Epicor Service Connect User Guide

EPICOR

Business Inspired™

Epicor Service Connect User Guide

About Epicor Software Corporation

Epicor Software Corporation is a global leader delivering business software solutions to the manufacturing, distribution, retail and services industries. With nearly 40 years of experience serving midmarket organizations and divisions of Global 1000 companies, Epicor has more than 20,000 customers in over 150 countries. Epicor enterprise resource planning (ERP), point of sale (POS), supply chain management (SCM), and human capital management (HCM) enable companies to drive increased efficiency and improve profitability. With a history of innovation, industry expertise and passion for excellence, Epicor inspires customers to build lasting competitive advantage. Epicor provides the single point of accountability that local, regional and global businesses demand. The Company's headquarters are located in California, with offices and affiliates worldwide. For more information, visit www.epicor.com.

Disclaimer

Epicor Worldwide Headquarters
4120 Dublin Boulevard
Dublin, CA 94568
Phone 949.585.4000
FAX 949.585.4091
http://www.epicor.com

Release 9.05.700
Distribution April 2012

Table of Contents

Introduction

This User Guide gives you an in-depth exploration of Epicor Service Connect, a business integration platform for secure workflow orchestrations within Epicor applications, as well as external connectivity to Epicor and non-Epicor applications. Users can automate tasks and processes within the application to promote lean principles, continuous performance initiatives, and Six Sigma quality within the organization or across the supply-chain. Service Connect lets employees focus on value-added activities and management by exception instead of repetitive data (re)entry tasks.

The focus of this guide is twofold: to explain Service Connect and to get you started with your own Service Connect integrations.

- Chapter 1 introduces you to the application and explains Service Connect concepts.
- Chapter 2 covers basic administration tasks, such as how to manage users and multiple Service Connect installations.
- Chapter 3 offers detailed information about the Administration Console and Task Manager. In this chapter, you will learn how to set up communication channels and route information to workflows.
- Chapter 4 covers the Workflow Designer, the tool at the center of Service Connect. Each available workflow activity is explained with examples and context for use.
- Chapter 5 contains step-by-step instructions that you can follow to integrate Service Connect with your Epicor solution.

Use this guide as a starting point to learn about Service Connect and as a reference for later use. This guide is a crucial resource for anyone who needs to leverage Service Connect for both managing and enhancing your organization's unique business practices.

Additional Service Connect documentation is available on EPICweb for Epicor 9, Enterprise Financials and Supply Chain Management, Epicor for Service Enterprises, and Epicor iScala. Each additional document contains start-to-finish instructions on how to create a sample workflow with your Epicor solution. A backup of the sample workflow is bundled with the documentation.

Chapter 1 Epicor Service Connect Overview

Epicor Service Connect (SC) is a workflow and application integration environment. You can use Service Connect to run a workflow within a single application or to run workflows that span multiple applications. Because it uses documents as its primary interface and leverages a Service Oriented Architecture (SOA), Service Connect simplifies the data conversion process from one application to suit the needs of other applications.

Service Connect uses open, industry-wide standards and technology such as:

- XML – Service Connect supports the syntax and semantics of XSLT 1.0 language limited by Microsoft XML Core Services (MSXML6)
- Web Services (Web Service Enhancements 2.0, 3.0 and Windows Communication Foundation®)
- Microsoft® .NET Framework

The emphasis of Service Connect is more on document exchange and less on the enforcement of business rules, although, it can do both. Service Connect is designed to convert business data by mapping structures and manipulating data through formatting, basic math functions, direct database operations, and other miscellaneous functions. To satisfy business rules in Service Connect, you can call out to web services, local .NET assemblies, or windows workflows and can use external interfacing (connectivity) to both Epicor and non-Epicor software applications.

The various ways you use Service Connect:

- When you enter a sales order into a customer relationship management (CRM) solution, such as Epicor Clientele, you can create an invoice and a purchase order in Epicor Enterprise. The invoice and purchase order use some of the sales order data, and some of the sales order data may be converted to match what the financial application expects.
- When you enter a new employee into the human resources (HR) system, you can send a request to set up a new account to the payroll application as well as send an email welcoming the employee to the company.
- You can create a new customer record using a spreadsheet and then run a query against the database to retrieve updated customer table data.
- When a credit hold is placed on a customer, you can send an email to the customer's sales account manager.
- You can regularly import queued external support requests into an automated customer support system.

This guide covers topics of interest to system administrators, consultants, and developers who are using or thinking of using Service Connect with Epicor or non-Epicor applications. This chapter describes the main Service Connect applications and the basics of what makes up a Service Connect solution.

The Service Connect Environment

The Service Connect environment consists of various server files, tools, and Windows services. You manage them, for the most part, in a Microsoft Management Console (MMC) called the Epicor Service Connect Administration Console. The Service Connect design environment consists of an application called the Workflow Designer that you use to create and modify workflows.

The Service Connect server can be on a different machine than the applications with which it interacts, or it can be on the same machine.

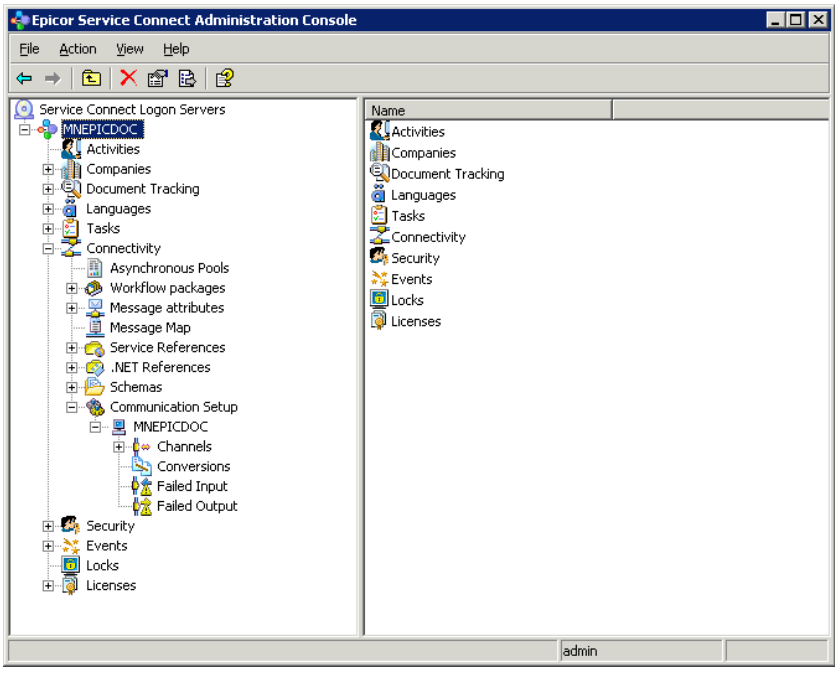
Service Connect Solutions

The basic components of a Service Connect solution and terms used throughout this guide are:

- **Documents** – XML files, CSV files, or Microsoft Excel files made available from an application. Use custom conversion plug-ins to work with other formats. The parts of an XML document are referred to as nodes.
- **Web Services** – Software components designed to support interoperable machine-to-machine interaction over a network.
- **.NET References** – You use .NET references to call .NET object methods inside workflows.
- **Database Operations** – You use these workflow elements to perform application database transactions.
- **Channels** – You configure channels in the ESC Administration Console to receive documents from an application and, optionally, to send documents to other applications. Channels that receive documents are referred to as input channels and channels that send documents are referred to as output channels.
- **Message Maps** – These route incoming documents to the appropriate workflow.
- **Workflows** – These orchestrate automated processes. Workflows manipulate documents to pass along data to another application, either by calling out to another application in the middle of the workflow or by producing a new document at the end of the workflow. A workflow is represented in the Workflow Designer as a diagram that consists of various activities. Each activity performs a specific operation, such as transforming a document to a new format or calling a web service to obtain information.

Epicor Service Connect Administration Console

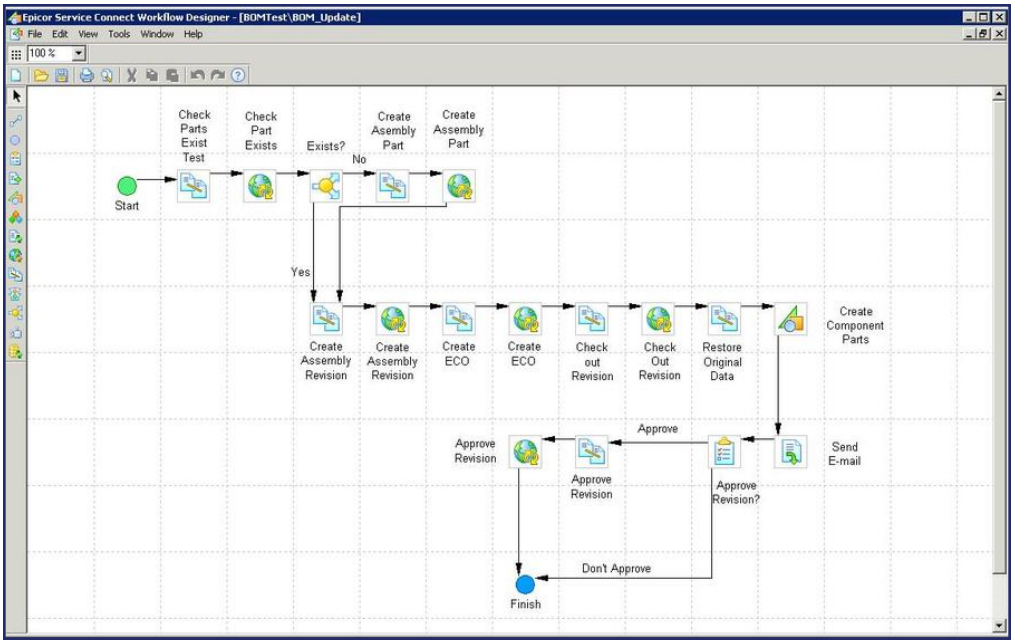
All system administration tasks can be performed from within the ESC Administration Console. This screen shows a single Service Connect installation managed on a server named MNEPICDOC.



Service Connect Workflow Designer

A workflow is a settings file, stored in XML, that directs and uses the various capabilities of Service Connect to manipulate the data sent to it. Some Epicor applications have Epicor-authored workflows that ship with the product.

This screen displays the Service Connect Workflow Designer with a complete workflow. The individual items, such as Create ECO and Send E-mail, are known as workflow activities or elements.



Service Connect Document Transformation

The most common use of Service Connect is to process a document by transforming its data for use in another application.

Two conversion types take place in Service Connect:

- Convert a document to a format Service Connect recognizes. This conversion type is explained in the Document Submission section.
- Convert a document so it can be used for business transactions in another application. This conversion type is explained in the Document Conversion section.

Document Submission

Three methods exist to submit documents to Service Connect:

- Use an input channel.
- Call a workflow directly from an Epicor application.
- Call a workflow exposed as a web service or through SC Integration Service web service.

When you use an input channel, documents are converted to the internal message format, if they are not already in the internal format, and then linked to a workflow. When you call a workflow directly from an Epicor application, the pre-update dataset of the calling business object is sent directly to the workflow. When you call a workflow exposed as a web service, the business object that makes the call must pass an XML document to the web service.

The Internal Message

Once a document is submitted to Service Connect, and until it leaves Service Connect, data must be converted to an internally recognizable format before Service Connect can process it. To meet this requirement, incoming documents are placed in the Service Connect internal message envelope. The internal message envelope is an XML document that encapsulates the original document in a single XML node called the business data (dta) node. The other message nodes contain data for routing, error handling, and tracing. Review the next section, Internal Envelope Structure, for details about each node.

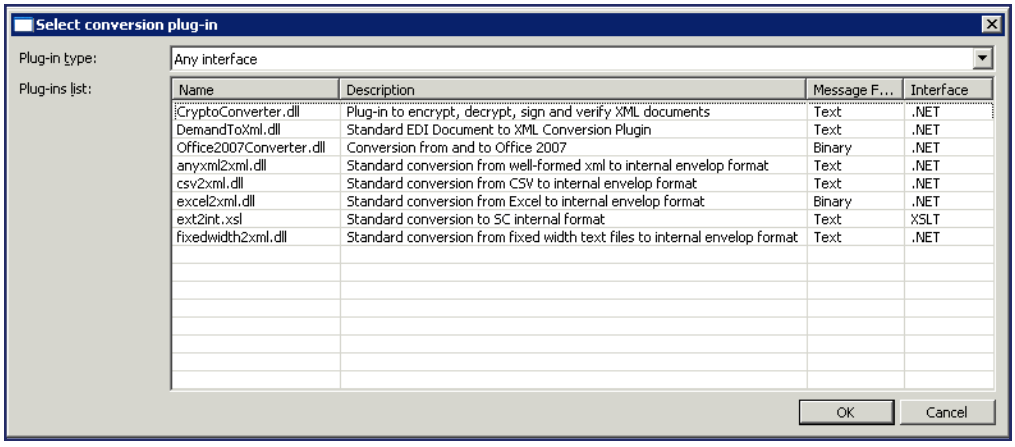
Documents are wrapped in the internal envelope at one of the entry points - an input channel or a workflow exposed as a web service. In general, most integration scenarios use input channels because they are less complex than calling workflows directly or calling workflows exposed as web services.

Input channels are configured in the Epicor Service Connect Administration Console. When you configure an input channel, you select the transfer protocol of the channel, such as MSMQ, and its connection details. You also select a conversion type appropriate for the document that is received. Conversion plug-ins available include standard XML file to ESC internal message, CSV file to ESC internal message, Excel file to ESC internal message, ESC external message to ESC internal message, Office 2007 file to ESC internal message, fixed width text file to ESC internal message, and cryptographic conversion. The converter interface is documented, so you can develop new converters for various data formats.

The Service Connect external message envelope is a schema that is available to format documents either before they enter Service Connect or after they leave Service Connect. Conforming to the external schema, either for incoming or, more commonly, outgoing documents, is not required to use Service Connect. The external schema is a suggestion of a schema to use if you need to supply or receive the Service Connect data that would be exposed by using it. Conversion plug-ins to convert the external message to an internal message, and vice versa, are available.

If the document is already in the internal format, you do not need to select a conversion plug-in. The input channel configuration also involves entering values used to populate the internal message metadata. When you set this metadata, such as Sender or Message Type, Service Connect routes an incoming document to the appropriate workflow. Values for this metadata are defined in the Message attributes node of the Epicor Service Connect Administration Console, so you can select them when you configure an input channel.

This screen displays the input channel configuration options for conversion plug-ins.



Using a workflow exposed as a web service as the entry point for a document requires more custom work than using input channels. It also requires that any document submitted to the workflow must be an XML document. Service Connect uses Internet Information Services (IIS) to expose its web services. Once published, the web services can be consumed in applications such as Microsoft InfoPath®, BizTalk®, Visual Studio® 2010, or another instance of Epicor Service Connect. When you call a workflow that has been exposed as a web service through its Execute web method, you send parameters that allow the document to be wrapped in the ESC internal message envelope.

Internal Envelope Structure

The following table displays the main nodes in the internal envelope structure of a Service Connect document.

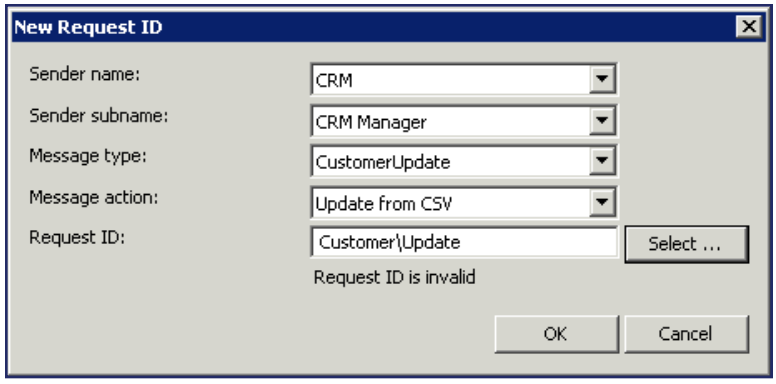
Node	Description
msg	The top level node of the internal envelope XML document.
req	The request node. This node contains the **dta**, **ers**, **wfl**, **ctx**, and **cfg** nodes.
dta	The business data node. This node holds the data used for most processing, such as stock items or sales orders. Information that enters Service Connect is stored in this node. The data displays as child nodes.
ers	The error messages node. If Service Connect or a web service returns an error, the error number and description display as child nodes.
wfl	The workflow data node. This contains the **usr** node, plus internal information, such as the message type and sender, which is used for document processing.
usr	The user node. This node contains message extensions and process variables. Each message extension and process variable displays as a child node. Message extensions and process variables are custom data containers that can be used to store values in a workflow until the information is ready to be used as part of a business process. Review Chapter 4: Workflow Designer for more information about message extensions and process variables.
ctx	The element configuration node. Values in this node are defined by the incoming document or process properties settings.
cfg	A second configuration node. This node can contain the same configuration data as the **ctx** node, but has lower precedence.
trc	The tracing information node. This node is used for internal purposes.

Workflow Linking

When a workflow exposed as a web service is used for the entry point for a document, linking to a workflow is direct since you are calling it specifically. However, as noted before, this is a less common way to submit documents. You can also call a workflow directly from an Epicor application, such as Epicor 9, Vantage, or Epicor Portal. Each Epicor product that calls a workflow directly uses its own system to access the workflow. Many of these systems are described later in this user guide. The following paragraph describes the most common method to submit documents to Service Connect, through input channels that are mapped to a workflow.

As noted previously, user-defined message attributes can be added to documents as they arrive in input channels. Or, if the document from the sending application is already in the internal message format, the sending application could have added the appropriate message attributes. Upon arrival in the input channel, Service Connect evaluates the message attribute combination in a document and selects a message map that has the matching set of attributes. The message map, in turn, has a definition for which workflow to run. Thus, a message map is a specific combination of message attributes that point to a specific workflow. This is how, for example, you can get a document that represents an update routed to the workflow that performs an update.

This screen displays a message map.



Document Conversion

After a document is submitted (converted to an internal message and linked to a process), Service Connect can transform the business data while it is inside a workflow.

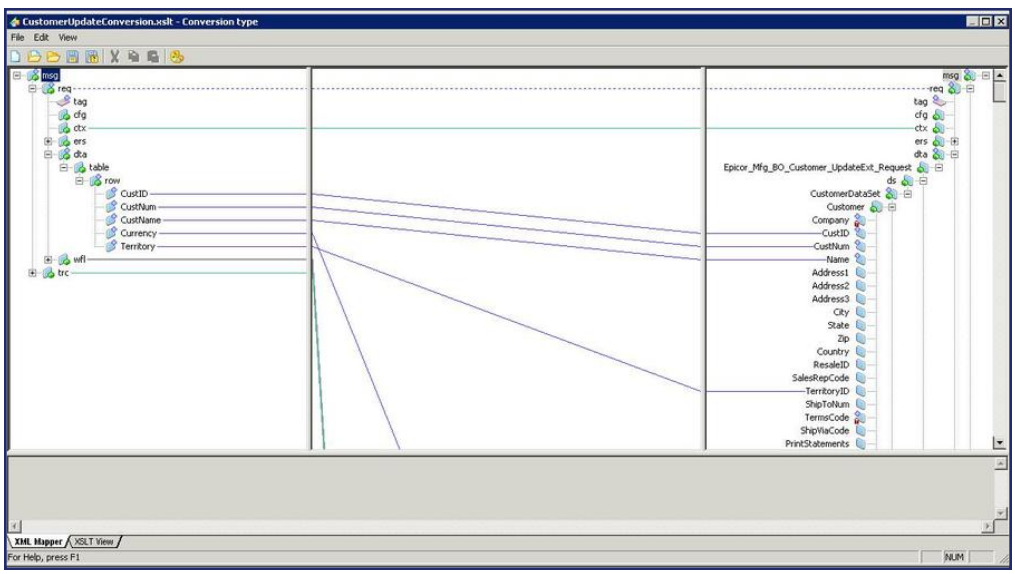
The Workflow is the key component of Service Connect. A workflow organizes a sequence of automatic activities or user-performed tasks to perform on a Service Connect internal message. In general, the goal of a workflow is to transform data via conversion, web services, .NET assemblies, and database operations so another application can use it. Other business activities are also supported such as tasks and conditional flows. Every workflow has a starting point and an ending point. The starting point begins with the message sent to the workflow and the ending point ends with a transformed message. Workflows are organized into folders called Workflow packages.

A Service Connect workflow is stored in a single XML file that is saved, by default, on the Service Connect server in C:\Program Files\Epicor Service Connect\System\Services\DES\Processes\Custom\Packages\<Package Name>\<Workflow Name>. The workflow XML file may depend on XSLT transformations and e-mail templates. XSLT transformations are located in a Transformations folder where the workflow is saved, and, by default, e-mail templates are located in C:\Program Files\Epicor Service Connect\System\Services\DES\EmailTemplates.

The workflow XML file is built and maintained using the Workflow Designer. The Workflow Designer uses a graphical user interface that allows you to place workflow activities into a diagram that represents the workflow process. Every document sent to a workflow has been converted to the internal message, which is XML, so all actions within a workflow are XML to XML operations. Workflow activities generally perform operations on the business data (dta) node of the message.

The Conversion is the most commonly used workflow activity. The Conversion activity uses Extensible Stylesheet Language Transformation (XSLT) to transform a document from one format to another. When you use a Conversion, you can use the XML Mapper utility to create an XSLT file that maps and transforms data. The XML Mapper utility includes components called functoids, which are pre-packaged functions that simplify the execution of common transformations for field values, such as string formatting and basic math functions.

This screen dipslays the XML Mapper utility.

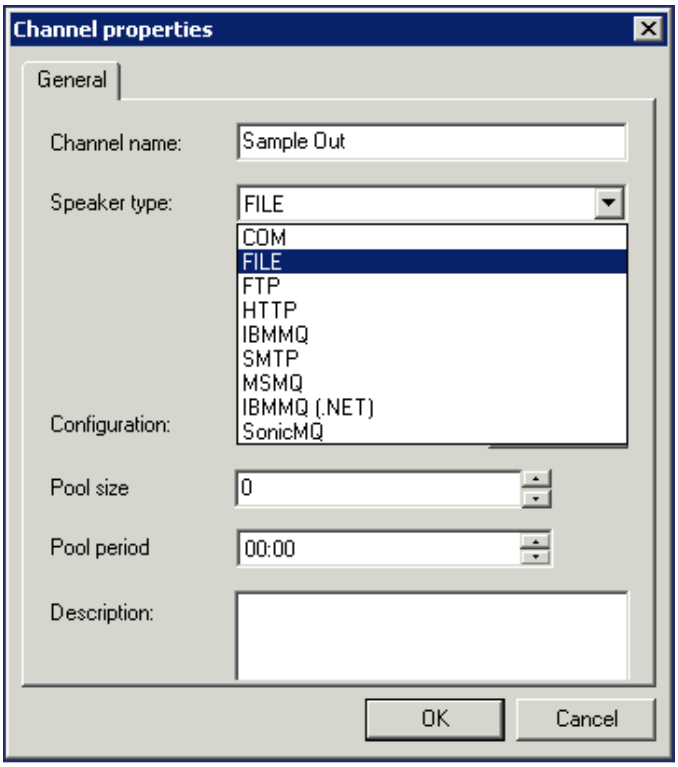


The Task and Web Method activities are also commonly used. The Task activity pauses a process and alerts particular users so they can decide how the process should proceed. All tasks are queued in the browser-based Task Monitor. The Web Method activity allows the sending and receiving of business data to web service methods that have been configured in the ESC Administration Console. Review Chapter 4: Workflow Designer for detailed descriptions of the workflow activities.

Transformed Data Availability

Once the business data is transformed by a workflow, several options exist to make the transformed data available to integrated applications including:

- **Poster activity** – You can specify an output channel to which to send messages. For output channels you can configure a transfer protocol, such as MSMQ, set up channel connection details, and select an appropriate conversion to apply on theinternal message. Conversion options include stripping the ESC internal message envelope or converting to the ESC external message envelope. You can also send the message along in the ESC internal envelope.
- **Requester activity** – This is similar to the Poster activity, except the Requester waits for a response after a message is sent so additional actions can be taken inside the workflow. This screen displays the protocols available for an output channel speaker type.
- **Web Method activity** – You can send XML data to a web service method. The web service response is also available inside the workflow.
- **Windows Workflow Foundation (WF)** – This option is a Microsoft technology for defining, executing, and managingworkflows. WF Foundation is part of .NET Framework 3.0. You can integrate Service Connect with WF Foundation workflows created in MS Visual Studio 2010. You can call WF workflows as sub-workflows inside Service Connect workflow processes. Use the WF Workflow call activity to send XML data from the ESC workflow to a WF workflow. You can set up the ESC workflow to consume information that the WF workflow returns.
- **Sub-Workflow activity** – You can send a message to other workflows which can then act as a subroutine for the main workflow. For documents that contain multiple records, you can design a Sub-Workflow to cycle through each record. If you called the Execute method of a workflow exposed as a web service through your own custom built application, the Execute method returns the message that hits the end point of the workflow, or the message that stopped at a Task activity.



- **.NET Call activity** – You can use the standard Workflow Designer engine to call any method of any .NET object registered in a workflow. Use this feature to extend the scope of the functionality within the workflows and to create the functionality in any .NET language.
- **DB Operation activity** – You can perform SQL statements against the application database. Once you establish the connection, you can create and run the SELECT, UPDATE, INSERT, and DELETE operations against the database.

