

WebSphere Transformation Extender



Integration Flow Designer

Version 8.1

Note

Before using this information, be sure to read the general information in "Notices" on page 83.

30 June 2006

This edition of this document applies to WebSphere Transformation Extender, 8.1 and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, e-mail DTX_doc_feedback@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 2006. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Integration Flow Designer overview	1
When to Use the Integration Flow Designer	1
Chapter 2. Using the Integration Flow Designer.	3
Chapter 3. Stages of using the Integration Flow Designer	5
Stage 1: defining systems	5
Stage 2: verifying component relationships	5
Stage 3: preparing systems to run	6
Chapter 4. Integration Flow Designer basics	7
Filename extensions	7
Starting the Integration Flow Designer.	7
To start the IFD	7
To disable the Startup window	7
Converting files from previous versions	7
System definition files	8
Launcher system files	8
Chapter 5. Integration Flow Designer user interface	9
Navigator	9
Navigator icons	9
Navigator: List view	10
Navigator: Composition view	10
Using the Navigator	10
Displaying components from the navigator.	12
Dragging components	12
System windows	12
Toolbox.	14
Menu Commands	14
Menu Bar	15
File menu	15
Edit menu.	15
View Menu (Alt+V)	16
System menu.	16
Server menu	17
Tools menu	17
Window Menu (Alt+W)	18
Help menu	19
Customizing your work environment.	19
Shortcut keys.	19
Options.	19
Finding and replacing in the Integration Flow Designer	20
Chapter 6. How the Integration Flow Designer works	25
Theory of operation	25
Map components	25
Command Server or Launcher	26
Map component execution settings	26
Systems	27
Component relationships in a system.	29
Chapter 7. System definition files	31
Systems	31

System definition files	31
System definition files and map source files	31
About relocating system definition files	31
Working with system definition files	32
To create a new system definition file.	32
To open a system definition file	32
To display the systems contained in a system definition file	32
To close a system definition file.	32
To save a system definition file.	32
To create a system	32
To copy a system to another system	33
To rename a system	33
To delete a system	33
Importing and exporting system definitions	33
Exporting system definitions	33
Importing system definitions	34
Import and export options	35
Audit log	35
Chapter 8. Map components	37
Source map components	37
Modifying a map component name	38
Compiled map components	38
Pseudo map components	38
Defining sources and targets for a pseudo map	39
Editing input and output cards.	39
Creating a source map from a pseudo map.	39
Chapter 9. Viewing sources and targets of a map component	41
Card tooltips	41
Viewing multiple tooltips.	42
Chapter 10. Working with map components	43
Map names	43
Execution settings	43
Viewing execution settings	43
Integration Flow Designer execution settings	44
MapServerLocation.	44
MapSettings	44
MapDelay	45
PendingExpiration	45
Priority setting	45
Max Concurrent Map Instances setting	45
TimeEvent.	46
Pending Instance Thresholds setting	47
Input and output settings.	47
Overriding map settings	48
Overriding card settings	49
Chapter 11. Subsystem components	51
Subsystem component name.	51
Execution settings	51
ServerName	52
ExecutionType	52
Inputs and outputs	52
Default settings	53
Saving map card positions	53
Overriding card source or target	53
Working with subsystems	53

Chapter 12. Overview of server definitions and execution modes	55
Server implementation examples	55
Server definitions	56
Working with server definitions	56
Assigning a server to a system	57
Modifying a server definition	57
Deleting a server definition	58
Importing and exporting server definitions	58
To export a server definition	58
To import a server definition	58
System execution modes	59
Chapter 13. Generating system files	61
Building maps	61
To build all maps referenced by all source map components in a system	61
To build one map referenced by a source map component in a system	61
To build a collection of maps referenced by a collection of source map components within a system	61
To view the last build results at any point in time	61
Analyzing a System	62
Generating executable system files	62
To generate a command file	62
To generate a Launcher system file	63
Unresolved settings	63
Processing unresolved settings	64
System definition file differences	64
Comparing system definition files	64
Comparing systems	65
Comparing servers	65
Comparing deploy scripts	65
Comparing components	65
Comparing compiled map components	65
Comparing source map components	65
Comparing a psuedo map component	66
Comparing subsystem components	66
Comparing document links	66
Comparing system definition files	66
Viewing and printing system definition file differences	66
Chapter 14. Links	69
Internal links	69
External links	69
Pseudo links	70
Using a pseudo link	71
Doc links	71
Using doc links	71
Overriding a card source or target	72
Chapter 15. Deploying systems	73
Using deploy scripts	73
To access deploy scripts	73
To configure a script command	73
To add a deploy script	73
To remove a deploy script	73
To rename a deploy script	74
Build and transfer maps	74
Generate and transfer files	76
Transfer additional files	77
Running a deploy script	78
To run a deploy script	78
To view deploy script results	78

Chapter 16. Return codes and error messages	79
Return codes and error messages	79
System analysis structure error messages	79
Systems using wildcards analysis structure error messages	80
Troubleshooting tips for error messages	81
"Not enough memory to execute" error message	81
"Version Mismatch" error message.	81
Notices	83
Programming interface information	85
Trademarks and service marks	85
Index	87

Chapter 1. Integration Flow Designer overview

The Integration Flow Designer is a component of the Design Studio that provides a graphical facility to combine collections of maps and run them as a single unit.

The Integration Flow Designer has an easy-to-use graphical interface. You can create system diagrams, what-if scenarios, generate process control information, and see how the systems will behave at execution time. Additionally, the automation of the Integration Flow Designer helps reduce the number of errors that often result from manual operations.

The term for a collection of maps is *system*. Using the Integration Flow Designer, you create a system file by combining map components and then saving them as a single file known as a system definition file (.msd).

The Integration Flow Designer provides both local and client functionality. Use the Command Server or Launcher to execute a system's process control information on both Windows and UNIX operating systems.

When to Use the Integration Flow Designer

Use the Integration Flow Designer when you have multiple maps to manage within your enterprise. Just one click can build or port as many maps as you have defined in any system.

Note: You must use the Integration Flow Designer to generate systems if you plan to use the Launcher. The Integration Flow Designer is the client definition facility for the Launcher.

You do not need the Integration Flow Designer if you are using a Command Server. However, you will find it to be very useful as a client facility for managing maps that will be run by a Command Server. The Integration Flow Designer generates process control information for Command Servers in the form of command files. Generating these command files manually is tedious and error-prone. Using the Integration Flow Designer eliminates possible manual errors.

Chapter 2. Using the Integration Flow Designer

When you use the Integration Flow Designer to create systems from collections of maps, you will typically have maps already created and ready to use. However, using the Integration Flow Designer, you can create placeholder maps called *pseudo* maps. You can add input and output cards to these pseudo maps and create source maps as you move forward with your design.

For each system, you can select a server and choose the execution mode: **Launcher** or **Command**. If you are using the Launcher, you must generate a Launcher system file (.msl). The .msl file is the only file that the Launcher recognizes. If you are using the Command Server, you can generate text command files.

When you select a server for the system, the system is called an *executable system*. For each executable system, you can create and run deploy scripts that automate the process of generating and deploying a system to a different server.

After you generate an .msl file, you are ready to run the system using the Launcher.

Chapter 3. Stages of using the Integration Flow Designer

The three main stages of using the Integration Flow Designer are discussed.

Stage 1: defining systems

Use the Integration Flow Designer at design time to model what-if scenarios and visualize the results.

In the first stage, you use the Integration Flow Designer to construct system definition files. A system definition file is a model of a system of maps that you want to execute. System definition files are maintained in various system windows that you open and view. The diagrams include map and subsystem components and their execution properties. Data flow relationships are automatically derived and displayed. The system definitions are stored in system definition files (.msd) based on how you decide to organize them. The Navigator helps you to graphically interact with system definition files and the various systems that they contain.

The Integration Flow Designer includes a Doc Links feature that helps you document your design. You can link any text or Windows-registered document type to both map and system components. Then you can view and edit this document by referencing it through the system component to which it is linked. For example, you can link a Microsoft Word document to a map component that specifies design information for this map. You can link a document from a preferred project management tool to a map or a system to assist in managing the project plan. In addition to Doc Links, the Integration Flow Designer enables you to enter free-form text directly to the system definition file.

While you are defining systems, you can use the Integration Flow Designer to open the Map Designer and/or the Database Interface Designer using a map you select in a system. You can also open the Type Designer using a type tree associated with a map input or output card that you select.

Stage 2: verifying component relationships

The Integration Flow Designer displays a system graphically, which allows you to visualize your designs and verify that relationships between components are configured properly.

You can view the contents of maps and subsystems by double-clicking the icons in the system window. This allows you to view the details associated with a component's sources or targets. Visual cues, such as tool tips for specific sources and targets, help you to see key pieces of setup information. Displaying information graphically reduces the need to hunt for information buried within a system definition; however, a find-and-replace operation can also be used.

Stage 3: preparing systems to run

After you design your systems and verify component relationships, you use the Integration Flow Designer to prepare them to run. The Integration Flow Designer assists you in this process by building maps and porting them to designated servers. You can easily build and port entire systems, or specific groups of maps or individual maps.

The Integration Flow Designer also generates process control information based on the execution mode you specify for a system and the execution settings you have specified for the various components within the system. The Integration Flow Designer produces command files for systems that will be run by a Command Server and Launcher system files (.msl) for systems that will be run by an Launcher.

The Integration Flow Designer includes an Analyzer that checks your system definitions for logical consistency to ensure they can be executed. The Analyzer detects any inconsistencies you may have introduced in the definition while experimenting with system models.

Systems that will be run by a Launcher can be designed to run independently on one or more servers or as a distributed system across multiple servers. Event coordination among multiple servers is analyzed at design time before deployment. During execution, data from one system component to another can flow seamlessly without the need for a global controller.

Chapter 4. Integration Flow Designer basics

Filename extensions

The Integration Flow Designer views a map as an indivisible unit recognizing both source and compiled maps. For design purposes, it also enables you to define placeholders for maps that have not yet been implemented (pseudo maps). The Integration Flow Designer generates or uses the following files:

Filename Extension	Description
--------------------	-------------

- | | |
|-------------|---|
| .msd | System definition file generated by the Integration Flow Designer |
| .msl | Launcher system file, a system definition file generated for the Launcher |
| .mmc | Compiled map generated by the Map Designer |
| .mtt | Type tree file generated by the Type Designer |

Note: Pseudo maps and external systems do not have filename extensions.

Starting the Integration Flow Designer

This procedure assumes that you have installed the Design Studio.

To start the IFD

Select Integration Flow Designer from the **Design Studio** program menu.

The Startup window is displayed with the following options:

Open an existing system definition file

Select this option to browse for a specific .msd file.

Create a new system definition file

Select this option to create a new .msd file.

Open a recently used system definition file

Select this option to choose one or more .msd files from the displayed file list. You can also double-click on a file from this list to open it.

To disable the Startup window

To disable the Startup window, enable the **Do not show this at startup** check box. You can always access this dialog from the **Help** menu by choosing **Startup Window**.

Converting files from previous versions

When upgrading your version of the Design Studio, some file conversions must take place. Always create backups of your files before converting to a new version.

System definition files

System definition files (.msd) from previous versions are upwardly compatible with newer versions of the Integration Flow Designer. When you open an .msd file that is older than the Integration Flow Designer version you are using, the IFD automatically attempts to convert the file to the current version upon opening.

System definition files are not necessarily downward compatible; you may not be able to use a system definition file with an older version of the Integration Flow Designer. Any system definition file that is opened and saved with this version of the Integration Flow Designer cannot be used with previous versions. When the new version of the system definition file is saved, the older version is saved with a file name extension of .osd (original system definition).

Launcher system files

All existing systems must be regenerated for the current version of the software that you are using. Launcher system files (.msl) generated by previous versions of the Integration Flow Designer cannot be executed on the current version of the Launcher. You must use the same version for both the Integration Flow Designer and the Launcher or the software will not work as intended.

Things you should know about opening files of different versions:

- Make a backup copy of all type tree, map source, database/query, and system definition files before opening them.
- Opening a map source file in the Map Designer will *automatically* update the format of the type trees referenced by that map. Backup files for those type trees will be created with an .omt extension.
- Opening a system definition file (.msd) in the Integration Flow Designer will automatically attempt to update the format of the file if it is from an older version. If the .msd was created for an older version, you are prompted upon opening the file to update map source files and type trees referenced by all systems in that file. If you do not want an automatic update, you can replace map and type tree paths before requesting an automatic update to occur. Backup files for type trees have the .omt extension. Backup files for maps have the .oms extension.

Chapter 5. Integration Flow Designer user interface

The Integration Flow Designer interface includes a navigation pane (Navigator), floating toolbox, and main window for each open system. The menu bar and toolbar are customizable.

Navigator

The Navigator is the navigation pane for the Integration Flow Designer. The Navigator displays a list of open system definition files (.msd) and the systems they contain.

Generally, the Navigator is displayed automatically docked on the left side of the Integration Flow Designer window when you start the application. However, if it is not open or if you need to reposition it, proceed with the following steps:

To view and position the Navigator:

1. From the **View** menu, choose Navigator.
The Navigator is displayed.
2. To reposition the Navigator, do one of the following:
 - Double-click the top border of the Navigator (gripper) to dock or float the navigation pane.
 - Right-click on the gripper to view a context menu for positioning the Navigator.

There are two tabs that represent the **List** view and the **Composition** view.

Navigator icons

In addition to the alphabetical and compositional structure showing the contents of each system file, information about entries in the Navigator is conveyed visually with icons.

The following table lists and briefly describes some of the icons that appear in the Navigator.

Icon	Description
------	-------------



system definition file (.msd)



system or subsystem (relative to positioning)



system component



log file



document link



deployment definition

Navigator: List view

The List view displays open system definition files. This hierarchy includes:

- Defined Servers
- Systems
- Subsystems
- Components within a system
- Cards within a map component
- Doc Links (within a map component)
- Log Files (within a map component)
- Deploy Scripts

A subsystem is displayed under the system that contains it, a map component is displayed under the system that contains it, and so forth. Components at the same level are listed alphabetically.

Click any item to display its components in the system window. Right-click on any icon to view a context menu for that component.

Navigator: Composition view

The Composition view presents open system definition files sorted alphabetically. Each system definition file can be expanded to present all executable systems that it contains. An *executable system* is one that has an assigned server. Systems that do not have an assigned server do not appear in Composition view.

Click any system to display its components in the system window. Right-click on any icon to view a context menu for that component.

Using the Navigator

Use the Navigator to open and interact with one or more system definition files. To view the systems in the Navigator, expand or compress the node using + or -.

Once a system definition file is selected, you can use **File** menu commands, toolbar tools, or a context menu to close, save or copy the file.

Drag-and-drop functionality

The following drag-and-drop operations are available in the Integration Flow Designer.

- Servers can be copied/moved across system files or can be copied within the same system file.
- Components can be copied/moved across system files or copied within the same system file.
- Systems can be copied/moved across system files or copied within the same system.
- Doc Links can be copied/moved across system files or within the same system file.

Clipboard functionality in the Integration Flow Designer

- Servers can be copied into the clipboard and/or pasted across system files and within system files.
- Components (systems or maps) can be copied and/or pasted across system files and within the same system file.
- Doc Links can be copied and/or pasted across system files and within the same system file.

