserClass Method' in the WebAdmin page shown by selecting **[Admin] > [DIS Configuration]** and enter the handler to the SSO handler.

Monitoring Event Handler

Monitoring Event Handler is called when a monitoring event occurs. The **event** method, called when an event occurs, gets the event type with the first argument and monitoring event information with the second argument as a map. Information that can be obtained from the map can be checked in the **'Event Message'** item in the WebAdmin page shown by selecting **[Configuration] > [Event] > [Events]**.

The following handler outputs occurred event information as log.

```
package com.tmax.pkg.handlerCase1;

import java.util.Map;

import com.tmax.anylink.api.event.DefaultMonitoringEventHandler;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;
import com.tmaxsoft.schemas.anylink.MonitoringEventType;

public class MonitoringEvent extends DefaultMonitoringEventHandler {

 private static final Logger logger =
Logger.getLogger(MonitoringEvent.class.getName());

 @Override
 public void event(MonitoringEventType arg0, Map<String, Object> arg1)
   throws AnyLinkException {
  logger.info(arg0.toString());
 }

}
```

After implementing the handler, deploy it and then search, configure, and save the handler in the **'Event Handler'** item in the WebAdmin page shown by selecting **[Configuration] > [Event] > [Events]**.

Message Management Handler

Message Management Handler is called by clicking **[Send]** in the WebAdmin page shown by selecting **[Configuration] > [Message]**. The **callHandler** method, called by clicking **[Send]**, receives a checked message list as an argument and gets the message information.

The following handler outputs sysid of messages to send as log.

```
package com.tmax.pkg.handlerCase1;

import java.util.List;

import com.tmax.anylink.api.msgtransfer.DefaultMsgTransferHandler;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.dis.msg.ResourceInfoPK;
import com.tmax.anylink.logging.Logger;

public class MsgTransfer extends DefaultMsgTransferHandler {

 private static final Logger logger = Logger.getLogger(MsgTransfer.class.getName());


 public String callHandler(List<ResourceInfoPK> arg0)
   throws AnyLinkException {
  for(int i = 0; i<arg0.size(); i ++) {
   ResourceInfoPK resource = arg0.get(i);
   logger.info("resource sysid = " + resource.getSysId());
  }
  return null;
 }
}
```

After implementing the handler, deploy it and then enter the handler class in the **'Message Handler'** item in the WebAdmin page shown by selecting **[Admin] > [DIS Configuration]**. Select messages to send in the WebAdmin page shown by selecting **[Configuration] > [Message]** and then click **[Send]** to call the handler.

Administrator Handler

Administrator Handler is called by clicking **[Send]** in the WebAdmin page shown by selecting **[Configuration] > [External Contact] > [Administrators]**. The **callHandler** method, called by clicking **[Send]**, gets a checked administrator list.

The following handler outputs selected administrators' names as log.

```
package com.tmax.pkg.handlerCase1;

import java.util.List;

import com.tmax.anylink.api.partneraddress.DefaultPartnerAddressHandler;
import com.tmax.anylink.api.partneraddress.PartnerUser;
import com.tmax.anylink.common.AnyLinkException;
import com.tmax.anylink.logging.Logger;

public class ParnerAddress extends DefaultPartnerAddressHandler {

 private static final Logger logger = Logger.getLogger(ParnerAddress.class.getName());

 @Override
 public String callHandler(List<PartnerUser> partnertUserList)
   throws AnyLinkException {
  for(int i = 0; i<partnertUserList.size(); i++) {
   PartnerUser user = partnertUserList.get(i);
   logger.info("User Name = " + user.getUserName());
  }
  return super.callHandler(partnertUserList);
 }
}
```

After implementing the handler, deploy it and then enter the handler class in the **'Organization Administrator Handler'** item in the WebAdmin page shown by selecting **[Admin] > [DIS Configuration]**. Select administrators to call in the WebAdmin page shown by selecting **[Configuration] > [External Contact] > [Administrators]** and then click **[Send]** to call the handler.

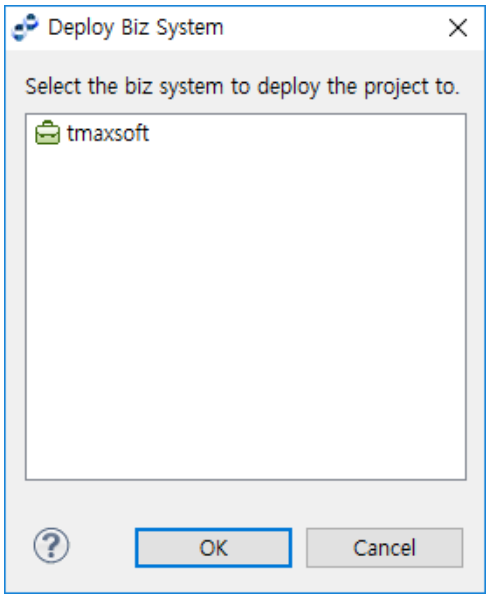
Chapter 8. Common Business Project

This chapter describes the functions and uses of the Common BizTx/BizTx Group function. A common business project is created to define a set of resources that can be shared by projects created for the same business system.

8.1. Creating a Common Business Project

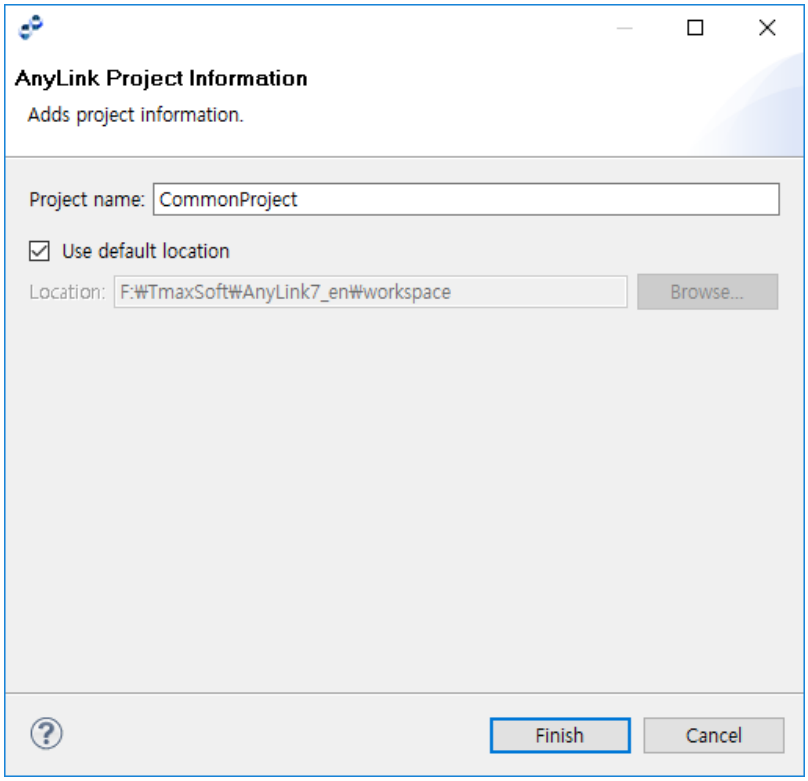
To create a Common Business Project, select **[New] > [Common Business Project]** from the context menu of the **Project Navigator**. Select a business system for the project in the **Select Biz System** window, and then click **[OK].** Only one Common Business Project can be created for each business system.

[Figure 8.1] Deploy Biz System



Enter the required items in the **Create Project** window, and then click **[Finish]**.

[Figure 8.2] Create Project



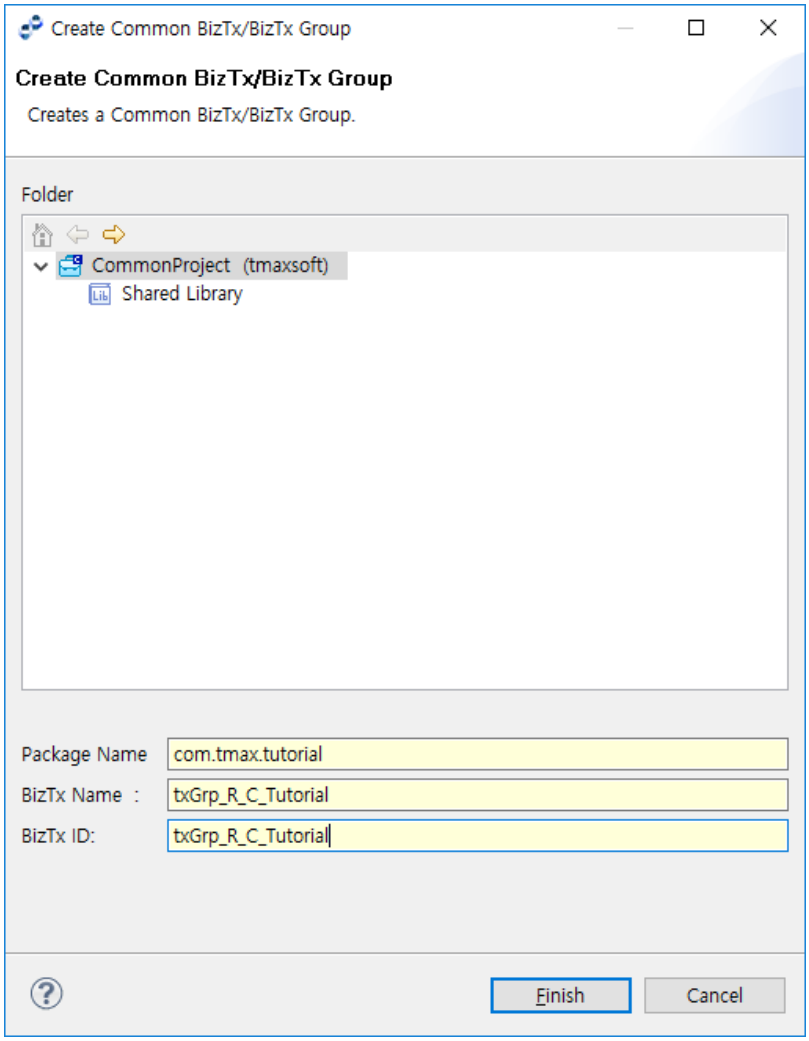
Item	Description
Project Name	Project name. Special characters (? , <, > , ", * ,|) are not allowed.