Summary

This chapter introduced Service Connect, the main pieces that make up Service Connect, and the basics of how to construct a Service Connect solution. The next chapter describes the administration aspects of running a Service Connect installation.

Chapter 2
Epicor Service Connect Administration

This chapter describes user management for Service Connect administrators and developers, Service Connect server management, and connectivity settings for backups and restores.

You will learn how to log in to Service Connect, change existing accounts, and set up new users. In Epicor Service Connect you can import multiple Windows users and multiple users from a file.

The Installation Management section of this chapter explains how to register Service Connect servers in the Epicor Service Connect Administration Console, manage user sessions and Service Connect services, and set up Events logging functionality.

The Connectivity Administration section shows how to backup connectivity settings, such as message maps, message attributes, workflows, user schemas, and so on, and how to restore them.

User Management

This section describes how to change default passwords for created administrative accounts, set up new user accounts, and add multiple users.

Service Connect user account management, when used with Epicor iScala, is not covered in this chapter. Refer to the online Service Connect documentation, the iScala documentation, and the iScala supplementary guide available on EPICweb for more information on how to set up users with iScala.

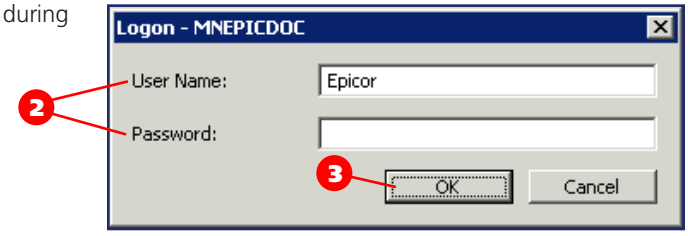
Online Service Connect documentation does not have the Company Specific User properties help topic.

Log in to Service Connect

Use the instructions in the Epicor Service Connect Installation and Implementation Guide to install Service Connect. The installation program creates two administrative user accounts - Epicor and Admin. During installation, you have the option to set the passwords for these accounts or leave the passwords blank.

To log on after Service Connect is installed:

1. From the **Start** menu, select **All Programs > Epicor Software > Epicor Service Connect > Service Connect Administration Console**.
2. In the **User Name** field, enter **Epicor**. Enter the **Password** set during the installation process.
3. Click **OK**.



You can also log in with the Admin account.

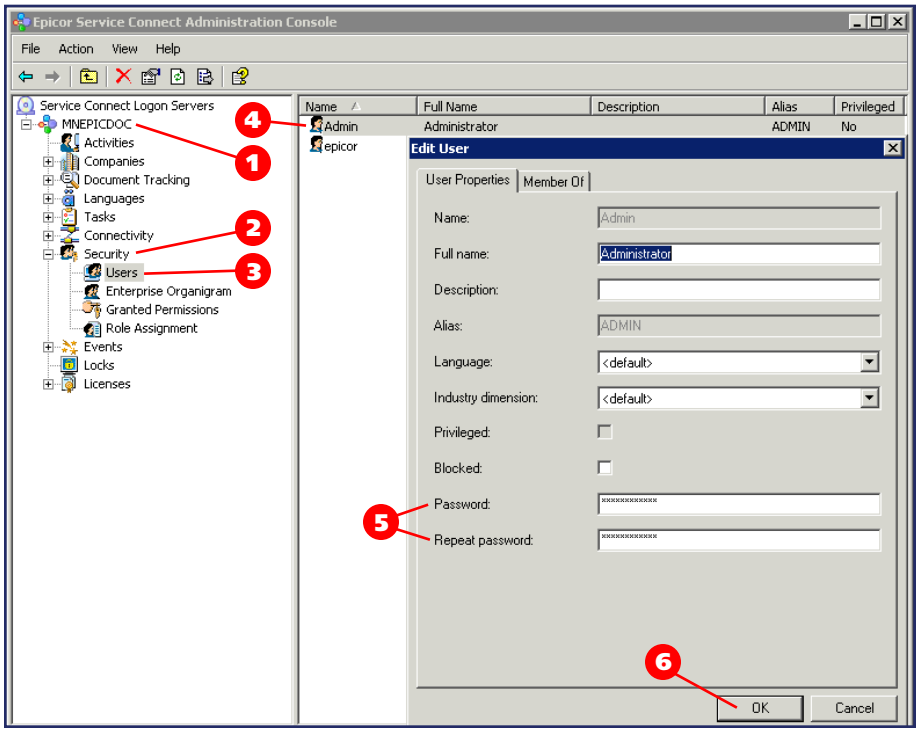
Change the Admin and Epicor User Passwords

If you left the passwords for the Epicor and Admin users blank during the installation process, you should change them to keep your server secure.

To change the passwords:

1. In the **Tree View**, expand the computer name node of your Service Connect installation.
2. Expand the **Security** node.
3. Click **Users**.
4. Double-click **Admin**.
5. Enter your new **Password** and enter it again the the **Retype password** field.
6. Click **OK**.

Repeat these steps to change the **Epicor** user.



Set Up a User

When you set up Service Connect to use with applications other than iScala, you need to set up user accounts for users who require access to the browser-based Task Monitor, for administrators who will be working in the Epicor Service Connect Administration Console, and for developers who will be working with the Workflow Designer. To do this, you can use the built-in System Administrator and Workflow Administrator roles.

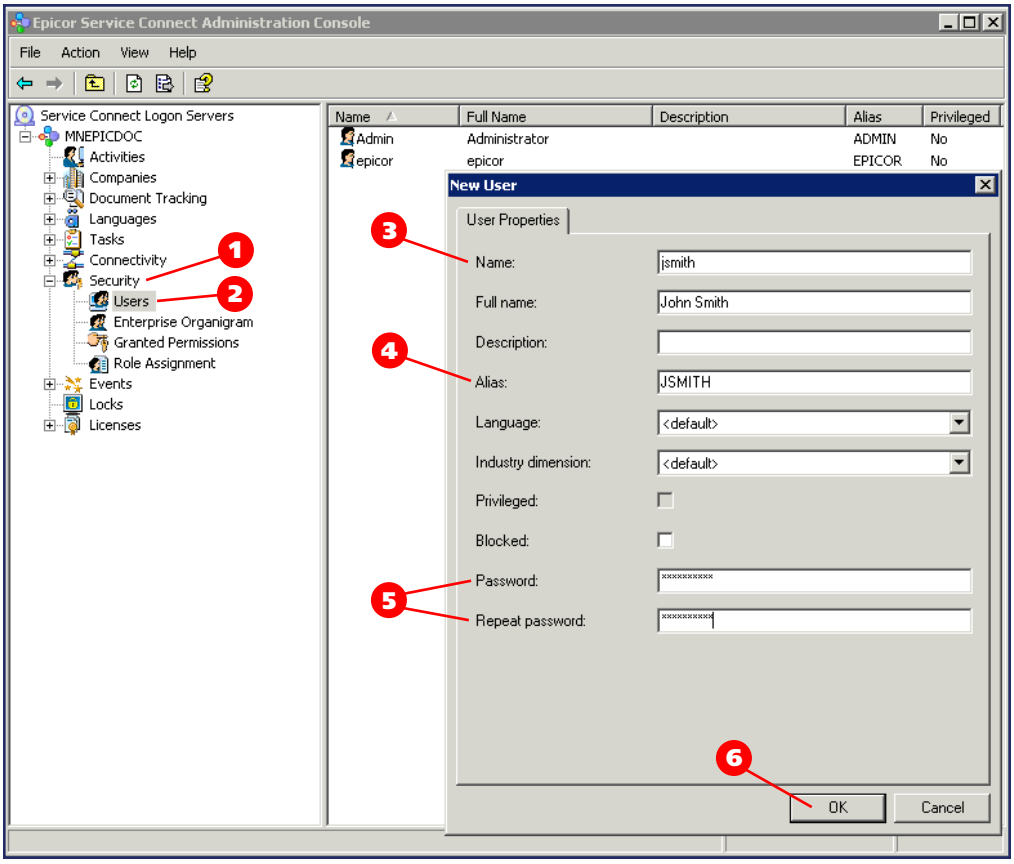
Create new user accounts and assign built-in roles to them.

To create new use accounts:

1. In the Tree View, expand the **Security** node.
2. Right-click **Users** and select **New User**.
3. The **New User** window displays. Enter the user **Name**.
4. Enter an **Alias**, which is either the same name or a different name.

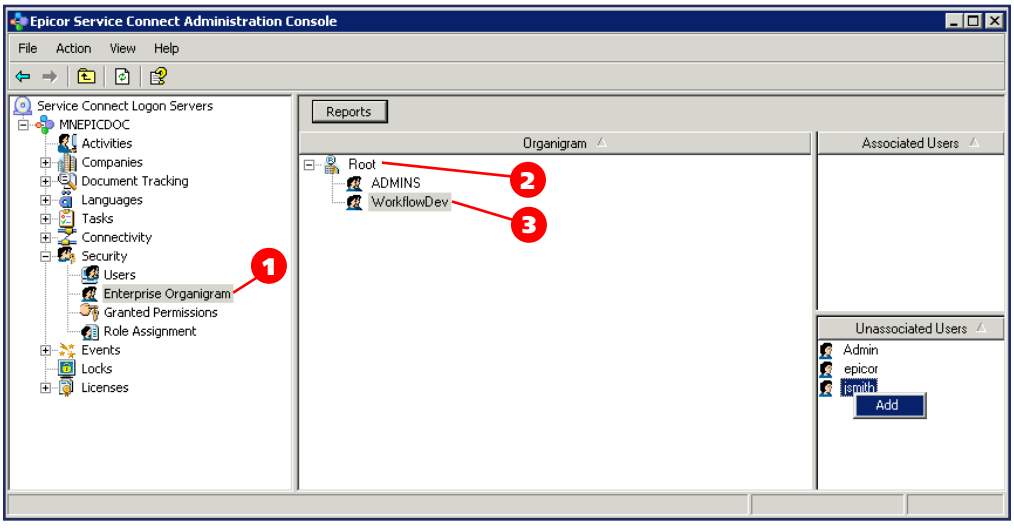
   Only iScala uses the Alias field. For more information, refer to iScala documentation.
5. Enter your new **Password** and enter it again the **Retype password** field.
6. When you finish entering the user properties, click **OK**.

   Repeat these steps to add other users.

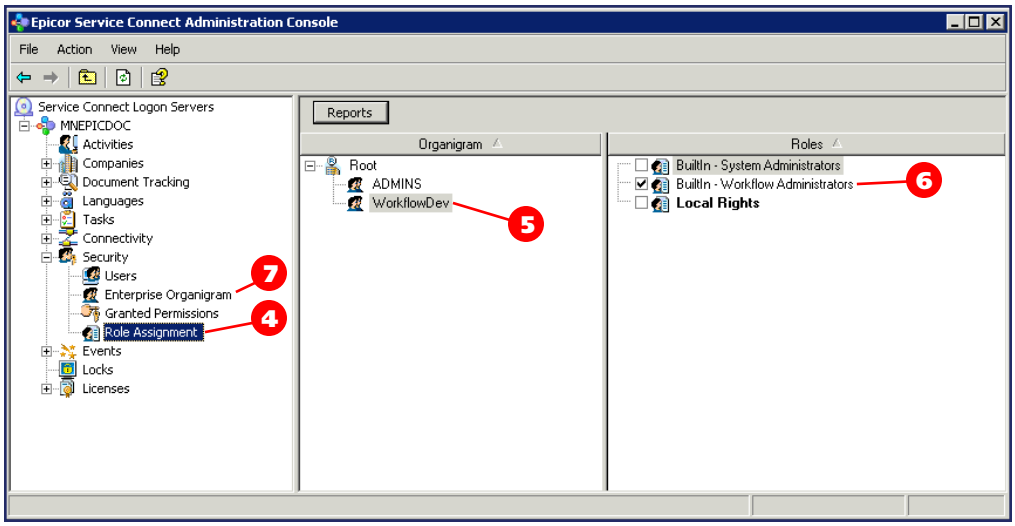


To set up user accounts using built-in roles:

1. In the **Tree View**, click **Enterprise Organigram**.
2. Right-click **Root** and select **Add**.
3. Enter **WorkflowDev** to replace the default NewGroup1 text.

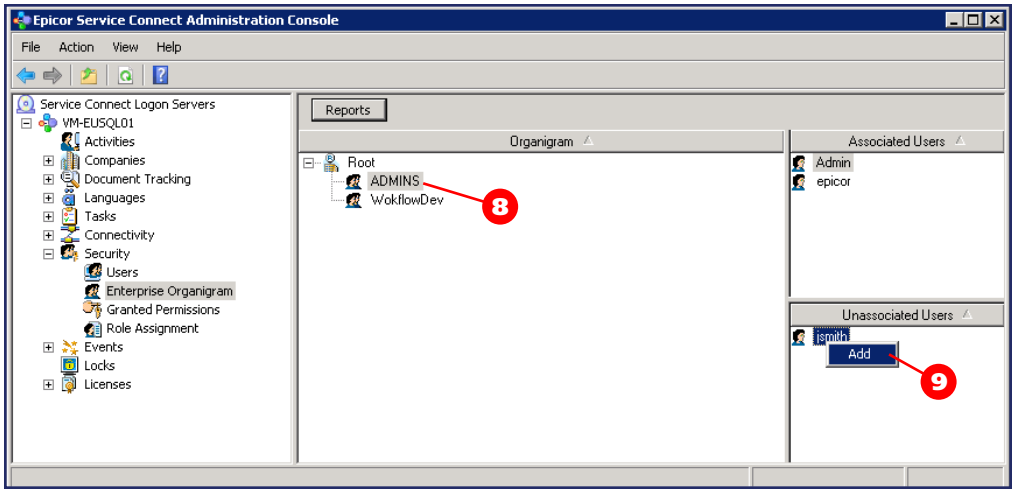


4. Click **Role Assignment**.
5. In the **Organigram** column, select the **WorkflowDev**.
6. In the **Roles** column, select the **BuiltIn – Workflow Administrators** role.
7. Click **Enterprise Organigram** again.



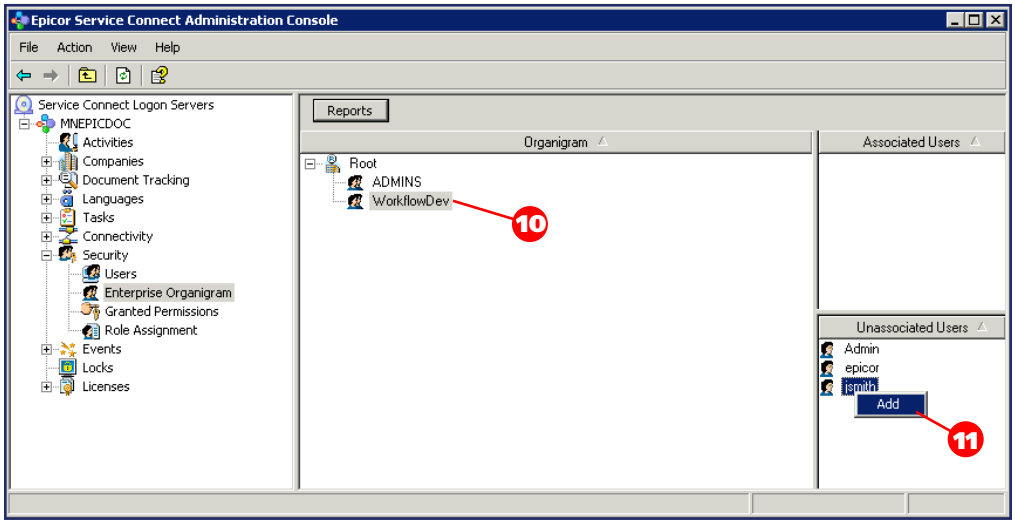
8. Under the **Root** node, click **ADMINS**.
9. Under **Unassociated Users**, right-click the user you want as a system administrator and select **Add**.

   Repeat this step for all users you want as system administrators. These users are able to do anything in both the ESC Administration Console and the Workflow Designer.



10. Under **Root**, click **WorkflowDev**.
11. Under **Unassociated Users**, right-click the user you want as a workflow developer and select **Add**.

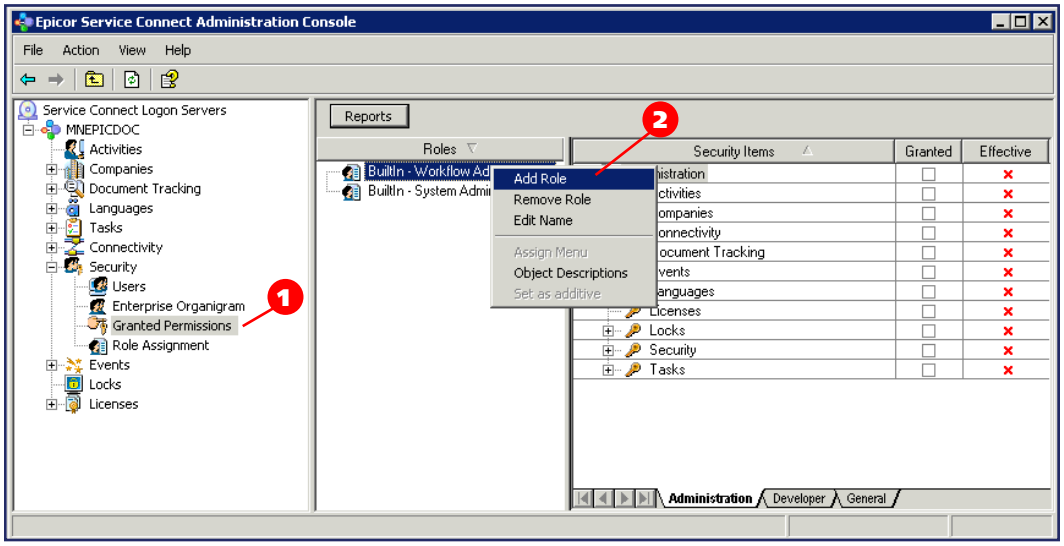
    Repeat this step for all users you want as workflow developers. These users are able to do anything in the Workflow Designer.



Grant Permissions

1. If you need to create your own unique roles for more restricted privileges, click **Granted Permissions** under the **Security** node.

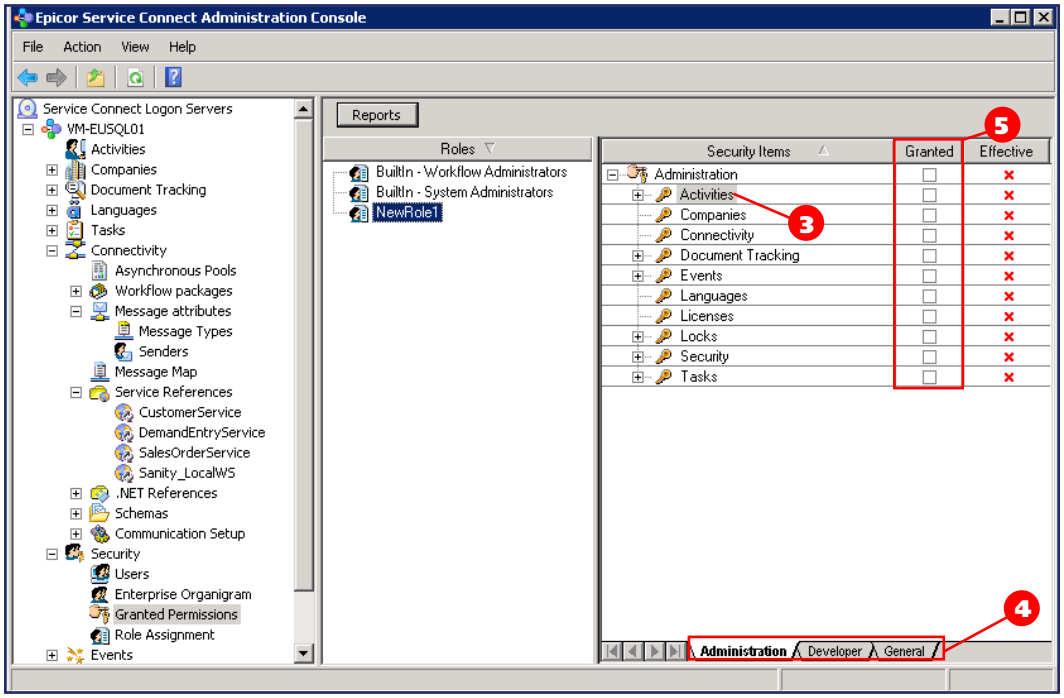
2. In the **Roles** column, right-click a role and select **Add Role**.



3. Highlight the new role.

4. On the bottom right, review the tabs, which group different categories of security items.

5. To grant rights to the selected role, in the **Granted** column, select the check box for the specific security items you want.

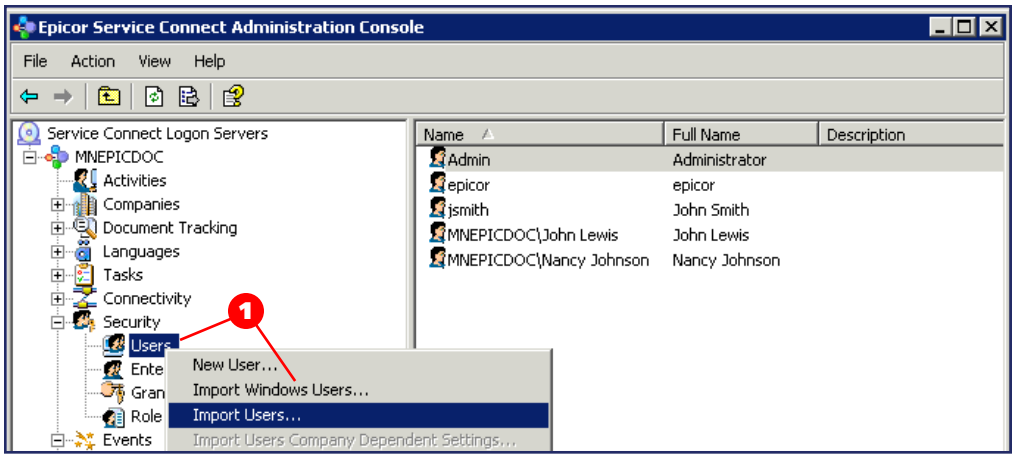


Add Multiple Users

When you set up Service Connect, you can add multiple users at the same time. Use this function to import users from a file or to import Windows users. Windows users are able to log on automatically, that is, they are not prompted for credentials based on a Windows account. Automatic logons only work when the Authentication setting for the installation uses the default setting of Windows & internal.

To add multiple Windows users:

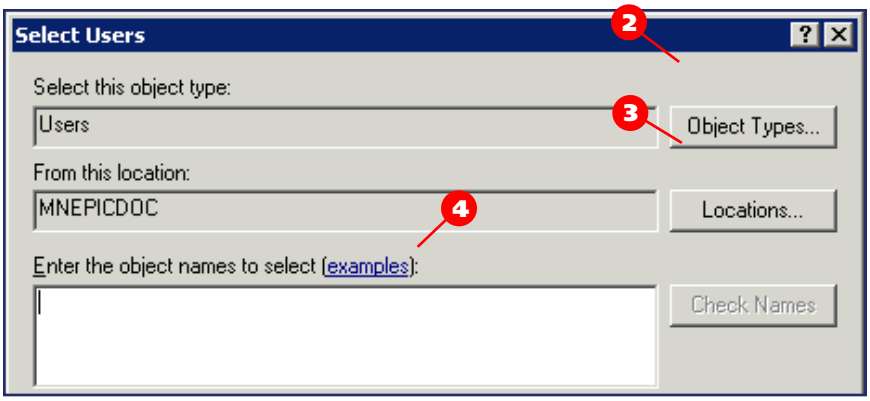
1. In the **Tree View**, right-click **Users** and select **Import Windows Users**.



2. The **Select Users** window displays. Click the **Object Types** button to find and select Users.

3. Click the **Locations** button to find and select the object type location.

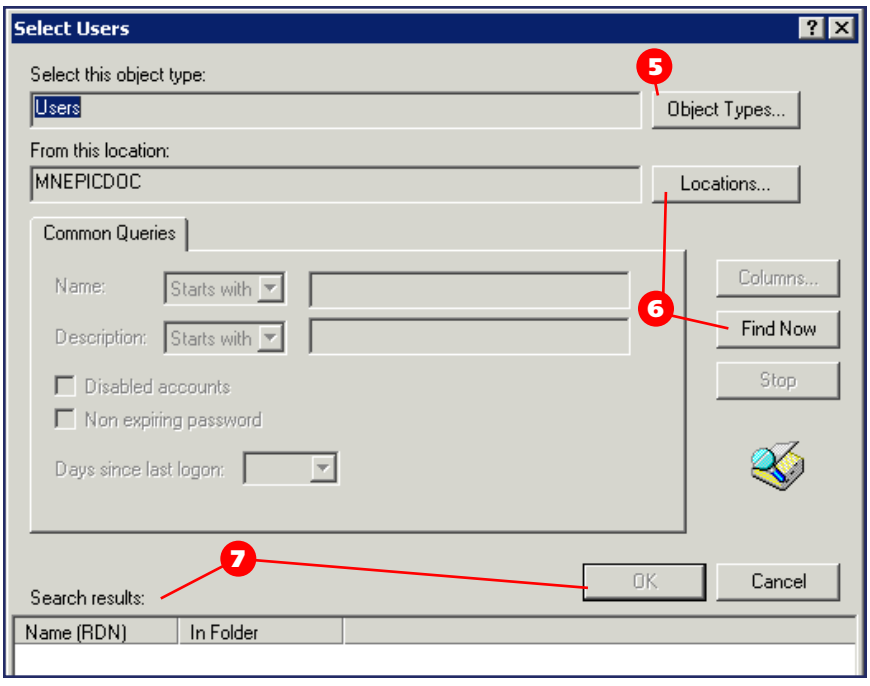
4. Click the **Advanced** button to find and select the object names. You can also **Enter the object names to select** in the field.