System and subsystem components

- The symbols next to the component names in the Navigator indicate the type of component. Even though both systems and subsystems have the same symbol, it is their position in the Navigator that determines the reference. For example, system symbols always appear directly under a system definition file symbol. A subsystem symbol can appear anywhere below a system symbol. Right-clicking on these symbols yields a different context menu depending on where the symbol is located.
- A question mark is displayed on the subsystem component icon when unresolved references exist. An unresolved reference is when the subsystem references a system that does not exist.

Focus options

The following focus options are available when using the Integration Flow Designer.

- When you double-click a component in the Navigator, it selects the component, opens the system window containing that component (if it is not already open), and refocuses the system window on the selected component.
- Place the cursor on a system, subsystem, or map component symbol in the Navigator and double-click or press Enter to synchronize the focus of the Navigator and the active system window.

Resizing/moving options

- Use the Navigator gripper to dock or float the Navigator.
- Right-click on a non-client area in the Navigator to view a context menu that enables options to undocked, float, or hide the Navigator window.
- To re-dock the Navigator, drag-and-drop the Navigator to the left or right edge of the main window of the Integration Flow Designer.
- The Navigator can be sized horizontally when docked, or it can be sized as a normal window when undocked.

Displaying components from the navigator

From the Navigator you can display a specific system, subsystem component or map component. When you double-click a component in the Navigator, the corresponding system window opens and becomes the active window.

Dragging components

In the Integration Flow Designer, you can perform drag-and-drop operations on servers, components, and document links within the Navigator. To move items, simply click-and-hold and drag the item to the desired location. To copy an item, you must also press CTRL while dragging the item.

You can only move or copy items to items of similar nodes.

System windows

The Integration Flow Designer main window displays any open system windows. Each system window is composed of a work area, a title bar, drop-down lists to specify the system server and execution mode, and a platform indicator displaying the type of platform associated with the assigned server. Specify the server and type of platform in the Add Server or Edit Server dialog by selecting **New** or **Edit** from the **Server** menu.

Multiple system windows can be open at the same time so you can view and work on several system definition files. The size and position of each system window is saved in its system definition file; when reopened, each system window is presented in its last state.

System Window Element Description

Title bar

The title bar of each window, as shown in the preceding image, displays the system name and the path of the containing system definition file (.msd).

Because you can have multiple system definition files open concurrently, a system window's title bar helps you to identify which system is being viewed. The same system name can be used throughout the various system definition files that you have open. The combination of the system name and the system definition file that contains the system is unique.

Work area

The work area is the main viewing area of the system window and is where you construct the system definition file, which is the graphical representation of the components that compose the system. It can also show links that represent data flow between the components. Components can be source maps, compiled maps, pseudo maps, or other systems (also known as subsystems).

Property Bar

The property bar is displayed below the title bar to show the assigned **Server**, the **Execution Mode**, and the **Platform** of the assigned server.

Server From the **Server** field, you can select a server on which the system will run

from drop-down list. This drop-down list contains the list of default servers as well as each server that is defined within the currently selected system definition file

Execution Mode

Assign the execution mode to define the type of Command or Launcher that will run the system. The **Execution Mode** drop-down list includes **Command** and **Launcher**. Selecting **Command** specifies that this system will be run using a Command Server. Selecting **Event Server** specifies that this system will be run using a Launcher.

Platform

This field shows the platform type associated with the assigned server. Specify the platform type in the Add Server or Edit Server dialogs which appear by selecting **New** or **Edit** from the **Server** menu.

Working with system windows

Work with system windows by using the **System** menu and the Navigator. The **System** menu commands operate on systems contained within the system definition file that is currently selected in the Navigator. The Navigator always shows you which item is currently selected regardless of operations that you may be performing throughout the Integration Flow Designer.

Using the Window Menu:

The IFD includes a **Window** menu that enables you to switch between windows that are currently open. The name of the system and its system definition file appear at the bottom of the **Window** menu. Selecting one of these items displays the corresponding system window on top of any other window that is currently in view.

To open a system window

1. In the Navigator **List** view, select the system definition file that contains the system you want to open and expand the list to view the system name.
2. Double-click the name of the system in the Navigator.
The system window opens showing the current state of the system definition file, the server assignment, and the execution mode.

Splitting windows

You can split the view in a system window to view two or four areas of a system file simultaneously.

To split a system window:

1. Place the cursor over the split box (in the upper right corner or lower left corner of the system window). The cursor changes to a split pointer.
2. Drag the divider to a new location.
3. Use the scroll bars to reposition the view to include the components you want to view.

You can also repeat the procedure to split the window again so that both horizontal and vertical splitters are displayed, or use the **Split** command.

4. From the **Window** menu, choose **Split**.
The cursor turns into a crosshair icon and vertical and horizontal lines appear.
5. Position the crosshair icon and click the mouse button.

All actions performed within a split window apply to the active system.

To remove a split from a window:

Double-click the separator (or the separator junction).

The system window displays as a contiguous space (a single window).

Toolbox

When enabled from the **View** menu, the toolbox is displayed automatically when a system window is opened. It provides the tools needed to create or modify a system definition file.

The toolbox location is saved when the Integration Flow Designer is closed. The next time you open the Integration Flow Designer, the toolbox is displayed at the previously saved location.

To use each tool, you must click the button and then click in the system window. Each tool button is described in the following table.

Tool Name	Use This Tool To...
Select	Draw a box around one or more components in the system definition file to select them. You can then perform operations on the selected group.
Add Text	Add a text block to your system definition diagram.
Override	Override sources and targets. The source or target of the first card selected will override the source or target of the second card selected.
Add Pseudo Link	Draw links between system components. Pseudo links serve a documentation purpose and are represented by a red dotted line.
Add Source Map	Add source map components.
Add Compiled Map	Add compiled map components.
Add Pseudo Map	Add pseudo map components.
Add Subsystem	Add subsystem components.
Add External System	Add an external system placeholder.

Menu Commands

You can perform actions in the IFD through menu commands, tools, and shortcut keys. Not all menu commands have tool icons and not all tool icons are represented in the command menus. When working with systems, you can activate commands several ways:

- Select functionality from the menu bar
- Click the tools on the toolbar
- Right-click an icon in the Navigator
- Double-click objects in the Navigator
- Right-click in the system window or component in the system window
- Click the execution settings button on any map icon in the system window

Note: IFD commands are available as listed above; however, most procedural information in this guide uses the menu access as a default method. Please use the access method most convenient for you.

Menu Bar

The IFD menu bar provides the common menu structure for commands generally used with Windows programs as well as commands specific to the IFD.

File menu

The **File** menu provides the options that are generally available in Windows applications.

Using the keyboard, press Alt F to view the **File** menu.

Command	Key stroke	Description
New	Alt+F, N or Ctrl+N	Creates a new system definition file and opens a system window
Open	Alt+F, O or Ctrl+O	Displays the Open dialog for opening an existing system definition file
Close	Alt+F, C	Closes the active system definition file
Save	Alt+F, S or Ctrl+S	Saves the active system definition file
Save As	Alt+F, A	Saves the active system definition file with a new file name
Source Control	Alt+F, L	Displays a submenu with Source Control options
System Definition File Differences	Alt+F, D	Begins the compare system definition file operation
Print	Alt+F, P or Ctrl+P	Prints a copy of the system definition diagram shown in the active system window
Print Preview	Alt+F, V	Displays the active system window in preview mode
Print Setup	Alt+F, R	Changes the printer and printing options
Recently Used File List	Alt+F, 1 Alt+F, 2 ...	Presents up to eight of the most recently used system definition files
Exit	Alt+F, X	Quits the application; prompts to save system files

Edit menu

The **Edit** menu provides the editing commands that are generally available in Windows applications. It also contains the **Find** and **Replace** commands, which are useful for adapting systems to new server environments.

Using the keyboard, press Alt E to view the **Edit** menu.

Command	Key Stroke	Description
Undo	Alt+E, U or Ctrl+Z	Reverses the last action, such as cut, copy, paste, and delete operations and the visual positioning of a component within the system window
Delete	Alt+E, D or Del	Deletes the selected component or components
Cut	Alt+E, T or Ctrl+X	Cuts the selection and places a copy on the clipboard
Copy	Alt+E, C or Ctrl+C	Copies the selection to the clipboard
Paste	Alt+E, P or Ctrl+V	Inserts the contents of the clipboard into the active system window
Select All	Alt+E, A or Ctrl+A	Selects all components in the active system window
Find	Alt+E, F or Ctrl+F	Displays the Find dialog for locating system component names in the Navigator
Replace	Alt+E, R or Ctrl+R	Displays the Replace dialog for replacing system component names in the Navigator.
More Find/Replace	Alt+E, M	Displays the More Find/Replace dialog for locating and editing system component names and paths in an entire system definition file

View Menu (Alt+V)

The **View** menu offers you choices of viewing specific items.

Using the keyboard, press Alt V to view the **View** menu.

Command	Key	Description
Toolbars	Alt+V, T	Displays the Customize dialog for toolbars
Status Bar	Alt+V, S	Shows or hides the status bar at the bottom of the system window
Navigator	Alt+V, N	Shows or hides the Navigator
Toolbox	Alt+V, X	Shows or hides the toolbox
Grid	Alt+V, G	Toggles the display of the grid in the active system definition diagram
Build Results	Alt+V, B	Presents the latest build results
Deploy Results	Alt+V, D	Presents the latest deploy results

System menu

The **System** menu contains commands that initiate actions for a selected system.

Using the keyboard, press Alt S to view the **System** menu.

Command	Key	Description
New	Alt+S, N	Creates a new system and displays its window
Delete	Alt+S, D	Deletes the active system.
Copy	Alt+S, C	Displays the Copy System dialog to copy the active system as a new system in the same or a different system definition file (.msd)
Rename	Alt+S, R	Displays the Rename System dialog to rename the active system
Build Maps	Alt+S, B	Builds all maps that are referenced by source map components and subsystem components contained in the active system window
Analyze	Alt+S, A	Analyzes the active system.
Generate	Alt+S, G	Generates an Launcher system file (if the system's execution mode is set to Launcher) or generates a command file for a Command Server (if the execution mode is set to Command)
Deploy	none	Displays a submenu for defining deploy scripts. Selecting Definitions from the submenu displays the Define Deploy Scripts dialog to manage deploy scripts for the active system. Selecting a specific deploy script name from the submenu runs the selected deploy script.
Unresolved Settings	Alt+S, U	Processes unresolved settings for the active system definition file

Server menu

The **Server** menu contains commands that initiate actions for servers.

Using the keyboard, press Alt R to view the **Server** menu.

Command	Key	Description
New	Alt+R, N	Displays the Add Server dialog to create a new server
Edit	Alt+R, E	Displays the Servers dialog for editing an existing server
Delete	Alt+R, D	Displays the Servers dialog for deleting an existing server

Tools menu

The **Tools** menu contains commands that initiate actions for defining Integration Flow Designer tools.

Using the keyboard, press Alt L to view the **Tools** menu.

Command	Key	Description
Short Cuts	Alt+L, S	Displays the Shortcut Keys dialog for changing the shortcut key assignments
Options	Alt+L, O	Displays the Options dialog for changing program options
Align to Grid	Ctrl+G	Aligns each selected component to its nearest respective grid intersection point

Window Menu (Alt+W)

The **Window** menu provides viewing control of the open system windows.

Using the keyboard, press Alt W to view the **Window** menu.

Command	Key	Description
New Window	Alt+W, N	Opens a new system window with the same contents as the active system window. The new system window has a similar name followed by a numeral (1 for the first duplicate, and so on).
Split	Alt+W, S	Splits the active system window into four panes, useful for working on components that are spread out. When the split grid is displayed, click your mouse to position the junction of the boundaries. After the grid is set, click and drag either the horizontal divider, or the vertical divider to a new position.
Cascade	Alt+W, C	Arranges the system windows so that they overlap in a descending pattern
Tile Horizontally	Alt+W, H	Arranges system windows as horizontal non-overlapping tiles
Tile Vertically	Alt+W, V	Arranges system windows as vertical non-overlapping tiles
Next	Alt+W, X or Ctrl+F6	Activates the next system window
Previous	Alt+W, P or Ctrl+Shift+F6	Activates the previous system window
Arrange Icons	Alt+W, A	Arranges system window icons at the bottom of the main window
Close All	Alt+W, L	Closes all open system windows
Open Windows List	Alt+W, 1 Alt+W, 2 ...	Displays a list of windows that are currently open. Select the one you want to view. A check box indicates which system window is currently active. When ten system windows are open, a More Windows command is displayed to access a dialog to activate any open system window.

Help menu

The **Help** menu offers information about the IFD version.

Using the keyboard, press Alt H to view the **Help** menu.

Customizing your work environment

You can customize the Integration Flow Designer interface to suit your preferences while working in the application. For example, if you do not want receive confirmation messages upon deletion; you have the ability to turn off that setting. Or, you can create a shortcut for a keystroke you use often.

Shortcut keys

From the Shortcut Keys dialog, you can configure shortcut keys for Integration Flow Designer functions.

Under **Select a macro** is a list of the Integration Flow Designer menu selections. When you click on an item in the list, its description and currently assigned shortcut is displayed. You can create or remove shortcuts as needed.

Refer to Design Studio Introduction documentation for more information about configuring shortcut keys.

Options

From the Options dialog, you can customize options related to viewing, saving, and confirmation messages. To access the Options dialog, select **Tools** → **Options**.

Click on each item in the list to view available options.

Maps

The Maps option specifies how cards are displayed when a map component is expanded.

Miscellaneous

The only **Miscellaneous** option available is **Backup On Save**, which is enabled by default. When enabled, the Integration Flow Designer automatically creates a backup copy of a system definition file (.msd) when it is saved and places it in the same directory.

Navigator

The Navigator option allows you to do the following:

- Select the font for the text in the Navigator
- Select the style of lines that connect components in the treeview format in the Navigator. Options include **None**, **Solid**, or **Dotted**.
- Enable or disable tooltips to appear when you mouse-over the name of a component that is not fully displayed.

Subsystem

The Subsystems option allows you to specify how cards are displayed when a subsystem component is expanded.

Confirmations

The Confirmations option allows you to enable or disable confirmation messages from appearing upon copy, move, or delete operations. All options are enabled by default, which is the recommended setting.

Import and export

The Import and Export option allows you to enable logging and set up preferences for import and export operations. See "Import and Export Options" for more information.

Finding and replacing in the Integration Flow Designer

The Integration Flow Designer supports standard search facilities. There are two find commands:

- **Find** searches for the selected name in the Navigator.
- **More Find/Replace** searches an entire system definition file.

You can perform find and replace operations in the Navigator using the **Find** and **Replace** options separately; or you can perform a complete find and replace operation in a system definition file (.msd) using the **More Find/Replace** option.

Finding and replacing component attributes

A system that you define can include a complex arrangement of components. Subsystems can contain nested subsystems, so executable maps are represented at various locations within the Navigator.