8.2. Creating a Common BizTx/BizTx Group

A Common BizTx/BizTx Group, like other BizTx/BizTx Groups, can be created according to the node type.

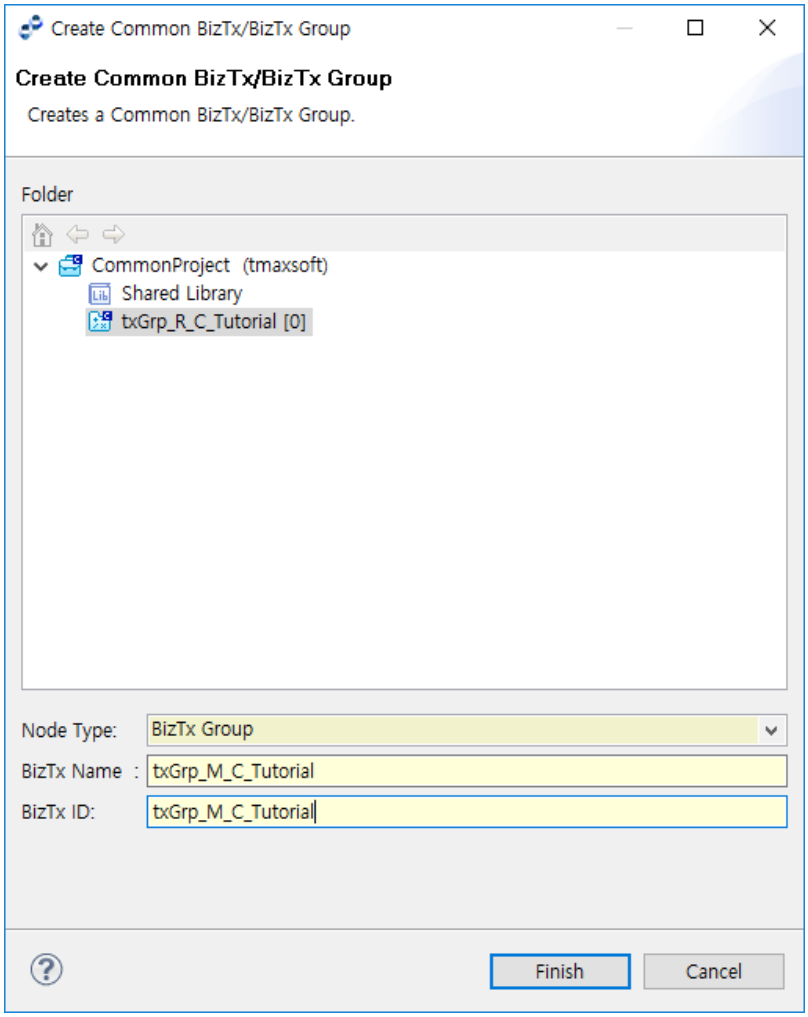
To create the root BizTx Group, select **[New] > [Common BizTx/BizTx Group]** from the context menu of a **Common Business Project** in the Project Navigator.

[Figure 8.3] Create Common BizTx/BizTx Group - Root BizTx Group



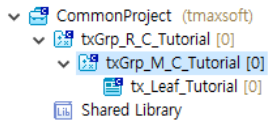
To create a BizTx/BizTx Group select **[New] > [Common BizTx/BizTx Group]** from the context menu of a **Common BizTx Group** resource. The input items are the same as those for a BizTx Group.

[Figure 8.4] Create Common BizTx/BizTx Group - BizTx Group (Interim), BizTx (Leaf)



Once a Common BizTx/BizTx Group is created, it is added to the Project Navigator.

[Figure 8.5] Checking the Result



For information about the editor, which is the same as that for other BizTx/BizTx Groups, refer to "Chapter 2. BizTx/BizTx Group Editor". Multi-binding and WSDL cannot be created in a Common Business Project.

Chapter 9. Deployment & Undeployment

This chapter describes how to deploy/undeploy resources to/from DIS.

9.1. Overview

AnyLink can be used to add resources to a BizTx/BizTx Group and deploy them, and the resources can be rolled back to a previous version through versioning.

In AnyLink, resources can be added to BizTx/BizTx Group and then deployed to DIS and runtime server. The resources in the deployed BizTx/BizTx Group can be modified and redeployed or undeployed from DIS and runtime server.

Deployed resources are registered on the runtime server. The runtime server has information about most of the resources and deployment does not affect the operation of the runtime server engine. However, shared libraries must be deployed before they can be used on the runtime server engine. For more information, refer to the relevant sections.

9.2. Resource Deployment

A resource is deployed in the IAR format, which is a compressed archive file format. The resource version is 0 before deployment, and is changed to 1 when it is deployed for the first time.

Each time a resource is redeployed the version number is incremented by 1, and the resource can be rolled back to a previous version. When attempting to redeploy a resource that is modified after a rollback, DIS will reject the request because of a version conflict between the local version and the latest version on the server.

In such a case, the **'Skip Version Check'** option can be used to deploy the local version to the server as the latest version, but this workaround should only be used when absolutely necessary.

9.2.1. Deployment Unit

All resources in a BizTx/BizTx Group are deployed as a unit. When deploying a BizTx Group, any child BizTx or BizTx Groups are not deployed by default. The 'Include Subgroups' option can be set to deploy the child BizTx or BizTx Group.

9.2.2. Deployment

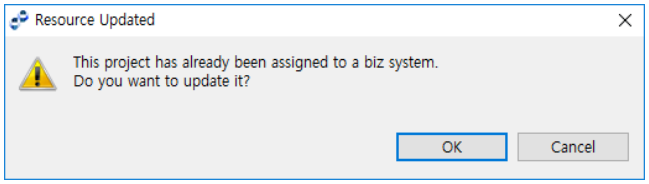
This section describes the deployment process that involves assigning a business system and deploying the resources.

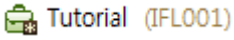
Assigning a Business System

Before deploying the resources, the project must be assigned to a business system.

To assign a business system, select **[Assign Biz System]** from the context menu of the **Project** in the Project Navigator. If one is already assigned, the following message is displayed. Click **[OK]** to modify the current assignment.

[Figure 9.1] Warning Message When Assigning a Business System

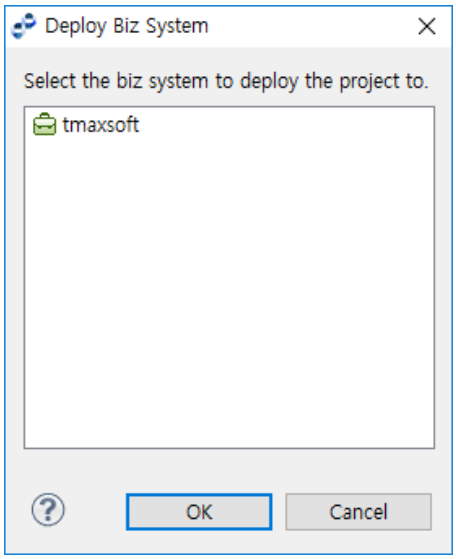


To check whether a business system has been assigned to the project, check the project name in the **Project Navigator**. If one is assigned, the project is shown as "Project Name (Business System)" 

.

Select a desired business system from the **Deploy Biz System** window, and then click **[OK]** to assign the system to the project.

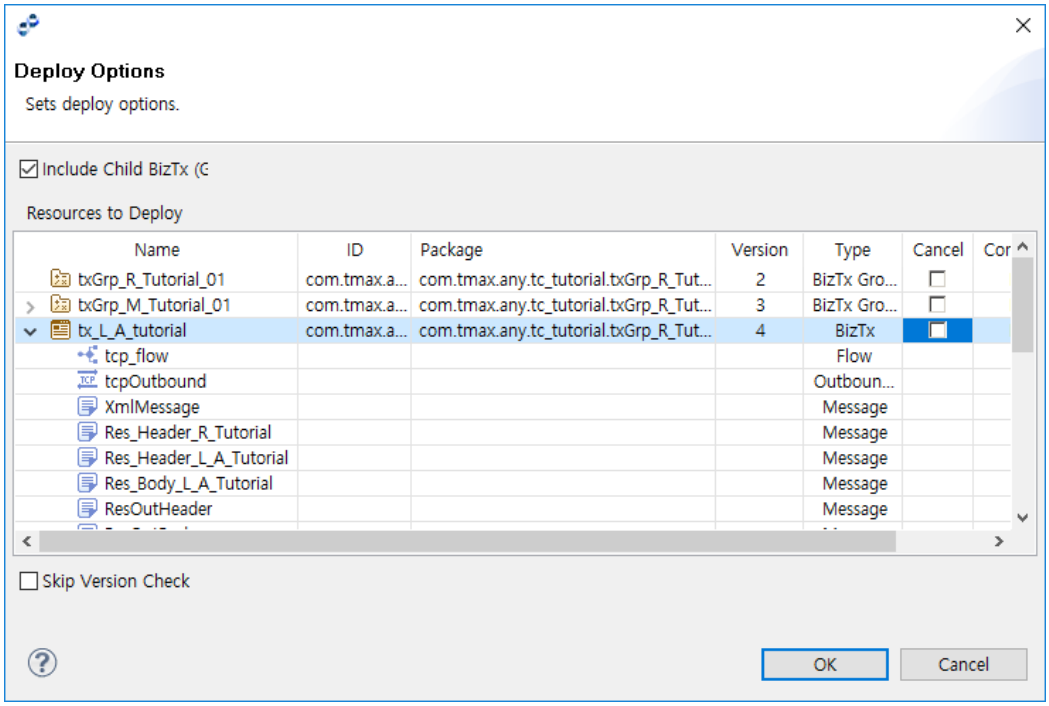
[Figure 9.2] Deploy Biz System



Deploying Resources

Select BizTx to deploy and then click **[Deploy]** from the context menu of the **BizTx/BizTx Group** in the Project Navigator to open the **Deploy Options** window. Enter the required items, and then click **[OK]**. The resources are deployed to DIS, and registered on the runtime server. To deploy multiple BizTxs, hold down the **Ctrl** key and select multiple BizTxs and then click **[Deploy]**.