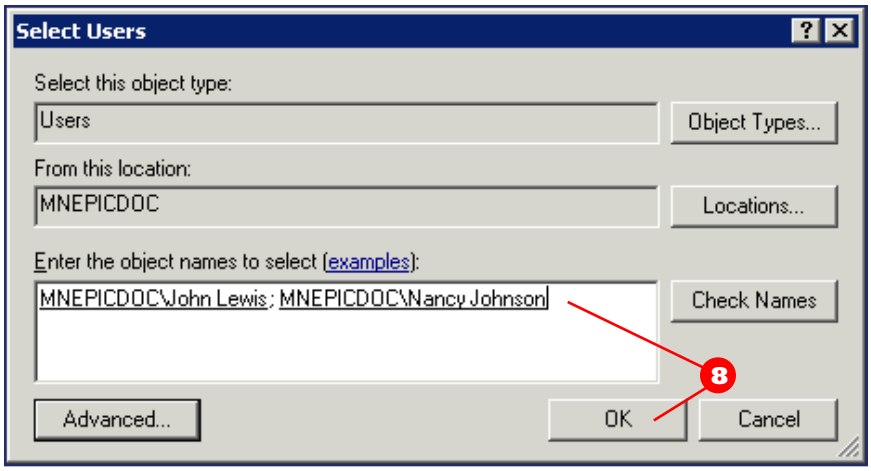
5. If you clicked the **Advanced** button and the **Select Users** Search form displays, click the **Object Types** button to find and select **Users**.

6. Click the **Locations** button to find and select the object type location. and click **Find Now**.

7. From the **Search Results** grid, select the users you need and click **OK**.

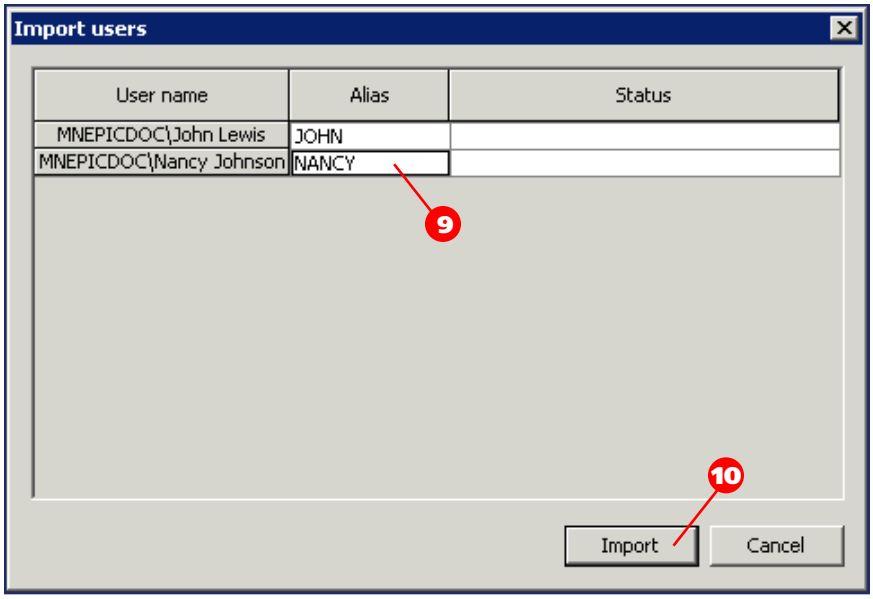


8. Your selected users display. To confirm your selection, click **OK**.



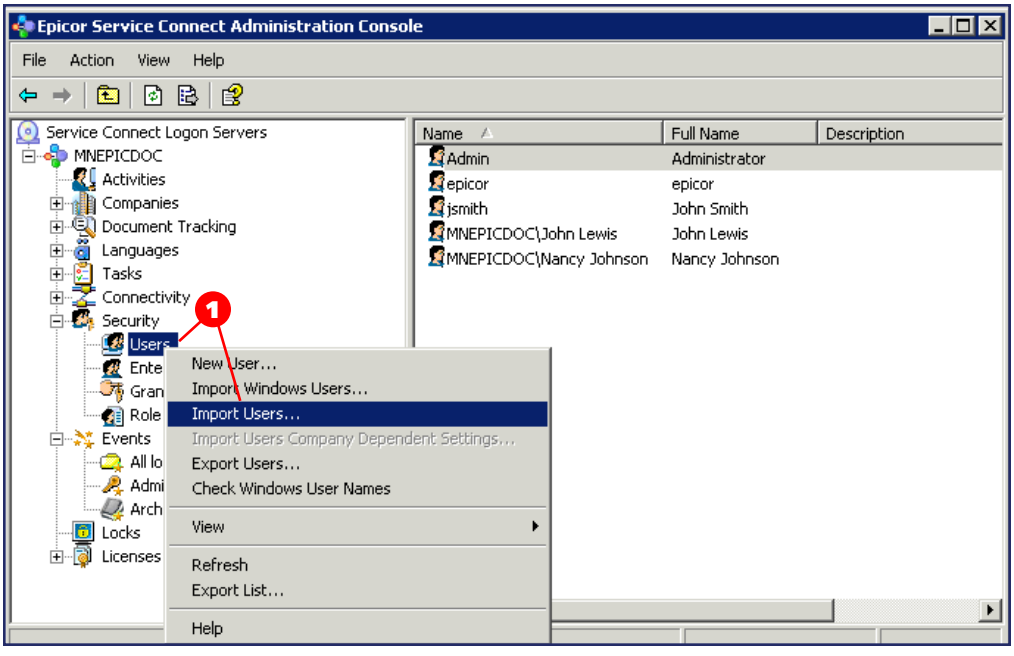
9. The **Import users** window displays. Enter an **Alias** for the users.

10. Click **Import**.

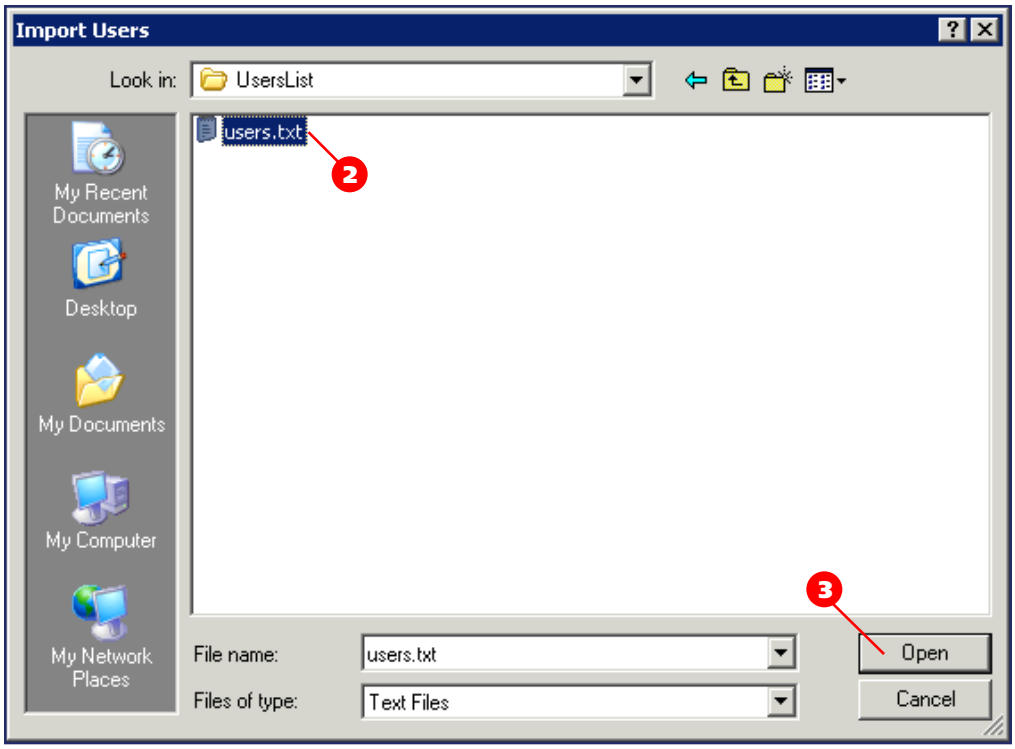


To import a list of users from a file:

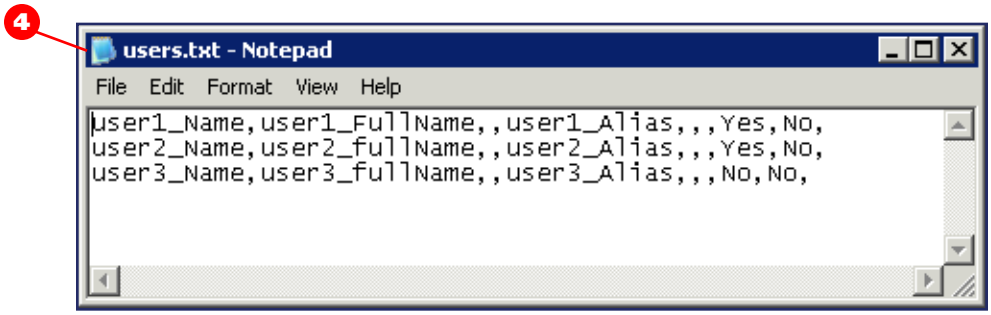
1. In the **Tree View**, right-click **Users** and select **Import Users**.



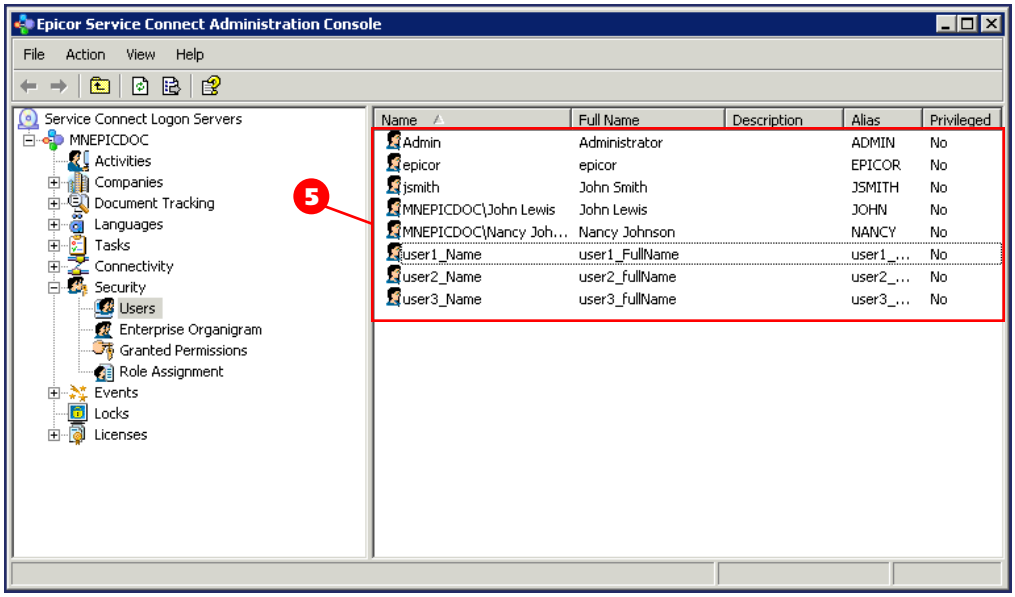
2. Select the text file that contains the user list you want to import.
3. Click **Open**.



4. This example shows the import file.



5. The imported users display in the right pane.



Service Connect Installation Management

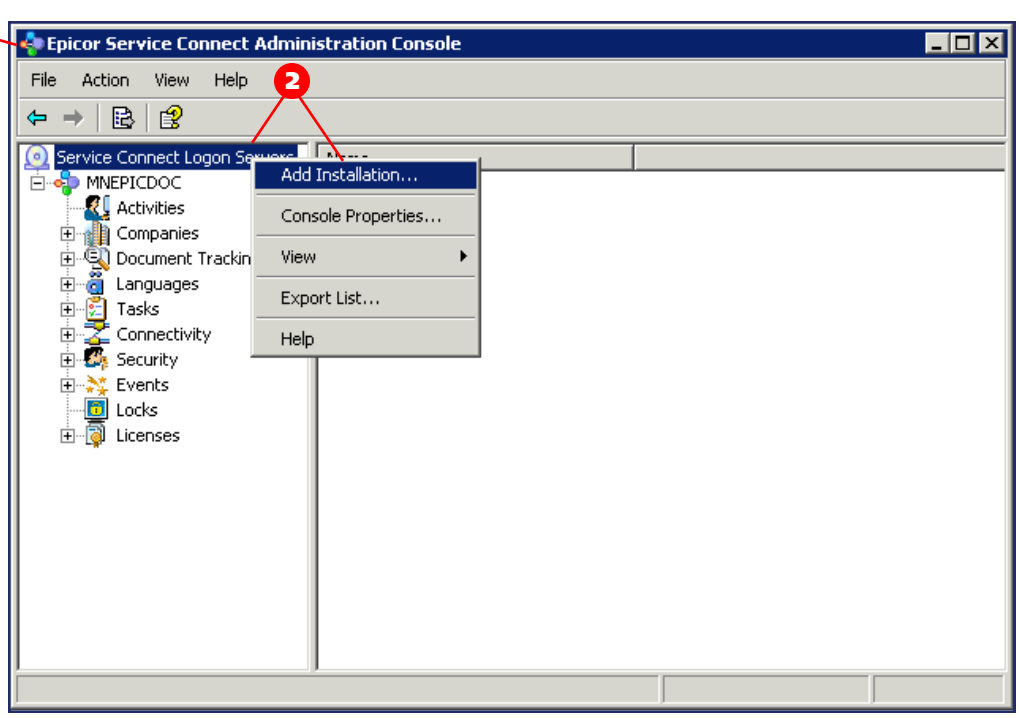
This section describes common installation-wide settings and procedures.

Register Servers

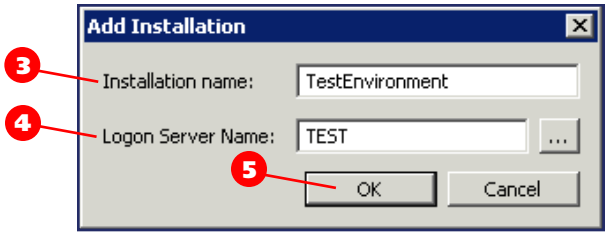
You can install Epicor Service Connect Administration Console on computers that are not the Service Connect server. You can then register one or more Service Connect installations to administer.

To register a Service Connect server:

1. Open the **Epicor Service Connect Administration Console**.
2. In the **Tree View**, right-click **Service Connect Logon Servers** and select **Add Installation**.



3. Enter an **Installation name**.
4. In the **Logon Server Name** field, enter the computer name of the Service Connect server.
5. Click **OK**.



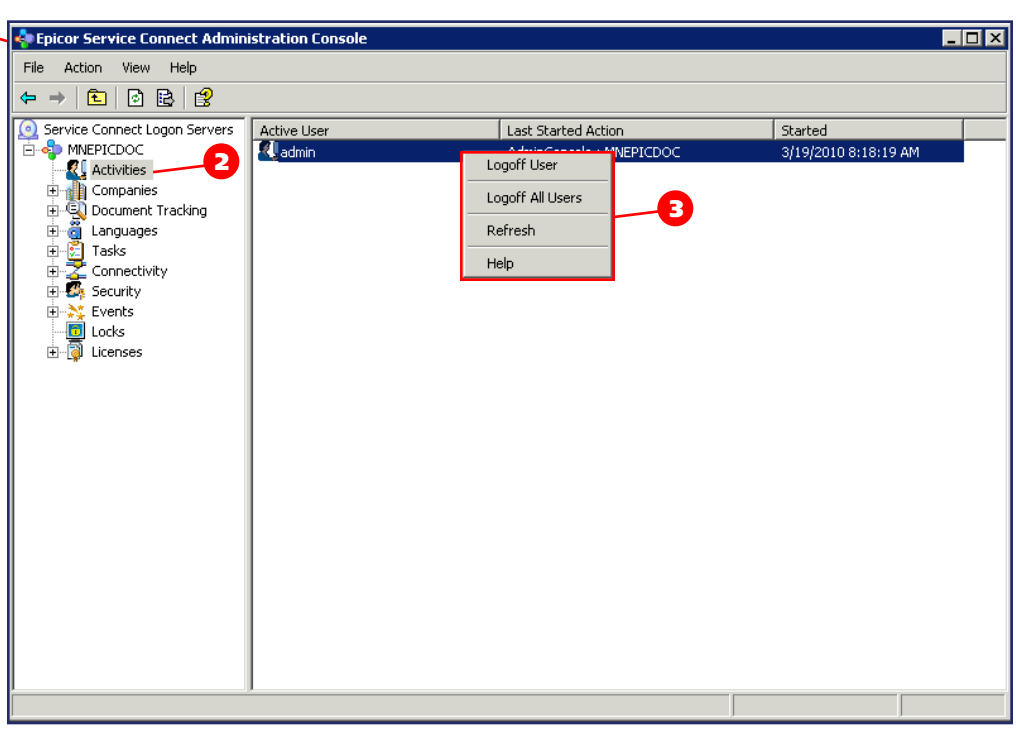
Manage User Sessions

You can view users who are logged in to the ESC Administration Console or Workflow Designer to see their status and log them off if necessary.

To manage user sessions:

1. Open the **Epicor Service Connect Administration Console**.
2. In the Tree View, click **Activities**.

   Active Users and their Last Started Actions are listed.
3. Right-click a user to display the context menu options: **Logoff User**, **Logoff All Users, Refresh,** and **Help**.



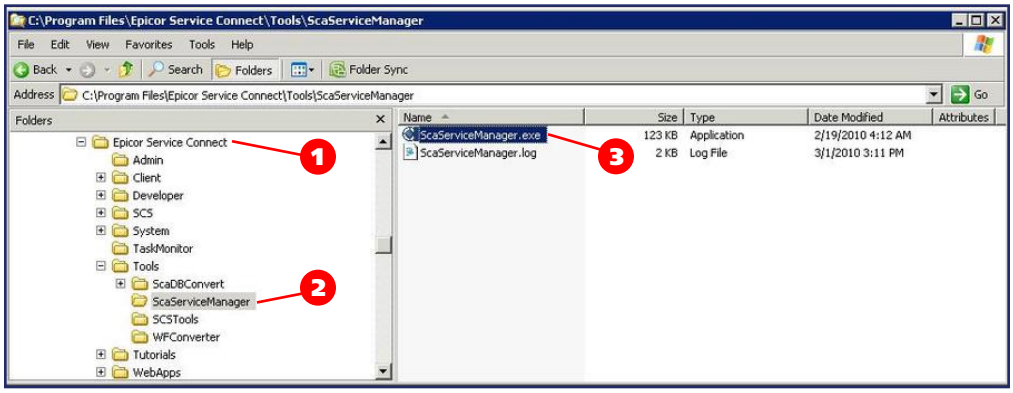
Manage Service Connect Services

Use the Service Manager utility to manage the services associated with Service Connect.

To start the Service Manager:

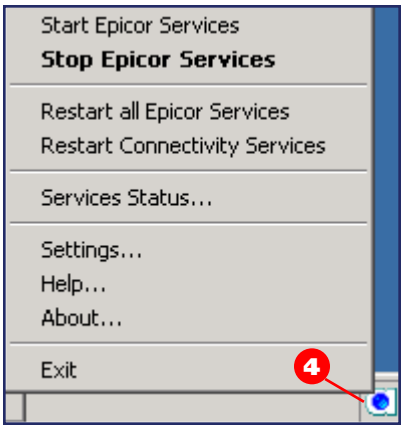
1. On the computer where Service Connect is installed, open Windows Explorer and navigate to the folder where **Epicor Service Connect** is installed.
2. Browse to the **ScaServiceManager** folder.

   The default path is: C:\Program Files\Epicor Service Connect\Tools \ScaServiceManager.



3. Double-click **ScaServiceManager.exe**.

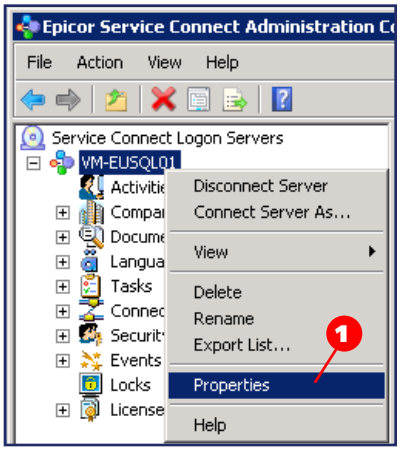
4. The **Service Manager** icon displays in the System Tray at the end of the Windows Taskbar. To access the menu, right-click the icon and select from the options.



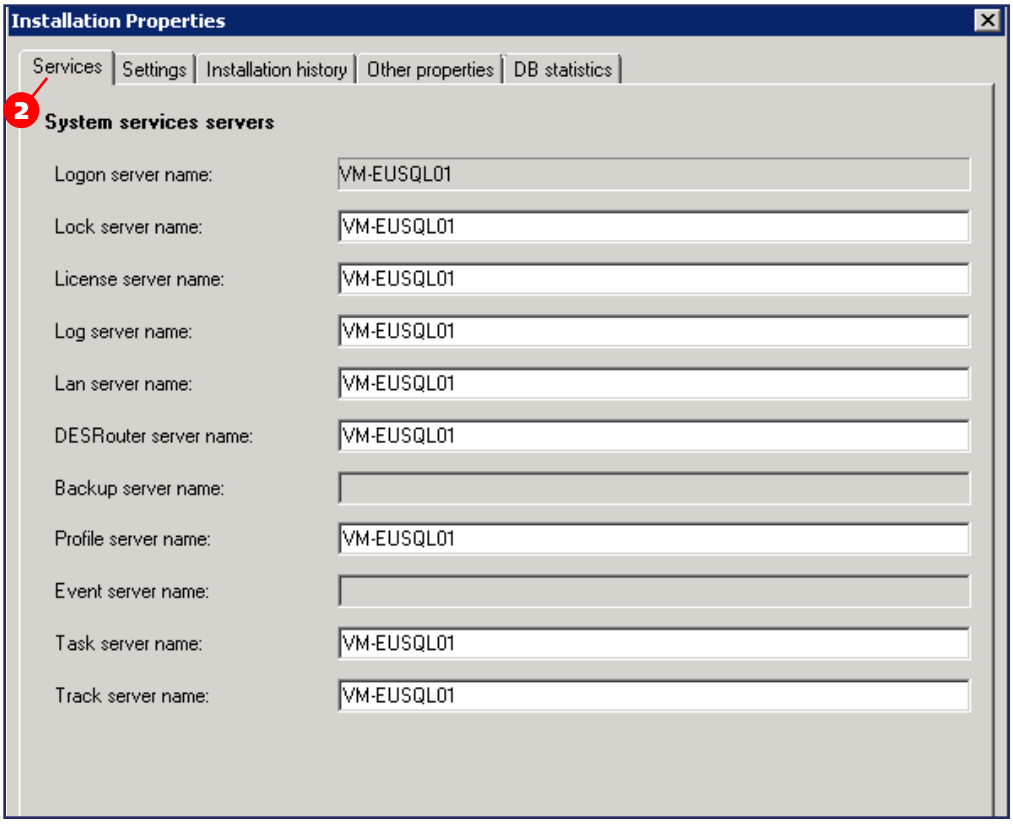
You can check statuses, Stop Epicor Services, Start Epicor Services, or Restart the Service Connect Windows services. You can also select the Settings menu option to set options for the Service Manager utility.

View Installation Properties

1. To display the installation properties, in the **Tree View**, right-click the main node (server name), and select **Properties**.



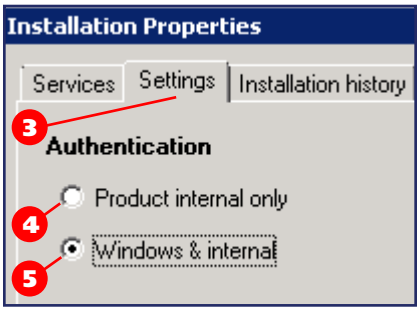
2. The **Installation Properties** window displays. Use the **Services** tab to set system services servers to different server names.