Using the More Find/Replace operation, you can easily locate and modify specific attributes associated with components defined in systems. Use the Find operation to locate card names, path names, or component names within a system; use the Replace operation to change the names. You can also use the Find and Replace operations to change specific settings across a set of components in order to adapt existing systems to new server environments.

Note: You can find and replace component names and path names, however you can only find card names. The replace operation cannot be used to replace card names.

The goal of the Find operation is to locate specific attributes associated with system components based on the search criteria that you define. For example, by using the Find and Replace operations, you can change the paths that specify where executable maps are located on target servers. From the **More/Find Replace** dialog, there are two types of Find options:

- To list only those attributes that match a specific value based on your search criteria, enter a value in the **Find What** field and click **Find**.
- To find all the attributes associated with the components in a system that match your specified search criteria, click **Find All** without entering a value into the **Find What** field.

Note: The text on the **Find All** button changes to "Find" when a value is entered in the **Find What** field.

You can change the display width of a column by moving your cursor on the border between column titles and then clicking and dragging the border to the desired width.

Find and replace operations:

To begin a find operation:

1. Select the system window from where you want to perform a find operation.
2. Select **Edit** → **Find**.
The Find dialog is displayed.
3. Enter text in the **Find What** field, and click **Find**.
4. To list only those attributes that match a specific value based on your search criteria, enter a value in the **Find What** field and click **Find**.

To begin a replace operation:

1. From the **Edit** menu, choose **Replace**.
The Replace dialog is displayed.
2. After you have an entry in the **Find what** field, enter the new name that is replacing the existing name in the **Replace with** field.
3. Click **Find Next** to begin the operation.
4. Click **Replace** or **Replace All** to replace the names.

Sorting by column:

You can sort the results listed in the More Find/Replace dialog by clicking the column headings. When you click a column heading, the list is sorted in alphabetic order. Clicking again reverses the sort order.

Using the more find and replace option

Use the More Find/Replace option to search an entire system definition file.

To find component names:

1. From the **Edit** menu, select **More Find/Replace**.
2. Enter the component name in the **Find What** field.
3. Select **Component Names** from the **Search Criteria** drop-down list.
4. Click **Find**.
Results appear under the **Component Name** heading on the **Find** tab.
5. To go directly to a resulting item, select it from the list and click **Go To**.

To find specific component names:

1. From the **Edit** menu, select **More Find/Replace**.
2. Select **Component Names** from the **Search Criteria** drop-down list.
3. Click **Settings** to specify any further search criteria you require.
4. When the **Component Name Settings** dialog is displayed, enable the components that you would like to find, and click **OK**.
5. From the Find/Replace dialog, enable **Match Case** and or **Include Referenced Systems** as optional search criteria.
6. Click **Find** to display the search results.

Results appear under the **Component Name** heading on the **Find** tab.

7. To go directly to a resulting item, select it from the list and click **Go To**.

To find card names:

1. From the **Edit** menu, select **More Find/Replace**.
2. Enter the card name in the **Find What** field.
3. Select **Card Names** from the **Search Criteria** drop-down list.
4. Click **Find**.

The default is to find all card names associated with the system. You can add further settings to find specific card names.

5. To go directly to a resulting item, select it from the list and click **Go To**. This allows you to view the component in the associated system definition diagram.

To specify settings for card names:

1. From the **Edit** menu, select **More Find/Replace**.
2. Select **Card Names** from the **Search Criteria** drop-down list.
3. Click **Settings**.

The **Card Name Settings** dialog is displayed.

4. Enable or disable the options to search for input or output cards, and click **OK**. (By default, both options are selected.)

To find path names:

1. From the **Edit** menu, select **More Find/Replace**.
2. Enter the path name in the **Find What** field.
3. Select **Path Names** from the **Search Criteria** drop-down list.
4. Click **Find**. The default is to find all path names associated with the system. You can add adjust the settings to find specific path names.

Results appear under the **Component Name** heading on the **Find** tab.

5. To go directly to a resulting item, select it from the list and click **Go To**. This allows you to view the component in the associated system definition diagram.

To find specific path names:

1. Enter the path name in the **Find What** field.
2. Select **Path Names** from the **Search Criteria** drop-down list.

The **Path Name Settings** dialog is displayed.

3. Enable or disable path name search options as desired and click **OK**.
4. From the More Find/Replace dialog, click **Find** to display the search results.

Results appear under the **Component Name** heading on the **Find** tab.

5. To go directly to a resulting item, select it from the list and click **Go To**. This allows you to view the component in the associated system definition diagram.

To replace:

1. From the **Edit** menu, select **More Find/Replace**.

The More Find/Replace dialog is displayed.

2. From the **Find** tab, begin a find operation.
3. When the results appear in the list, select the check box for each item that you want to change.
4. Go to the **Replace** tab.
5. In the **Replace** field, enter what is being replaced.

6. In the **With** field, enter the replacement text, or select a historical entry from the drop-down list.

If you do not enter a value in the **With** field, the replace operation will substitute a blank string for the string you entered in the **Replace** field.

7. Click **Find**.

The replace operation completes based on your selection criteria.

Text in a system definition diagram

You can easily add text to a system definition diagram in order to help document your design.

To add text to a system definition diagram

1. From the toolbox, click on the Add Text tool.
2. In the system window, click on the location where you would like to enter text.
3. Enter text.

You can move or delete the text by selecting it and choosing the appropriate action from the menu or toolbar.

Printing a system definition diagram

In the Integration Flow Designer you can print a snapshot of the active system window.

To print system details:

1. From the Navigator, select the system you want to print.
2. In the system window, arrange the system components to display how you would like them to print. For example, double-click map source files to expand or hide map components.
3. For best results, select **File** → **Print Preview** to view what will print, and then click **Print**. Otherwise, select **File** → **Print**.

The Print dialog is displayed.

4. Select a printer and click **OK**.

Using the grid

Components in a system diagram are positioned on a grid that can be viewed by selecting **Grid** from the **View** menu or by clicking the grid icon on the tool bar.

To view the grid:

From the toolbar, click the Grid tool to view or hide the grid.

You can also cause subsequent actions that add or move components to position the components on the nearest grid point by enabling the snap to grid mode. Enable or disable snap-to-grid mode by selecting the Snap To Grid Mode tool on the toolbar. When snap-to-grid mode is enabled, existing components are not repositioned.

To cause new components to be aligned to a grid point:

From the tool bar, click the Snap To Grid Mode tool to enable snap-to-grid mode.

When each new component is added, it will be positioned to its nearest respective grid point.

To align existing components to grid points:

1. Select the components to be aligned to the grid.
2. From the **Tools** menu, choose **Align to Grid** or press Ctrl G.
Each selected component will move to its nearest respective grid point.

Chapter 6. How the Integration Flow Designer works

When you use the Integration Flow Designer, you are creating visual process flows and from them creating executable systems that are processed by the Launcher or Command Server.

The Integration Flow Designer enables you to easily manage collections of related maps by graphically organizing them into logical collections of systems. The Integration Flow Designer turns a system into an executable file that is run by the Launcher or Command Server.

Theory of operation

You use the Type Designer and Map Designer to produce maps. A map transforms data content from source formats to target formats according to the rules you specify in the map. An executable map is stored in a source map file and built into a compiled map file. Command or Launchers run compiled maps.

The Integration Flow Designer views a map as an indivisible unit, recognizing both source and compiled maps. For design purposes, it also enables you to define placeholders, or pseudo maps, for maps that have not yet been implemented. Consider the Integration Flow Designer as one layer up from the Map Designer.

Map components

A map component is an Integration Flow Designer object that represents an executable map with a defined set of interfaces based on sources, targets, and a surrounding layer of execution settings used at run time.

There are three classes of map components.

A map component references an executable map; it does not consume it. That is, the executable map exists externally and is reusable such that another map component can also reference it. You define an executable map once and then use it as many times as you want in the Integration Flow Designer. Since the Integration Flow Designer treats every map component as a unique instance, you can define multiple map components that reference the same executable map.

The Integration Flow Designer enables you to work from the bottom up. You can roll existing executable maps up into systems. The Integration Flow Designer also enables you to work from the top down. You can start at the top defining systems of maps that have not yet been implemented and you work your way down later to implement these maps. You can also do a bit of both: reference existing executable maps and maps that will be implemented later within the same system.

A map component is an executable unit. The Integration Flow Designer maintains the execution settings that you specify separately with each map component. The referenced compiled map files and source map files are static and are not modified directly with this information. The settings are eventually used on designated servers to execute the maps.

Command Server or Launcher

The Integration Flow Designer manages collections of maps at design time while a Command Server or Launcher manages the execution of maps at run time. The Integration Flow Designer treats a map as an indivisible unit at design time while a Command Server or Launcher treats maps as indivisible units at run time.

The Command Server manages the execution of one system at a time. The Launcher manages the execution of as many systems as you have defined concurrently.

Command Servers

A Command Server executes maps based on commands that you supply using a predefined command-line syntax. You start a Command Server, providing it with one or more commands, and the Command Server terminates when it has completed the last command. The commands for a Command Server can be produced by the Integration Flow Designer from the execution settings that you specify for map components. You can also define the commands manually.

You can use a Command Server to execute one map or multiple maps. A Command Server executes multiple maps in a single-threaded sequential fashion, a scheme that is sufficient for many applications. A Command Server expects all the sources of a map to exist when the map is executed.

Note: Refer to the Command Server documentation for more information.

Launchers

The Integration Flow Designer is the tool you use to create and generate systems for the Launcher to run. A Launcher executes maps based on a sophisticated event-driven model. Maps are launched only when specific events occur and only when required resources to run the map are available. You can specify event triggers, source and target resources, and other settings for a map from the Integration Flow Designer.

For example, the Launcher for Windows NT is a service that runs maps in a multithreaded fashion. The Launcher is started and allowed to run continuously to monitor events and run maps accordingly.

Note: For more information about the Launcher, refer to the Launcher documentation.

Map component execution settings

You maintain the execution settings for map components by using the Integration Flow Designer. These settings are eventually embedded into the process control information, which drives the corresponding Command or Launcher. Some execution settings that you specify are used by both Event and Command Servers at execution time, including:

- The location of the map on the server to be executed
- The source settings for input data
- The target settings for output data
- The same map settings available in the Map Designer
- Retry options that apply if resources (such as work files or log files) that the map requires to execute are in use. Each source and target of the map also has retry options

Launcher settings

Some map component settings are specific to an Launcher, such as time event triggers and input event triggers. Examples of an input event trigger include when a file's timestamp changes, or when a messaging system receives a message.

Note: If you are using cluster support for the Launcher, the system definition file (.msd) is a key factor in the behavior of file triggers. The **OnSuccess** card setting must be set to **Delete** to prevent the file from being processed more than once.

Source and target settings

A source is used to read input data from a file, query a database, retrieve a message from a messaging system, or receive the results of a call to an application. A target is used to write output data to a file, insert or update data into a database, send a message to a messaging system, or call an application to send data to it.

A map component that references an executable map has a default set of sources and targets defined within the referenced source or compiled map file. The sources and targets are the interface points of the map component used within a system. The Integration Flow Designer displays sources and targets of a map component.

You can use the Integration Flow Designer to change a map component's sources or targets by changing its source or target settings. This capability enables you to reuse the functionality of a map and couple it to desired source and target pipelines.

Retry and rollback settings are also available for each source and each target. The retry setting specifies how many attempts will be made to recover from a source or target fault condition that might occur. These attempts are spaced over a specified interval of time. The rollback setting specifies whether source or target data should be rolled back to its original state if the map fails as a transaction unit.

Map settings

Map settings are initially defined for a map in the Map Designer. You can use the Integration Flow Designer to override these map settings. Map settings include definitions for audit log creation, paging values, trace file creation, validation settings, and work file configuration.

Systems

Systems of maps are executable units. The Integration Flow Designer enables you to use a system as a component of another system. This type of system is a subsystem component. A system can contain two types of components: map components and subsystem components. A subsystem component is a reference to another system that you have already defined. As with map components, the Integration Flow Designer displays subsystem components graphically.

A system can consist of map components and subsystem components. Subsystems can also be nested within subsystems. The Integration Flow Designer Navigator displays systems and the contained subsystems in a hierarchical and alphabetical list.

A map component and a subsystem component are conceptually similar. The difference between them is that they operate at different layers. As with a map component, a subsystem is an Integration Flow Designer object. A subsystem does the following:

- Encapsulates an executable system and includes a surrounding layer of execution settings used at run time
- Can include a subsystem component with a defined set of interfaces based on the sources and targets of the various maps that it contains

At design time, you decide which subsystem components you want to add to a system. A subsystem component can *reference* another system you have defined within a system definition file (.msd).

A subsystem component can also reference a design document that you have produced by using a preferred tool (a type is registered with the Windows platform). This facility helps you to keep track of design-related and implementation-related information at a system level.

As with map components, it is important to recognize that while a subsystem references another system, it does not consume it. The system being referenced is reusable. Another subsystem can also reference the same system.

Systems and the subsystems that reference them have execution settings. The execution settings include the server that will execute the system and the type of Command or Launcher that will run the system.

The following table lists some similarities of maps and systems of maps:

Feature	Map	Systems of Maps
Executable	✓	✓
Contains execution settings	✓	✓
Reusable	✓	✓
Include sources and targets that serve as component interfaces	✓	✓

Single-server systems

You can configure a system to execute on a single server; all maps contained within the system are executed on the designated server. Although all maps are executed on this single server, the actual sources or targets of these maps may cause data to be gathered from and sent to other servers.

A Command Server manages the execution of one system at a time. A Launcher manages the execution of as many systems as you have defined concurrently.

Distributed systems

You can also configure a system so its subsystem components are distributed across a set of servers. Each server has a Command or Launcher that runs the corresponding maps.

Data flows across servers between the distributed system components as indicated in the following example of a distributed system. This data flow is possible because maps can directly communicate data across servers when they execute.

There are four basic source and target types: file, database, message, and application. Within these four types, wide ranges of options are available to retrieve data from a remote location and to send data to a remote location, similar to the following scenarios:

- A map source or target can be a file that is in a shared location, allowing multiple servers to have direct access to it.
- Using the application adapter architecture, you can use any connectivity method to retrieve data from a remote location and to send data to a remote location. For example, you can use sockets, remote procedure calls, or directly call a local application that connects indirectly to another server to pass data. Databases and messages within messaging systems can also be accessed remotely.

The primary benefit of a distributed system is that you decide which subsystem components run on specific servers so that you can balance the overall execution load of the system across the servers in your enterprise. The data flow between servers is managed automatically as maps are executed. You do not need a global manager to coordinate this data flow.

You can easily add more servers to a distributed system at a later time. The Integration Flow Designer enables you to rearchitect the distribution of subsystem components across a new set of servers.

Component relationships in a system

You use the Integration Flow Designer to define the relationship between system components. As soon as you add map components to a system, an implied relationship exists. Based on the sources and targets you specify, the Integration Flow Designer deciphers if data flows between components at execution time. Data flow links automatically appear between components, drawing out a diagram of how the system will behave at execution time.

There are three types of links: internal, external, and pseudo.

Chapter 7. System definition files

A system definition file contains the definitions for one or more systems and servers.