[Figure 9.3] Deploy Options



Item	Description
Include Subgroups	Option to deploy the child BizTx and BizTx groups that belong to the BizTx Group.
Resources to Deploy	List of resources to deploy. Check the list with the following information. – Name: resource name. – ID: SYS ID of the BizTx node. – Package: package name of the BizTx node. – Version: current version of the BizTx node. – Type: Resource type. – Cancel: option to exclude from deployment. – Check Deploy: deploy status, if currently deployed.

Item	Description
Skip Version Check	Option to deploy the local version as the latest version if a later version exists on the server.

9.3. Resource Undeployment

This section describes how to undeploy a resource. When a resource is undeployed, all data related to the resource is changed to the undeployed state on DIS.

9.3.1. Undeployment Unit

Similar to deployment, all resources in a BizTx/BizTx Group are deployed as a unit.

9.3.2. Undeployment

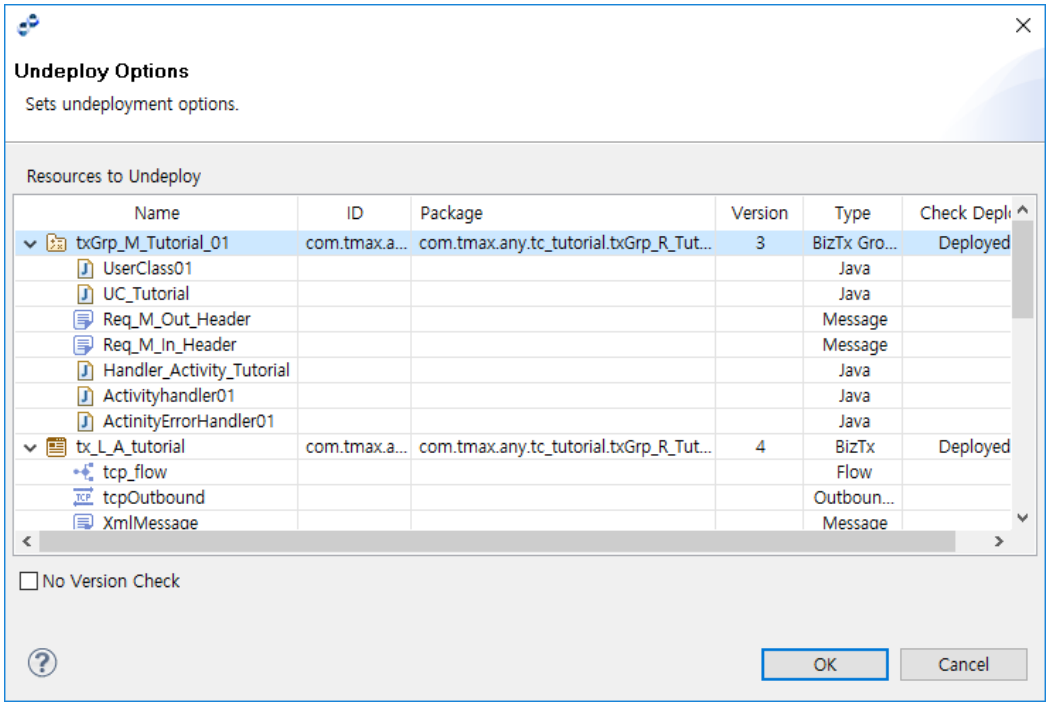
Undeployment also unassigns the resources from the business system.

1. **Undeploying Resources**

   Click **[Undeploy]** from the context menu of the **BizTx/BizTx Group** in the Project Navigator to open the **Undeploy Options** window.

   Enter the required items, and then click **[OK]**. The resources are undeployed from DIS, and they are also stopped and removed from the runtime server engine. If a BizTx Group is undeployed, its child BizTx/BizTx Group nodes are also undeployed.

[Figure 9.4] Undeploy Options



Item	Description
Resources to Undeploy	List of resources to undeploy. Check the list with the following information. – Name: resource name. – ID: SYS ID of the BizTx node. – Package: package name of the BizTx node. – Version: current version of the BizTx node. – Type: Resource type. – Check Deploy: deploy status, if currently deployed.
Skip Version Check	Option to undeploy the local version if it is not the latest version.

2. **Unassigning a Business System**

To unassign a business system, select **[Unassign Biz System]** from the context menu of the **Project** in the Project Navigator. For a common business project, a business system cannot be unassigned. The assignment can only be modified. For more information, refer to "9.2.2. Deployment".

9.4. Shared Library Deployment & Undeployment

This section describes how to deploy and undeploy a shared library.

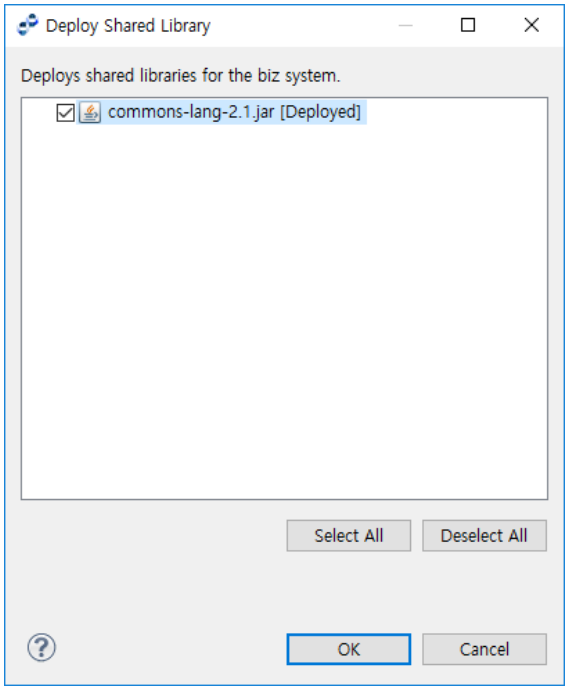
A shared library is a jar file that includes Java compiled library files. Once the jar file is deployed and registered on the runtime server, it can be used by all BizTx nodes providing extended use of various classes. Note that a deployed library may conflict with another library on the runtime server so that the server cannot distinguish which library to call. Since such a conflict can incur problems in AnyLink, care should be taken when deploying a shared library.

To deploy a shared library, specify the target deploy path in the **'Shared Library Path'** item in the WebAdmin page shown by selecting **[Admin] > [DIS]**.

9.4.1. Deploying a Shared Library

Click **[Deploy]** from the context menu of the **Shared Library** in the Project Navigator to open the **Deploy Shared Library** window. Select the library to deploy, and then click **[OK]** to deploy the shared library.

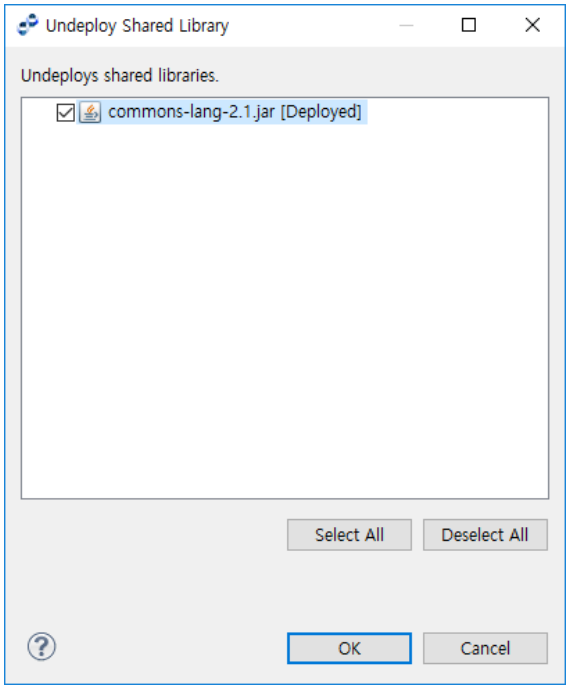
[Figure 9.5] Deploy Shared Library



9.4.2. Undeploying a Shared Library

Click **[Undeploy]** from the context menu of the **Shared Library** in the Project Navigator to open the **Undeploy Shared Library** window. Select the library to undeploy, and then click **[OK]** to undeploy the shared library.

[Figure 9.6] Undeploy Shared Library



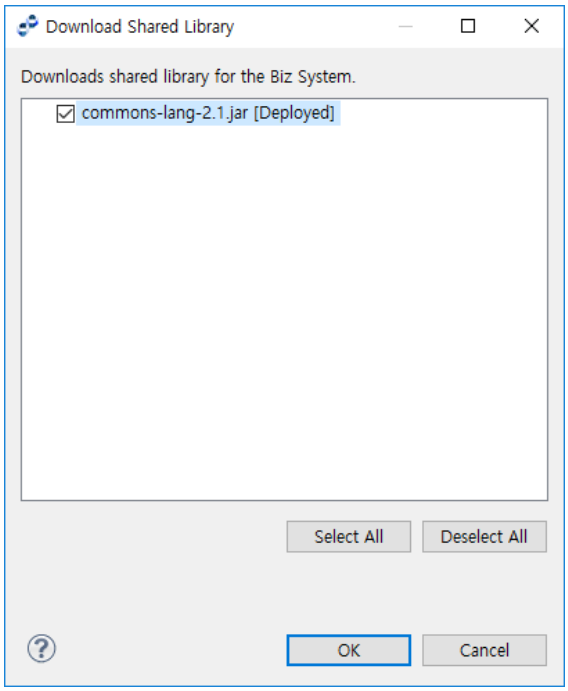
9.4.3. Adding a Library

Click **[Add Library]** from the context menu of the **Shared Library** in the Project Navigator to open the file selection window. Select a jar file, and then click **[OK]** to add the library to the shared library.

9.4.4. Downloading a Library

Click **[Download Library]** from the context menu of the **Shared Library** in the Project Navigator to open the **Download Shared Library** window. Select the library to download, and then click **[OK]** to download the shared library.

[Figure 9.7] Download Shared Library



Chapter 10. Resource Search & Download

This chapter describes how to search for and download system resources.