3. Use the second tab, **Settings**, to select the **Authentication** type.

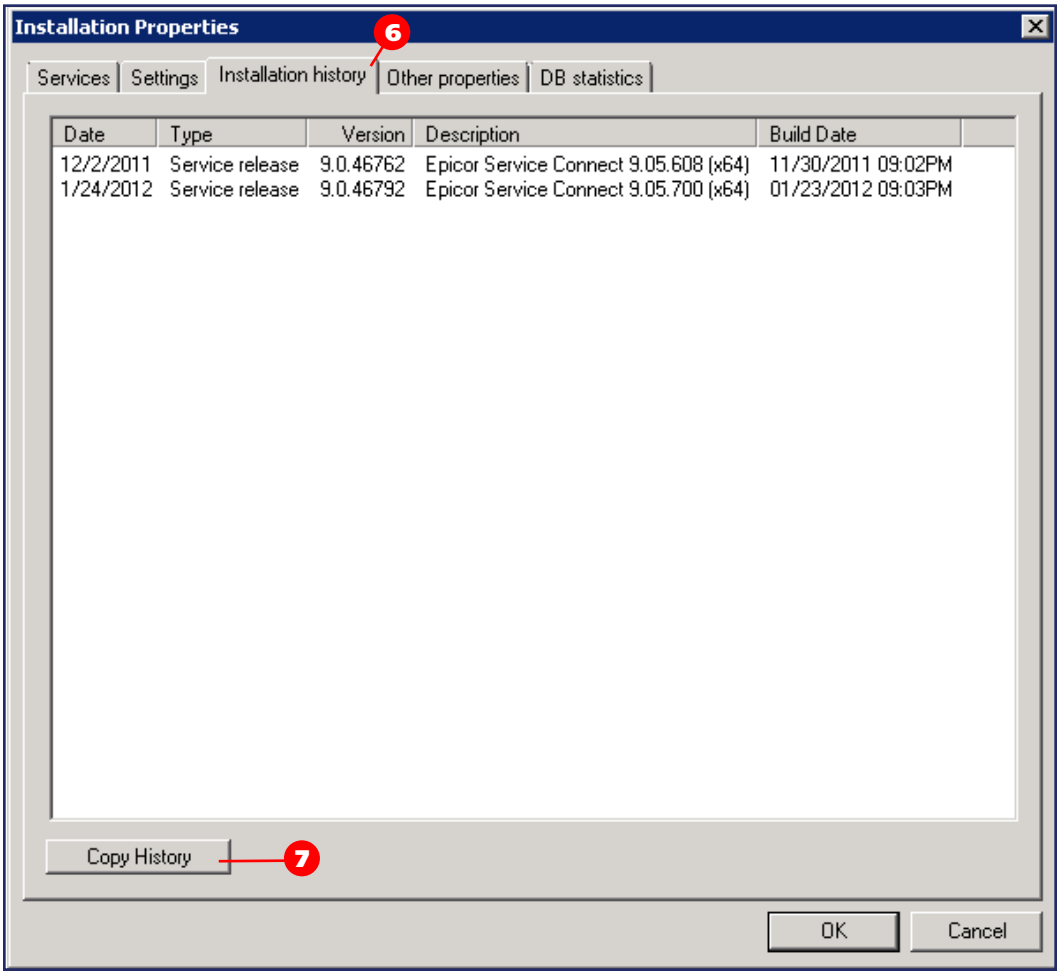
4. If you want users to enter their credentials to log in to ESC Administration Console and Workflow Designer, select the **Product internal only**.

5. If you select the **Windows & internal** check box, you can select the Import Windows user option to bypass entering credentials, and, instead, to automatically verify users' Windows account when they start the ESC Administration Console or Workflow Designer.

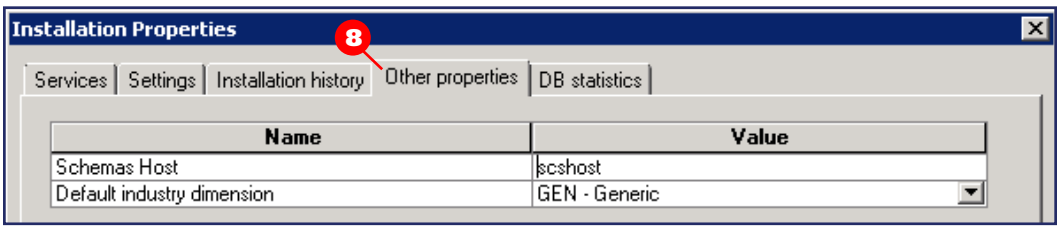


6. Use the **Installation history** tab to track progress of your Service Connect installations.

7. To copy the grid content to the clipboard, click the Copy History button.



8. Use the **Other Properties** tab to view and adjust the Service Connect installation properties. Available options:

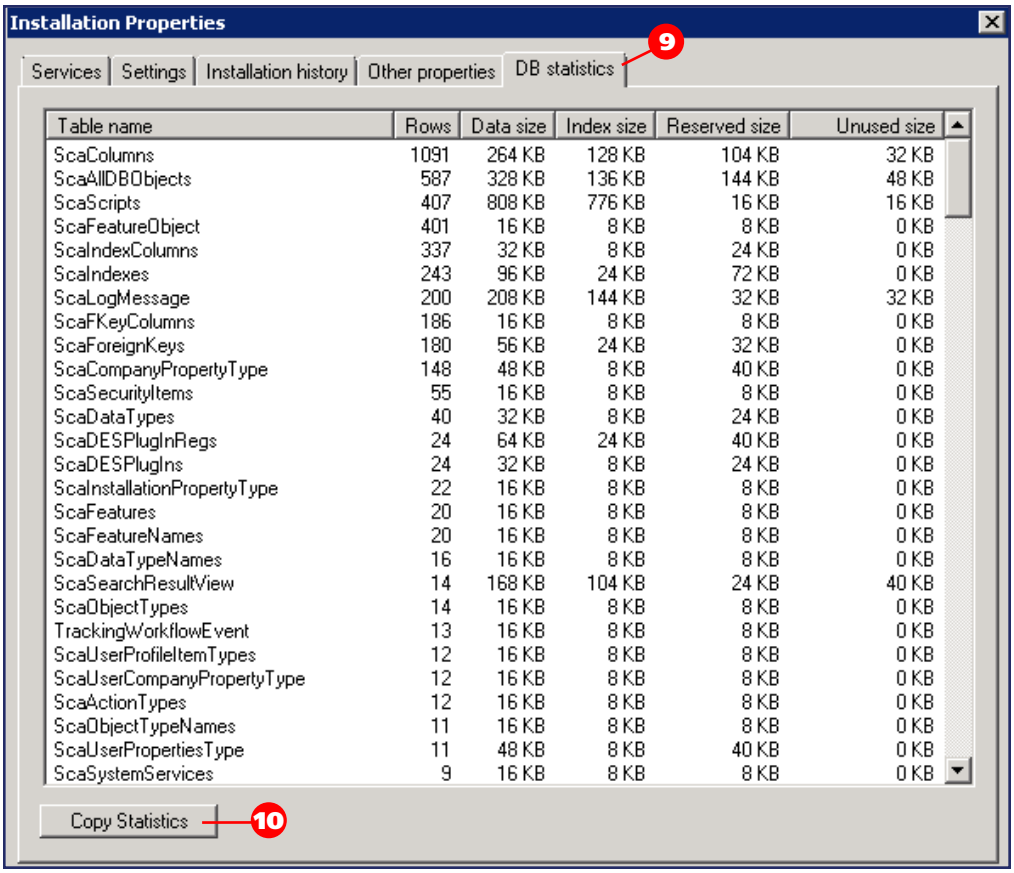


- **Schemas host** - The location from which schemes are loaded if scshost is not listed in the hosts file in the \\WINDOWS\system32\drivers\etc folder.

- **Default industry dimension** - Default industry dimension for the installation. Different industry dimensions provide industry specific terms for menu names, forms and dialog boxes.

9. Use the **DB Statistics** tab to view Service Connect data base statistics and analyze space allocation for all tables (Document Tracking, Task, Event) in ESC DB.

10. To copy the grid content to the clipboard, click the **Copy Statistics** button.



Use Event Logs

In Epicor Service Connect Administration Console, under the Events node, you can view a Service Connect events log.

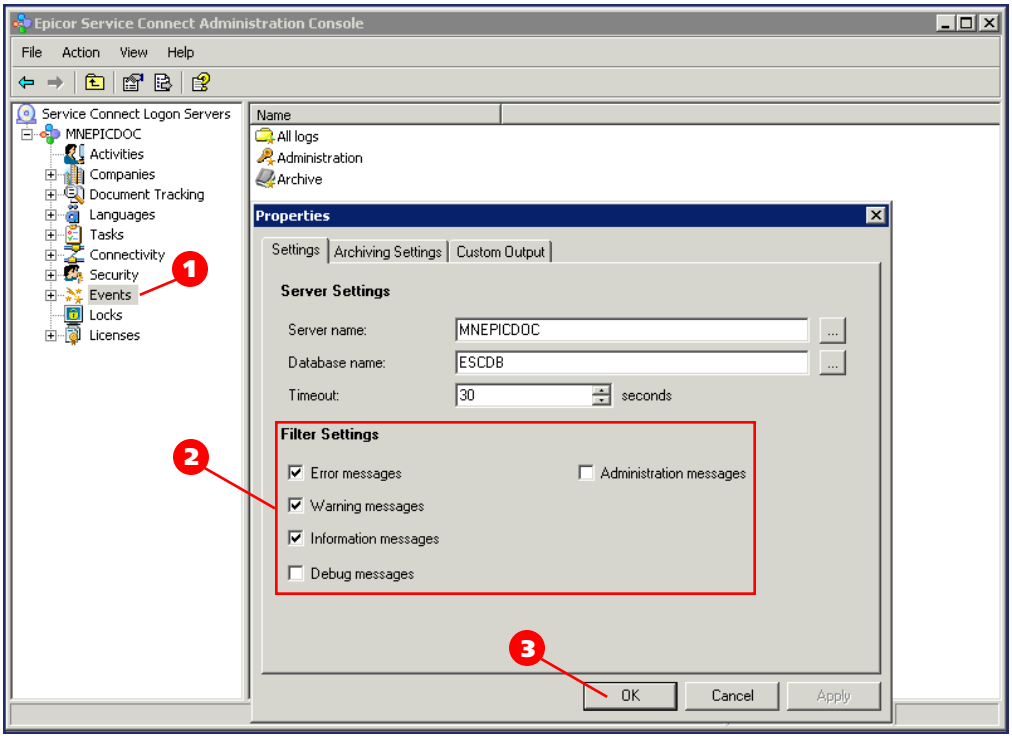
To define which events are logged:

1. In the **Tree View**, right-click the **Events** and select **Properties**.

2. The **Properties** window displays. In the **Filter Settings** section, select which type of messages to include in the log.

   Selecting **Debug messages** can be helpful with workflow development.

3. Click **OK**.

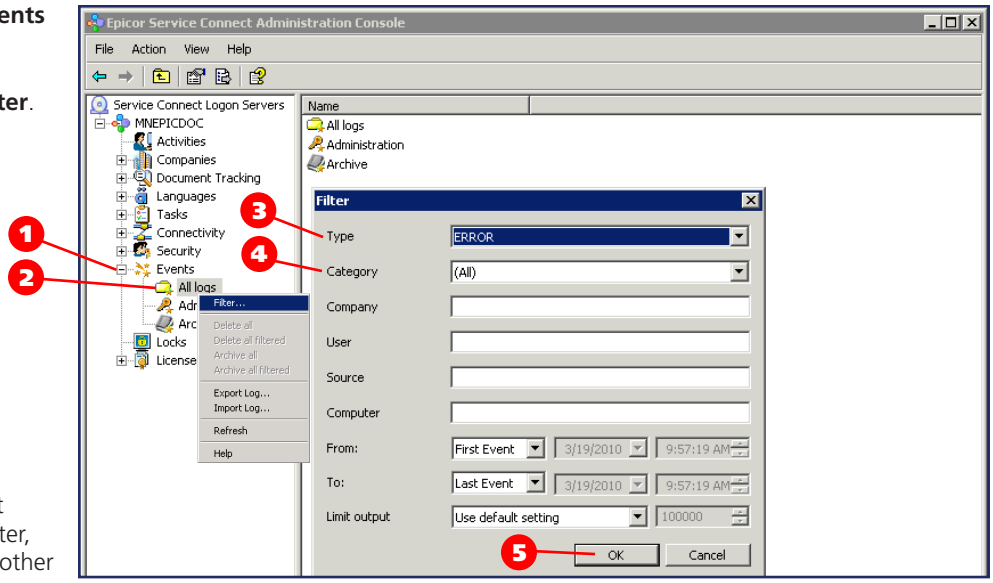


Filter Log Entries

To establish filter parameters:

1. In the **Tree View**, expand the **Events** node.
2. Right-click **All logs** and select **Filter**.
3. The **Filter** window displays. Select a **Type**.

   For example, select ERROR to see only the Error entries.
4. Optionally, select a **Category**.
5. Click **OK**.



The log displays the entries that fit the selected criteria. Filter settings remain between logons to the Service Connect Administration Console. To reset the filter, change the filter Type back to All or another setting.

Connectivity Administration

The Connectivity node in the Epicor Service Connect Administration Console holds site-specific and user-generated information for a particular Service Connect installation. You can back up and restore this information. The restore is flexible, so you can restore and reconfigure specific information. The settings in the Connectivity node are described in Chapter 3: Connectivity Components.

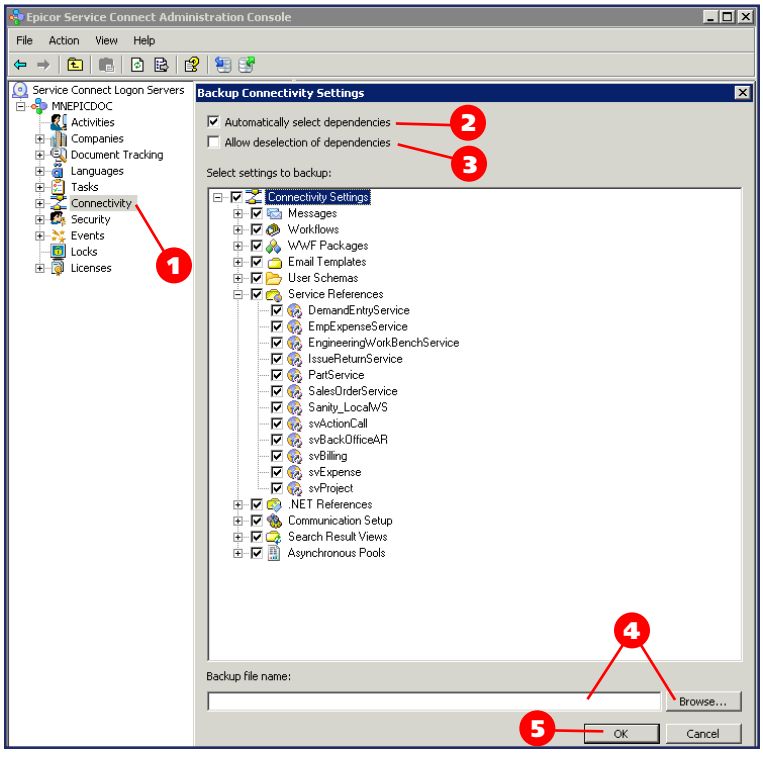
Create a Backup

To backup connectivity settings:

1. In the **Tree View**, right-click the **Connectivity** node and select **Backup**.
2. The **Backup Connectivity Settings** window displays. Leave the **Automatically select dependencies** check box selected.

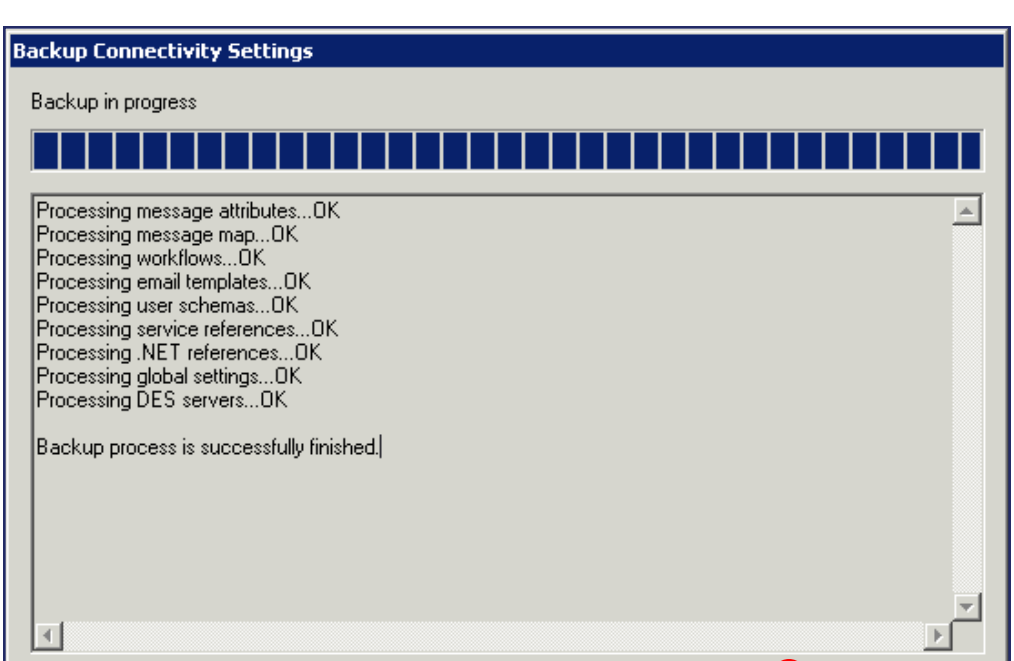
   This setting enforces object selection depending on other objects. For example, a workflow that uses a service reference cannot be omitted if the associated reference is selected.
3. Select **Allow deselection of dependencies** if you want the ability to select or clear any item regardless of its dependencies.

   If you select both options, when you select an object with a dependency, the dependency will also be selected until you manually clear it.
4. Click **Browse** to navigate to a folder and enter a filename for the backup file.
5. Click **OK**.



6. The **Backup Connectivity Settings** window displays. Once the backup process is complete, click **OK**.

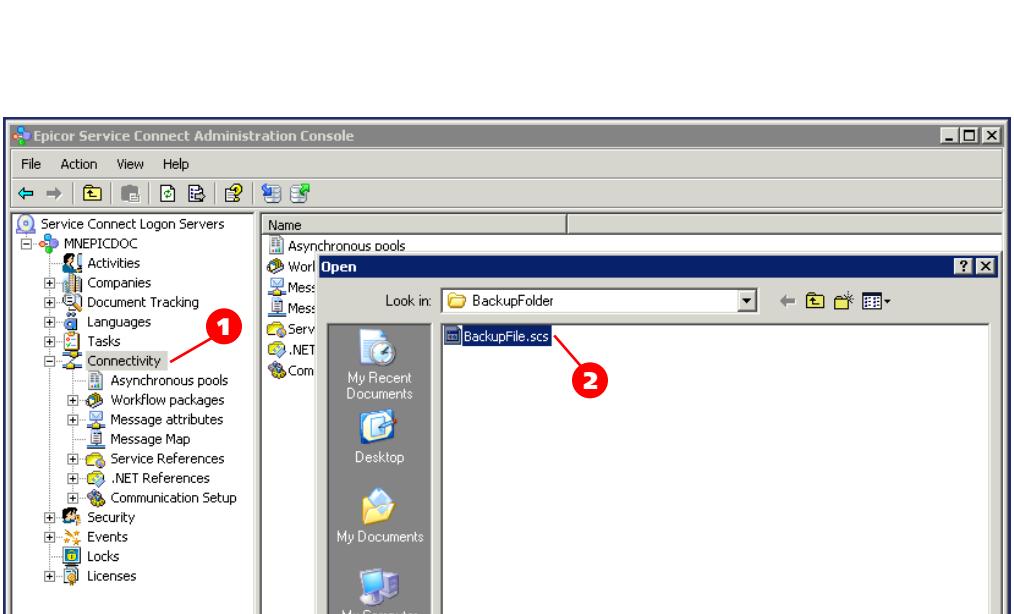
The backup file is saved. Include this backup file in your regular system backup routines.



Restore from a Backup

To restore connectivity settings:

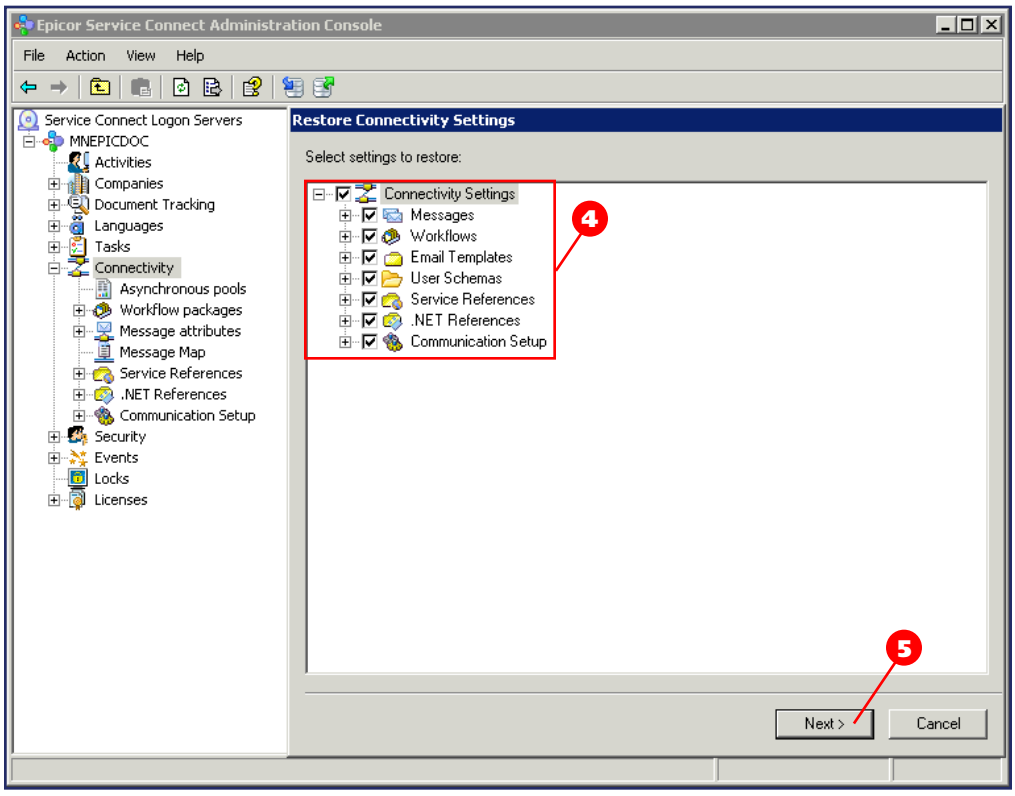
1. In the **Tree View**, right-click the **Connectivity** node and select **Restore**.
2. Browse to select your backup file.
3. Click **Open**.



4. The connectivity settings that were backed up display in the **Restore Connectivity Settings** window. Clear the check box next to the items you do not want to restore.

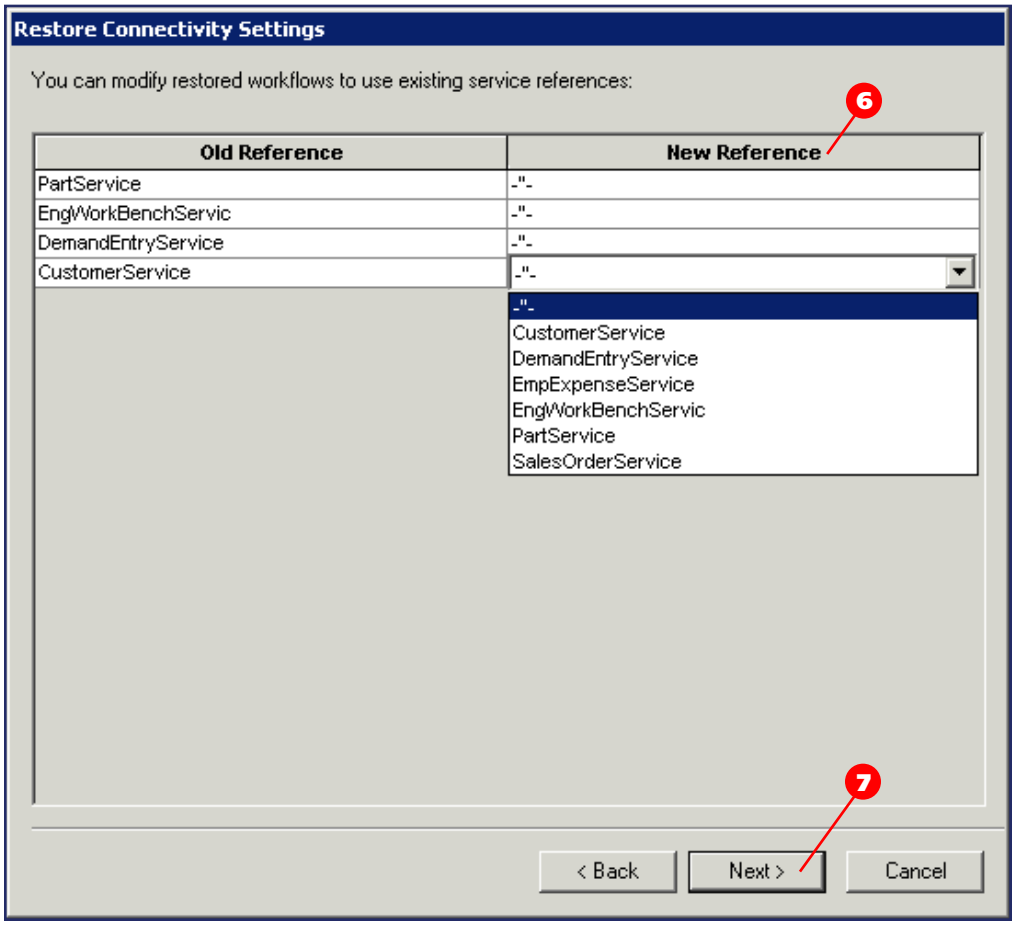
5. Click **Next**.