Systems

A *system* is a collection of maps that you organize into a unit. A system can include components that reference executable maps and components that reference other systems. Systems of maps are executable units. The systems you define using the Integration Flow Designer are executed by Command or Launchers.

System definition files

A *system definition file* contains the definitions for one or more systems and the servers on which those systems are to execute.

System definition files and map source files

The following table notes the parallels between a system definition file (.msd) and a map source file (.mms).

system definition file (.msd)...	map source file (.mms)...
Normally contains one or more executable systems. Because a system contains a collection of maps, you can view a system definition file as a level up from a map source file	Normally contains one or more executable maps.
Maintains systems in source format for design purposes. Systems are not executed directly from a system definition file. You analyze and generate process control information for a particular system to execute it	Maintains maps in source format. Maps are not executed directly from a map source file. You build a compiled map to execute it

About relocating system definition files

If you want to copy system files (.msd) from one machine to another, be aware of the dependencies or references to other source map files (.mms), compiled map files (.mmc), and text or Windows-based design documents that you have produced.

The server map location is used to resolve paths specified as relative to the map location, such as the audit log, trace file, work file, and source and target files.

Whenever an .msd file is copied or moved to a different location or a different machine, the set paths for the contained source files (such as map source, compiled map, or subsystem component) change. When you open the system, the source

files cannot be found. In this case, you must change the paths to reflect the new file location. You can change paths by either performing a find-and-replace operation or by editing the component.

Working with system definition files

This section describes the basic procedures for working with .msd files, including opening, creating, closing, and so forth.

To create a new system definition file

From the **File** menu, choose **New**.

A new system definition file icon named **SystemFileX** (where *X* is a sequential number) is displayed in the Navigator. A new system window opens with the default name **System1**.

To open a system definition file

1. From the **File** menu, choose **Open**.
The Open dialog is displayed.
2. Select an .msd file and click **Open**.

To display the systems contained in a system definition file

To display systems in a system definition file, choose one of the following options:

- In the Navigator, click on a system definition file name.
Systems assigned a server of **None** are not executable.
- In the Navigator, click **+** to expand the system and view its components.
- Double-click on the system node or select the system node and press Enter to open the system window.

To close a system definition file

1. In the Navigator, select a system to close.
2. From the **File** menu, choose **Close**.
If you made changes to a system since it was last saved, a confirmation dialog is displayed giving you the option to save changes.

To save a system definition file

1. In the Navigator, select a system definition file name.
2. From the **File** menu, choose **Save**.
If this is a new system definition file, the Save As dialog is displayed so you can enter a file name.
3. Enter the name of the file to be saved and click **Save**.

To create a system

1. In the Navigator, select the system definition file that is to contain the new system.
2. From the **System** menu, choose **New**.
The New System dialog is displayed with a default name of **SystemX**, where *X* is a sequential number.

3. Accept the default name or enter a unique name for the system, and click **OK**.
A system window is displayed with the new system name and the path name of the system definition file displayed in the title bar.

To copy a system to another system

1. In the Navigator, select the system that you want to copy.
2. From the **System** menu, choose **Copy**.
The Copy System dialog is displayed.
3. Enter a unique name and click **OK**.
The new system name is displayed in the Navigator.

To rename a system

1. In the Navigator, select the system that you want to rename.
2. From the **System** menu, choose **Rename**.
3. When the Rename System dialog is displayed, enter a unique name and click **OK**.

To delete a system

1. In the Navigator, select the system that you want to delete.
2. From the **System** menu, choose **Delete**.
The Delete System dialog is displayed.
3. Click **OK** to delete the system.

Importing and exporting system definitions

The Integration Flow Designer permits the import and export of system definition files as XML. When you import or export a system definition file (.msd), the logging option is enabled by default and both the server and system definitions are included.

There are options available for password encryption and for overwriting existing definitions.

Exporting system definitions

For information about the export command line utility, refer to the Utility Commands documentation.

The following items are included in the definition when exporting an .msd file:

- deployment scripts
- components (maps, subsystems, external)
- pseudo links
- text components
- component positions within the system window

However, the state of a system window, such as the size or grid setting, is not exported.

Password encryption is an optional setting. To change default settings or for additional information about import and export options, see "Import and Export Options" .

To export a system definition:

1. From the Navigator, select a system to export.
The Save As dialog is displayed.
2. Accept the default or enter a new file name and click **Save**.
The system definition is saved as an .xml file.

The following XML example displays a simple system definition with one map component:

```
<?xml version="1.0" encoding="UTF-8"?>
<MSDFile FilePath="" ServerCount="0" SystemCount="1" Version="1">
  <System ComponentsCount="1" ExecutionMode="Command" Name="cmndsrvr" ScriptCount="1"
  ServerName="Local Server" Version="1">
    <DeployScripts Name="cmndsrvr" Version="1">
      <BuildAndTransferMaps>
      </BuildAndTransferMaps>
      <GenerateAndTransferMslFile FileName="install_dir\systems\evntsrvr.msl"/>
      <TransferAdditionalFiles>
      </TransferAdditionalFiles>
    </DeployScripts>
    <MapComponents ComponentName="cmndsrvr" ExecutableMapName="cmndsrvr"
    Expanded="No" LogFileName="install_dir\examples\general\ifd\cmndsrvr\cmndsrvr.log"
    MapCompiledFile="install_dir\examples\general\ifd\cmndsrvr\cmndsrvr.mmc"
    MapSourceFile="install_dir\examples\general\ifd\cmndsrvr\cmndsrvr.mms"
    Type="Source Map" Version="1">
      <EventServerSettings ConcurrentInstances="0"
      MapServerLocation="install_dir\examples\general\ifd\cmndsrvr\cmndsrvr.mmc"
      Mapdelay="0" Pendinghigh="0" Pendinglow="0" Priority="Normal" Version="1">
        <Pending_Hour>0</Pending_Hour>
        <EventServer_TimeEvent BeginAfter-Hour="0" BeginAfter-Minute="0"
        BeginAfter-Second="0" SkipIfBusy="Yes" Switch="OFF">
          <Trigger_Once></Trigger_Once>
        </EventServer_TimeEvent>
      </EventServerSettings>
      <ComponentPosition X="281" Y="129"/>
    </MapComponents>
  </System>
</MSDFile>
```

Importing system definitions

Note: For information about the import command line utility, see the Utility Commands documentation.

To change the default settings or for additional information about import and export options, see "Import and Export Options" .

To import a system definition:

1. From the Integration Flow Designer menu, select **File** → **Import**.
The Open dialog is displayed, prompting you to select an .xml file.
2. Choose a file and click **Open**.
The Save As dialog is displayed.
3. Enter a name for the system and click **Save**.
The system is displayed in the Navigator.

Import and export options

The Integration Flow Designer settings for importing and exporting system and server definition files are located under **Tools** → **Options** → **Import and Export**. Each option is described below.

Enable Logging: Enables or disables logging for the import and export process.

Import

Overwrite Servers: Enable this option to overwrite any server definitions already present in the .msd file. If the option is disabled, the import process fails when a duplicate server name is detected and an error is logged.

Overwrite Systems: Enable this option to overwrite existing system names already present in the .msd file. If the option is disabled, the import process fails when a duplicate system name is detected and an error is logged.

Referenced Servers: Enable this option to import any servers defined within the system to be imported.

Referenced Systems: Enable this option to import any referenced systems defined within the system to be imported.

Export

Encrypt Password: Enable this option to encrypt the server password.

Referenced Systems: Enable this option to include any referenced systems associated with the system definition file to be exported.

Referenced Servers: Enable this option to include any referenced servers defined within the system to be exported.

Audit log

When the logging option is enabled, the log file is generated by default into the directory of the map that is being imported or exported.

The name of the file begins with the map name, followed by **imexport.log**. For example, **cmndsrvrimport.log**.

To view the import/export audit log

Select **View** → **Import\Export Results**.

The import/export log is displayed.

The following is an example of an export log:

```
Map cmndsrvr exported successfully.  
System cmndsrvr exported successfully.  
15:23:33, March 02, 2004  
Output File: install_dir\examples\general\ifd\cmndsrvr\cmndsrvr.xml.  
Map cmndsrvr exported successfully.  
System cmndsrvr exported successfully.
```

Chapter 8. Map components

A map component is an Integration Flow Designer object that represents an executable map with a defined set of interfaces based on sources, targets, and a surrounding layer of execution settings used at run time.

The Integration Flow Designer treats each map component as a unique object. Consequently, the name, execution settings, corresponding source and target interfaces, and document links are maintained separately for each map component that you add to a system.

Source map components

Source maps reference executable maps defined within a map source file.

Compiled map components

Compiled maps reference executable maps in compiled file format.

Pseudo map components

Pseudo maps reference executable maps that have not yet been implemented.

Source map components

A source map component references an executable map that is defined in a map source file. You can add map components to single-server systems only. Single-server systems run on one server rather than multiple servers and cannot have a distributed server assigned.

A distributed system consists of subsystems that are assigned to run on different servers.

To add a source map component to a system:

1. In the system window, place the focus on a system to add a source map component to.
2. From the toolbox, click the Add Source Map tool.
3. Click on the location in the system window where you want to position the map.

The Add Source Map(s) dialog is displayed.

4. Enter the name of the map file or click the browse button to locate the file.

All executable maps in the opened map source file appear in the Add Source Map(s) dialog. Note that you can click browse again and open another map file. You can repeat this step as many times as needed.

5. Enable the check box next to each executable map you want to reference with map components. Later, when you are finished with the Add System(s) dialog, one map component will be added for every executable map you enabled.

To add source map components that reference all executable maps displayed in the list, click **Select All**.

6. Click **OK**.

Each new source map component is displayed in the system window.

Modifying a map component name

You can modify the names of source map components, compiled map components, and pseudo map components.

To modify a map component name:

1. Right-click the map component icon and select **Edit Map Component** from the context menu.
2. Enter the new name in the **Component Name** field (replacing the old name).
3. Click **OK** to save changes.

Compiled map components

A compiled source map component references an executable map within a compiled map file. You can use compiled maps when source maps are not available. You can add compiled map components to single-server systems only.

Use the following steps to add a compiled map component to a system.

1. In the system window, place the focus on a system that you want to add a compiled map component to.
2. From the toolbox, select the Add Compiled Map tool.
3. In the system window, click where you want to insert the compiled map. The Add Compiled Map(s) dialog is displayed.
4. Enter the name of the map file.

The executable map is displayed in the Add Compiled Map(s) dialog. Note that you can click the browse button again and open another map file. You can repeat this step as many times as needed.

5. Enable the check box next to each executable map you want to reference with map components. After you close the Add Compiled Map(s) dialog, one map component is added for every executable map that you selected.

To add compiled map components that reference all executable maps displayed in the list, click **Select All**.

6. Click **OK**.

Each new compiled map component is displayed in the system window.

Pseudo map components

Pseudo maps are placeholders for executable maps that have not yet been implemented. You can add pseudo map components to single-server systems.

To add a pseudo map component to a system:

1. In the system window, place the focus on a system.
2. From the toolbox, click on the pseudo map.
3. In the system window, click where you want to insert the pseudo map. The Add Pseudo Map(s) dialog is displayed.
4. Enter a name for the pseudo map component and click **Add**.

The component name displays in the **Component Name** column in the Add Pseudo Map(s) dialog.

5. Continue to add as many names for pseudo maps that you want to include as components in your system definition diagram.

A check mark automatically appears next to the component name to indicate that it will be added to the system file. You can also do the following:

- Change the component name by clicking **Modify Name**.
- Add a document link by clicking **Browse Link**.

Component names must be unique within a system. A warning message appears if you try to add a duplicate component name.

6. Click **OK**.

Defining sources and targets for a pseudo map

After a pseudo map is added to a system file, you can define sources and targets by adding input and output cards.

To add an input card:

1. Right-click the pseudo map icon and select **Add Input Card** from the context menu.

The **Add Input Card** dialog is displayed.

2. In the **CardName** field, enter a unique name for the input card. This is the only required field, however you can specify other values as needed.

If you choose to specify the **TypeTree** and **Type** values, they must reference a valid type in an existing tree. See the Map Designer documentation for more information.

3. Click **OK** to add an input card.

Editing input and output cards

The settings for an input card or an output card can be modified any time after the card has been added.

To edit an input or output card

1. Right-click the input or output card icon and select **Edit** from the context menu.
2. Make your changes to the card settings and click **OK** to save changes.

To delete an input or output card

1. Right-click the input or output card icon and select **Delete** from the context menu.
2. Click **Yes** when a dialog is displayed to confirm the deletion.

Creating a source map from a pseudo map

Before creating a source map, you must first add at least one output card with a valid type and type tree to the pseudo map. This procedure assumes that you have already added the input and output requirements.

After the pseudo map component is defined, you can convert it into a source map component. Pseudo maps are the only components in the Integration Flow Designer that allow you to create source maps and add input and output cards.

During source map creation, you can either create a new map source file (.mms) or select an existing map source file. Within the map source file, you can either create a new executable map or overwrite an existing map.

1. Right-click the pseudo map icon and select **Create Source Map** from the context menu.
The **Create Source Map** dialog is displayed.
2. In the **Map Source File** field, do one of the following:
 - If you want to create a new .mms file, accept the default map source file or enter a new name.
 - To select an existing .mms file, click GRAPHIC (browse) next to the **Map Source File** field and choose the file.
3. In the **Executable Map Name** field, do one of the following:
 - If you are creating a new .mms file, either accept the default executable map name (which is the same name as the pseudo map), or type a new map name.
 - If you selected an existing .mms file and you want to overwrite one of the existing executable maps, click the browse button next to the **Executable Map Name** field to display the list of executable maps within that file. Select the executable map and click **OK**.
 - If you selected an existing .mms file and you want to create a new executable map, type the new name in the **Executable Map Name** field.
4. To edit the input or output cards, select the card from the list and click **Edit**.
5. To create the .mms file, click **Create**.
6. When a message is displayed to confirm the map was created successfully, click **OK**.

The pseudo map component is converted to a source map component (the icon color changes).

Chapter 9. Viewing sources and targets of a map component

When you are working with systems in the Integration Flow Designer, you can view and modify sources and targets (inputs and outputs).

To display sources and targets of a map component

You can display map component sources and targets (inputs and outputs) by doing one of the following:

- Double-click a map component.
- Select the map component and click the Expand/Contract button on the toolbar. The expanded view of a map component displays (or hides) the input and output cards with corresponding card names and numbers. If the audit log file option is enabled for the map, the audit log file is presented as an output (**MapSettings** → **MapAudit** → **Switch = ON**).

From the system window, after you expand the contents of a map component, you can right-click on a card to view a context menu. From the context menu, you have the option to **Open Type Designer** or **View Card Details**.

- If you choose **Open Type Designer**, the selected card opens in the Type Designer.
- If you choose **View Card Details**, the **View** window is displayed for the card.

The following illustration shows a system with three expanded map components. The compiled map component, **CreditCheck**, has two inputs and one output, and a pseudo link representing data flow to the compiled map **Activity Report**. The input and output card names are displayed at the bottom of the card icons. Card sequences are defined in the Map Designer.

When a map component is expanded, you can right-click to view a context menu. From the toolbox, you can also use the Override tool to override the source or target of a card with the source or target of another card. Each input and output card is represented by an icon that represents its adapter class.