10.1. Overview

In AnyLink, a resource, such as BizTx, BizTx Group, user class, and adapter, is an element that runs on or provides information to the engine. Various resources in AnyLink can be created and deployed through WebAdmin and Studio. For efficient resource management, resources are grouped in a BizTx/BizTx Group and resource versioning is provided through DIS.

Deployed resources are also registered on the runtime server. The runtime server has information about most of the resources and deployment does not affect the operation of the runtime server engine. However, shared libraries must be deployed before they can be used on the runtime server engine. A deployed resource that is registered on DIS can be downloaded and used by multiple users.

This chapter describes how to find and use AnyLink resources. For information about registering resources on DIS and resource versioning, refer to "Chapter 9. Deployment & Undeployment".

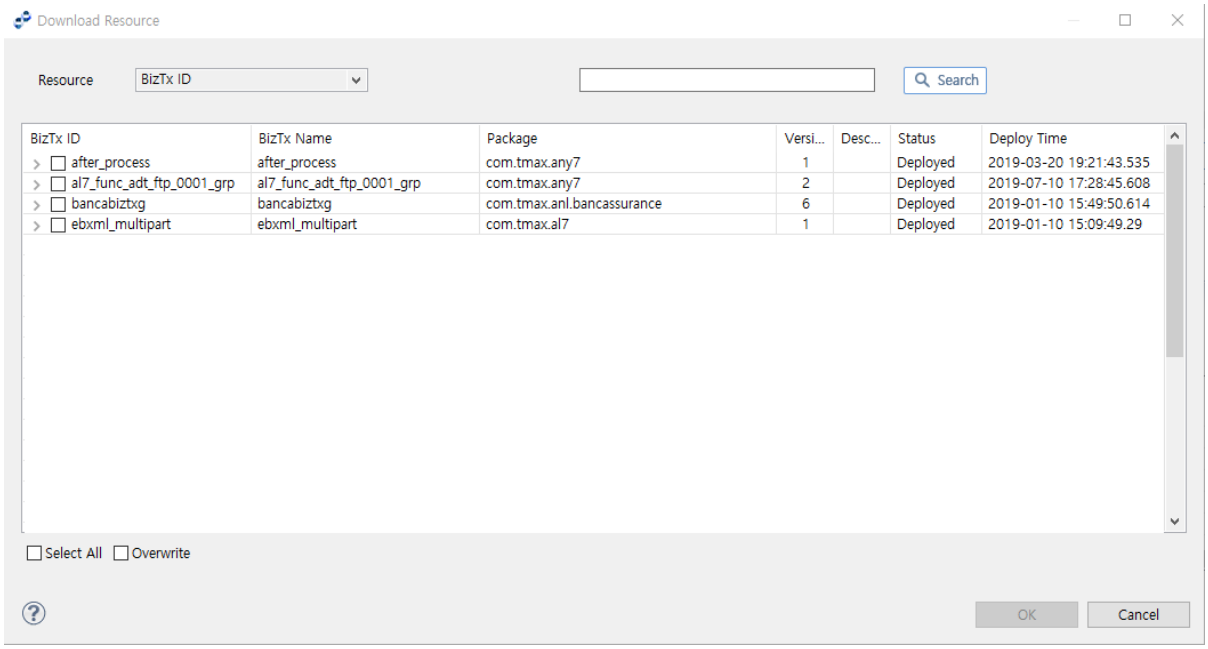
Note

For detailed information about resources, refer to *AnyLink Runtime Engine Server Guide*.

10.2. Resource Search & Download

Click **[Download]** from the context menu of a project, BizTx Group, or BizTx in the Project Navigator to open the **Download Resource** window. Find and check the resources to download, and then click **[OK]** to download them.

[Figure 10.1] Download Resource



Item	Description
Resource	Search keyword type (BizTx Name, BizTx ID, or Package). Enter the keyword in the textbox, and then click **[Search]** to execute the search.
Resource List	List of resources retrieved by the search. If a resource has child resources, the **[**  **]** icon will appear before the resource name. Click this icon to expand or collapse the child resources. The following are the context menu items of a resource: – **[Include Parent BizTx's]**: Download the parent BizTx Group with the selected resource. – **[Include Child BizTx's]**: Download the child BizTx/BizTx Group nodes with the selected resource. – **[Deselect Group]**: Deselect any selected parent or child BizTx/BizTx Group nodes of the selected resource. Click on the 'Version' column to display the **[∇]** icon. Click this icon to display version history of the resource. For a version rollback, select an earlier version to download.
Select All	Selects all resources in the list for download.
Overwrite	Option to overwrite the latest version on DIS with the local version of the same resource.

Appendix A. Dialog Boxes

This appendix describes how to use various dialog boxes in Studio that are shared by multiple pages. For information about each dialog box, refer to the relevant description in this chapter.

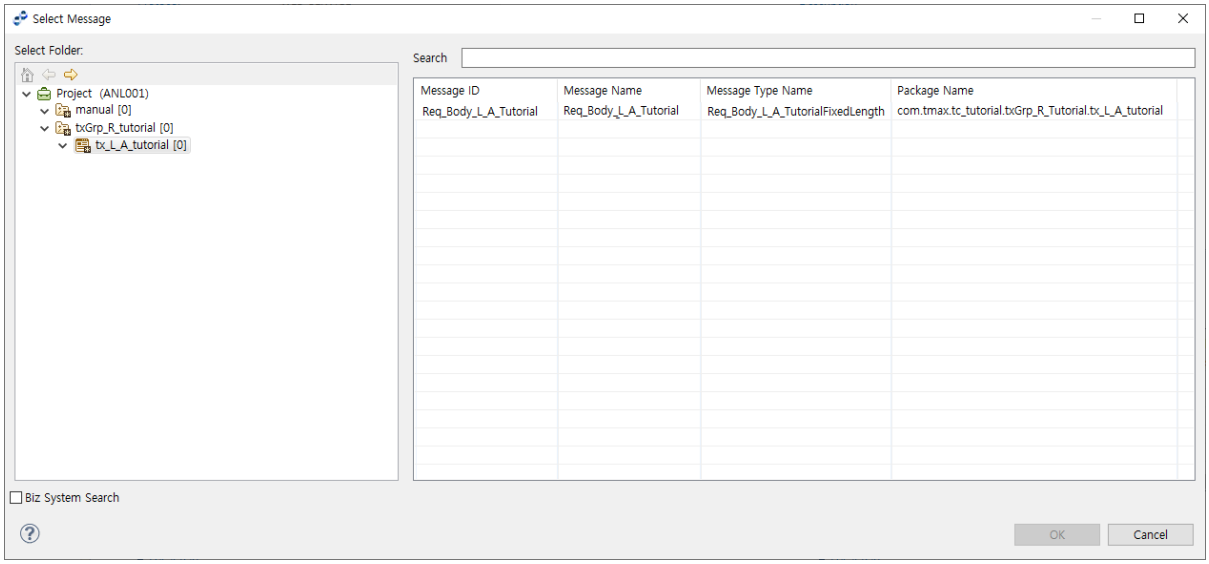
A.1. Message Dialog Boxes

A.1.1. Select Message

The Select Message dialog box is used to select a message. Select a **BizTx/BizTx List** from the **Project Navigator** on the left to view all the messages that belong to it in the **Message List** on the right. Select a message from the list, and then click **[OK]**.

Enter a message name in the Search textbox to search for a specific message.

[Figure A.1] Select Request Message Dialog Box

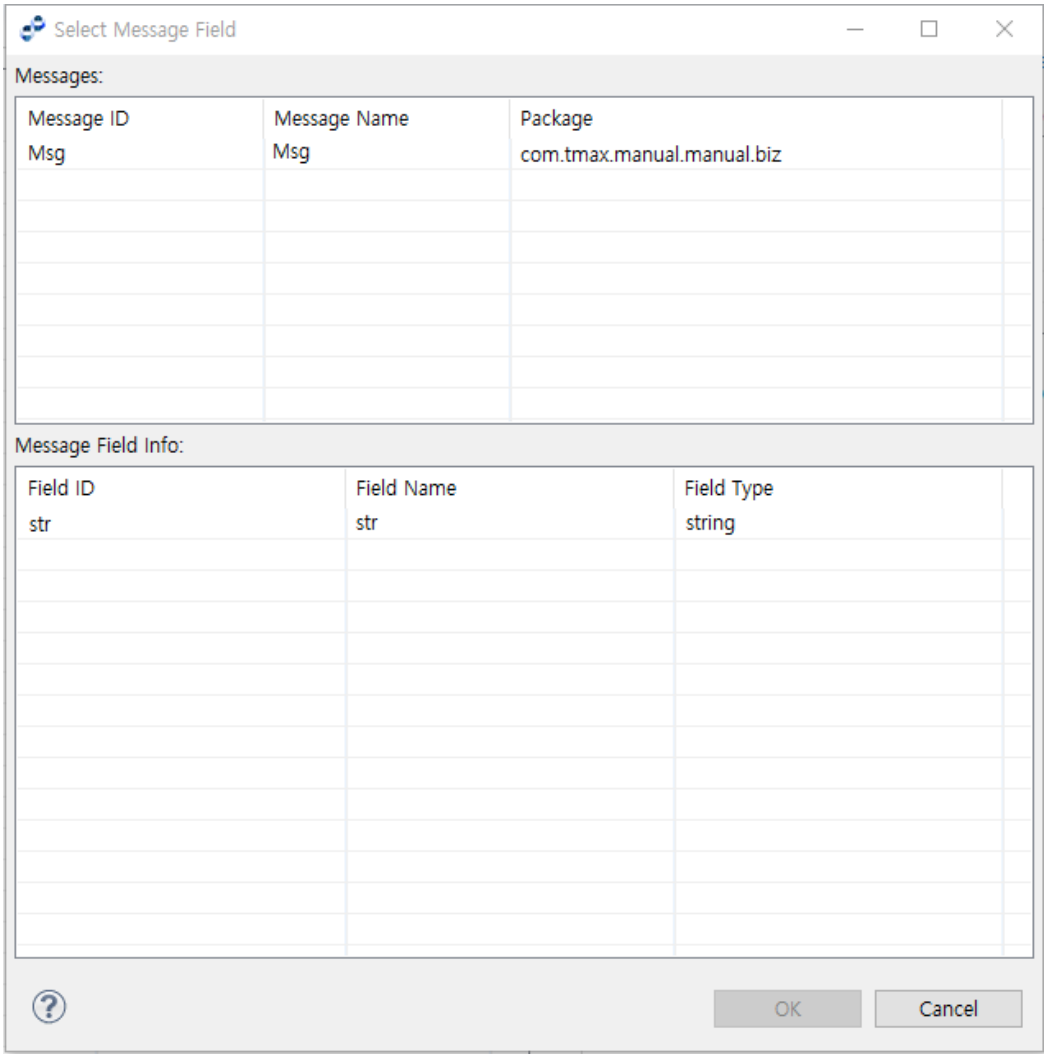


A.1.2. Select Message Field

The Select Message Field dialog box is used to select a field of the selected message.