   If you select to restore Workflows, Service References, Communication Setup, and .NET References, you must perform additional steps.



6. If you restore a workflow that uses web-methods or .NET References, you can adjust these registered references in the **New Reference** column. Select the service references to use instead of the imported one.

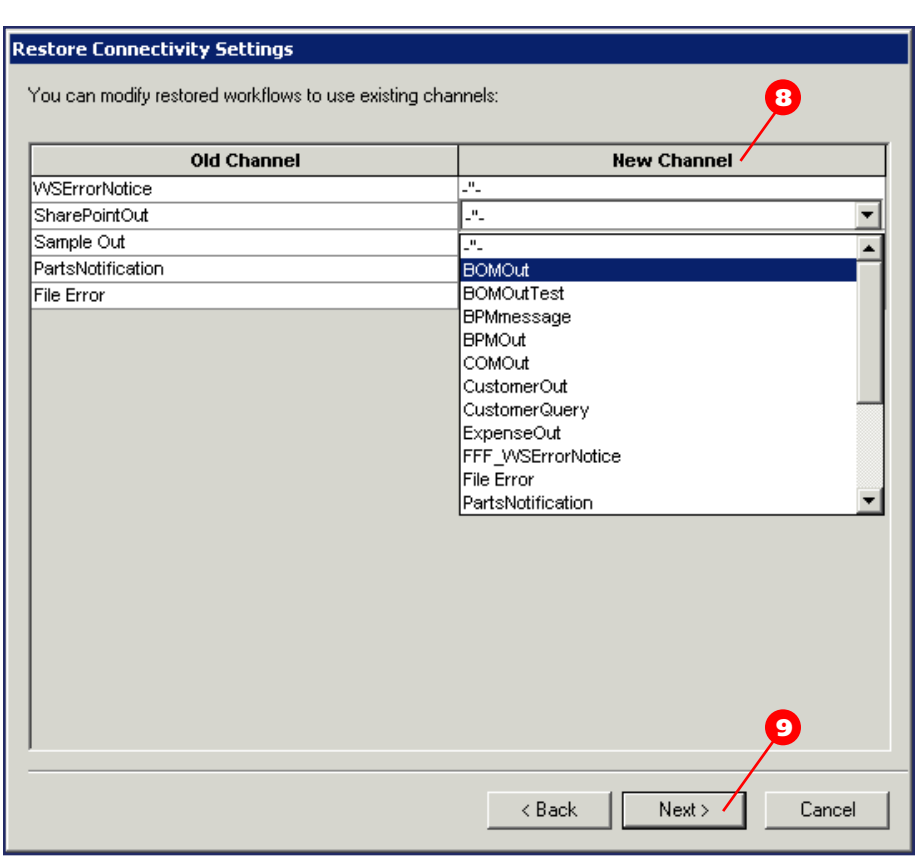
7. Click **Next**.



8. To adjust the communication channels that you restore, in the **New Channel** column, select the channel to use instead of the imported one.

9. Click **Next**.

   You may need to modify the web references being restored, that is, change the web-service URLs and Windows credentials to gain access to the specified server. Also, some changes in web service handlers may be required.



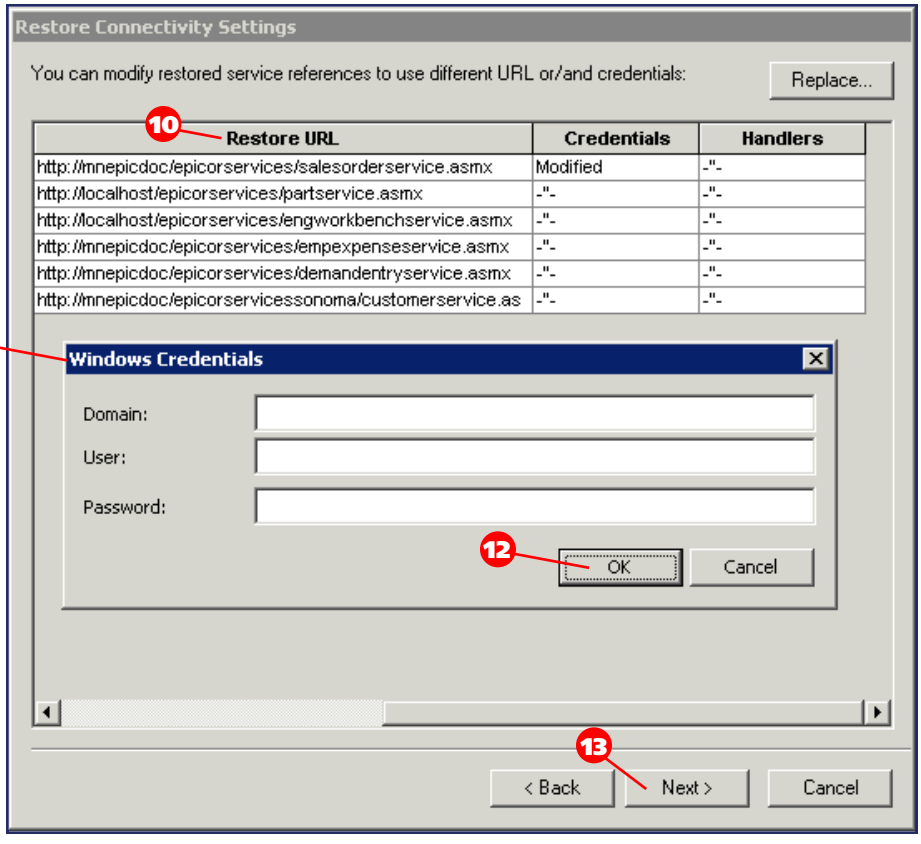
10. In the **Restore URL** column, enter the adjusted URL for the web reference.

    The restored URL for web references is checked for validity. An empty URL or a URL with incorrect syntax is not accepted.

11. If you need, in the **Credentials** column, select **Modified** to modify the **Windows Credentials** settings.

12. The **Windows Credentials** window displays. Click **OK**.

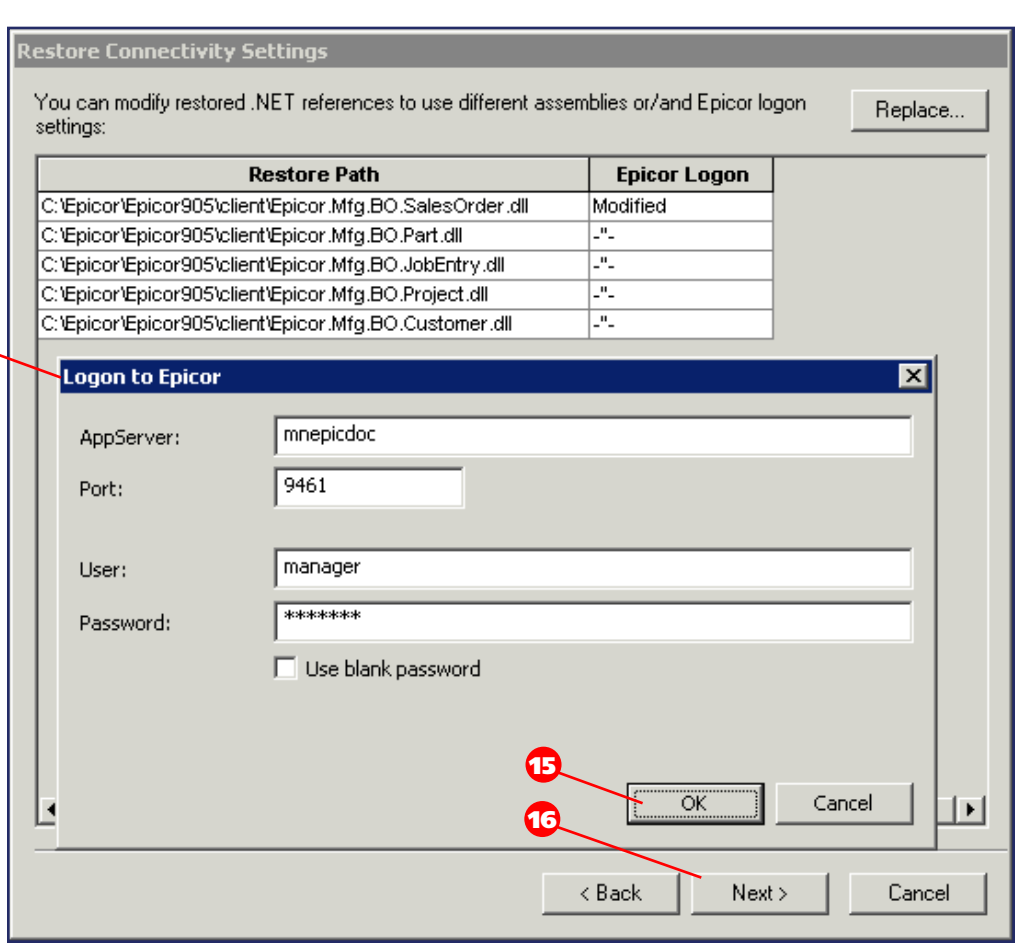
13. Click **Next**.



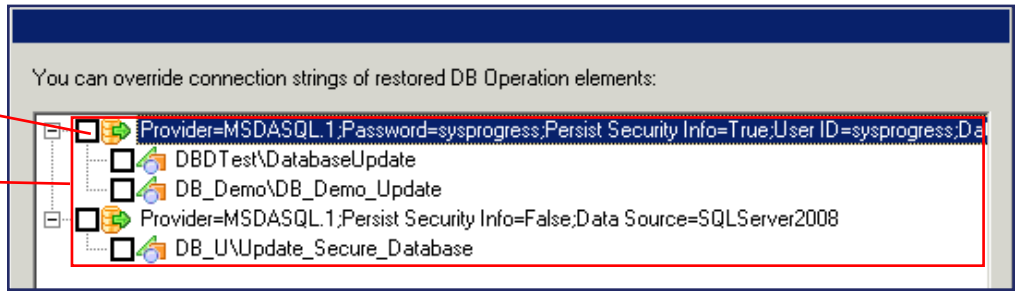
14. You may need to modify the .NET References being restored, that is, modify the **Restore Paths** to assemblies. If you restore Epicor Assembly .NET Reference, you may need to modify Epicor configuration settings . To do so, select **Modified** in the **Epicor Logon** column and in the **Logon to Epicor** window, enter the new information to access the application server.

15. Click **OK**.

16. Click **Next**.



17. View the connection strings for DBOperation workflow elements. Connection strings display as a tree. Multiple root nodes are connection strings, and their children are workflows that use them. The workflows that use the same connection string are grouped under one root node.



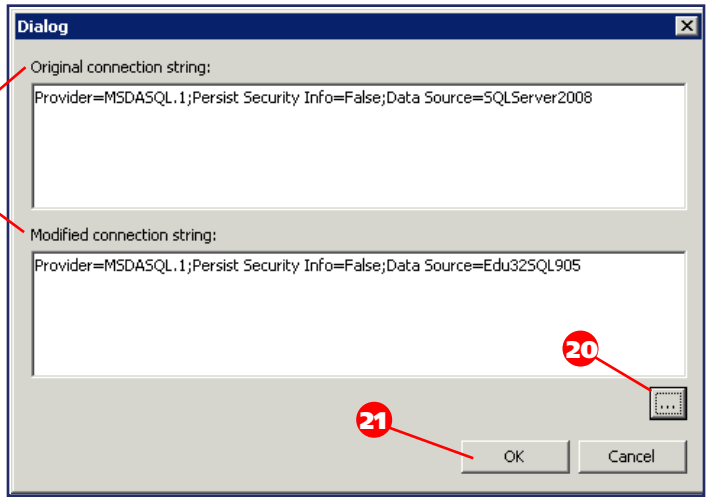
18. Select the connection string or strings to edit. To edit a connection string for all the workflows that use it, select the root node in the tree. All the subordinate nodes are selected automatically.

If you want to edit a connection string only within some of the workflows that use it, select the leaf nodes corresponding to particular workflows.

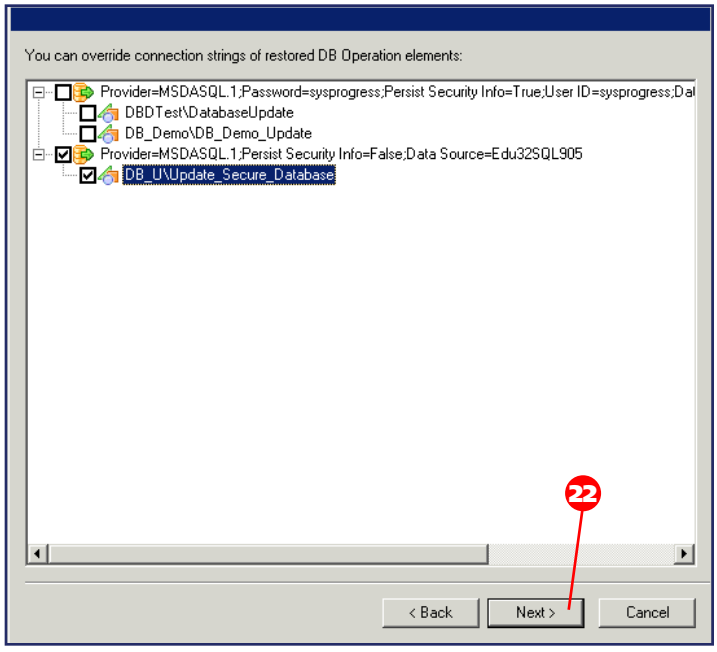
19. The dialog with the **Original connection string** and **Modified connection string** fields displays.

20. To Adjust the connection string in the **Modified connection string** field, click the **Ellipse** button to display the **Data Link Properties** window and make the changes.

21. Click **OK**.



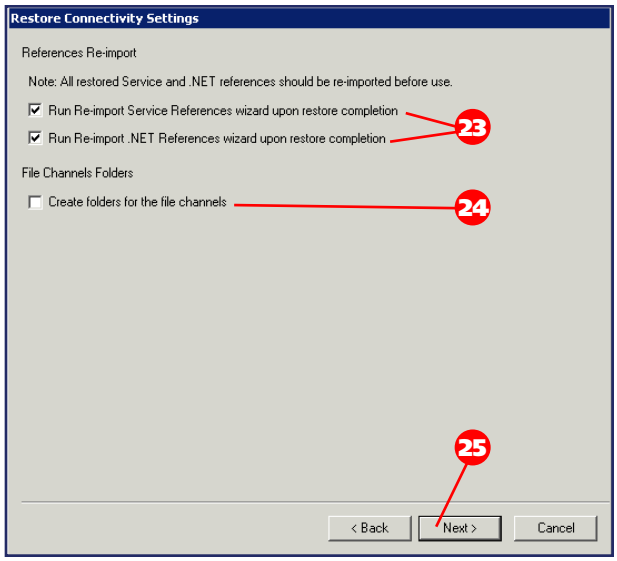
22. Click **Next**.



23. The **Restore Connectivity Settings** window displays. To automatically re-import service references and .NET References, select the **Run Re-import Service References wizard upon restore completion** and **Run Re-import .NET References wizard upon restore completion** check boxes.

24. If you selected to restore Input or Output channels, select the **Create folders for the file channels** check box to restore channel folders during the process.

25. Click **Next**.

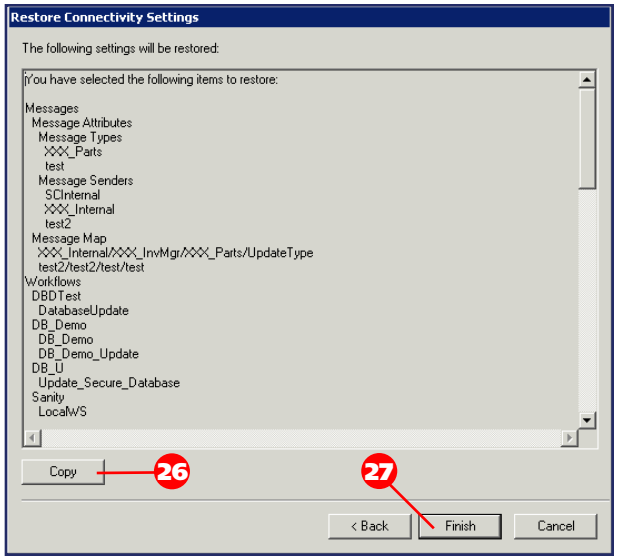


26. If you want to copy the list of all restored items to the clipboard, click the **Copy** button.

27. Click **Finish** to complete the wizard.

At the end of the restore process, the results display.

> Apart from the backup and restore process, you can also re-import selected web service and .NET References manually. Refer to Chapter 3: Connectivity Components for information on the re-import process.



Summary

This chapter covered basic administration tasks performed on a Service Connect installation. The next chapter describes the components involved in creating Service Connect solutions.

Chapter 3
Connectivity Components

This chapter describes the parts of the Epicor Service Connect Administration Console that receive documents into a workflow, publish documents from a workflow, route incoming documents, and manage external services used during workflow processing. This chapter also explains the browser-based Task Monitor and the Administration Console's document tracking feature, which can be used to trace workflow execution.

Connectivity Components

The components under the Connectivity node in the ESC Administration Console configure, route, and process messages. Refer to Chapter 2: Epicor Service Connect Administration for information on how to back up and restore all items under the Connectivity node.

Workflows

Workflows are complex units that organize a sequence of automatic activities or user-performed tasks to accomplish on a Service Connect internal message. Workflows are the foundation of Service Connect. Most of the connectivity components exist to route messages to and from a workflow. To learn how workflows interact and how to develop workflows refer to Chapter 4: Workflow Designer and to the product-specific chapters available on EPICweb. This section describes the connectivity components that route messages to workflows and interact with workflows.

Message Attributes and Message Maps

Documents are submitted to Service Connect in three ways:

- Through an input channel
- Directly from an Epicor application
- Through a workflow exposed as a Web Service

Message attributes are used to help route documents submitted to an input channel. A document that drops into an input channel has one of two formats:

- Service Connect internal XML message format, also called internal envelope.
- Format other than internal envelope. These messages should first be converted to the format SC understands.

Review Chapter 1: Epicor Service Connect Overview for more information on the internal message format. If documents are already in the internal message format, then you do not need to define message attributes in the ESC Administration Console because the sending application already added the attributes to the message. In a typical scenario, incoming documents are not formatted for Service Connect and require the addition of message attributes.

After you define message attributes, you use them to configure an input channel. A document sent to this input channel is stamped with those attributes as the document is converted to the internal format. The documents sent to the input channels configured with message attributes are stamped with those attributes as the documents are converted to the internal format.

Finally, message attributes are used to create message maps, which contain specific combinations of attributes, and route documents to workflows according to the attributes stamped on the document.

When you design a Service Connect solution that relies on converting documents to the Service Connect internal message format through input channels, you must plan the message flow. To do it, use two message attributes:

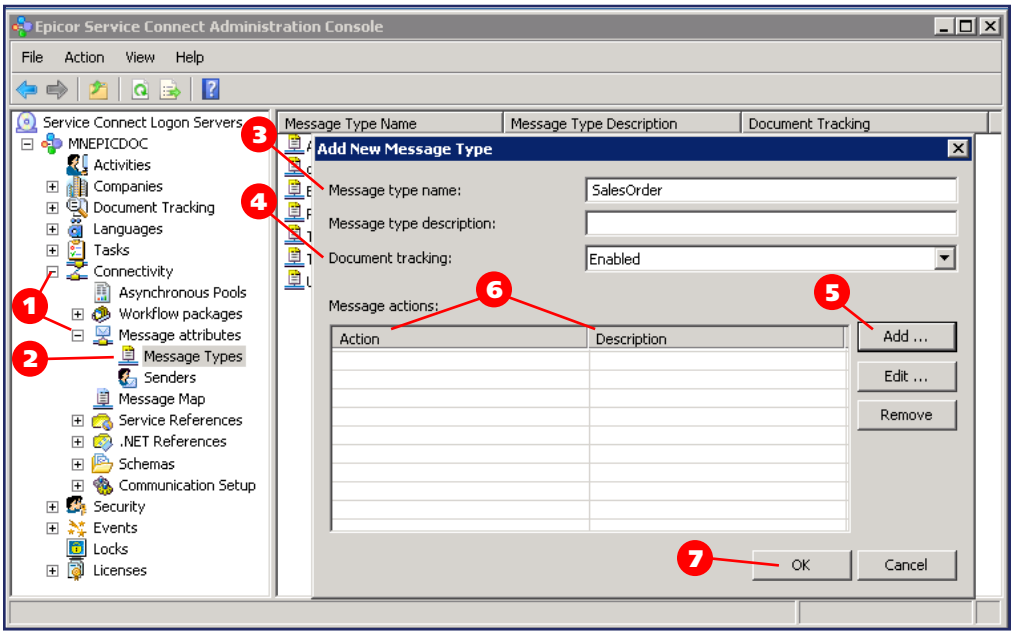
- Message types
- Senders

Use message types to classify the documents you plan to process in Service Connect. For example, if you plan to send documents that represent orders to Service Connect, you might create a message type called Order. Within a message type, you can have sub-classifications that define message actions, such as enter order, update order, and delete order.

Use senders to define the origin of documents sent to Service Connect. For example, you can set up an FTP input channel to receive shipment information from a supplier. In this case, the Sender field would likely contain the supplier name. The Sender can also be an application name like Epicor 9, a place name like Denver, or a computer name. Within a Sender definition, you can also define subnames to more narrowly define the document origin.

Add a Message Type

To add a new message type:

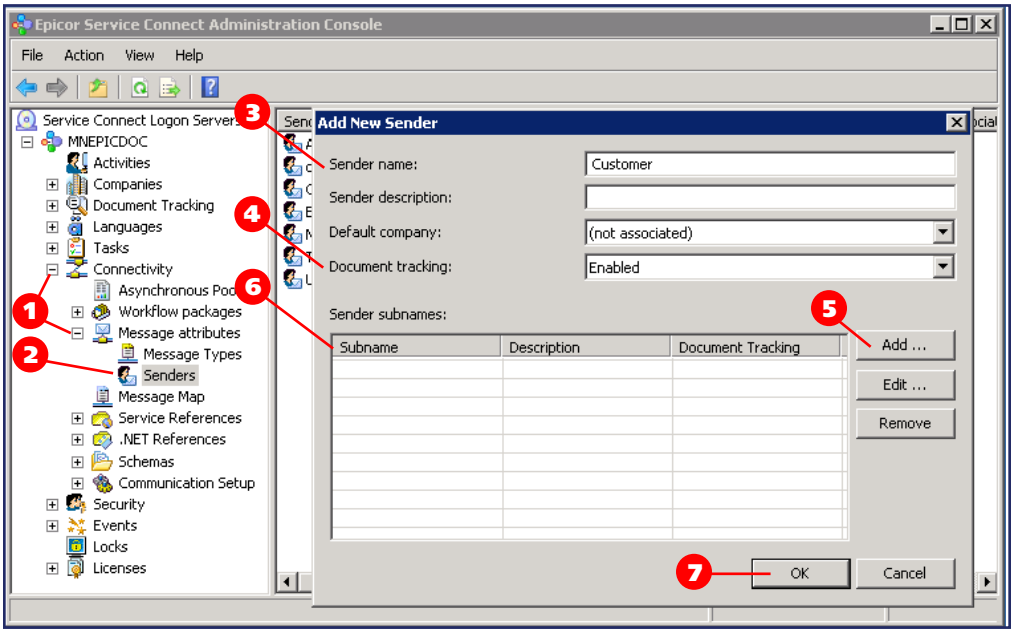


1. In the **Tree View**, expand the **Connectivity > Message** attributes node.
2. Right-click **Message Types** and select **Add new Message Type**.
3. The **Add New Message Type** window displays. Enter a **Message type name** that describes a document such as an order or parts list. In this example, enter Sales Order.
4. Leave **Document tracking** set to **Enabled**.
5. Click **Add** to add a message action.
6. The **Message actions** window displays. Enter an **Action** name and a **Description**.

   In general, actions are verbs that describe typical database actions, such as create, retrieve, update, or delete.
7. Click **OK** until you exit all dialog boxes.

Add a Sender

To add a new sender:



1. In the **Tree View**, expand the **Connectivity > Message** attributes node.
2. Right-click **Senders** and select **Add New Sender**.
3. Enter a **Sender name** that describes the origin of a document. For example, if the sender is the Clientele application, you might use Clientele as the sender name. In this example, enter Customer.
4. Leave **Document tracking** set to **Enabled**.
5. Click **Add** to add a sender subname.

6. The **Sender subnames** grid displays. Enter a **Subname** that represents a smaller category or use a catch all name like All.

7. Click **OK** until you exit all dialog boxes.

Add a Message Map

Message maps are used to automatically determine where to route a document once the document is received in an input channel. The route is determined by matching the message attributes stamped on the document to the message map with the same combination of message attributes and destination. In a message map, you set the Sender Name (and optional Subnames), Message Type, Message Action, and Request ID. The Request ID is the destination where the document is sent. Request IDs are usually workflows but can also be channels, or web methods.