Card tooltips

The Integration Flow Designer provides card tooltips that display information about the source of data for an input card or the target of data for an output card. The type of information displayed depends on the adapter type.

To view card tooltips, move the mouse cursor over a card and wait approximately two seconds.

Resource type

Tooltip information

File Displays the file name specified for the card

Database Adapter

Displays the type of database adapter, connectivity information (connect string, data source name, and so on) and table(s) affected

Messaging Adapter

Displays the type of messaging adapter and the **AdapterCommand** defined for the card

Application Adapter

Displays the type of adapter and the **AdapterCommand** defined for the card

Viewing multiple tooltips

When working in a system definition diagram, it is often useful to view the card tooltips of several of the components simultaneously. You can do this by moving the cursor over each card tooltip you want to display. These card tooltips will continue to be displayed until you click anywhere in the system window or move the cursor over any open area in the system window and wait for several seconds. Performing either of these actions clears the card tooltips and restores the card names.

Chapter 10. Working with map components

Three basic characteristics of map components include the component name, execution settings (including source and target interfaces), and document links.

If a map component contains a trigger, the appropriate symbol is displayed on the execution settings button.

The Integration Flow Designer treats each map component as a unique object. The name, execution settings, corresponding source and target interfaces, and document links are maintained separately with each map component that you add to a system.

Map names

You choose a map component name when adding the component to a system. It is common to name a map component the same as the executable map that it references. You can also rename a map component within a system. Duplicate component names are not accepted within a system.

Execution settings

This section presents the basic concepts of map component execution settings and describes the available settings according to the Command or Launcher that runs the map.

Map components are executable units. Using the Integration Flow Designer, you can specify the execution settings for a map component, which in turn are used by the Command Server or Launcher to run the map. Because the Integration Flow Designer treats each component as a unique instance, these execution settings are maintained separately for each map component.

Some of the execution settings that you specify are used by both Launchers and Command Servers at execution time. Settings that apply to both engines include the location of the map on the server to be executed, source settings for input data, target settings for output data, and the same set of map settings available in the Map Designer.

Viewing execution settings

You can define execution settings for each map component in your system. There are two ways to access execution settings.

To view execution settings:

In the system window, do one of the following:

- Right-click the map component icon and select **Edit Launcher Settings** (or **Edit Command Server Settings** if the execution mode is **Command**) from the context menu.
- Click the execution settings button on the map component icon.

The Launcher Settings dialog is displayed (or Command Server Settings dialog if the execution mode is **Command**).

Integration Flow Designer execution settings

- "MapServerLocation"
- "MapSettings"
- "MapDelay"
- "PendingExpiration"
- "Priority"
- "Max Concurrent Map Instances"
- "TimeEvent"
- "Pending Instance Thresholds"
- "Input and Output Settings"

MapServerLocation

The **MapServerLocation** setting identifies the path of the executable map file on the server. The default value for the **MapServerLocation** setting is the path of the selected map.

MapSettings

MapSettings specify properties for map execution that are not specific to a particular input or output. These include settings for auditing a map, tracing data content, performance settings, validation settings, and map-specific warnings and errors.

Because maps executed with a Command Server can only override those **MapSettings** that have corresponding execution commands, some **MapSettings** are not configurable in the Integration Flow Designer. The following table displays a list of which Launcher and Command Server settings are available in the Integration Flow Designer.

MapSetting	Command Server Settings	Launcher Settings
MapAudit		
Data Log	✓	✓
Execution Log	✓	✓
MapSettings Log		✓
CardSettings Log		✓
MapTrace	✓	✓
WorkSpace	✓	✓
Century	✓	✓
Validation	✓	✓
Retry	✓	✓
Warnings	✓	✓
DocumentVerification	✓	✓

See the Map Designer documentation for information about configuring map settings.

MapDelay

The **MapDelay** setting defines the amount of time in milliseconds that the Launcher waits, after all event triggers are satisfied, before running the map. The default value for the **MapDelay** setting is **0**.

PendingExpiration

The **PendingExpiration** setting specifies the amount of time that a map can stay in an initiation pending state before expiring (that is, being dropped and forgotten). The default value for the **PendingExpiration** setting is **Hour**.

The **PendingExpiration** → **Value** setting specifies how many units in the **PendingExpiration** setting value (**Hours**, **Minutes**, or **Seconds**) to use for measuring the waiting period. The default value for the **Value** setting is **0**.

If the map associated with a given watch has multiple triggers, and at least one event trigger occurs, the map is placed in an *initiation pending* state until all triggers occur. A *watch* pertains to the map and the set of events that initiate it. Maps in an initiation pending state are reevaluated each time another event trigger occurs. The initiation pending state occurs when any of the following conditions are true:

- The instance of the map is defined as single-threaded and one occurrence of the map is already running.
- The map requires an event trigger that has not yet occurred.
- The maximum number of concurrent maps are already running.

If all preceding conditions are eliminated, then the initiation pending state can be removed from the map; if the **PendingExpiration** period has not elapsed, then the map can run.

After a map instance is initiated, a Source not available message might appear if that source is used by an application unknown to the Launcher and one of the following conditions apply

- One of the previous source event triggers of a pending map no longer exists
- all event triggers occur and another input is not an event trigger, and that source does not exist

The message Target not available may appear if a target message, database, or application adapter does not exist.

Priority setting

Do not use. This setting is reserved for future development.

Max Concurrent Map Instances setting

The **Max Concurrent Map Instances** setting specifies the maximum number of concurrent mappings per watch.

This setting overrides the **MaxThreads**, **WatchMaxThreads**, and **DisableMaxThreads** Event Server settings.

TimeEvent

The **TimeEvent** settings initiate map components of a system that is time-based. Each map component is run based on a time event, a source event, or a combination of multiple events. If a map component is not assigned an event trigger, the Launcher will not initiate that map to run. The **TimeEvent** settings are described in the following table.

Note: When you analyze a Launcher system using the Integration Flow Designer, a warning is registered for any source or compiled map component that has no associated event trigger.

TimeEvent Setting Description

Switch

The **Switch** setting enables or disables **TimeEvent** options. The default value for the **Switch** setting is **OFF** (Disables **TimeEvent** options).

BeginAfter

The **BeginAfter** settings specify the period of time in hours, minutes, or seconds to wait (from the time the Launcher starts) before enabling the time trigger.

- **Hour** - The **Hour** setting is used to specify how many hours to wait to begin the map. The default value for the **Hour** setting is 0.
- **Minute** - The **Minute** setting is used to specify how many minutes to wait to begin the map. The default value for the **Minute** setting is 0.
- **Second** - The **Second** setting is used to specify how many seconds to wait to begin the map. The default value for the **Second** setting is 0.

Trigger

Use the **Trigger** setting to define the type of time event trigger. You can trigger a map based on a singular time event or cycle, according to a specified time increment. The default value is **Once**.

Trigger → Interval

Use the **Interval** setting to describe the recurring period to use for the trigger. The **Interval** setting is available only if the value of the **Trigger** setting is **Every**. The default value for the **Interval** setting is 1 **Day(s)**.

Interval → At

Use the **At** settings to set a specific time for the trigger to start after the start of the interval defined in **Interval** settings. The **At** settings are only available if the value of the **Trigger** setting is **Every**. The time periods specified in the **Hour**, **Minute**, and **Second** settings describe a time based on a 24-hour clock set to system time.

- **Hour** - The **Hour** setting specifies the hour in which to run the map. The default value for the **Hour** setting is 0.

- **Minute** - The **Minute** setting specifies the minute in which to run the map. The default value for the **Minute** setting is **0**.
- **Second** - The **Second** setting specifies the second in which to run the map. The default value for the **Second** setting is **0**.

OmitDays

Use the **OmitDays** setting to set the days to omit.

- **Monday** - The **Monday** setting is used to omit Mondays from the time trigger schedule. The default value for the **Monday** setting is **No**.
- **Tuesday through Sunday** - These settings work the same way as the **Monday** setting.

OmitHours

Use the **OmitHours** setting to set the hours to omit.

- **0** - The **0** setting is used to omit the 0 hour (midnight to 1:00 am). The default value for the **0** setting is **No**.
- **1 through 23** - These settings work the same way as the **0** setting (above).

SkipIfBusy

Use the **SkipIfBusy** setting to ignore the time event trigger if it occurs when a watch instance is already in process. The default value is **Yes**.

EventCoordination

Use the **EventCoordination** setting to control the coordination of time event and input event triggers. The **EventCoordination** setting is available only if the map has both a time event trigger and one or more input event triggers; and the **SourceEvent** setting for one of the input cards is **ON**. The input event trigger can have wildcards. The default value for the **EventCoordination** setting is **FirstAvailable**.

Pending Instance Thresholds setting

The **Pending Instance Thresholds** setting enables a listener to pause or resume for a specific watch.

Setting Description

High Value (no limit) for the watch to pause the event(s) for that watch. The default value is **0**.

Low Value (no limit) for the watch to resume the event(s) for that watch. The default value is **0**.

Note: If the **InitPendingHigh** and **InitPendingLow** settings in the **Launcher** section of the **dtx.ini** file are also set, the **Pending Instance Thresholds** Launcher settings will take precedence.

Input and output settings

Because maps executed with a Command Server can only override those input and output settings that have corresponding execution commands, some input and output settings are not configurable in the Integration Flow Designer.

Note: If you are using cluster support for the Launcher, the system definition file (.msd) is a key factor in the behavior of file triggers. The **OnSuccess** card setting must be set to **Delete** to prevent the file from being processed more than once.

Input settings

The following table lists input settings.

Input Setting	Command Server Settings	Launcher Settings
FetchAs	-	✓
WorkArea	✓	✓
Backup	-	✓
SourceEvent	-	✓*
Source		
Command	✓	✓
OnSuccess	-	✓*
OnFailure	✓	✓
Retry	✓	✓
Scope	-	✓
Warnings	-	✓
FetchUnit	-	✓

*Availability varies by adapter.

Note: See the Map Designer documentation for information on how to configure these settings.

Output settings

The following table lists output Launcher and Command Server settings.

Output Setting	Command Server Settings	Launcher Settings
Backup		✓
Target		
Command	✓	✓
OnSuccess		✓*
OnFailure	✓	✓
Retry	✓	✓*
Warnings		✓
Scope		✓

*Availability varies by adapter.

Overriding map settings

Map settings specify properties for map executions that are not specific to a particular input or output. Map settings defined during the design phase are saved in the map source file (.mms) and the compiled map file (.mmc). Using the Integration Flow Designer, you can override these map settings. Any time you change a map setting from the Integration Flow Designer, it is considered an

override. This override value is used during the creation of system files for the Launcher or command files for the Command Server.

When you place an override on a map card, the override propagates to the uppermost system level where that card is visible. When an override is placed on a subsystem card for the corresponding map card, the higher-level subsystem card override takes precedence, and is propagated to the top. The override at the highest level always takes precedence.

To override a map setting:

1. In the system window, right-click on a map component and select **Edit Launcher** settings.

The Launcher Settings dialog is displayed.

2. Select a map setting and choose a new value.
3. Click **OK** to save changes.

The new map settings will be used during execution.

Overriding card settings

Card settings, which are specific to input and output cards, define the data object the card represents and how that data is retrieved or routed. A map component that references an executable map has card settings that are defined in the map source file (.mms) and the compiled map file (.mmc). Using the Integration Flow Designer, you can specify execution settings that override the card settings.

To override card settings:

This procedure assumes that **Launcher** is the execution mode for this system.

1. In the system window, double-click on a map component to view the inputs and outputs.

2. On the map component icon, click the execution settings button.

The Launcher Settings dialog is displayed.

3. Select an input or output setting and choose a new value.
4. Click **OK** to save changes.

The new card settings will be used during execution.

Chapter 11. Subsystem components

Systems can contain any combination of map components and subsystem components. A subsystem component is an Integration Flow Designer object that represents or references another system that you have already defined. The concept of a subsystem component is that it enables you to reuse a system. As such, a subsystem is an executable unit for which you can specify execution settings for using the Integration Flow Designer.

The Integration Flow Designer treats each subsystem component as a unique instance. Consequently, the name, execution settings, source and target interfaces, and document links are maintained separately with each subsystem component that you add to a system definition file.

The Integration Flow Designer displays a subsystem component graphically in the system window, as shown in the following diagram.

Subsystem components, which are displayed graphically in the system window, have four basic characteristics: name, execution settings (including sources and targets), and optional document links.

Subsystem component name

When you add a subsystem component to a system definition file, you are selecting a system from a list of systems that you have already defined. By default, the name of the subsystem component is the name of the referenced system.

After a subsystem component is added to a system, you can rename it. The name can be any length and can contain spaces. A subsystem component name must be unique because the Integration Flow Designer does not accept duplicate component names within a system file.

Execution settings

A subsystem component has a defined set of interfaces that are based on the sources and targets (inputs and outputs) of the various maps that it contains. The inputs and outputs of a subsystem are derived according to the following rules.

- An input of a subsystem is a source that is not internally produced by one of its components.
- An output of a subsystem is a target that is internally produced by one of its components, and is not deleted.

To view execution settings for a subsystem

From the system window, do one of the following:

- Click the execution settings button on the subsystem icon.
- Right-click on the subsystem icon and select **Edit Execution Settings** from the context menu.

The System Launcher Settings dialog is displayed.

ServerName

The **ServerName** setting specifies the type of server on which the system is deployed. The values include **Local Server**, **Distributed**, and **None**. You can also define additional servers that appear in the drop-down list. The default value for the **ServerName** setting is **Local Server**. Each server option is described in the following table.

Option Description

Local Server

A system is going to run on the local workstation, usually where the Integration Flow Designer is installed. You might want to assign **Local Server** to systems to use your workstation as a test environment to verify execution. Or, you may plan to use **Local Server** as a part of your production environment for system execution.

Distributed

A system is composed of subsystems that are assigned to run on different servers in a coordinated manner to distribute the processing. Components in a distributed system must be subsystems, not maps.

None No server selected. Because there is no server assigned, the system is not executable.

Systems that have an assigned server value of **None** do not appear in the **Component** view in the Navigator. These nonexecutable systems can be used as subsystems within a system that has an assigned server.

A subsystem with an assigned server value of **None** derives its server from its parent system that has a server assigned. Subsystems always display in the **Composition** view unless the parent system also has a server value of **None**.

ExecutionType

The **ExecutionType** setting specifies the execution mode for the system. Choices include **Launcher** or **Command**. The default setting is **Launcher**, which indicates that the system is deployed for execution by the Launcher. If **Command** is selected, the system is deployed for execution by the Command Server.

Inputs and outputs

A subsystem component has a set of sources and targets that are based on the sources and targets of the various maps that it contains. The sources and targets (inputs and outputs) are the interface points of the subsystem component used within a system.

In general, a system input is a component input that is not internally produced and a system output is a component output that is not deleted. When a system is presented as a component of another system, the subsystem inputs and outputs serve as interface points, allowing it to be connected to other components in the view.

Double-clicking a subsystem icon expands the view to show inputs and outputs. Double-clicking an input or output card displays the execution settings for the subsystem component.

Default settings

The default settings for inputs and outputs are based on the inputs and outputs of the various maps contained within the subsystem.