All messages in the current project are listed in the **Messages** list box. Select a message to display its fields in the **Message Field Info** list. Select a field, and then click **[OK]**.

[Figure A.2] Select Message Field Dialog Box



A.2. Mapping Dialog Boxes

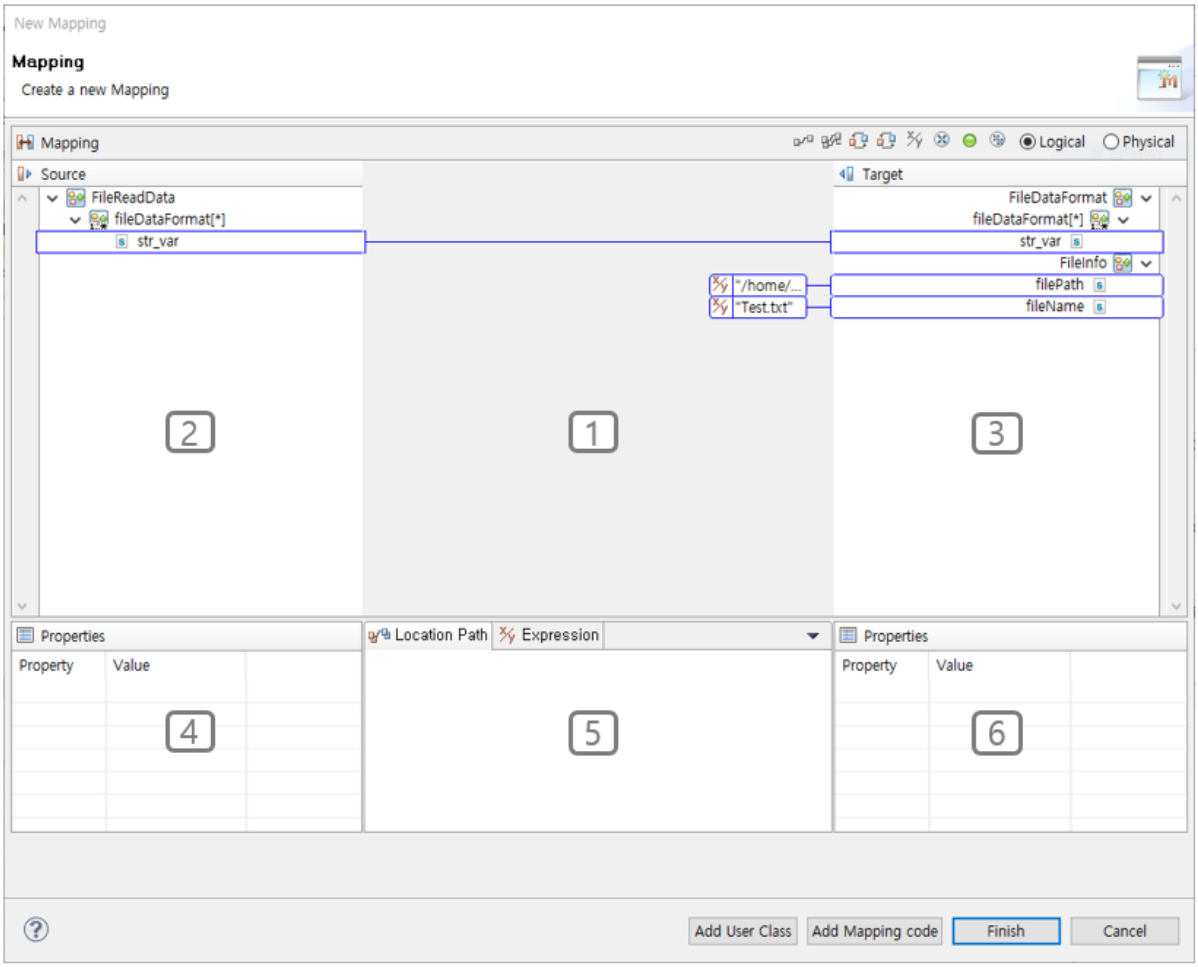
A mapping defines a source-to-target relationship used to transform messages. Mapping is frequently used in AnyLink, and it is usually created through a wizard.

AnyLink provides various adapters to establish and navigate links between resources. An adapter transforms the resource format to a message format that can be used in AnyLink by using a mapping that is defined through the **Mapping Wizard**. The same **Mapping Wizard** is used for all resource types and in all sections, such as adapter or flow, where mapping is used.

A.2.1. Mapping Wizard

The following describes the **New Mapping** dialog box of the Mapping Wizard. The wizard is divided into the mapping area (1, 2, 3) and properties and editing area (4, 5, 6).

[Figure A.3] New Mapping Dialog Box



Mapping Area

The mapping area is divided into the source tree (2), mapping line (1), and target tree (3) areas.

The following describes the toolbar icons and buttons on the top of the mapping area.

Item	Description
 (Create New Mapping)  (Create New Mappings)	Create a new mapping.
 (Auto-Generate Mappings/ Auto-Generate Mappings with Physical Name)	Automatically maps nodes with same names. If there are multiples nodes with the same name in the source or target tree, the first node that appear in the tree is mapped. Hence, the user must check auto-generated mappings to ensure that they are mapped correctly. **[Auto-Generate Mappings with Physical Name]** automatically maps using physical names instead of the logical names that are shown in the tree.
 (Create New Expression)	Create new function or arithmetic expression for the selected node in the tree. If a mapping already exists for the selected target, the **[Create New Expression]** menu is disabled. The target elements that are involved in a function mapping is shown inside a rounded rectangle. A function is shown on the mapping line with a rounded rectangle function icon. When a function mapping is created, it is automatically shown as selected.
 (Delete Mapping)	Delete mapping and expression.
 (Show Painted Tree)	Highlight the mapped fields in purple.
 (Delete All Mappings)	Delete all mappings.
Logical	Display data as logical names in the source and target areas.
Physical	Display data as physical names in the source and target areas.
Add User Class	Open the dialog box to search for a user class that defines the mapping logic.
Add Mapping Code	Open the dialog box to edit the Java mapping code.

Mapping lines, which represent mapping relationships, are shown in blue and currently selected line is shown in red. If the mapped nodes are visible in the source and target trees, they are connected with a solid line. If they are hidden (collapsed), they are connected with a dotted line.

Configuration Area

The configuration area is divided into the source properties pane (4), editing pane (5), and target properties pane (6).

- **[Location Path]**

  Tab with a list of mapping paths. This tab is read-only.

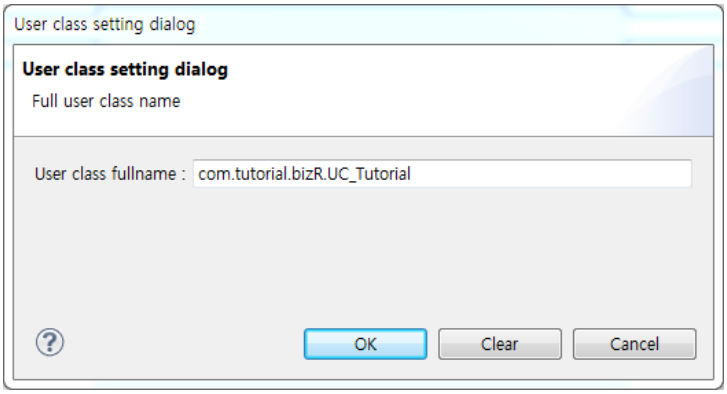
- **[Expression]**

  Tab with function mapping. Constant values can be edited.

- **[Add User Class]**

  Click **[Add User Class]** to open the following dialog box to add a user class. A user class is used to implement more complex logic that cannot be created in the editing area with **[Add New Function]**.

  [Figure A.4] User Class Properties Dialog Box



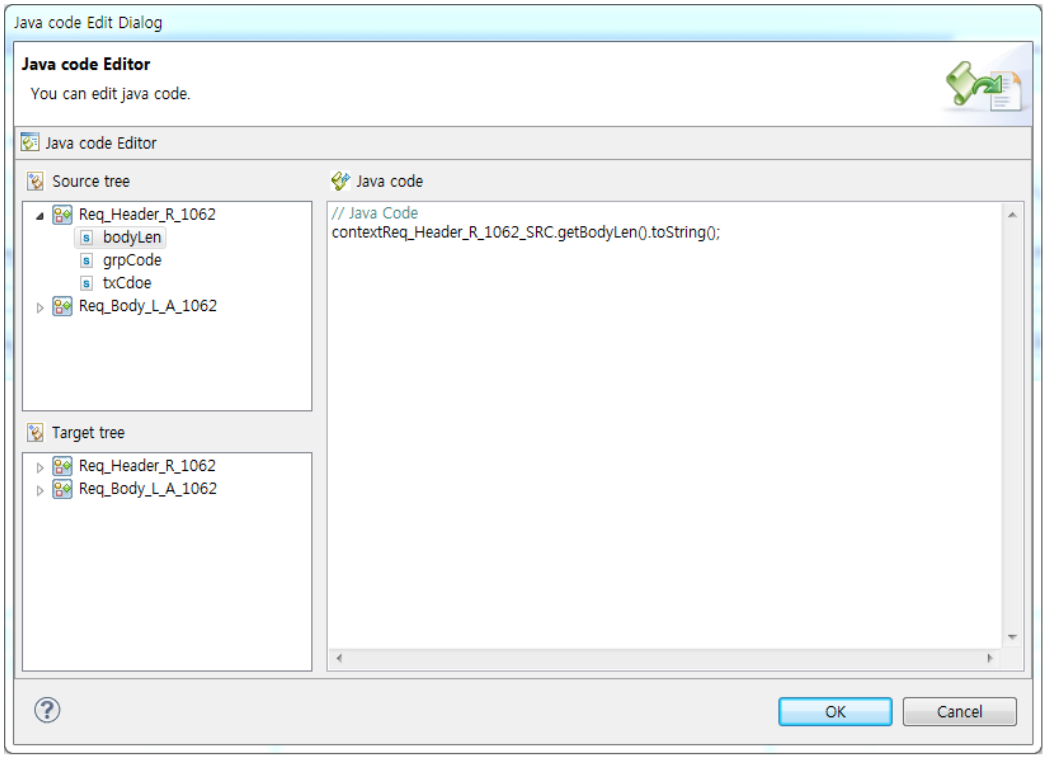
  Enter the full user class name including the package name in the **'Full User Class Name'** textbox. If the user class does not exist on the server, use **[Upload]** to upload a new library to the server. Otherwise, click **[OK]** to add the selected user class or **[Clear]** to clear the entry. To add a simple function to the mapping, additional code can be added to the mapping.