If you choose not to call a workflow directly from an Epicor application or not to call a workflow exposed as a web service, you will need to set up message maps to route incoming documents.

Adding a message map is typically one of the last components configured for a Service Connect solution because most of the other components in the solution must be configured first.

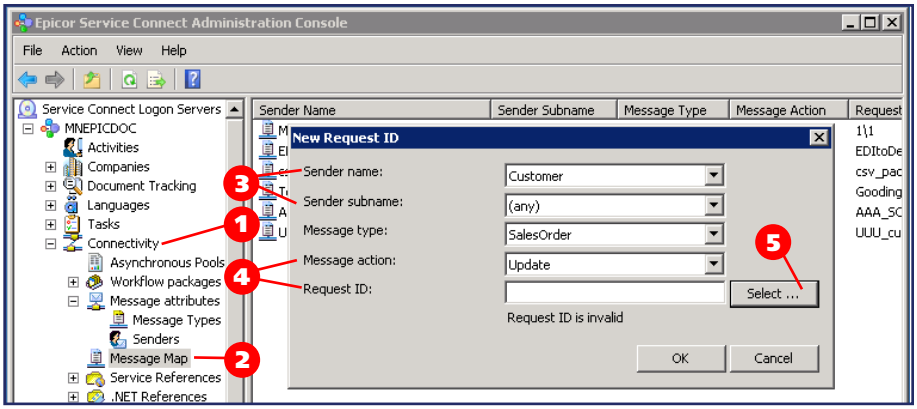
To add a Request ID to a message map:

1. In the **Tree View**, expand the **Connectivity > Message** attributes node.

2. Right-click **Message Map** and select **Add new Request**.

   The map selections define a combination of attributes that helps route an incoming document to the Request ID in the last field.

3. In the **New Request ID** window, select the appropriate **Sender name** and **Sender subname** for the intended Request ID. In this example, select Customer.



4. Select the appropriate **Message type** and **Message action** for the intended Request ID. In this example, select SalesOrder and Update.

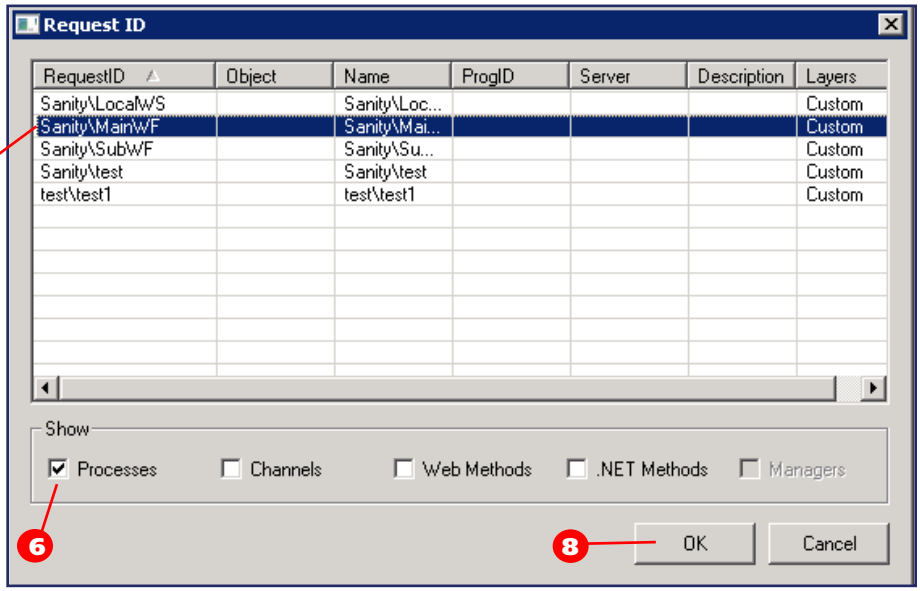
5. Click the **Select** button to find and select the workflow that will process the incoming document.

6. The **Request ID** window displays. At the bottom of the window, leave the **Processes** check box selected and clear all other check boxes. Now the window lists only workflows also referred to as processes.

   The RequestID column lists workflows in the <workflow package>/<work flow name> format. Workflow packages are equivalent to physical folders. Use them to store your workflows logically grouped.

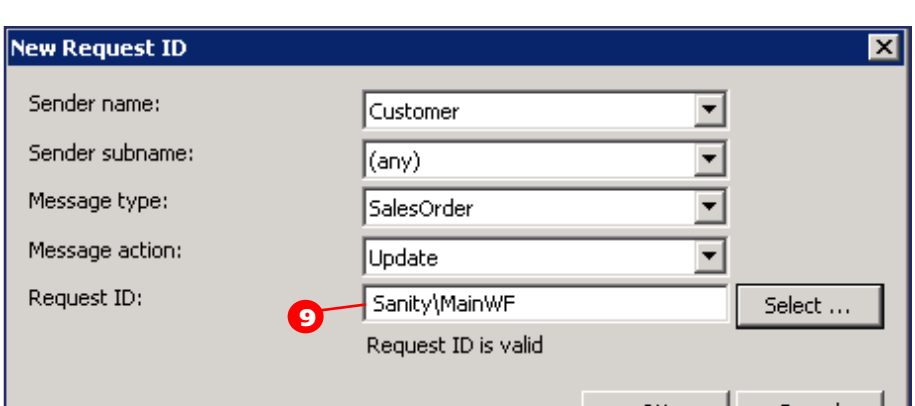
7. Select the workflow that should process the incoming document.

8. Click **OK**.



9. Notice the **Request ID** field now dis plays the workflow package and workflow name.

10. Click **OK**.



Service References

Service references are configured links to the published application services, such as web services or Windows Communication Foundation® (WCF) services, which are outside of Service Connect. After you add service references in the ESC Administration Console you can pass and receive datasets to and from the web methods from inside workflows. A wizard helps you import service references, configure security, and handle other miscellaneous details regarding the exchange of datasets. The wizard has some pre-configurations available for the Epicor applications' web services. For any service, you have the option to import all or just a subset of its web methods. Review Chapter 4: Workflow Designer and the product-specific chapters for more details and examples on how to use service references in a workflow.

Add a Service Reference

To add a service reference:

1. In the **Tree View**, expand the **Connectivity > Message** attributes node.

2. Right-click **Service References** and select **Add Service Reference**.

3. The **Add Service Reference** window displays. Click **Next**.

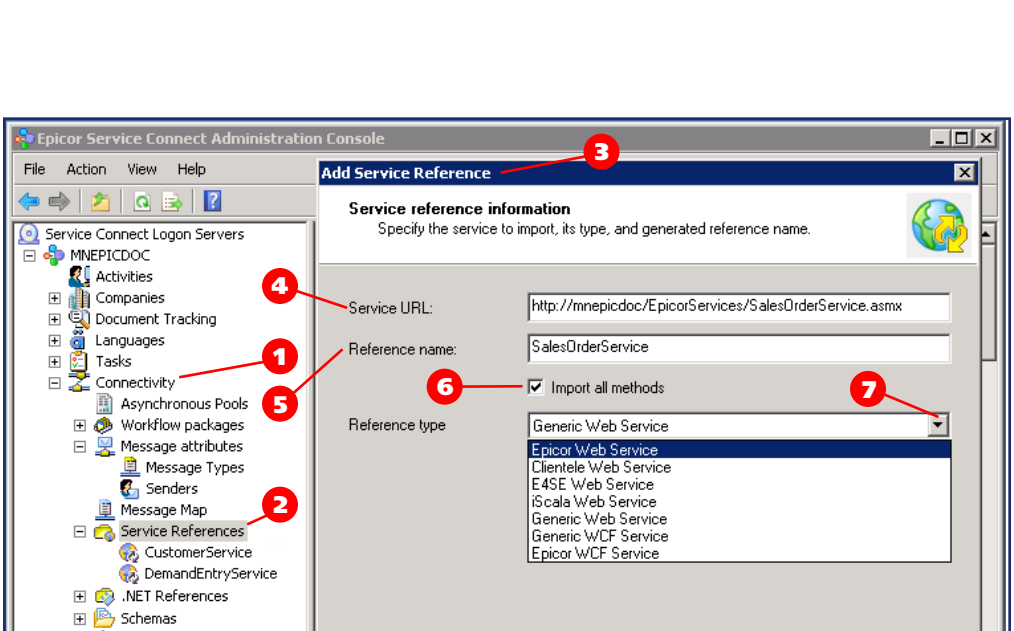
4. Enter the **Service URL**.

5. Enter a **Reference Name**.

   The name will be used when referencing the service. Epicor recommends you use the service name.

6. To import all service methods, leave the **Import all methods** check box selected.

   If you do not select Import all methods, you can add the methods one by one later and configure each individually.



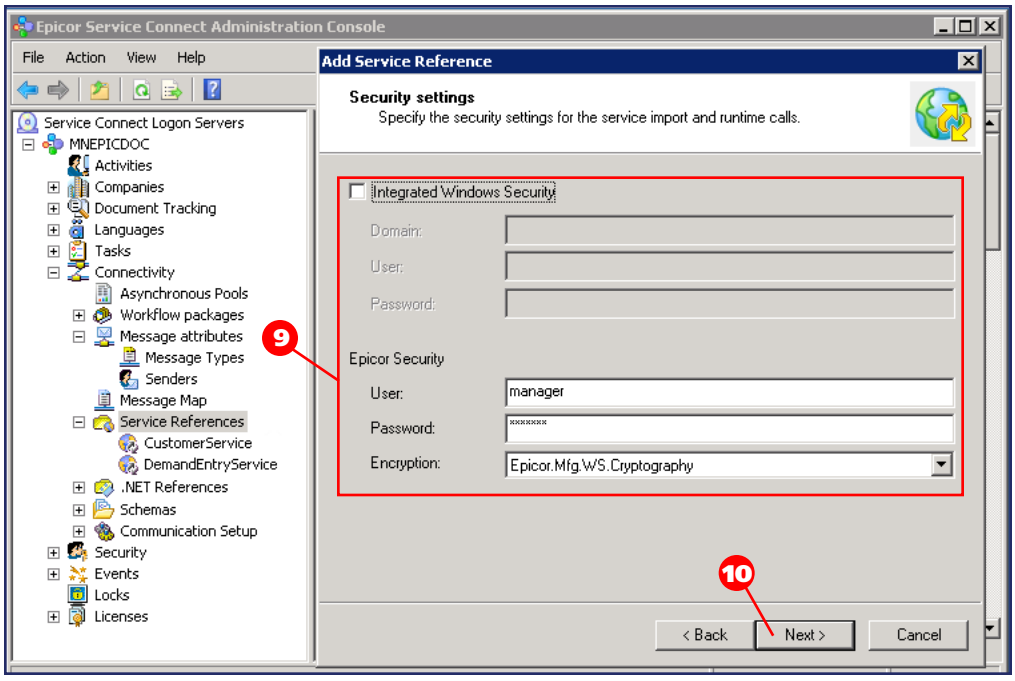
7. Click the **Reference Type** drop-down list to select the appropriate Epicor product. You can select **Generic** for any service.

8. Click **Next**.

9. Complete the **Security settings**.

   Depending on the Service Reference Type you selected, you may have different options on your Security settings window. Refer to the product-specific chapters for details on how to set up security for your product.

   If you selected Generic as the Service Reference Type, you have these security setting options:

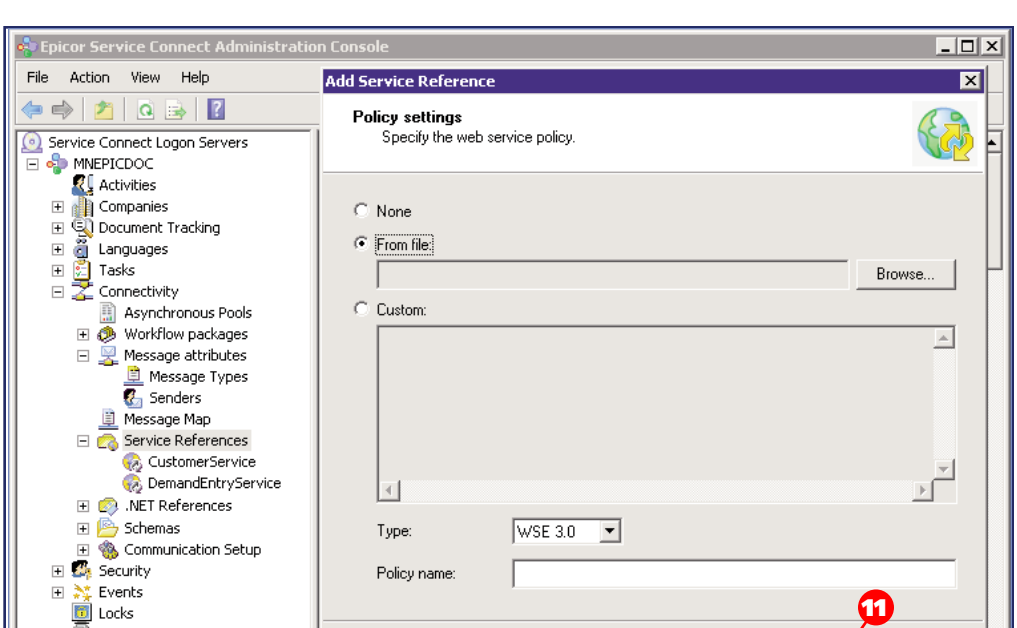


   - Select **Integrated Windows Security** and enter a **Domain**, **User**, and **Password** to supply the Windows credentials needed to access the service.
   - Select **Integrated Windows Security** and do not enter credentials. In this case, the Windows account used to run the ScaDESRouter service is used.
   - Do not select Integrated Windows Security. In this case, the Windows account used to run the ScaDESRouter service is used.
   - If you selected Generic Web Service type, and the service is configured to use a Web Services Enhancements (WSE) for Microsoft .NET policy file, click Next and on the Policy settings window, select a policy file to use.

10. Click **Next**.

For Epicor ERP and Vantage, the Integrated Windows Security option is used only when you have set up Windows Authentication for the services. You can select the Integrated Windows Security check box and leave the Domain, User, and Password fields blank to grant access to the web services that use the same account that Service Connect uses to run ScaDESRouter. For more information on how to use Windows Authentication with the Epicor ERP or Vantage Web Services, review the Enabling Windows Authentication section in the Web Services Developer Guide. This document is located in the Web Services folder of your Epicor ERP or Vantage server.

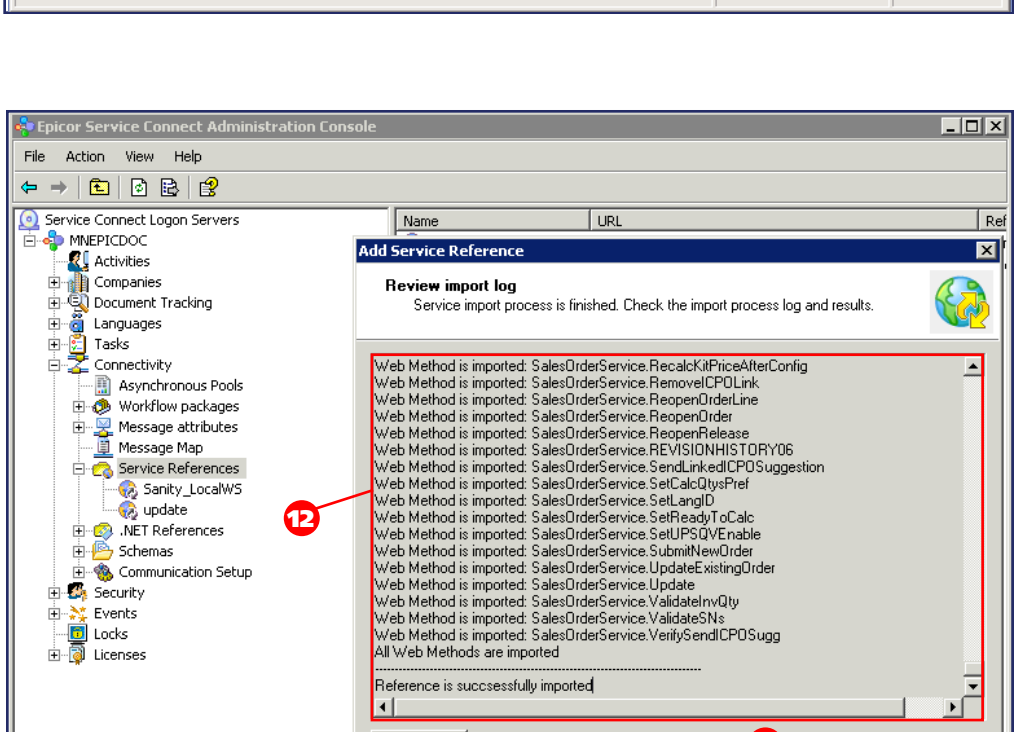
11. Review the Policy settings and click **Next**.



12. When the import is complete, a detailed log displays on the **Review import log**.

You can use the **Copy** button to copy and save a copy of the log.

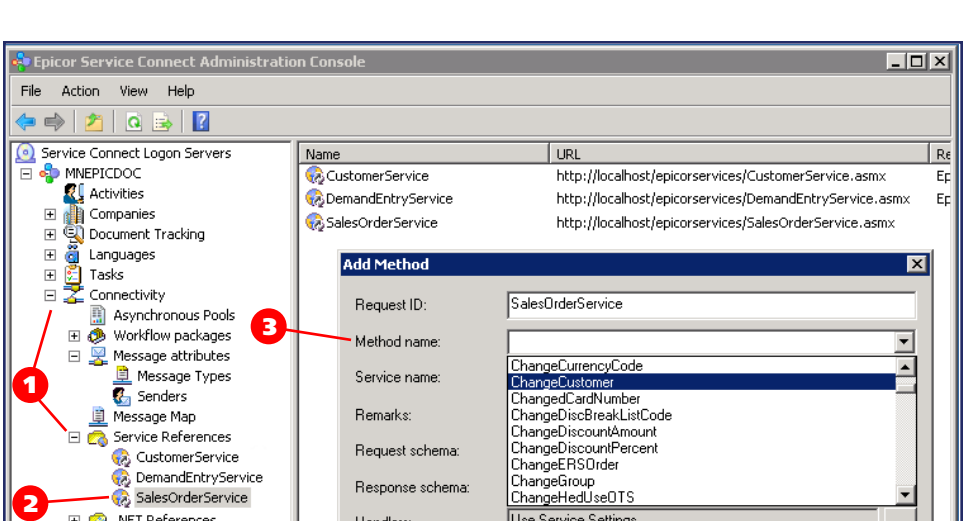
13. Review the import log and click **Finish**.



Add Service Methods

If you did not select Import all methods when you added a service reference, follow these instructions to add an individual service method:

1. In the **Tree View,** expand the **Connectivity > Service References** node.
2. Right-click the service reference to which you want to add a method and select **Add Method**. In this example, select SalesOrderService.
3. The **Add Method** window displays. Select a **Method Name**.
4. Click **OK**.

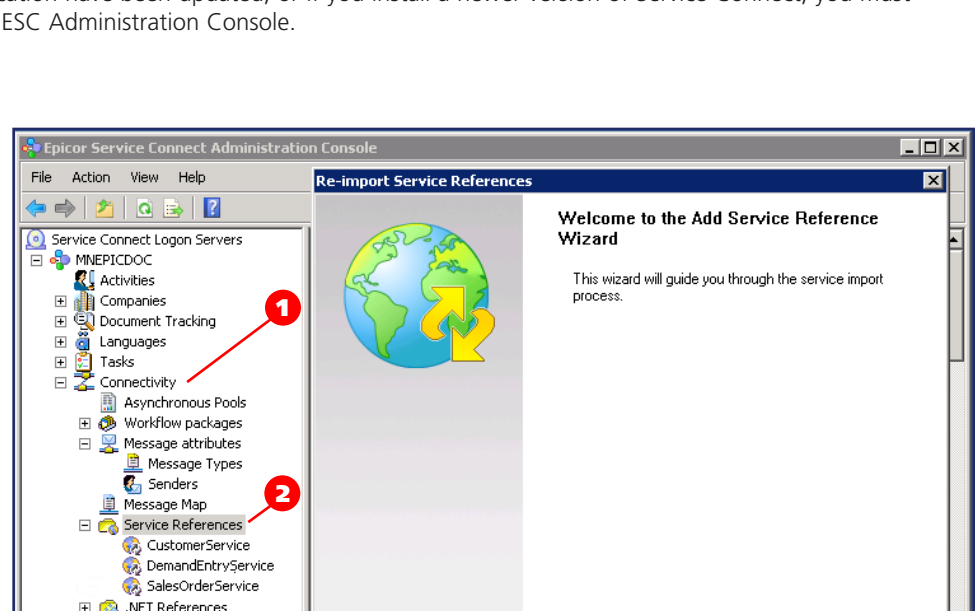


Re-Import Service References

If the Web Services for the outside application have been updated, or if you install a newer version of Service Connect, you must re-import the service references into the ESC Administration Console.

To re-import services:

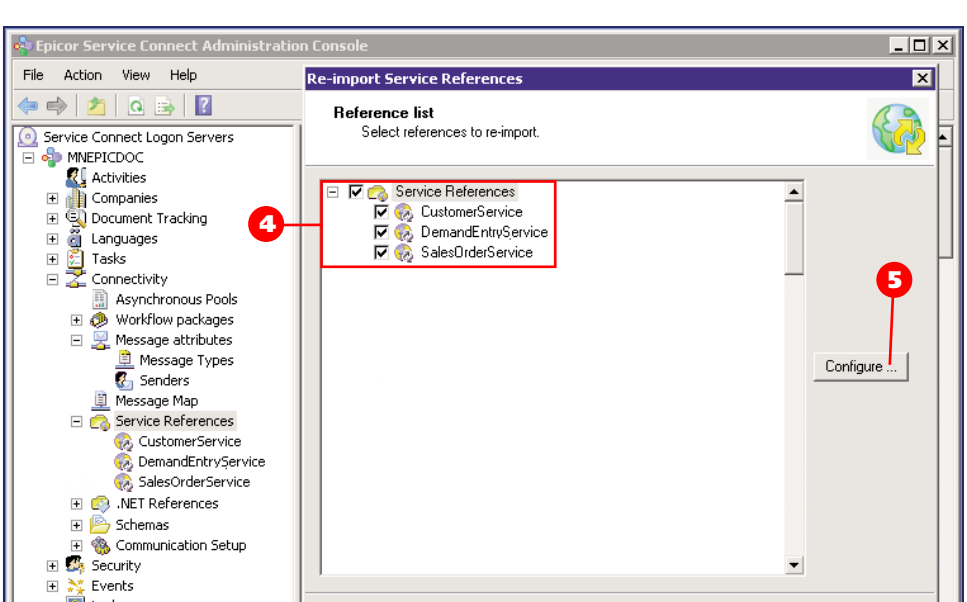
1. In the **Tree View**, expand the **Connectivity** node.
2. Right-click **Service References** and select **Re-import References**.
3. The **Re-import Service References** window displays. Click **Next**.



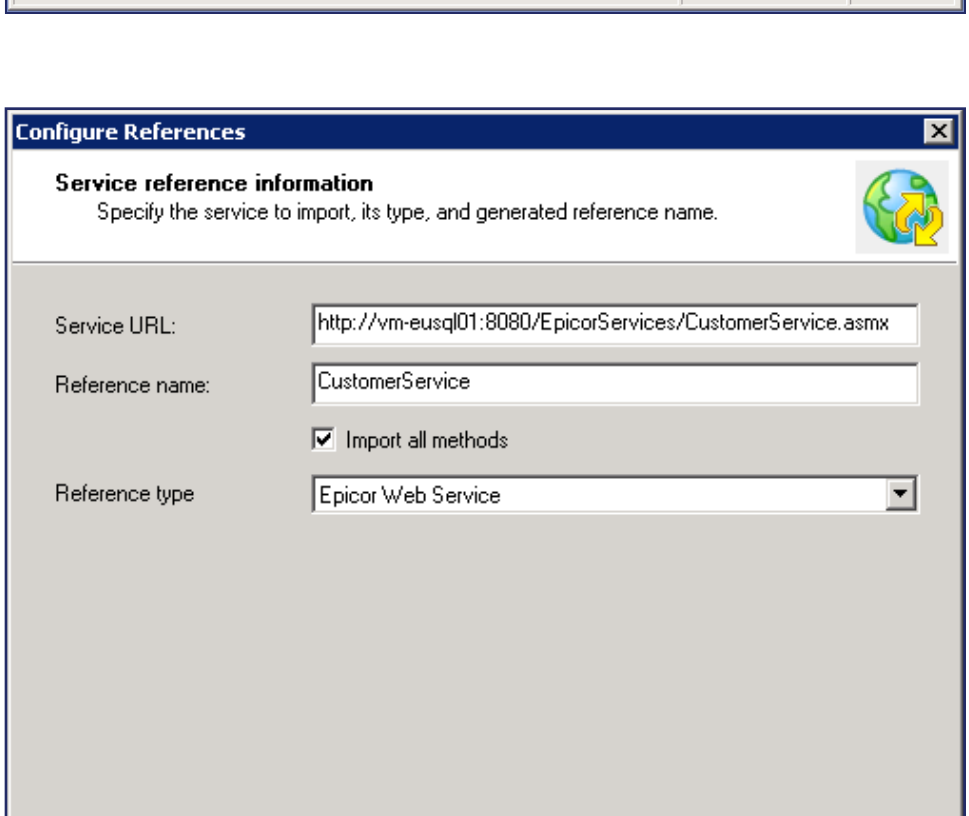
4. From the **Reference list**, select the references to re-import.