In the system diagram above, the following applies:

- A system input card correlates to a specific input card of a map component that it contains.
- A system output card correlates to a specific output card of a map component it contains.

If any of the inputs can trigger the map, then you will see an input event trigger displayed on the map card icon.

Note: The time trigger icon and input event trigger icon are not displayed on the execution settings button of a subsystem component icon.

The following convention is used to identify the subsystem card in the area below the card icon.

<MapCardName> : <MapComponentName>

Saving map card positions

When you initially expand a subsystem component, the input and output cards are cascaded with first card on top by default. Similar to map components, you can move subsystem input or output cards to specific positions and the card positions are retained when you save the file.

The following pertains to card positioning:

- Moving the subsystem component in contracted view to a new position resets the card position to the default.
- If the subsystem component is expanded and it is moved to a new location, the card positions are not changed.

Overriding card source or target

In a subsystem, you can override the source and target of a card with the source or target of another card by using the override function. This works the same as with map component input or output cards.

Working with subsystems

The following procedure details adding a subsystem to a system.

To add a subsystem component to a system

1. Open an open .msd file and ensure that it is focused in the system window.
2. From the toolbox, click the Add System tool and then click the location in the client area of the system window where you want to add the subsystem.

The Add System(s) dialog box is displayed.

3. Choose an .msd file and click **OK**.

In the Add System(s) dialog box, the .msd and a list of systems that it contains is displayed.

4. Do one of the following:
 - Click **Select All** to reference all systems listed in the new subsystem component.
 - Select only the systems in the list that you want to reference in the new subsystem component.
5. To change the name of a system in the list, click to select it and click **Modify Name**.
6. To link a document to this subsystem, click **Browse Link** and navigate to the file.
7. Click **OK** to save changes.
8. When a message is displayed to confirm that the subsystem was generated, click **OK**.

Recursion is not allowed. The subsystem that you add to a parent system cannot contain the parent system as a component. The Integration Flow Designer alerts you if you try to add a subsystem that contains the system you are defining.

Chapter 12. Overview of server definitions and execution modes

The distribution of systems among your enterprise servers plays a key role in the behavior of these systems at execution time. You can define servers using the Integration Flow Designer. These server definitions represent servers within your enterprise that execute systems. After a server is defined, you can assign it to a system. You also assign an execution mode to a system to specify the type of Command or Event Server that will run this system. The server and execution mode that you assign to a specific system provides the Integration Flow Designer information it needs to perform key operations. For example, building maps depends on the target server platform. Also, generating appropriate process control files for execution purposes depends on the type of Command or Launcher that will run the system.

Server implementation examples

Use the Integration Flow Designer to define systems of varying complexity. The simplest case involves one system that runs on one server. The following sections describe various examples of systems with increasing complexity.

Multiple independent systems running on one server

You can define multiple systems that have no data dependencies among their components and are all executed on the same server.

Although all systems in this configuration are executed on the same server, data can still be gathered from and sent to remote server locations. This behavior is possible because you can set the sources and/or targets of maps contained in the systems to directly cause data to flow from and to remote servers when these maps are executed.

You can also have multiple independent systems running on multiple servers.

Multiple dependent systems running on one server

You can use the Integration Flow Designer to define systems that have data dependencies among their components and are all executed on the same server.

Data sources and targets can exist on remote servers.

Multiple dependent systems running on multiple servers

You can use the Integration Flow Designer to define distributed systems. A distributed system consists of only subsystem components, not map components directly. These subsystem components are distributed across more than one server. Data flows across servers from one subsystem to another based on the sources and targets of maps that you specify to cause this flow.

Complex combinations of systems

You can define varying combinations of the previously described systems to yield very complex systems. For example, you might have a system that is distributed across servers as well as completely independent systems running on the same servers. The design of your systems depends on your requirements.

Server definitions

The Integration Flow Designer includes three predefined server definitions: **Distributed**, **Local Server**, and **None**.

In addition to the predefined servers, you can create custom server definitions for each system. When you create your own server definitions, they appear on the **Server** drop-down list in the system window.

A new system is automatically assigned the default server of **None** and is considered "non executable". Nonexecutable maps do not appear in the **Composition** view of the Navigator.

Each predefined server type is described in the following table.

Server Definition	Used When
-------------------	-----------

Distributed	A system is composed of subsystems that are assigned to run on different servers in a coordinated manner to distribute the processing. Components in a distributed system must be subsystems, not maps.
Local Server	A system is going to run on the local workstation, usually where the Integration Flow Designer is installed. You might want to assign Local Server to systems to use your workstation as a test environment to verify execution. Or, you may plan to use Local Server as a part of your production environment for system execution.
None	No server is assigned. The system is not executable, but can be used as a subsystem within a system. A subsystem with an assigned server of None derives its server from its parent system that has a server assigned. A new system automatically has None assigned by default.

Working with server definitions

This section contains the basic procedures for working with servers, such as adding, editing, deleting, and so forth.

To add a new server to a system:

1. In the Navigator, click the system that is to contain the server definition.
2. From the **Server** menu, choose **New**.
The Add Server dialog is displayed.
3. In the **Name** field, enter a name for the server that is unique within the system definition file. You should use the name of the server as defined by your network administrator.

4. In the **Platform** drop-down list, select the server platform. When building source maps for a system that has this server as the assigned server, the maps are built for the platform specified in this **Platform** field.
5. In the **User ID** field, enter the user name that has the appropriate privileges for copying files to this server. This information is used when you deploy systems.
6. In the **Password** field, enter the password corresponding to the **User ID** specified that has the appropriate privileges for copying files to this server. This information is used when you deploy systems.
7. In the **IP Address** field, enter the IP address corresponding to the server identified in the **Name** field. If the machine hosting the Integration Flow Designer has access to the server, click **Get Address** to automatically complete the **IP address** field.
8. Click **OK** to save the server definition.
A server definition that you add is saved when you save the system definition file that contains it.

Assigning a server to a system

In the Integration Flow Designer, there are two ways to assign a server to a system. You can either copy an existing server and drag it to a new server, or select the system in the Navigator and choose **New** from the **Server** menu. Both methods are described below.

To copy a server to system (drag-and-drop method)

Use this method if you want to add an existing server to a different system definition file (.msd). You must have both files open in the Navigator. **System File A** is the .msd file with the existing server; **System File B** is the .msd file receiving a copy of the server.

1. From the Navigator, click and hold on the server in **System File A**, and press **Ctrl**.
2. Drag the server to the root server node in **System File B**.
The mouse pointer will display a plus sign (+) to indicate correct positioning.
3. Release the mouse to drop the server.
4. When a message is displayed to confirm this action, click **Yes**.

You can also use this procedure to duplicate a server within the same system file. A new server with the same properties is generated, however, it has a unique name.

Modifying a server definition

The following procedures detail how to add or modify a server definition.

To change or assign a server in a system

To change or assign a server for a system, select a new server from the **Server** drop-down list in the system window.

To edit a server definition

1. Select a system containing the server definition that you want to edit.
2. Choose one of the following options:
 - From the **Server** menu, choose **Edit**.

- The Servers dialog is displayed.
- Select the server from the list and click **OK**.
- Or
- From the Navigator, right-click on the server and select **Edit** from the context menu.
- The Edit Server dialog is displayed displaying the current server information.
3. Make changes and click **OK**.

Note: After changing the server definition assigned to a system, the system should be analyzed and regenerated, as described in "Generating System Files".

Deleting a server definition

There are two ways to delete a server definition:

- From the **Server** menu
- By using the context menu in the Navigator

To delete a server definition:

1. Open the system that contains the server definition to be deleted.
2. Do one of the following:
 - From the Navigator, right-click the server and select **Delete** from the context menu.
 - From the Navigator, select the server and then select **Server** → **Delete**.
3. When a dialog is displayed to confirm the deletion, choose **Yes**.

Importing and exporting server definitions

The Integration Flow Designer permits the import and export of server definitions as XML files. For information about the Integration Flow Designer setting for the Import and Export operations, see "Import and Export Options".

To export a server definition

1. From the List view of the Navigator, select the **Servers** node to include all servers, or select a single server to export.
2. From the menu bar, select **Servers** → **Export**.
The Save As dialog is displayed.
3. Enter a file name and click **Save**.
An XML server definition file is created.

To import a server definition

1. From the **List** view of the Navigator, right-click the **Servers** node and choose **Import**. The Open dialog is displayed, prompting you to select an .xml file.
2. Choose a file and click **Open**. The Import - Servers dialog is displayed. Each server defined in the selected .xml file is displayed under **Servers**.
3. Under **Servers**, select the server to import and click the appropriate arrow button. The server names are displayed in the **Import** list.

4. To replace any existing servers with the same names as those you are importing, enable the **Overwrite Servers** check box. Otherwise, disable the check box to ensure that existing servers are not overwritten.
5. Click **OK** to complete the import process. The imported servers are displayed under the **Servers** node in the List view of the Navigator and on the **Server** drop-down list of the system window.

System execution modes

Each system you create has an execution mode. You specify the system execution mode by selecting **Launcher** or **Command** from the **Execution Mode** drop-down list.

The execution mode you select determines the type of execution server that will run the system.

- Selecting **Launcher** execution mode indicates that your system will run on a Launcher.
- Selecting **Command** execution mode indicates the system will run on a Command Server.

Chapter 13. Generating system files

To run an Integration Flow Designer system, you must generate a command (.txt) or Launcher system file (.msl) from your system definition file (.msd). An .msl file is the process control file for the Launcher. Before generating an .msl file, you should first build all maps and then analyze the system.

Building maps

The Integration Flow Designer enables you to easily build maps for a specified system. You can build a single map, a selected group of maps, or all maps referenced within a system. The build process is the same as that performed in the Map Designer.

To build all maps referenced by all source map components in a system

1. Open a system.
2. From the **System** menu, choose **Build Maps**.

All of the executable maps are built. The **Build Maps** dialog is displayed displaying the progress of the build process. After the Integration Flow Designer builds all the maps in the system, a dialog is displayed displaying the build results.

To build one map referenced by a source map component in a system

1. Place the focus on a system in the system window.
2. Right-click a source map component and choose **Build**.

The selected map is built.

To build a collection of maps referenced by a collection of source map components within a system

1. Place the focus a system in the system window.
2. From the toolbox, click the **Select** tool and select each source map component to include.
3. Select **System** → **Build Maps**.

The selected maps are built.

To view the last build results at any point in time

Select **View** → **Build Results**.

The results from the last build appear.

Analyzing a System

To ensure that a system will execute properly before actually executing it, use the **Analyze** function. This function analyzes the system for logical consistency and displays the results.

To analyze a system

1. Place the focus on a system in the system window.
2. Select **System** → **Analyze**.
The **Analyze System** dialog appears.
3. If errors are detected, click **Results** to view the details.
The **Analysis Results** dialog appears.

Generating executable system files

In the Integration Flow Designer, there are two execution modes you can use to create an Launcher system file: **Launcher** and **Command**. **Launcher** execution mode generates system files (.msl) for the Launcher. **Command** execution mode produces command files (.txt) for the Command Server. A Command Server runs one command file at a time.

To generate a command file

1. Place the focus on a system in the system window.
2. From the **Execution Mode** drop-down list, select **Command**.
3. Select **System** → **Generate**.
The Save As dialog is displayed.
4. Specify the path of the command file to be saved.
The generated command file includes one command for each map specified in the selected system. Options for each map are based on the Command Server settings.

```
; Date: Thu May 13 14:44:46 1999 (Version 2.0)
; File: C:\Examples\Business.msd
; System: Business
;
; Component: SourceMap
; Map File: C:\Examples\Business.mms
; Executable Map: SourceMap
;
E:\business\SourceMap.mmc
-WD
-Z14:18:21:26:27:28:29
;
; Component: business1
; Map File: C:\Examples\Business.mms
; Executable Map: business1
;
E:\business\business1.mmc
-ID1 '-T'
-WD
-Z14:18:21:26:27:28:29
;
; Component: business2
; Map File: C:\Examples\Business.mms
; Executable Map: business2
;
E:\business\business2.mmc
```

```

-ID1 '-Trig I trigttest1'
-WD
-Z14:18:21:26:27:28:29
;
; End System: Business
;

```

The commands within the generated command file are ordered based on the input and output dependencies of the maps in the system. Where no input or output dependencies exist, the commands are ordered based on the physical positioning of the maps in the system definition file (starting at the upper left point and moving left to right and top to bottom).

To facilitate the moving of components to different positions, use the **Align to Grid** option.

To generate a Launcher system file

1. Place the focus on a system in the system window.
2. From the **Execution Mode** drop-down list, select **Launcher**.
3. Select **System** → **Generate**.

The Save As dialog is displayed.

4. Specify the path and name of the Launcher system file (.msl) to be saved.

Other saving options:

- If you have a Windows share setup between your client and server machines, then you can save the .msl or command file directly to the server.
- You can use the Deploy option to copy the .msl or command file to the server running the corresponding Command or Launcher. (See "Deploying Systems".)
- You can manually copy the .msl or command file to the server running the corresponding Command or Launcher.
- For Launchers, place a copy the Launcher system file in your deployment directory.

When the Launcher runs, it manages the execution of all Launcher system files in the deployment directory. See the Launcher documentation for additional information.

Unresolved settings

If you use the Map Designer to delete or rename an input or output card that has override settings defined in a system, the Integration Flow Designer detects the change after the map source file is saved or after the compiled map file is produced. Within the Integration Flow Designer, settings that are no longer associated with any card are known as unresolved settings.

You know that unresolved settings exist in your system when one of the following scenarios occurs:

- You open a system and the **Unresolved Settings** dialog is displayed.
- You have a system open in the Integration Flow Designer and you use the Map Designer to delete or rename a card that is referenced by a map component in the system. Unresolved settings are detected when you save the map source file. When you return to the Integration Flow Designer, the **Unresolved Settings** dialog is displayed.

From the **Unresolved Settings** dialog, you can immediately delete or assign the unresolved settings to a different map card; or you can close the dialog and access it again at a later time by selecting **Unresolved Settings** from the **System** menu.

To view unresolved settings:

1. Open the system that contains unresolved settings.
2. From the **System** menu, choose **Unresolved Settings**.

The Unresolved Settings dialog is displayed.

The dialog displays the settings that are unresolved and are no longer associated with any card. These settings were defined in the Integration Flow Designer as the source or target settings for a card that no longer exists because it was deleted or renamed in the Map Designer.

Processing unresolved settings

You can process unresolved settings by deleting the card or by modifying the card settings by copying the settings to an existing card.

After you display the affected system in the Unresolved Settings dialog, you can immediately modify the card settings or delete the card.

If you modify card settings, you can drag the updated card to a map card in the system window, which updates the existing card settings in the map component.

To resolve map card settings

1. From the Unresolved Settings dialog, if you are not familiar with the map component that is displayed in the Unresolved Settings navigation pane, right-click the card and select **Go to component**.
The map component is displayed as a selected object in the system window.
2. Return to the Unresolved Settings dialog, and modify the card settings as required.
3. When the card settings are correct, click-and-hold on the card (in the navigation pane) and drag it to the corresponding card in the system window.
4. When the card in the system window is highlighted, release the mouse to drop the card.
5. Click **Yes** when you are prompted to confirm the copy operation.