- **[Add Mapping Code]**

  Click **[Add Mapping Code]** to open the following dialog box to add Java code to the mapping.

  The Java code is executed after executing the mapping. Drag and drop the variable values from the source and target trees. The code should be written with care since the compilation result is not shown immediately. Various functions are provided in the context menu of the source and target trees.

[Figure A.5] Java Code Editor Dialog Box



The size of each pane can be adjusted by placing the cursor on the border and dragging it. Double click on the tab of each pane to maximize or reset it.

The following describes each pane.

Pane	Description
Source Tree (Target Tree)	Same as a general tree control. Use the mouse to select a node. Use <+> and <-> keys to collapse and expand the tree. Each item in the source or target tree is shown with an icon that represents the item and the object count limit.
Java Code	Source/target properties. Properties vary according to the resource type of each node.

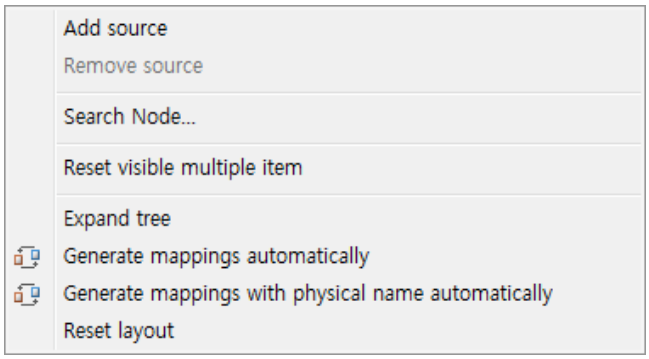
A.2.2. Using Mapping Wizard

The Mapping Wizard can be used to map one or more sources to one or more targets. The source and target trees show only the basic mapping items and not all possible items. The user can add or delete mapping items as needed.

Source Tree

The context menu of the source tree provides various functions.

[Figure A.6] Source Tree Context Menu

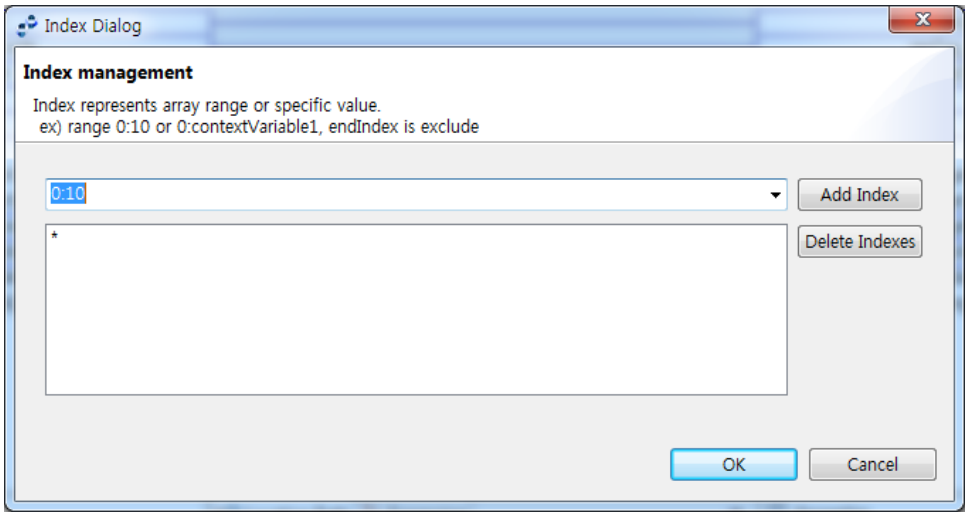


A source item can have child items in the tree. To hide a source tree, click **[Remove Source]** from the context menu or click **[Delete]**. If a source tree is hidden from the list, mappings defined for the deleted source item or its child items are automatically deleted.

Menu	Description
[Add Source]	Open the **Source Selection** dialog box for selecting source items that are not shown in the source tree. The **Source Selection** dialog box displays all items that can be added to the source tree. To select an item, click on the item or enter the item name. Use the <Shift> key to select multiple items. Double click on the selected item(s) or click **[OK]** to add them to the source tree.
[Remove Source]	Remove an item. Only enabled for removable items.
[Search Node]	Open the **Search Node** dialog box to find a node by name. Enter the search keyword in the textbox to find all node names that contain the keyword. Select the '**Regular Expression**' checkbox to use regular expression search. The search results are displayed and the nodes that are in the results are selected in the source tree.
[Reset Array Range]	Enabled for array type item. Opens the Manage Indexes dialog box to adjust the array range for mapping.
[Expand Tree]	Expand the tree to show all child items.
[Auto-Generate Mapping]	Generate mapping automatically.

Menu	Description
[Auto-Generate Mappings with Physical Name]	Generate mappings with physical name automatically.
[Reset Layout]	Reset the layout to its original size. To adjust the pane size, place the cursor on the border which changes the cursor to a pointer, and then drag the pointer.

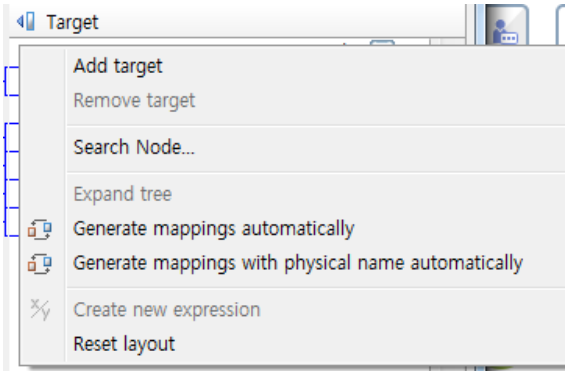
[Figure A.7] Manage Indexes Dialog Box



Target Tree

The context menu of the target tree provides various functions.

[Figure A.8] Target Tree Context Menu



Menu	Description
[Create New Expression]	Create a new function or arithmetic expression for the selected node in the tree.

Note

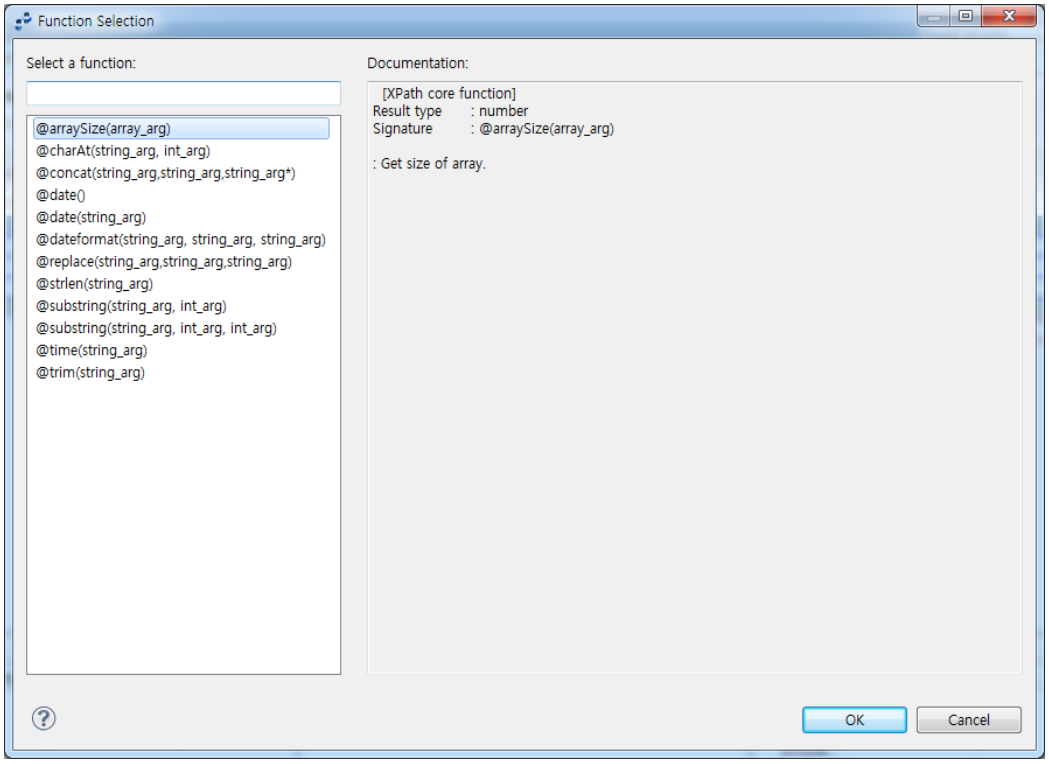
For information about other context menu items, refer to the source tree context menu description.

Edit

After creating a function mapping, the function must be edited in the editing pane.

The following is the context menu of the editing pane.