   All references are selected by default.

5. During service reference re-import process, Service Connect automatically uses the settings originally used to import the service. If you need to adjust the re-import settings, click **Configure** and change the settings.

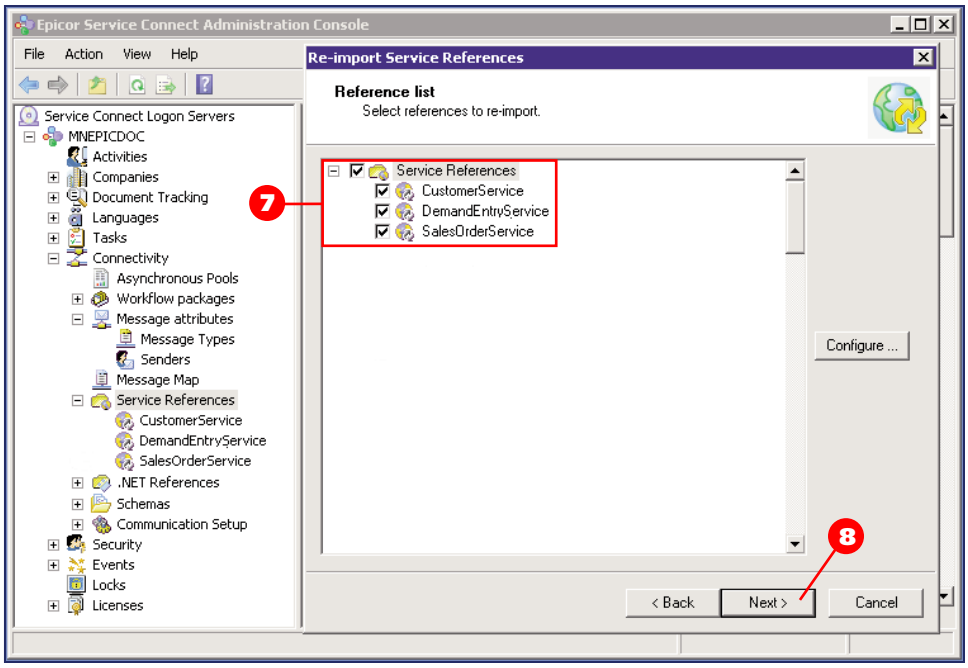


6. Click **Next**.

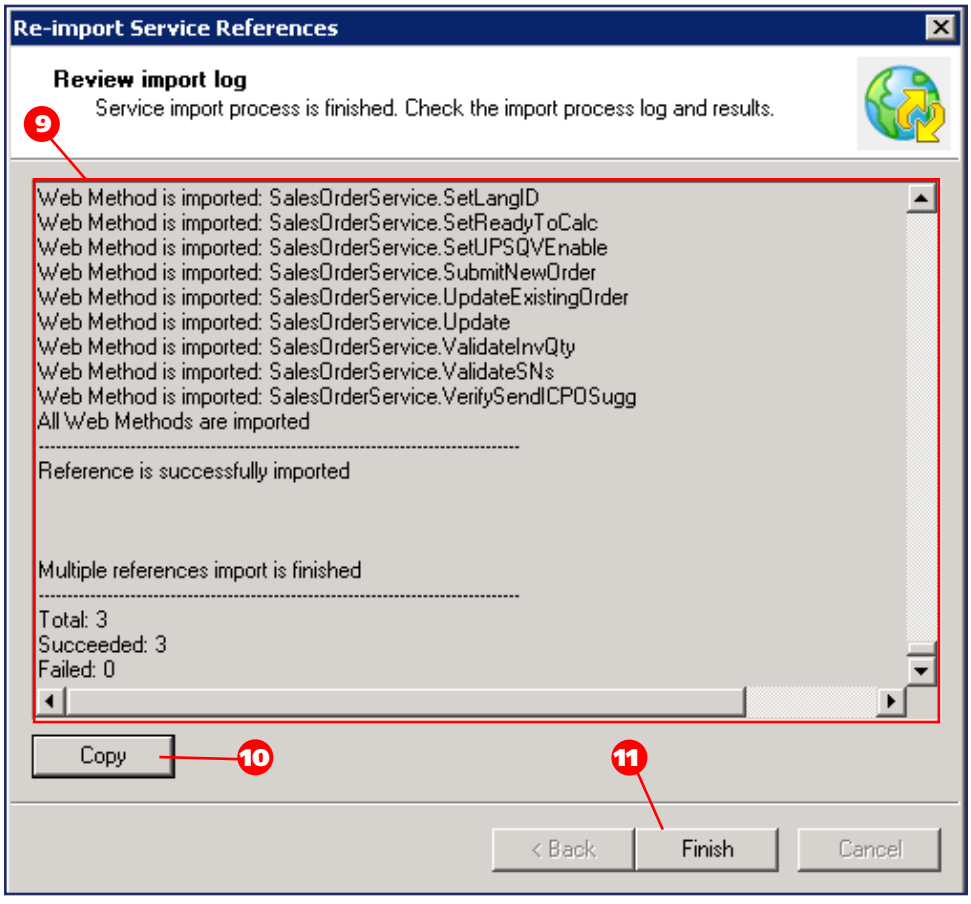


7. Review the re-import information.
8. Click **Next**.

   Depending on the number of services you re-import, the process can take several minutes.



9. When the process is complete, the application displays a dialog box that shows the settings used to re-import each service plus a summary that shows how many services were successfully re-imported and how many failed, if any.
10. You can use the **Copy** button to copy the re-import log to the clipboard.
11. Click **Finish**.

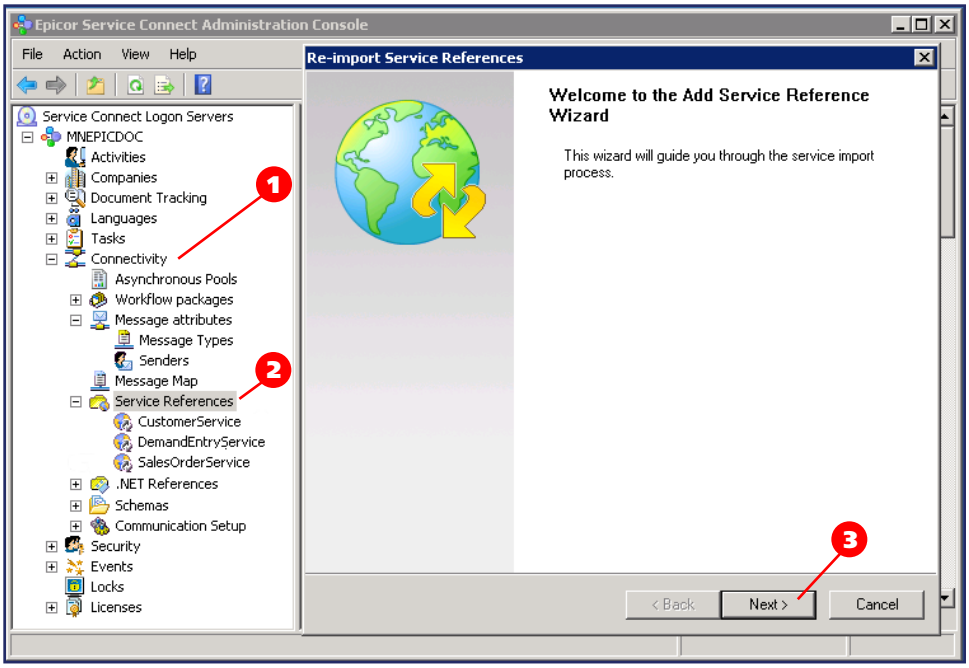


You can also re-import an individual web service.

1. In the **Tree View**, expand the **Connectivity > Service References** node.
2. Right-click the service reference to re-import and select **Re-import**.
3. The **Re-import Service Reference** window displays. Click **Next**.

   The re-import wizard takes you through the same windows you used when you first added the web service.

   During service reference re-import, Service Connect uses the settings originally used to add the service as default. You can adjust the re-import settings.

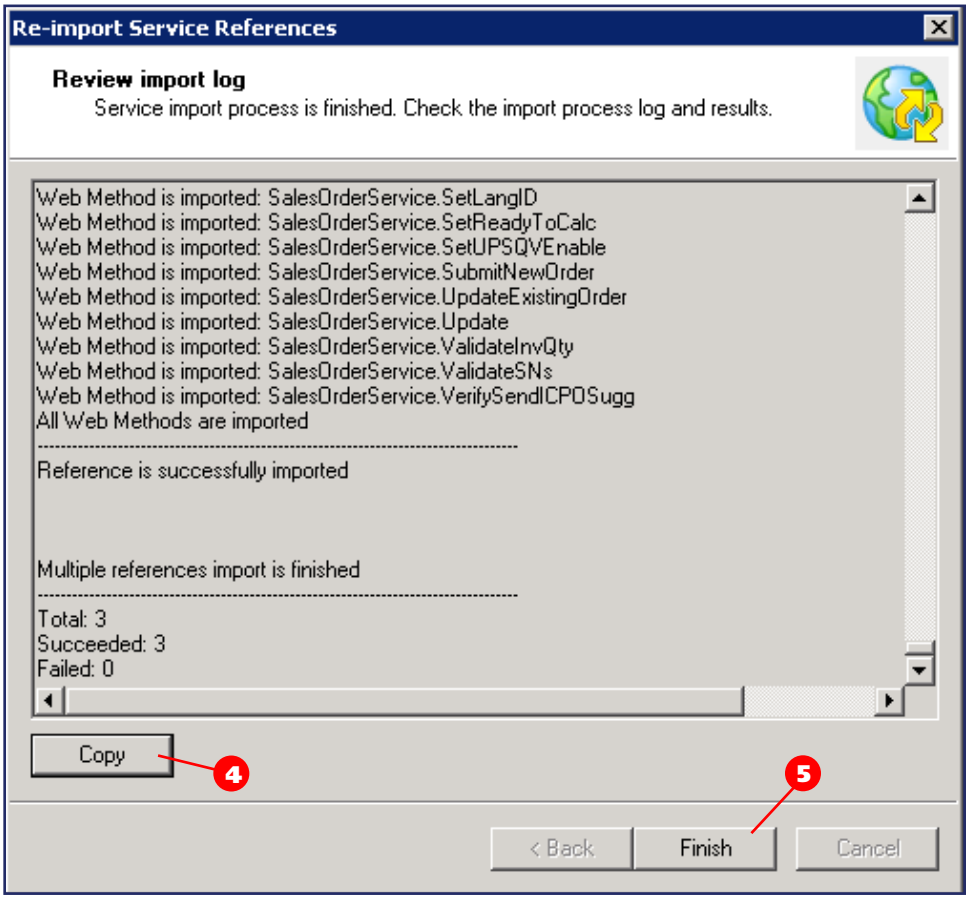


4. When the process is complete, the re-import log displays.

   You can use the **Copy** button to copy the re-import log to the clipboard.

5. Click **Finish**.

Depending on the number of services you re-import, the process can take several minutes. When the process is complete, the application displays a dialog box that shows the settings used to re-import each service plus a summary that shows how many services were successfully and unsuccessfuly re-imported.



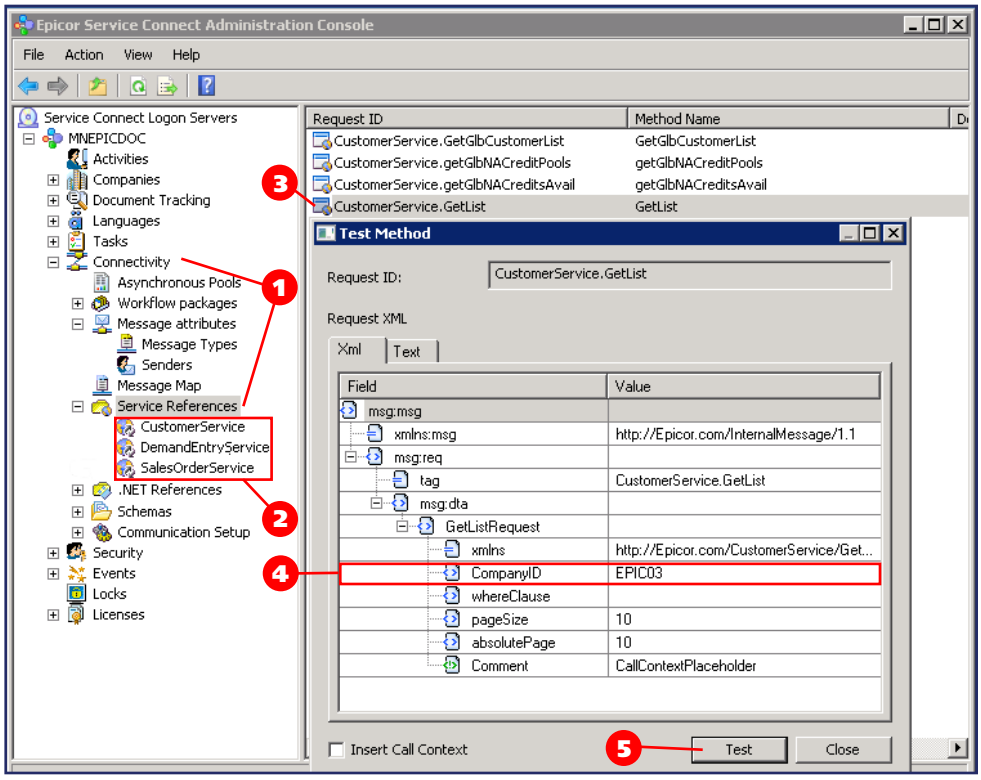
Test Connectivity

After you add a service reference, you can perform a connectivity test of the Web Service methods. The test functionality uses the schema created during the Web Service import to generate a sample xml. The response is displayed in xml format.

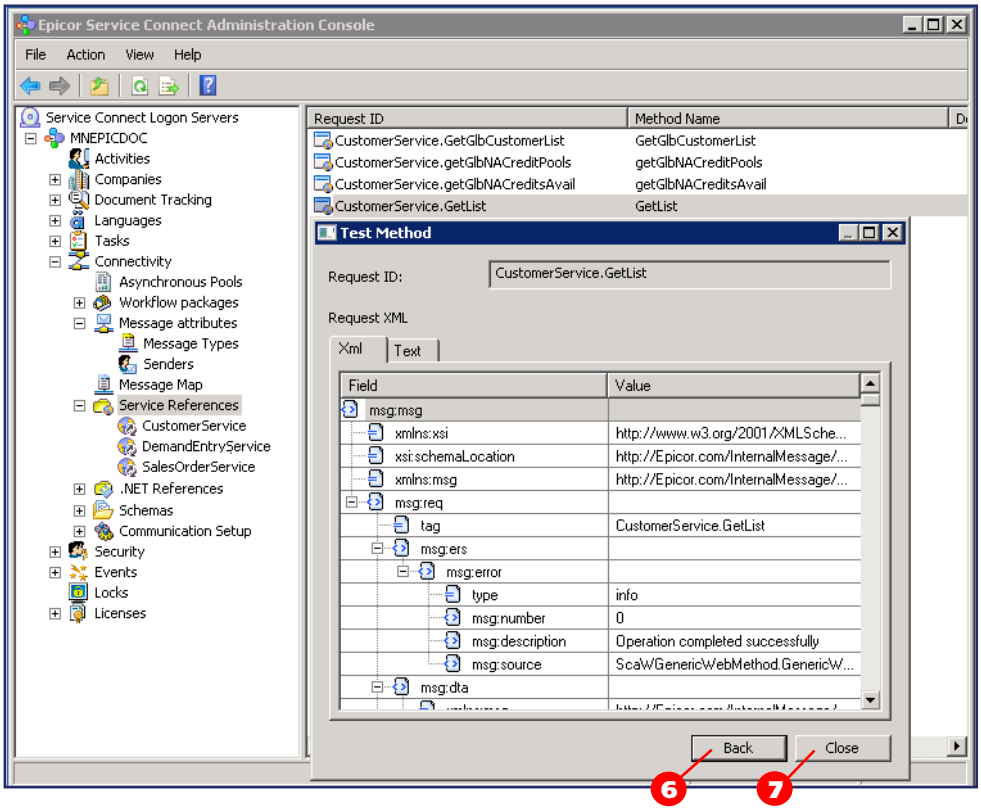
To test the connectivity of a Web Service and its methods:

1. In the **Tree View**, expand the **Connectivity > Service** reference node.
2. Select the Service Reference to test. In this example, select CustomerService.

   The list of its methods is displayed in the right pane.
3. In the right pane, right-click the method to test and select **Test**.
4. The **Test Method** dialog opens and displays the request XML structure (Xml tab) and the xml text (Text tab). In the Value column, enter the criteria the tested method expects. In this example, you specify the CompanyID for which you want to perform the connectivity check.
5. Click **Test**.



6. The connectivity check is performed and the response message displays. When the test is complete, you can change the request xml and test it again. To do this, click the **Back** button, edit the xml, and click **Test** again.
7. Click **Close**.



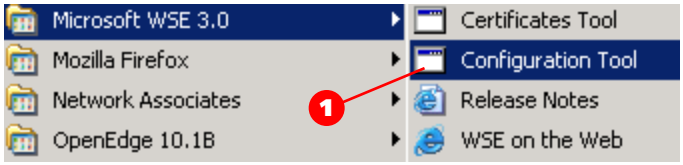
Use Web Services Enhancements 3.0

The **Web Services Enhancements** (WSE) 3.0 for Microsoft® .NET is an add-on to Microsoft Visual Studio® 2005 and the Microsoft .NET Framework 2.0 that enables developers to build secure Web services based on the latest Web services protocol specifications.

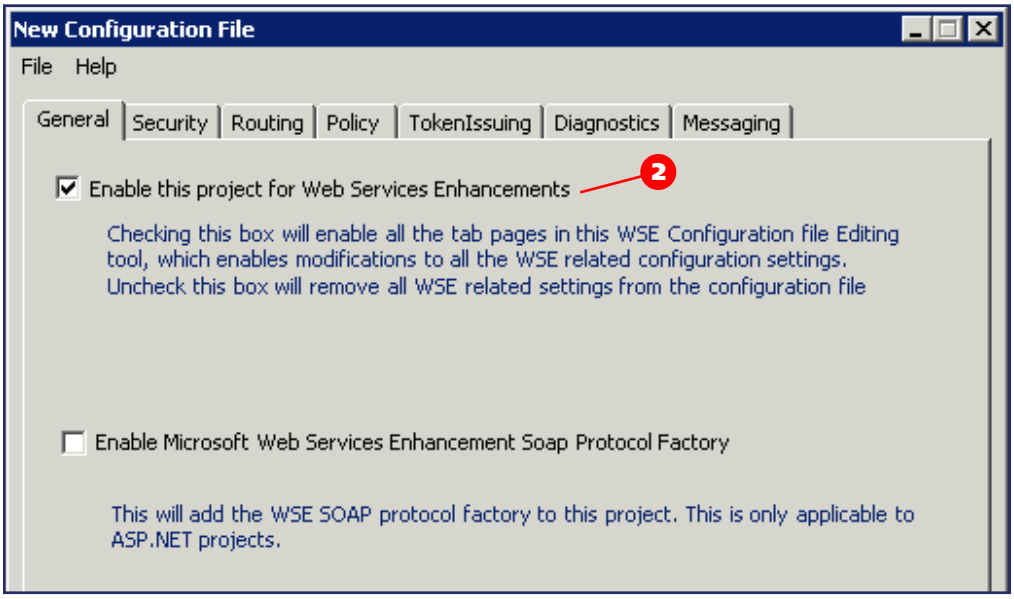
Epicor Service Connect supports WSE 3.0 server policies.

To create the WSE 3.0 server policy:

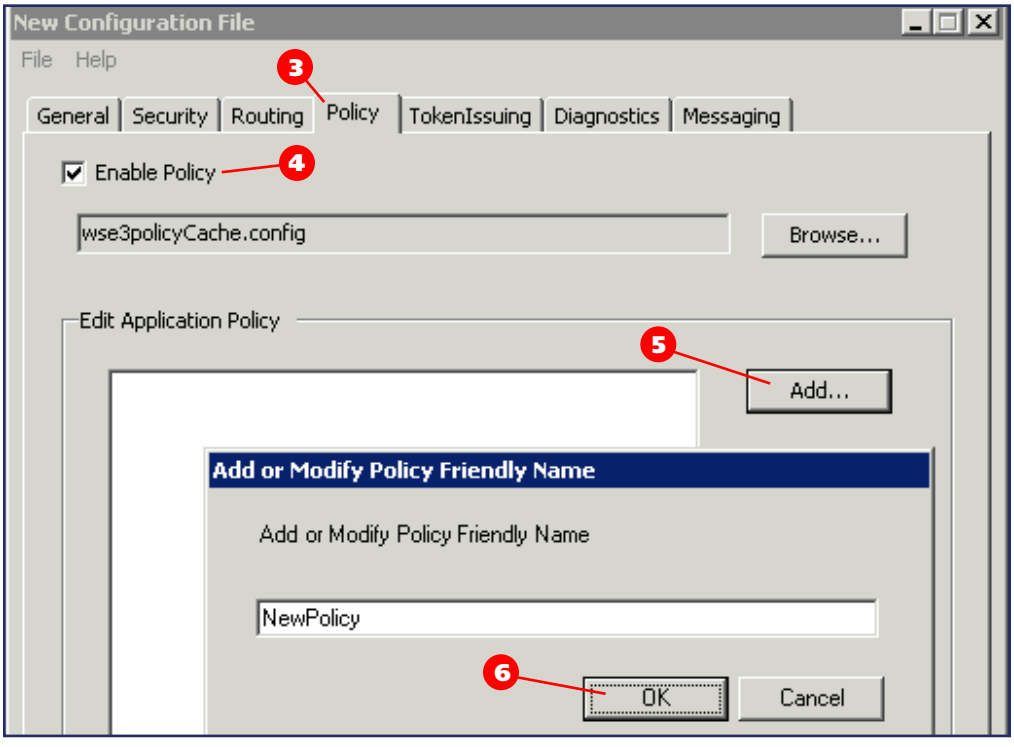
1. Start the **WSE 3.0 Configuration Tool** to create a WSE 3.0 policy file.



2. In the **New Configuration File** window, select the **Enable this project for Web Service Enhancements** check box.

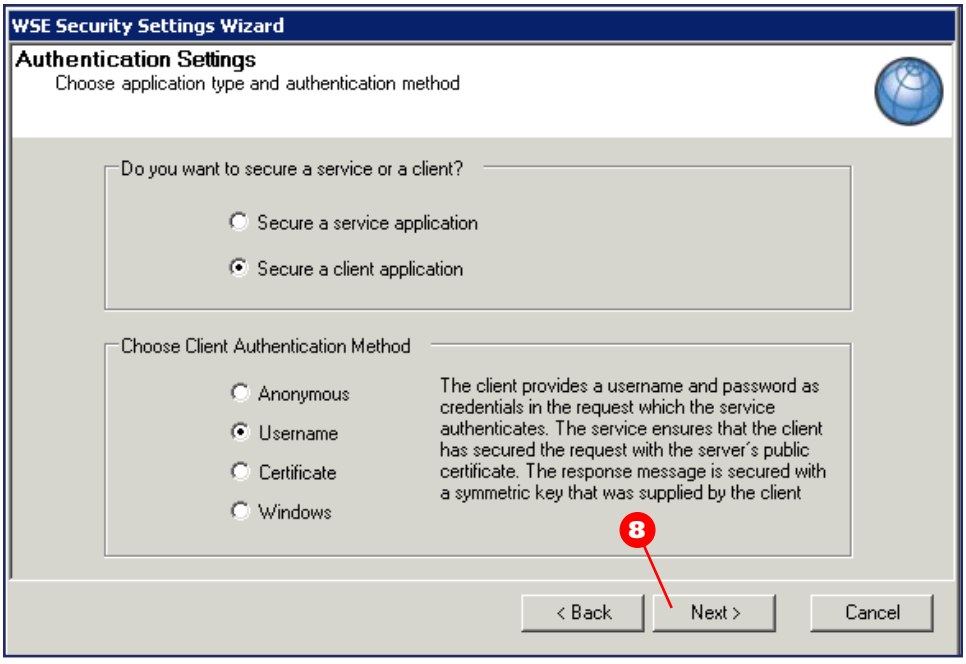


3. Navigate to the **Policy** sheet.
4. Select the **Enable Policy** check box.
5. Click the **Add** button.
6. Enter a name for your policy and click **OK**.