To delete unresolved settings

From the Unresolved Settings dialog navigation pane, right-click the affected card and select **Delete**.

System definition file differences

The Source Compare feature enables you to view differences between two system definition files.

Comparing system definition files

When comparing system definition files, the files are considered "different" when:

- Any of the systems that exist in one file and do not exist in the second file.
- Any of the systems are different.
- Any of the servers that exist in one file and do not exist in the second file.

- Any of the servers are different.

Comparing systems

When comparing systems, systems are considered "different" when:

- Any of the system properties are different.
- Any of the components of the system are different.
- A component exists in the system in the first file but does not exist in the same system in the second file.
- If any of the deploy scripts are different.
- If a deploy script exists in the file but not in the second file.

Comparing servers

When comparing servers, servers are considered "different" when any of the server settings are different.

Comparing deploy scripts

A deploy script is different if any of the script instructions are different.

Comparing components

A component is different if:

- If any of the doc links are different
- If any of the doc links are present in the component in the first file but do not exist in the same component in the second file
- If any of the unresolved settings are different
- If any of the log files are different
- If any of the input/output cards are different
- If any of the input/output cards are present in the component in the first file but do not exist in the same component in the second file

Comparing compiled map components

A compiled map component is different if:

- File path of compiled map is different.
- Map settings are different.
- Launcher or Command Sever settings are different.

Comparing source map components

When comparing source map components, components are considered "different" when:

- File path of map source is different.
- Map settings are different.
- Event or Command sever settings are different.
- Map name for source map component is different.

Comparing a psuedo map component

When comparing pseudo maps, a component is considered "different" when:

- Pseudo map name is different
- Event or Command sever settings are different
- Map settings are different

Comparing subsystem components

When comparing subsystem components, components are considered "different" when:

- Referenced system name is different.
- Referenced system definition file path is different.
- Event or Command sever settings are different.
- Any of the log files are different.

Comparing document links

A document link is different if the paths are different.

Comparing system definition files

System definition source file differences are compared for the source files that you select. You can view the results after the differences are identified.

To compare system definition source files

1. From the **File** menu, choose **System Definition File Differences**.
The Select First File dialog box is displayed.
2. Choose the first system definition source file to compare and click **Open**.
The Select Second File dialog box is displayed.
3. Choose the second map source file to compare and click **Open**.
The progress of the comparison is shown briefly in the System Definition Source File Differences Analysis dialog and then the System File Differences window is displayed.

To view system definition file differences

1. From the upper panes of the System Definition File Differences window, select the item you want to compare.
The settings display in the lower panes, based on your selection.
2. Press F8 to view the next difference or F7 to view the previous difference.

Viewing and printing system definition file differences

You can view and print all settings for an element that you select in the compared .msd file, regardless of the differences. For example, if you select a map source file, three tabs of information appear in the lower pane: **Properties**, **Map Settings**, and **Launcher Settings**.

- To expand a list of **Properties**, **Map Settings**, or **Card Settings**, right-click on the list and choose **Expand All**.

- To print **Map Settings** or **Card Settings**, right-click on the list and click **Print Properties**.

You cannot edit information from this window.

Chapter 14. Links

A link is a connection that the Integration Flow Designer creates between two components in a system definition file.

Links represent the direction of data flow among the components of a system at execution time. One of the linked components produces data and the other component uses it as a source. Links graphically present the source and target dependencies between system components.

Internal links

The Integration Flow Designer treats a map component as a self-contained indivisible unit. This basic building block is displayed graphically using various visual cues to convey information about it. Double-click on a map icon in the system window to expand or contract the view.

The expanded view of a map component uses links to distinguish its sources from its targets. These links are called internal links because sources and targets are an integral part of a map component. By default, the Integration Flow Designer displays the sources to the left of a map component and the targets to the right. However, you can drag any of the source or target icons to any new location within a system window. A map's internal links always help you to distinguish source icons from target icons regardless of where these icons are placed.

The Integration Flow Designer also displays the audit log of a map as a target if you enable the map's audit log option (**MapAudit** → **Switch = ON**). The audit log file is represented as a target by a dotted internal link.

In general, the Integration Flow Designer automatically derives and displays internal links for an expanded map component. It uses information in the referenced source or compiled map file together with any source and target overrides that may be specified.

If you change the default sources or targets defined within a source or compiled map file using the Map Designer, the Integration Flow Designer will display these modified definitions when you expand the corresponding map component. Also, if you change the source or target (input or output) overrides for a map component by using the Integration Flow Designer, the changed overrides are displayed when you expand the corresponding map component.

The Integration Flow Designer prevents you from deleting internal links as well as the corresponding source and target icons. You must use the Map Designer to change the number and type of inputs and outputs.

External links

Map components and subsystem components are the fundamental building blocks of a system. The Integration Flow Designer uses links to display data flow relationships among these components. A link that is displayed between two system components is called an external link because the link is not an integral

part of either component. An external link represents the fact that one of the components produces a data target and the other component uses this data as a source.

An external link can connect a map component to another map component, or a map component to a subsystem component, or a subsystem component to a subsystem component. External links are represented by a solid, directed line. An example follows. The map components are expanded, indicating that the internal links are visible. A visual clue that indicates a component is expanded is that the name is displayed in bold text. When the component is not expanded, the name text is regular, not bold.

An external link conveys more information when the map components that it connects are expanded. In this example, you can explicitly view which targets of one component feed which sources of the other component. When a link is displayed between contracted map components, the link implicitly displays the data flow relationship between the two components.

The Integration Flow Designer automatically derives and displays an external link between two components that share common resources as sources and targets. As with internal links, the Integration Flow Designer uses information in the referenced source map or compiled map files together with any source and target overrides that you may have specified to determine if any two components in a system definition file have a data flow relationship.

The Integration Flow Designer automatically derives and displays database links, messaging links, and file links.

A database link is drawn between two map components when the Select statement defined in the database source of one of the maps references the table associated with the database target of another map. You can view the links explicitly if both map components are expanded.

A messaging link is drawn between two map components when the messaging source of one of the maps references the messaging target of another map.

A file link is created between two map components when the file named in the source of one of the maps references the same file in the target of another map.

Changing the default sources or targets defined within a source map or compiled map file using the Map Designer, or changing the source or target overrides by using the Integration Flow Designer has a direct effect on the external links that are drawn.

The Integration Flow Designer prevents you from deleting external links that it has automatically derived. You must either delete one of the components or change the source or target settings to remove an external link.

Pseudo links

The Integration Flow Designer automatically derives external links and displays them using solid directed lines. The Integration Flow Designer also enables you to manually define external links between two components. These are called pseudo links. A pseudo link is displayed using a dotted directed line.

Pseudo links represent data flow relationships that cannot be derived statically at design time. They help you to visualize data flow relationships that are established at execution time. The Integration Flow Designer does not embed pseudo links that you define into the process control information, which drives Command or Launchers. Pseudo links serve a visual purpose only.

Using a pseudo link

This section contains the basic procedures for working with pseudo links.

To add a pseudo link

1. In the system window, place the focus on the system that contains the components to link.
2. From the toolbox, click on the Add Pseudo Link tool and click the source component or card in the system window.
The mouse pointer changes to a cross (+).
3. Click on the target component or card.
A dashed line is displayed between the two components or cards.

To delete a pseudo link

1. From the system window, select the link to delete.
Both ends of the pseudo link now have small bold rectangles indicating it is selected.
2. Press **Delete**.
The pseudo link is deleted.

To display or hide links within a system window

1. Place the focus on a system in the system window.
2. From the toolbar, click the Display/Hide Links tool.
All links will be displayed or hidden.

Doc links

The Integration Flow Designer includes a feature called Doc Links to assist you in documenting your design. You can link any type of text file or Windows-registered document to a map component. For example, you can link a Microsoft Word document or a project plan to a map component that specifies design information for the map. You can view and edit the document by referencing it through the map component to which it is linked.

Using doc links

You can define Doc Links for each map component and subsystem component as well as the sources and targets. A document link associates a document file of your choice, such as a design specification, with the component. For example, it is best to link a design specification to a subsystem component because it specifies information at the system level as opposed to the map level.

To add a document link for a component

1. Right-click the component icon where the document is to be linked.
2. From the context menu select **Doc Links** → **Add** and select the document to be linked by clicking **Open**.

The document is now associated with the component.

To view a document link

Doc Links can be viewed by launching the associated application from the Integration Flow Designer.

Right-click the map icon with the document link and select **Doc Links** → **document**.

Clicking the document will launch the associated viewing application provided that it is registered on the machine running the Integration Flow Designer.

To delete a document link

Deleting a document link does not delete the document, only the link between the component and the document.

1. Right-click the map icon with the document link and select **Doc Links** → **Remove**.

The Remove Doc Links dialog is displayed.

2. Select the document link and click **Remove**.

The document link is deleted.

Overriding a card source or target

You can override a card source or target with the source or target of another card; however, the card names cannot be overridden or changed. Links are created automatically between cards when a relationship is established.

To override an input or output card:

1. Place the focus on a system in the system window.
2. Double-click a map component to expand the view.
3. From the toolbox, click the Override tool and then click on the card source or target to copy.
4. Move the cursor to and click on the card you want to override.

The source or target of the first and second cards are now identical.

Chapter 15. Deploying systems

Using the Integration Flow Designer, you can create and run deploy scripts that automate the process of preparing systems to run on specific servers. For example, you can create a deploy script to build maps, generate system or command files, or transfer files to a designated server.

Using deploy scripts

Deploy scripts are only available for executable systems (systems with assigned servers).

To access deploy scripts

1. Place the focus on an executable system in the system window.
2. Select **System** → **Deploy** → **Definitions**.
If deploy definitions already exist for the system, they will appear in the submenu list.
3. Enter a name for the deploy script and click **OK**. The Define Deploy Scripts dialog is displayed.

The Define Deploy Scripts dialog lists the deploy scripts that have been defined. Clicking a deploy script name in the list causes the Script Commands group (at the bottom of the dialog) to be refreshed based on the definition of the selected script.

The Script Commands group has three commands with a check box next to each: Build and Transfer Maps, Generate and Transfer Files, Transfer Additional Files. Enabling a check box implies that the command will be executed when the script is run. Files are transferred by means of file transfer protocol (FTP) as binary data or ASCII text.

To configure a script command

Under **Script Commands**, do one of the following:

- Select the script command and click **Details**.
- Double-click the script command name.

To add a deploy script

1. Open the system that requires the deploy script.
2. From the **System** menu, select **Deploy** → **Definitions**. The scripts listed are associated with the system whose window was active prior to running the Deploy command.
3. When the Define Deploy Scripts dialog is displayed, click **Add**.
4. When the Add Deploy Script dialog is displayed, give the new deploy script a name and click **OK**. The script name must be unique within the active system.

To remove a deploy script

1. Open the system containing the deploy script to be removed.

2. From the **System** menu, select **Deploy** → **Definitions**. The scripts listed are associated with the system whose window was active prior to running the Deploy command.
3. Do one of the following:
 - Select the script to be removed, and click **Remove**.
 - Right-click the name of the script to be removed and select **Remove** from the context menu.

To rename a deploy script

1. Open the system containing the deploy script to be renamed.
2. From the **System** menu, select **Deploy** → **Definitions**. The scripts listed are associated with the system whose window was active prior to running the Deploy command.
3. Select the script to be renamed by clicking on its name and do one of the following:
 - When the **Define Deploy Scripts** dialog is displayed, click **Rename**.
 - Right-click the script to be renamed and select **Rename** from the context menu.
A Rename Deploy Script dialog is displayed to verify the action.
4. Click **OK** to rename the script.

Build and transfer maps

By using the **Build and Transfer Maps** script command, you can identify which maps to be built and to where they are transferred.

To build and transfer a compiled map

1. Select a system and select **System** → **Deploy** → **Definitions**.
The Define Deploy Scripts dialog is displayed.
2. Under **Deploy Scripts**, select a deploy script from the list.
3. Under **Script Commands**, double-click **Build and Transfer Maps**.
The Build and Transfer Maps dialog is displayed.
4. From the **Maps to Process** tab, select one of the two options:
 - The **All Maps** option (the default selection) to indicate that all maps in the selected system are to be built and transferred when this deploy script is run.
 - The **Maps Referenced Below** option to indicate that only the referenced maps in the list are to be built and transferred when this deploy script is run.

To use the Maps Referenced Below option

There are two drag-and-drop operations that you can use to move map references into the list: one from the **List** view and one from the system window. Instructions for both operations are provided in the following procedure.

To drag from the List view:

1. From the **List** view in the Navigator, click on a map and drag it to the component list on the **Maps to Process** tab.
2. When the mouse pointer displays a plus sign (+), release the mouse to drop the map component.

To drag from the system window:

To drag a map component from the system window, you must first ensure that the system window is not maximized. One way to make the active system window smaller is to go to the **Window** menu and select **Cascade**. Then you must arrange the open dialog boxes in such a way that they do not block your view of the map component in the active system window. Now you are ready to drag-and-drop from the system window.

1. From the system window, click on the map component and drag it to the component list on the **Maps to Process** tab.
2. When the mouse pointer displays a plus sign (+), release the mouse to drop the map component.

To modify the component list:

To modify the list of components in the Maps to Process tab, do one of the following:

- Select an item in the component list and right-click to view a context menu.
- Click a column title button to sort by ascending or descending order.

To specify the directory where the compiled map (.mmc) is transferred

From the Build and Transfer Maps dialog, go to the **Server Map Directories** tab.

On the **Server Map Directories** tab, the values in the **Server View** column are the directories of map server locations for the particular map components referenced on the **Maps to Process** tab.

Editing a path

The directories listed in the **Server View** column are for display only and cannot be edited. The **Integration Flow Designer View** column represents the target destination directory for the given source directory (under **Server View**) on the server.

The paths in the **Integration Flow Designer View** column can be edited either one at a time or by replacing text in a selected group of directories. Right-click to view the context menu.

One directory can be edited by selecting **Edit Integration Flow Designer View** (or by double-clicking a directory in the **Integration Flow Designer View** column), which allows inline editing directly in the list control. After the inline-editing mode is started, a box is displayed around the item in the list.

To edit multiple directories:

1. **Ctrl**-click each directory to include and right-click.
A context menu is displayed.
2. Select **Replace Integration Flow Designer View**.
The Replace dialog is displayed.
3. Enter the part of the selected string(s) to be replaced into the **Replace** field.
4. Enter the replacement string into the **With** field.
5. Click **OK** to modify the selected paths in the list.
If a directory cannot be changed based on the **With** field input, the directory stays highlighted and an error message is displayed.

Generate and transfer files

You can generate and transfer .msl or command files. Procedures for both are described in this section.

When deploying systems that contain maps with XML types to non-windows platforms, the XML Schema and XML input files must be transferred in binary format.

To generate and transfer a Launcher system file (.msl)

This procedure assumes that the execution mode of the active system is set to **Launcher**.

1. In the Define Deploy Scripts dialog, select the deploy script.
2. Under **Script Commands**, select **Generate and Transfer Launcher Control File** and click **Details**.