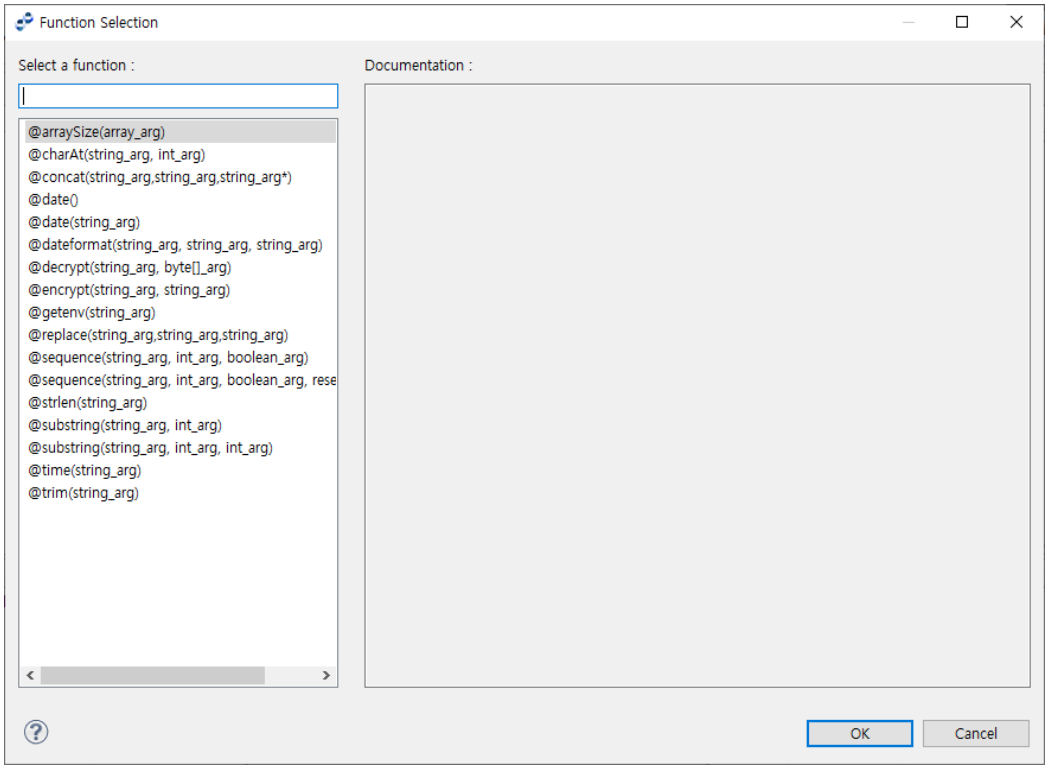
[Figure A.9] Edit Context Menu



- **[Add New Function]**

  To add a new function in the **[Expression]** tab, click **[Add New Function]**.

[Figure A.10] Select Function



Drag and drop a node object from the source tree to the function editing pane. Use this method to easily enter function parameters.

- Returns ArrayField size

```
@arraySize(arrayField)
```

- Returns String.charAt(i)

```
@charAt(String, index)
```

- Returns the result of concatenating specified strings

```
@concat(stringField1, stringField2, .....)
```

- Returns current date (String)

```
@date()
```

- Returns current date in the given format

```
@date("yyyyMMdd");
```

- Returns the given date in the given format

```
dateformat(stringField, "yyyyMMdd", "yyyy-MM-dd")
dateformat(stringField, "HHmmss", "HH:mm:ss")
```

- Decrypts and returns a string (String)

```
@decrypt(stringField, stringValue);
```

- Encrypts and returns a string (byte[])

```
@encrypt(stringField, byteArray);
```

- Returns AnyLink system variable or BizTx information (String)

```
@getEnv(stringKey);
```

- Returns the result of replacing the given string with the given pattern

```
@replace(stringField, "target", "replacement")
```

- Saves the sequence value to the system and increase the value by 1.

```
@sequence(stringField, int length, boolean padding)
```

– Saves the sequence value to the system and increase the value by 1. resetType determines whether to reset the value.

```
@sequence(stringField, int length, boolean padding, ResetType resetType)
```

– Returns the length of a given string

```
@strlen(stringField)
```

– Returns substring starting from the given index

```
@substring(stringField, int beginIndex)
```

– Returns substring in the range of the given indexes

```
@substring(stringField, int beginIndex, int endIndex)
```

– Returns current time in the given format

```
@time("HHmmdd")
```

– Returns the result of trimming the given string

```
@trim(stringField)
```

Note

For a more complex logic, click **[Add User Class]** to add a user class that implements the desired logic.

- **[Add Variable Data]**

  To add a variable value in the source code to the **[Expression]** tab, drag the variable from the source code to the **[Expression]** tab or click **[Add Variable Data]** from the context menu.

Mapping Lines

To create a mapping in general, either select a source item and a target item and then click **[Create New Mapping]** from the toolbar or drag and drop the source item on top of the target item.

The source and target items must be compatible types, and the target must not be defined in another mapping. Otherwise, **[Create New Mapping]** and drag and drop functionality are disabled.

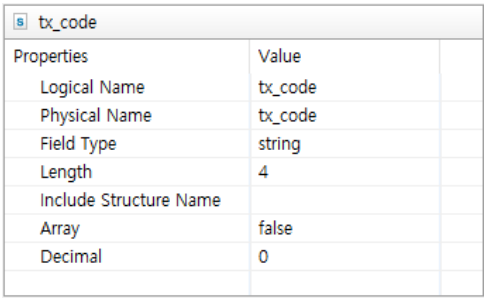
Click on a mapping line to select the mapping. For a function or constant mapping, click the inside of the rectangle that represent the mapping type to select the mapping. Hold down the <Ctrl> or <Shift> key to select multiple mappings.

If a single constant mapping is selected, the constant mapping tab is enabled. If a single function mapping is selected, the function mapping tab is enabled. Otherwise, the Location Path tab that shows the mapping paths is enabled.

Tree Properties

The source and target properties panes display the properties of the selected items in the source and target tree, respectively. The following is an example of the properties pane.

[Figure A.11] Properties Pane



A.3. Resource Dialog Boxes

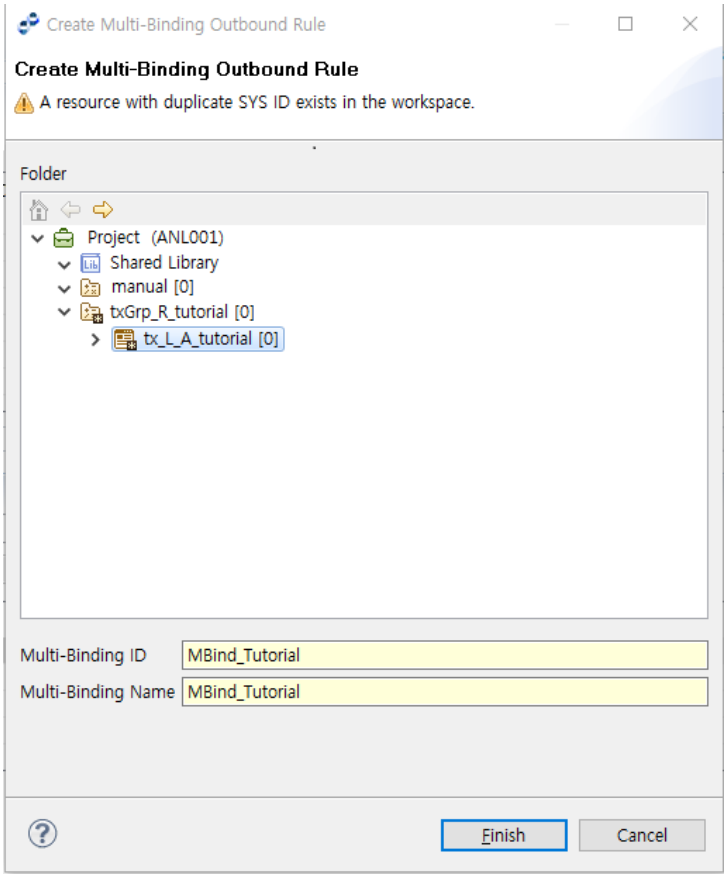
This section describes resource related dialog boxes.

A.3.1. Create Resource

The Create Resource dialog box is used to create a resource in a Studio project.

Select a BizTx/BizTx Group, enter the **[Resource Name]** and **[Resource ID]**, and then click **[Finish]** to add the resource to the current node. The configuration options and allowed ID and name are different for each resource type.

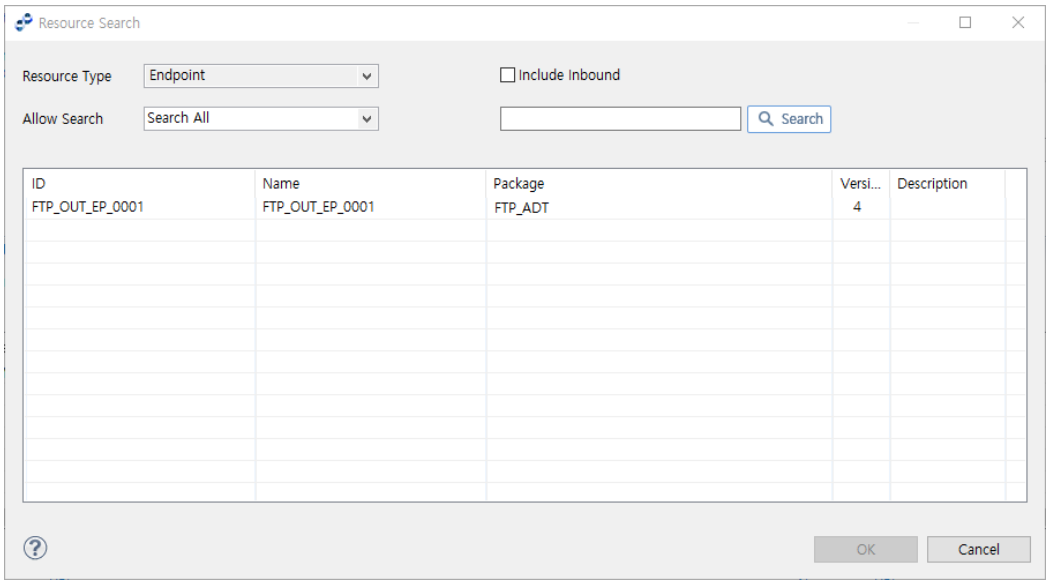
[Figure A.12] Create Resource Dialog Box



A.3.2. Search Resource

The Search Resource dialog box is used to select a resource for a field. To get a resource created in WebAdmin, the user must log in as the owner of the resource in WebAdmin. Select the desired resource and click **[OK]** to enter the resource in the field.