If the server is local, distributed, or undefined, the Event Server control file path on server dialog is displayed. (The Command file path on server dialog is displayed when using Command execution mode.) For this option, select or name the .msl file and click **Save**. This specifies the name and location of the .msl file that will be created by the deploy script.

If the server is completely defined, follow the prompts to select the target directory.

To generate and transfer a command file

This procedure assumes that the execution mode of the active system is set to **Command**.

1. From the active system, select **System** → **Deploy** → **Definitions**.
The Define Deploy Scripts dialog is displayed.
2. Under **Script Commands**, select **Generate and Transfer Command File** and click **Details**.

If the server is local, distributed, or undefined, the Command file path on server dialog is displayed. For this option, select or name the file and click **Save**.

If the server is completely defined, the Command control file path on server dialog is displayed. For this option, select the target directory and click **OK**. Click **OK** again from the Command file path on server dialog. This action specifies the name and location of the command file that will be created by the deploy script.

Moving a map component to the deploy dialog

Moving map components from a system window to the deploy dialogs is an easy drag-and-drop operation. A map component can also be moved from the Navigator to the dialog.

After the map component is moved to the dialog, the list is automatically refreshed.

Source map components as well as compiled map components can be moved to the dialog. When a script runs, the compiled maps are transferred. Pseudo maps cannot be moved; duplicates will be rejected.

Only map components that reside in the system being deployed can be moved into the dialog. For example, if you are defining a deploy script for system **ABC**, then only map components that are contained in the **ABC** system can be moved to the dialog.

Moving subsystems to the deploy dialog

Moving subsystems from a system window to the deploy dialog is an easy drag-and-drop operation. A subsystem can also be moved from the Navigator to the Build and Transfer Maps dialog. After the subsystem is moved to the dialog, the list is automatically refreshed.

Similar to map components, only subsystems that reside in the system being deployed can be moved to the dialog. If you are defining a deploy script for system **ABC**, then only subsystems that are contained in **ABC** can be moved to the dialog. If you open a system window unrelated to **ABC**, then subsystem components cannot be moved to the dialog.

If a subsystem has already been moved into the dialog, then any map component that the subsystem contains cannot be moved to the dialog. Also, if a map component has been moved to the dialog, then the subsystem that contains the map component cannot be moved to the dialog.

You can click one item in the list to select it or use the **Shift** and **Ctrl** keys to select multiple items. Right-click on one or more selected items in the list to display the following context menu.

If multiple items are selected, the **Go To Component** is disabled. If you select **Go To Component**, the system window where the component resides is displayed and the component is selected.

Transfer additional files

You can transfer files other than the compiled map, Launcher system file (.msl), or command file. For example, you can transfer a text file that contains notes about implementing the system. For each file, you can choose either an ASCII or binary transfer mode.

To transfer additional files

1. From the Define Deploy Scripts dialog, select **Transfer Additional Files** and click **Details**.

The Transfer Additional Files dialog is displayed.

2. Right-click on an item under **Source Location** or **Target Location** to view the context.
 - Selecting **Edit Source** or **Edit Target** allows inline editing of the source or target location. After the inline-editing mode is started, a box is displayed around the item in the list indicating that it can be edited.
 - Selecting **Replace Target Location** allows you to do a find-and-replace operation.
 - Selecting **Browse Target Location** allows you to browse for a target location.
3. To change the transfer mode from the default setting (ASCII) to binary, click on the **Mode** field, display the drop-down menu, and select **Binary** from the list.
4. Click **OK** to save changes and close the dialog.

Running a deploy script

After you create a deploy script, you can easily select and run it by selecting **System** → **Deploy**. Each deploy script listed on the submenu is associated with the active system.

To run a deploy script

1. Place the focus on a system in the system window.
2. From the **System** menu, select **Deployscript name**, where *script name* is the name of the predefined deploy script.

The script runs.

3. When the script ends, click the **Results** button to view all activity. In addition, all activities reported on the Deploy dialog are written to a log created in the installation directory that you can view at a later time.

If a deploy script fails, an error message is displayed.

To view deploy script results

Select **View** → **Deploy Results**.

The **Deploy Results** file is displayed in a text editor.

Chapter 16. Return codes and error messages

Return codes and error messages

This section contains return codes and error messages that result from an analysis of a system.

The System Analyzer checks the internal consistency of your systems to assure that they can be executed.

System analysis structure error messages

The following table lists the structure error messages that result from a structural analysis of a system:

Return Code	Message
-------------	---------

- | | |
|-------|--|
| SE100 | Map, <i>name1</i> , cannot be a component because system, <i>name2</i> , is distributed.
A distributed system cannot have maps as components. |
| SE101 | System, <i>name1</i> , contained in system, <i>name2</i> , must have a server.
Each component of a distributed system must be distributed or have a server assigned (i.e., the server for a component of a distributed server cannot be "None"). |
| SE102 | System <i>name</i> has unresolved settings that must be resolved - (error).
Unresolved settings must be resolved. |
| SE103 | Wildcards are used in map source/target name, but map will be run by Command Server - Map Component <i>name1</i> , Card Name <i>name2</i> (error).
Wildcards cannot be used in map source or target names if a Command Server will execute the map. |
| SE104 | Wildcards are used in directory name that is specified in source/target name - Map Component <i>name1</i> , Card Name <i>name2</i> (error).
Directories cannot have wildcards in their names. |
| SE105 | Wildcards are used in database source name - Map Component <i>name1</i> , Card Name <i>name2</i> (error).
A database source name cannot contain wildcards. |
| SE106 | Wildcard must be used in at least one source trigger because wildcards are defined for non-trigger sources and/or targets - Map Component <i>name</i> (error).
If wildcards are used as sources and/or targets, at least one source trigger must have a wildcard. |
| SE107 | Question mark wildcard is used in a target name - Map Component <i>name1</i> , Card Name <i>name2</i> (error).
Question marks cannot be used in target names. A target name can contain at most one asterisk. |

- SE108** Question mark wildcard is used in a source name that is not a trigger - Map Component *name1*, Card Name *name2* (error).
Question marks cannot be used in sources that are not event triggers. A source that is not an event trigger can contain at most one asterisk.
- SE109** Multiple wildcard characters are used in a source trigger name, but other source triggers are defined - Map Component *name1* (error).
Multiple wildcards can be used in a source trigger name of a map as long as there are no other source triggers.
- SE110** More than one source trigger name contains a wildcard character and the wildcard character used is different - Map Component *name1* (error).
If more than one source trigger name contains a wildcard, that wildcard must be the same wildcard.
- SE111** Wildcards are used in queue manager name and/or queue name specified for WebSphere MQ name - Map Component *name1*, Card Component *name1* (error).
- SE112** Source map component *name* references a non-existent MMS file - (error).
Unresolved MMS reference by source map components must be resolved.
- SE113** Compiled map component *name* references a non-existent MMC file - (error).
Unresolved MMC reference by compiled map components must be resolved.
- SE114** Subsystem component *name* references a non-existent system - (error).
Unresolved system reference by subsystem components must be resolved.
- SE115** A system contains itself as a component - System *name1* , Subsystem *name2* (error).
A system cannot contain itself as a component.
- SE116** Invalid *source/target name* - Map Component *name1* , Card Name *name2* (error).
Database, messaging and application adapters must have valid source/target names as checked by the individual adapter.
- SE117** Asterisk wildcard is used multiple times in a target name - Map Component *name1* , Card Name *name2* (error).
Only one wildcard per target name is permitted.

Systems using wildcards analysis structure error messages

Wildcards can be used to specify a variable name for a source or target. The asterisk and question mark characters are used according to the conditions described in the SE103 thru SE110 error messages detailed in "System Analysis Structure Error Messages" . The System Analyzer verifies that these conditions are satisfied. When a condition is not satisfied, it is flagged as an error.

Note: For more information about wildcards, refer to the Launcher documentation.

Return Code Message

SE500 Map component *name* has no triggers. [Warning].

Maps that will be executed by a Launcher should have a trigger, otherwise the map will never run.

SE501 System name has no components. [Warning].

A system should have at least one component.

SE502 Pseudo map component *name1* exists in system *name2* . [Warning].

Pseudo maps should be converted to source map components.

SE503 Triggers are defined for a map that will be executed by a Command Server - Map Component *name1* . [Warning].

Triggers should not be defined for a map that will be executed by a Command Server.

SE504 The execution mode of system *name* does not match the execution mode of subsystems that it contains. [Warning].

The execution mode specified for a system should be the same as the execution modes specified for its subsystems.

SE505 Drive letters should not be used in UNIX paths - Map Component *name1* , Card Name *name2* . [Warning].

Drive letters should not be used in UNIX paths.

Troubleshooting tips for error messages

"Not enough memory to execute" error message

Problem: A map is running on a UNIX platform but fails on a different platform displaying the following error message:

Not enough Memory to execute (2)

Solution: The most common reason is that the map being run is not built for that platform. Ensure that the executable map corresponds to the correct UNIX system where the engine is running.

"Version Mismatch" error message

If you deploy a system to an incompatible platform, the logged error message might incorrectly indicate a version mismatch, when it should instead indicate a platform mismatch.

The problem causing an incorrect error message is that the version is reverse byte ordering on UNIX (as opposed to Windows). To the Launcher, it is displayed as a version mismatch because it there is no indication that it was meant for Windows. The Launcher can only conclude that it is not appropriate for UNIX systems.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

i5/OS
IBM
the IBM logo
AIX
AIX 5L
CICS
CrossWorlds
DB2
DB2 Universal Database
Domino
HelpNow
IMS
Informix
iSeries
Lotus
Lotus Notes
MQIntegrator
MQSeries
MVS
Notes
OS/400
Passport Advantage
pSeries
Redbooks
SupportPac
WebSphere
z/OS

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (<http://www.eclipse.org/>).

WebSphere Transformation Extender, Version 8.1

Index

A

- align to grid option 18
- analyzing systems 62
- ASCII transfer mode 77
- audit log
 - for import/export 35

B

- backup on save setting 19
- binary
 - transfer mode 77
- build maps option 17
- build results
 - viewing 16, 61

C

- card names
 - finding 20, 22
 - specify settings for 22
- card settings
 - modifying 41
 - viewing 41
- command file
 - generate and transfer 76
 - generating 62
- comparing system definition files 15
- compiled map component
 - adding 38
 - defining a document link 71
- component attributes
 - finding 20
- component names
 - finding specific 21
- components
 - copying 16
 - deleting 16
- cut and paste tools 16

D

- database links 70
- deploy option 17
- deploy results
 - viewing 16
- deploy script
 - creating 73
 - removing 73
 - renaming 74
 - running 78
 - viewing results 78
- deploying systems 73
- distributed system 28
- document links
 - definition of 71

E

- edit menu 15
- error messages 79
- executable system 3, 73
- executable system files 62
- execution mode
 - overview of 59
- execution settings 51
 - defining 43
- external links 69
- external system 14

F

- file
 - closing 15
 - creating a new 15
 - differences 15
 - opening 15
 - printing 15
 - saving 15
- file menu 15
- filename extensions 7
- find and replace 16, 20
 - operations 20
- find operation
 - sorting results 21

G

- generate system 17
- grid 16, 18

H

- help menu 19

I

- implicit links 70
- incompatible platform 81
- initiation pending state 45
- input and output cards
 - viewing 41
- input card
 - overriding 72
- input or output card
 - tooltips 41
- Integration Flow Designer
 - theory of operation 25
 - using 5
- internal links 69

L

- Launcher settings
 - event coordination 47
 - pending instance thresholds 47

- Launcher system file
 - generate and transfer 76
 - generating 63
 - saving 63
- links
 - deleting pseudo 71
 - display and hide 71
 - document 71
 - external 69
 - for databases 70
 - implicit 70
 - internal 69
 - overview of 69
 - pseudo 71
- log See audit log 35

M

- map card settings
 - resolving 64
- map component
 - execution settings overview 43
 - naming conventions 43
- map components
 - moving 76
- map name
 - finding 21
- maps
 - build and transfer 74
 - building 61
- memory error message 81
- menu bar 15
- messages
 - source not available 45
 - target not available 45

N

- Navigator
 - overview 9
 - viewing 16

O

- options
 - accessing 18
- output card
 - overriding 72
- override card settings 72

P

- path names
 - finding 22
 - finding specific 22
- print
 - preview 15
 - setup 15
- process control files
 - generating 62
- pseudo links 71
 - definition of 71
- pseudo map component
 - adding 38
 - definition of 38

R

- recently used file list 15
- remote location 29
- remote procedure calls 29
- replace operation 22
 - starting 21
- return codes 79

S

- server
 - adding to a system 57
 - assigning 57
 - copying 57
 - drop-down list 12
 - duplicating 57
 - menu options 17
- server concepts
 - dependent systems on multiple servers 55
 - dependent systems on one server 55
 - independent systems on one server 55
- server definition
 - creating 56
 - deleting 58
 - editing 57
 - examples of 55
 - exporting 58
- server definitions
 - distributed 52
 - local 52
 - none 52
 - predefined 56
- server menu 17
- servers
 - distributed 28
- shared location 29
- shortcut
 - menu option 18
- single-server systems 37
- sockets 29
- sorting 21
- sorting find results 21
- source and target
 - overview of settings 27
 - types of 29
- source compare 64
- source control 15
- source map
 - creating from pseudo map 40
- source map component
 - defining a document link 71
- source not available
 - conditions for message 45
- status bar 16
- structure errors
 - system analysis 79
 - wildcards 80
- subsystem component
 - adding 53
 - characteristics of 51
 - execution settings 51
- subsystems
 - moving 77
 - overview of 27
- system
 - analyzing 17, 62

- system (*continued*)
 - copying 17, 33
 - creating 17, 32
 - deleting 17, 33
 - generating 17
 - renaming 17, 33
- system analysis
 - structure errors 79
 - wildcards 80
- system definition
 - exporting 33
 - importing 34
- system definition diagram
 - adding text 14, 23
 - moving text 23
 - selecting in 14
- system definition file
 - closing 32
 - comparing 66
 - creating 32
 - expanding 32
 - opening 32
 - saving 32
 - viewing differences 66
- system definition file differences 64
 - printing 66
- system file
 - executable 62
- system menu 16
- system window 12
 - opening 13
- systems
 - deploying 73
 - overview of 27
 - single-server 28

T

- target not available
 - conditions for message 45
- toolbox 14
 - add compiled map tool 14
 - add external system 14
 - add pseudo map tool 14, 38
 - add source map tool 14
 - add subsystem tool 14
 - add text tool 14
 - override tool 14
 - pseudo link tool 14
 - select tool 14
 - viewing 16
- tools menu 17
- tooltips
 - for expanded cards 41
- transferring additional files 77
- troubleshooting 79

U

- undo 16
- UNIX 81
- unresolved settings 17, 63
 - accessing 64
 - definition 63
 - deleting 64
 - displaying 64

- unresolved settings (*continued*)
 - processing 64

V

- version mismatch 81
- view menu 16

W

- window
 - cascading 18
 - closing all 18
 - splitting 18
 - tile 18
- window menu 18

X

- XML
 - server definitions 58
 - system definitions 33
 - types, deploying to non-Windows platforms 76



Printed in USA