[Figure A.13] Search Resource Dialog Box



Item	Description
Resource Type	Resource type to search. Any AnyLink resource type, such as endpoint, message, or flow, can be selected.
Allow Search	Search keyword type (Name, ID, or Package). Enter the keyword in the textbox, and then click **[Search]** to execute the search.
Resource List	List of resources retrieved by the search.

Index

Symbols

[Edit] submenus
- [Add Bookmark], 8
- [Add Task], 8
- [Copy], 8
- [Cut], 8
- [Delete], 8
- [Exit], 7
- [Find/Replace], 8
- [Paste], 8
- [Redo], 8
- [Select All], 8

[File] submenus
- [Close All], 6
- [Close], 6
- [Convert Line Delimiters To], 7
- [Exit], 7
- [Export], 7
- [Import], 7
- [Move], 7
- [New], 6
- [Open File], 6
- [Print], 7
- [Properties], 7
- [Recent Files], 7
- [Refresh], 7
- [Rename], 7
- [Restart], 7
- [Revert], 7
- [Save All], 7
- [Save As], 6
- [Save], 6
- [Switch Workspace], 7

[Help] submenus
- [About Tmax AnyLink Studio], 12
- [Cheat Sheets], 12
- [Check for Updates], 12
- [Dynamic Help], 12
- [Help Contents], 12
- [Install New Software], 12
- [Key Assist], 12
- [Search], 12

[Navigate] submenus
- [Back], 9
- [Forward], 9
- [Go Into], 8
- [Go To], 8
- [Last Edit Location], 9
- [Next], 8
- [Open from Clipboard], 8
- [Previous], 9
- [Show In], 8

[Project] submenus
- [Build All], 9
- [Build Automatically], 10
- [Build Project], 9
- [Build Working Set], 10
- [Clean], 10
- [Close Project], 9
- [Open Project], 9
- [Properties], 10

[Run] submenus
- [External Tools Configurations], 10
- [Organize Favorites], 10
- [Run As], 10

[Search] submenus
- [File], 9
- [Search], 9
- [Text], 9

[Window] submenus
- [Close All Perspectives], 11
- [Close Perspective], 11
- [Customize Perspective], 10
- [Hide Toolbar], 10
- [Navigation], 11
- [New Editor], 10
- [New Window], 10
- [Open Perspective], 10
- [Preferences], 11

[Reset Perspective], 11
[Save Perspective As], 11
[Show View], 10

A

Activity
Configuring an Activity, 100
Activity Error Handler, 150
Activity Handler, 149
Activity Preference
Activity Error Handler, 111
Activity Preferences
Activity Handler, 109
Activity Loop, 113
Basic Info, 101
Outbound Rule (Service), 103
Parameter Settings, 107
Skip Condition, 105
Adapter Message Handler, 152
Administrator Handler, 161
AnyLink Studio initial screen
Editor Panel, 6
Menu, 5
Navigator, 6
Perspective, 6
Title bar, 6
Tool bar, 5
View, 6

B

Batch Outbound Rule, 66
Configuring a Batch Outbound Rule, 68
Creating a Batch Outbound Rule, 66
Bitmap, 45
BizTx Invocation Service
Flow, 23
Outbound Rule, 23
BizTx Invocation Service(BizTx)
Multi-binding, 25
BizTx priority, 34
BizTx/BizTx Group Editor, 21
BizTx/BizTx Group Editor, 21
BizTx/BizTx Group Info, 21
BizTx/BizTx Group Options, 32
creating a BizTx/BizTx Group, 19
Parsing Info, 36
parsing options, 38
XML Viewer, 40
BizTx/BizTx Group Info
Basic Info, 22
BizTx Invocation Service (BizTx), 23
Error Response Handling, 28
Request/Normal Response/Business Error Response Message, 26
BizTx/BizTx Group Options, 32
block, 129
BPMN, 16
Bypass, 22

C

Child BizTx Identification, 36
Common BizTx/BizTx Group, 165
Common Business Project, 163
Creating a Common BizTx/BizTx Group, 165
Creating a Common Business Project, 163
configuration items for each message type, 50
Configuring a Batch Outbound Rule, 68
File List, 70
Outbound Rule Definition, 69
Request Message, 69
Response Message, 69
Style Options, 73
Configuring a Custom Log Outbound Rule, 56
Define Outbound Rule, 56
Log Table Setting, 57
Request Message, 57
Configuring a Multi-Binding Rule, 90
Configuring an Activity
Activity Preferences, 101
Activity Preferences Menu, 100
Creating a Batch Outbound Rule, 66
creating a BizTx/BizTx Group, 19
creating a BizTx/BizTx Group, 20
creating the root BizTx Group, 19

Creating a Message, 41
Creating a Multi-Binding Rule, 89
creating a project, 17
Creating a Service Flow, 95
Creating a User Class
  getEnv function, 146
  User Class, 145
  User Class Implementation, 145
CSV, 45
Custom Log Outbound Rule, 55
  Configuring a Custom Log Outbound Rule, 56
  Creating a Custom Log Outbound Rule, 55

D

DefaultActivityErrorHandler, 148
DefaultActivityHandler, 148
DefaultAdapterMessageHandler, 148
DefaultEncryptionHandler, 149
DefaultMessageValidityHandler, 149
DefaultMonitoringEventHandler, 149
DefaultMsgTransferHandler, 149
DefaultParsingHandler, 149
DefaultPartnerAddressHandler, 149
DefaultProcessErrorHandler, 148
DefaultProcessHandler, 148
DefaultRoutingHandler, 149
DefaultSsoLoginHandler, 149
DefaultUserMappingHandler, 149
DefaultUserTimer, 149
Delimeter, 45
Deployment & Undeployment, 167
  Resource Deployment (Redeploy), 167
  Resource Undeployment, 170
  Shared Library Deployment & Undeployment, 172

E

editing login data, 3
Encryption Handler, 153
Error Response Handling
  FLOW, 28
  MAPPING, 31
  NONE, 28
  PARENT, 28

F

FDL (Fixed Definition Language), 47
FixedLength, 45
Flow, 23
FML (Fixed Manipulation Language), 47

H

Handler, 147
  Creating a Handler, 147
  Handler Class Implementation, 148
Handler List, 148
Handlers
  Activity Error Handler, 150
  Activity Handler, 149
  Adapter Message Handler, 152
  Administrator Handler, 161
  Encryption Handler, 153
  Message Management Handler, 161
  Message Validation Handler, 157
  Monitoring Event Handler, 160
  Parsing Handler, 155
  Process Error Handler, 152
  Process Handler, 151
  Routing Handler, 158
  SSO Login Handler, 159
  Timer Handler, 156
  User Mapping Handler, 154

I

ISO8583, 47

J

JSON, 46

K

KeyValue, 46

L

Login Window, 2

M

Mapping, 92
  Adding/Deleting a Mapping, 92
Marshal, 41
menus, 6
  [Edit], 7
  [File], 6
  [Help], 12
  [Navigate], 8
  [Project], 9
  [Run], 10
  [Search], 9
  [Window], 10
Message, 41
Message configuration by type, 43
  Bitmap, 45
  CSV, 45
  Delimeter, 45
  FDL (Fixed Definition Language), 47
  FixedLength, 45
  FML (Fixed Manipulation Language), 47
  ISO8583, 47
  JSON, 46
  KeyValue, 46
  ValueObject, 46
  XML, 45
Message Editor, 41
  Creating a Message, 41
  message configuration by type, 43
Message Management Handler, 161
Message Validation Handler, 157
Monitoring Event Handler, 160
Multi-binding, 25, 89
Multi-binding Router, 89
Multi-Binding Rule, 89
Multi-Binding Rule Editor, 89
  Configuring a Multi-Binding Rule, 90
  Creating a Multi-Binding Rule, 89
  Mapping, 92
Multicasting, 89

N

navigator, 13
navigator context menus
  [Assign Biz System], 14
  [Compare With], 14
  [Delete], 14
  [Download], 14
  [Export], 14
  [Import Configuration], 14
  [Import], 14
  [New], 14
  [Paste], 14
  [Refresh], 14
  [Replace With], 14
  [Team], 14
  [Template], 14
  [Unassign Biz System], 14
  [Validate], 14
navigator resource display format modification, 14

O

Outbound Rule, 23, 55
Outbound Rule Editor, 55
  Batch Outbound Rule, 66
  Custom Log Outbound Rule, 55

P

Parsing Handler, 155
Parsing Info, 36
  BizTx Identifier Code, 36
  Child BizTx Identification, 36
parsing options, 38
  Encoding Type, 38
  Message Type, 39
  XML Self-Closing Setting, 39
Perspective, 5
Process Error Handler, 152
Process Handler, 151

R

Resource
  Handler, 16

Resource Deployment (Redeploy), 167
Resource Search & Download, 175
Resource Undeployment, 170
resources, 15
  BizTx/BizTx Group, 16
  Common BizTx/BizTx Group Project, 17
  External Mapping, 16
  Flow, 16
  Message, 17
  Multi-Binding, 16
  Outbound Rule, 16
  Project, 16
  User Class, 16
  WSDL, 16
Routing Handler, 158
Routing Method, 91

S

Service Flow Diagram, 95, 96
  [Palette], 97
  [Toolbar], 96
  Activity, 99
Service Flow Editor, 95
  Creating a Service Flow, 95
  Service Flow Diagram, 96
SFDL, 95
Shared Library Deployment & Undeployment, 172
SSO Login Handler, 159
Studio loading window, 1

T

Timer Handler, 156

U

Unmarshal, 41
User Class, 145
User Class/Handler Editor, 145
  Handler, 147
  User Class, 145
User Mapping Handler, 154
Utility, 130

V

ValueObject, 46
views, 12
  [Console], 12
  [Debug], 13
  [Error Log], 12
  [Outline], 12
  [Project Explorer], 12
  [Properties], 12
  [Relationships], 13

W

Workspace Launcher, 1
WSDL, 16

X

XA, 22
XML, 45
XML Viewer, 40