7. In the **WSE Security Settings** Wizard, click **Next**.

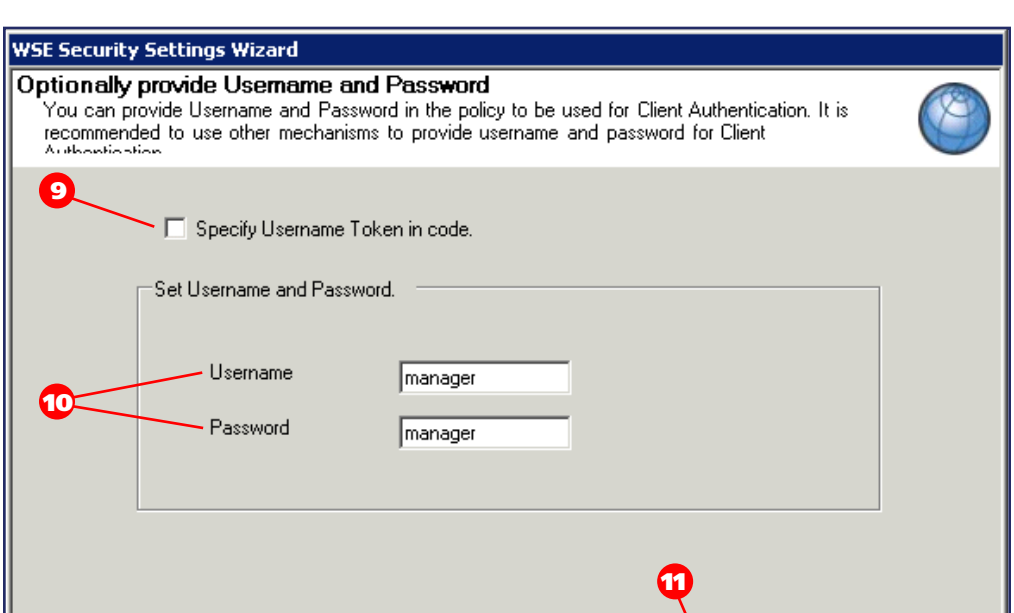
8. For the **Client Authentication Method**, select **Username** and click **Next**.



9. Clear the **Specify Username Token in code** check box.

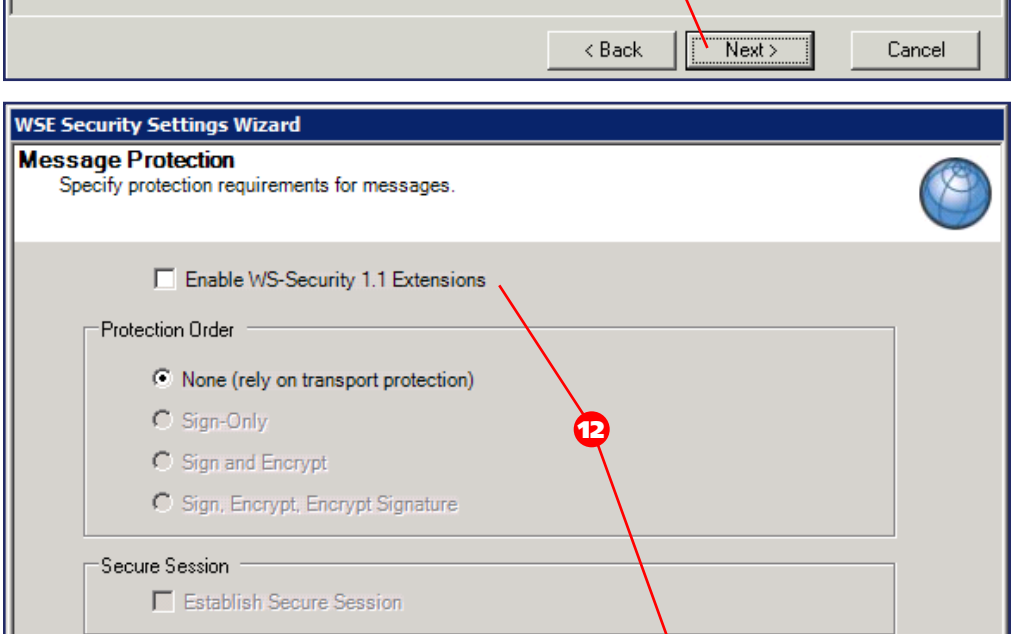
10. For the **Username** and **Password**, enter credentials you use to connect to your Epicor application, for example, manager / manager**.**

11. Click **Next**.

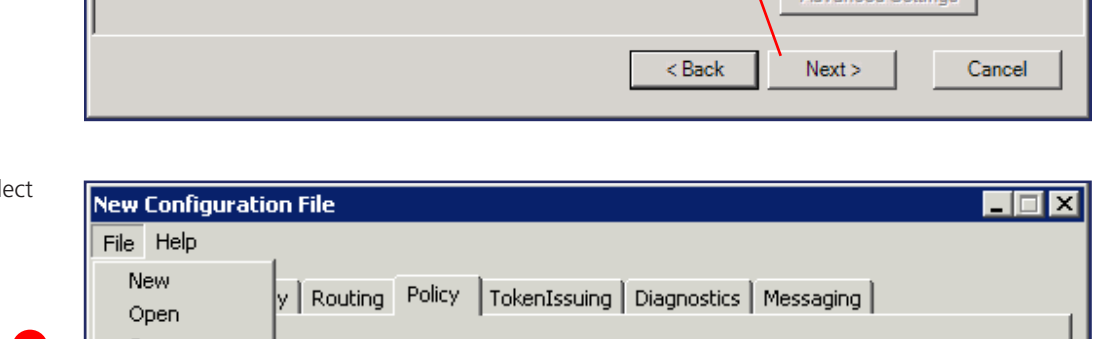


12. In the following window, clear the **Enable WS Security 1.1 Extensions** check box and click **Next**.

13. Click **Finish**.

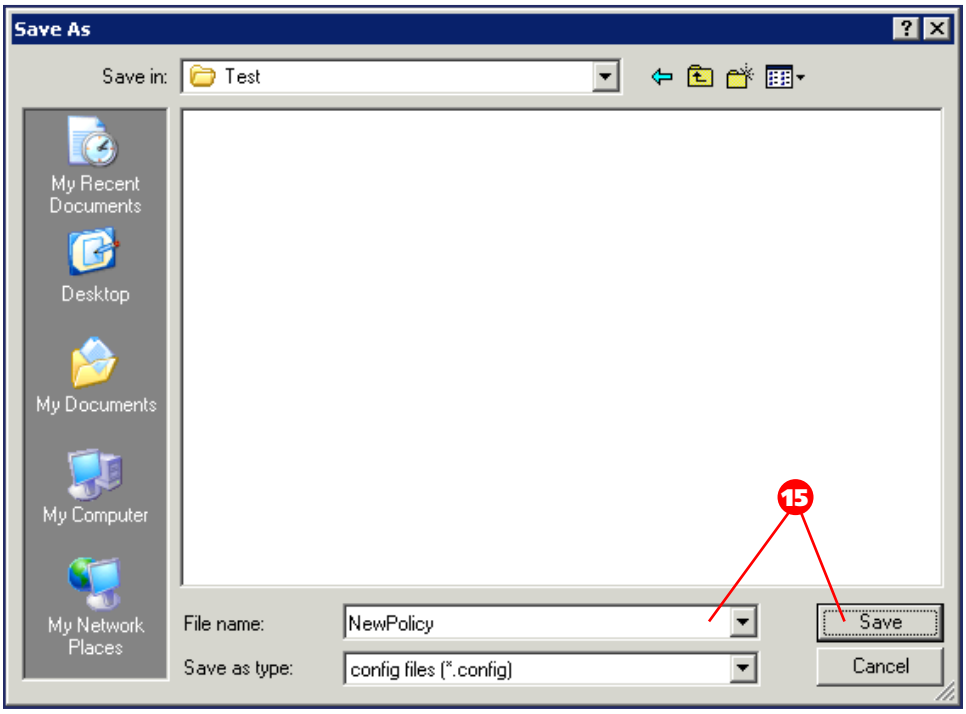


14. From the **File** menu, select **Save As**.



15. Enter a **file name** for the policy and **Save** the file.

16. **Exit** the WSE 3.0 Configuration Tool.



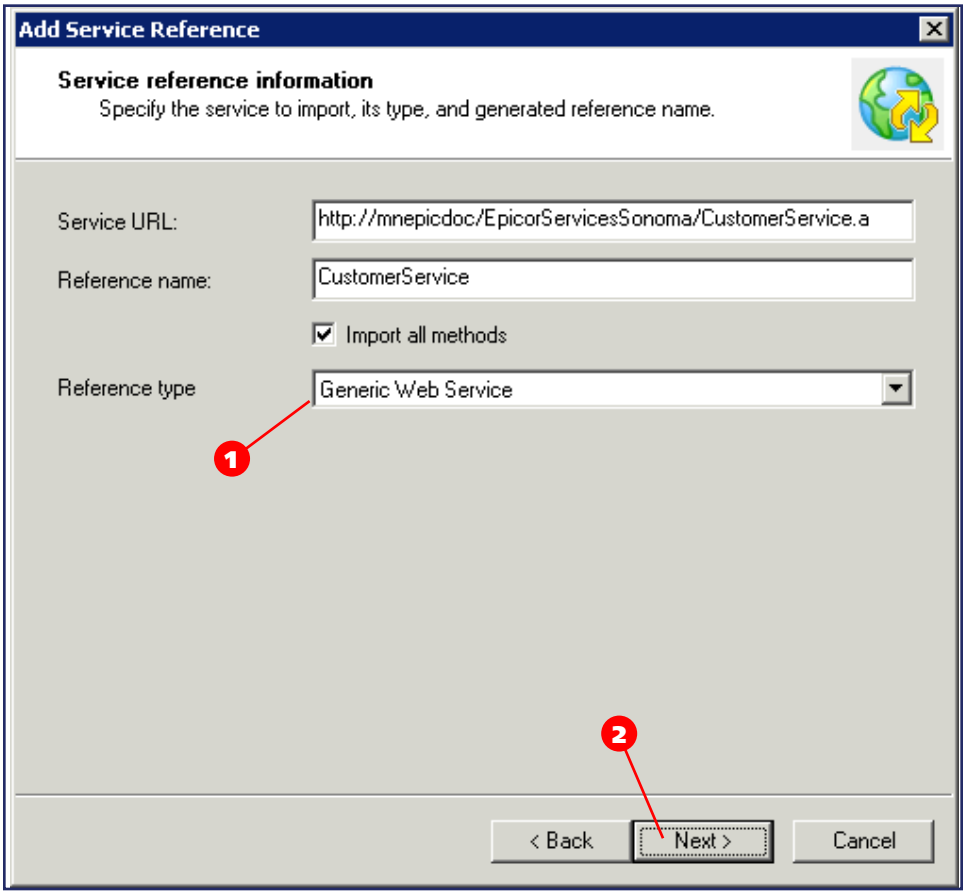
To apply the WSE 3.0 server policy:

For instruction on how to get to the Add Service Reference window seen here, review steps 1 and 2 of the Add a Service Reference section on page 29 of this manual.

1. While adding a new service reference, in the **Service Reference Type** field, click the **Reference type** drop-down list and select **Generic Web Service**.

   To apply the WSE 3.0 policy, you must select the Generic Web Service type.

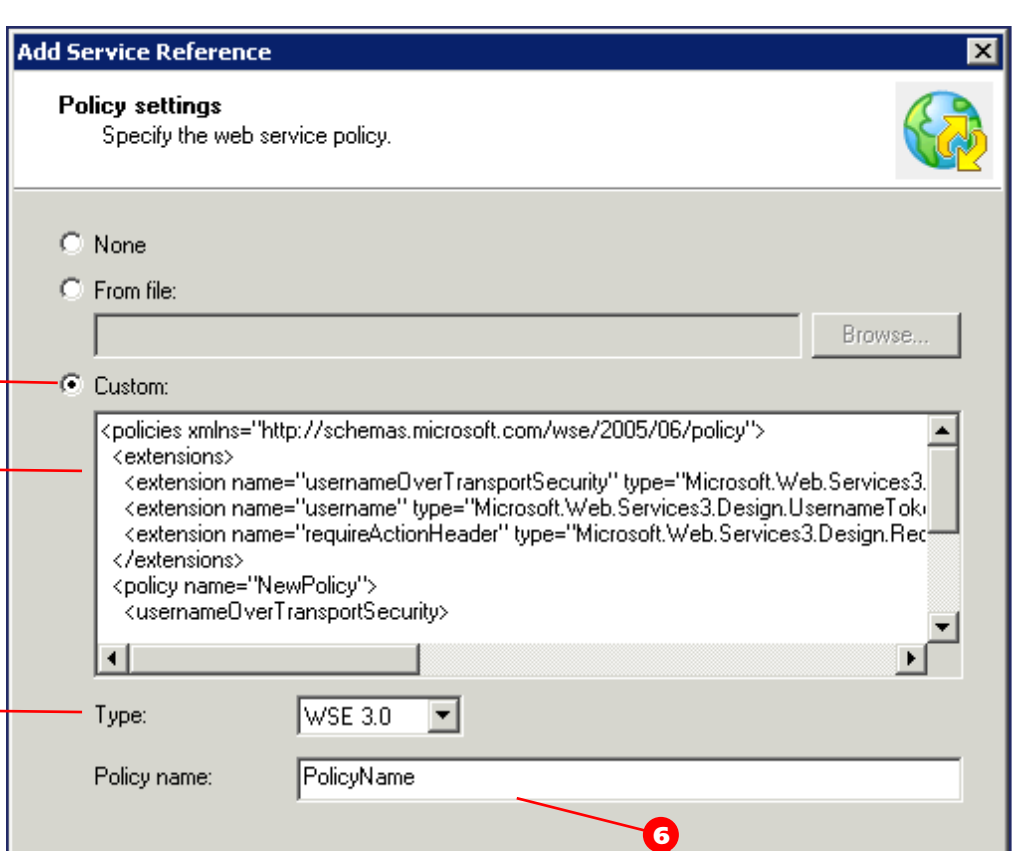
2. Click **Next**.



3. In the **Policy settings** window, select **Custom**.
4. Click the **Type** drop-down list and select **WSE 3.0**.
5. Copy the content of the policy file you created using the **WSE 3.0 Configuration Tool** and paste it in the **Custom** section.
6. Enter a **Policy name** and click **Next**.
7. Click **Next** until you finish importing the service.

The service reference is now using a created policy.

You may test a service method to verify the policy works properly.



.NET References

You can call .NET object methods inside Service Connect workflows.

The following object methods can be used in Service Connect:

- Public methods of public classes that have a default constructor
- Public static methods of public classes
- Epicor Business Objects

When you call .NET assemblies, performance improves and you can replace web service calls with direct Epicor .NET object calls. You can also create a wide range of customizations and custom functionalities in any .NET language. Similar to web service references, .NET references must first be imported. You may improve performance of imported .NET Epicor references by selecting the Cache Epicor sessions option in the Properties menu of the .NET References node.

Add a .NET Reference

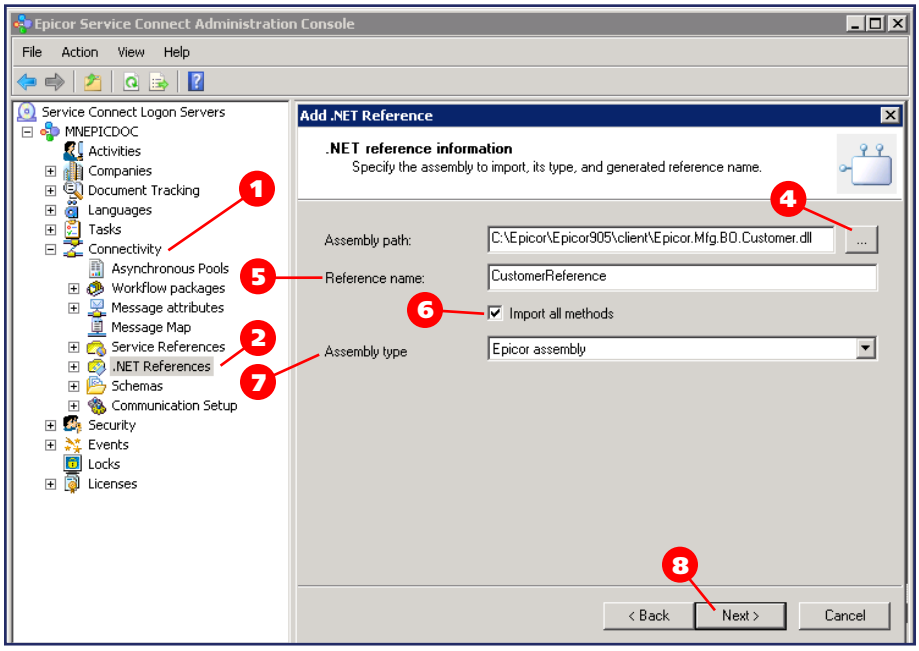
To add a .NET reference:

1. In the **Tree View**, expand the **Connectivity** node.
2. Right-click **.NET References** and select **Add Reference**.
3. On the **Add .NET Reference Wizard** welcome page, click **Next**.
4. Next to the **Assembly path** field, click the **... (Ellipse)** button, find and select the .NET assembly file. Click OK.

   The .NET reference name must start with a Latin letter, include only Latin letters, digits, and spaces, and not be longer than 40 symbols. It must be unique for an installation.

5. Enter the **Reference name**. This name will display in the imported .NET references list.

   This is a path on the Service Connect server.



6. Leave the **Import all methods** check box selected to import all service methods.

   If you do not select **Import all methods**, you can add the methods one by one later and configure each method individually.

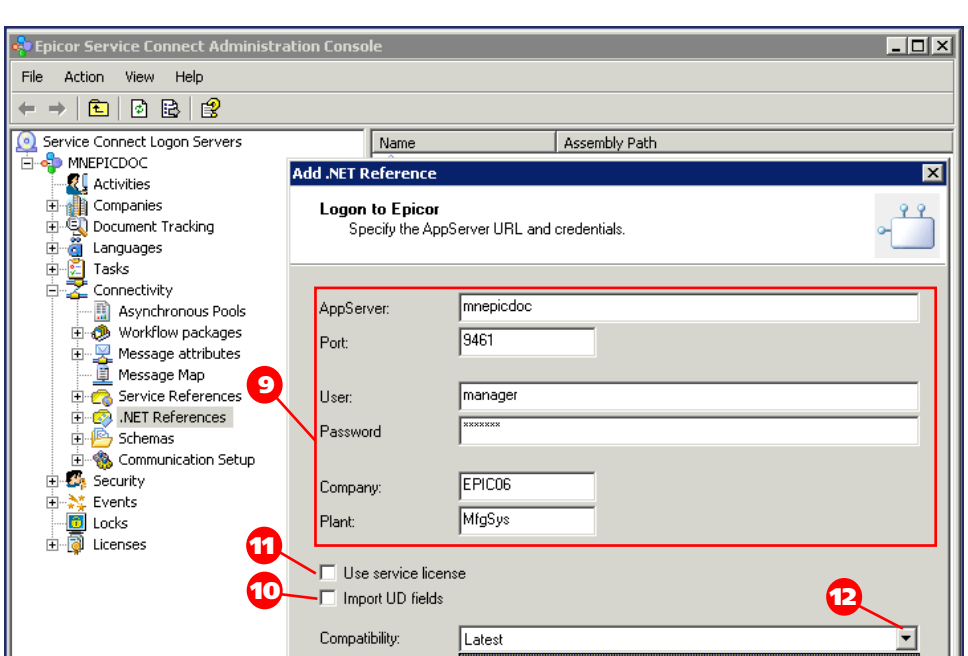
7. Click the **Assembly type** drop-down list and select select the .NET reference type. You can select Generic assembly or Epicor assembly.

   Epicor .NET references grant access to Epicor Business Objects. Service Connect performs some special operations like logon to Epicor system when working with Epicor .NET references.

8. Click **Next**.

9. If the **Assembly type** is **Epicor assembly**, enter the following Epicor logon settings:

   - AppServer
   - Port
   - User
   - Password
   - Company
   - Plant



10. If you want to import Epicor BOs with user-defined fields, select the **Import UD fields** check box.

    Service Connect identifies BOs with UD fields and makes them visible in BO dataset. User defined fields are then available for further processing.

11. Select **Use service license** check box to use a service license type for connection. If the number of users in the license is exceeded, connection is penalized (slowed down for 20 seconds) before being granted.

    If you leave this check box clear and service license is exceeded, Service Connect will use a default license.

12. Click the **Compatibility** drop-down list to select one of the following options:

    - **Latest** – Use this option if you are going to use Service Connect with the Epicor version 9.05.603 and later.
    - **Epicor 9.05.602** – Use this option if you are going to use Service Connect with the Epicor versions 9.05.600 – 9.05.602.

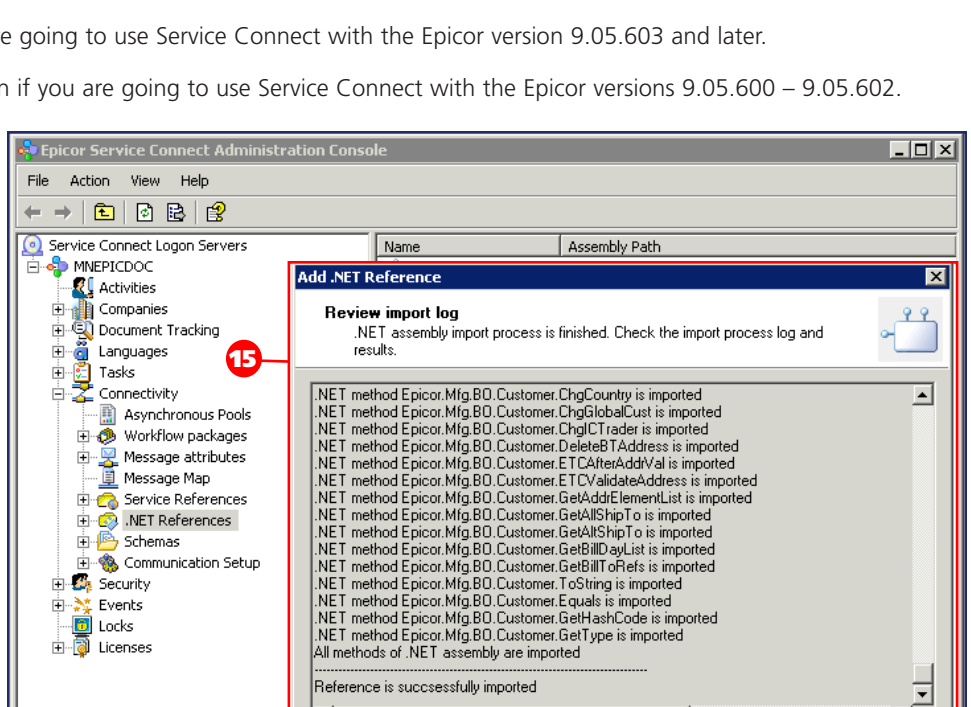
13. Click **Next**.

14. Review the .NET Reference Import information and click **Next**.

15. When the import is complete, detailed log displays on the **Review import log**.

    Use the **Copy** button to save the import log to the clipboard.

16. Review the log information and click **Finish**.



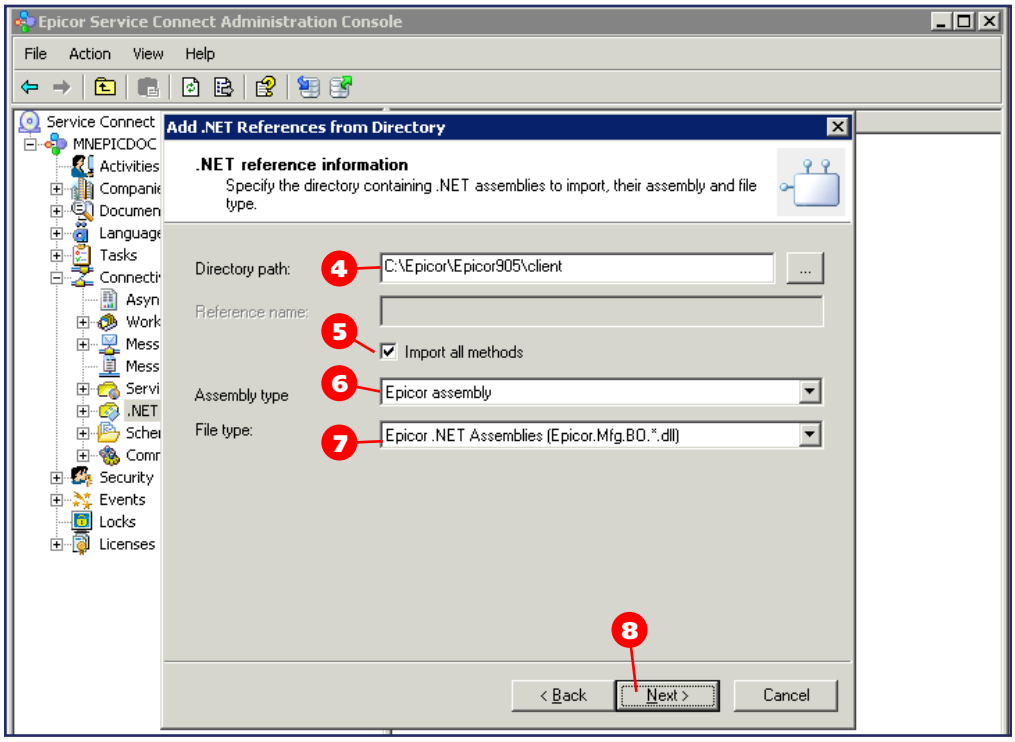
Add Multiple .NET References

To add multiple .NET eferences:

1. In the Tree View, expand the **Connectivity** node.
2. Right-click **.NET References** and select **Add References from Directory**.
3. On the **Add .NET Reference from Directory** welcome page, click **Next**.
4. Enter the **Directory path** to your local client.

   This is a path on the Service Connect server.
5. To import all service methods, leave the **Import all methods** check box.

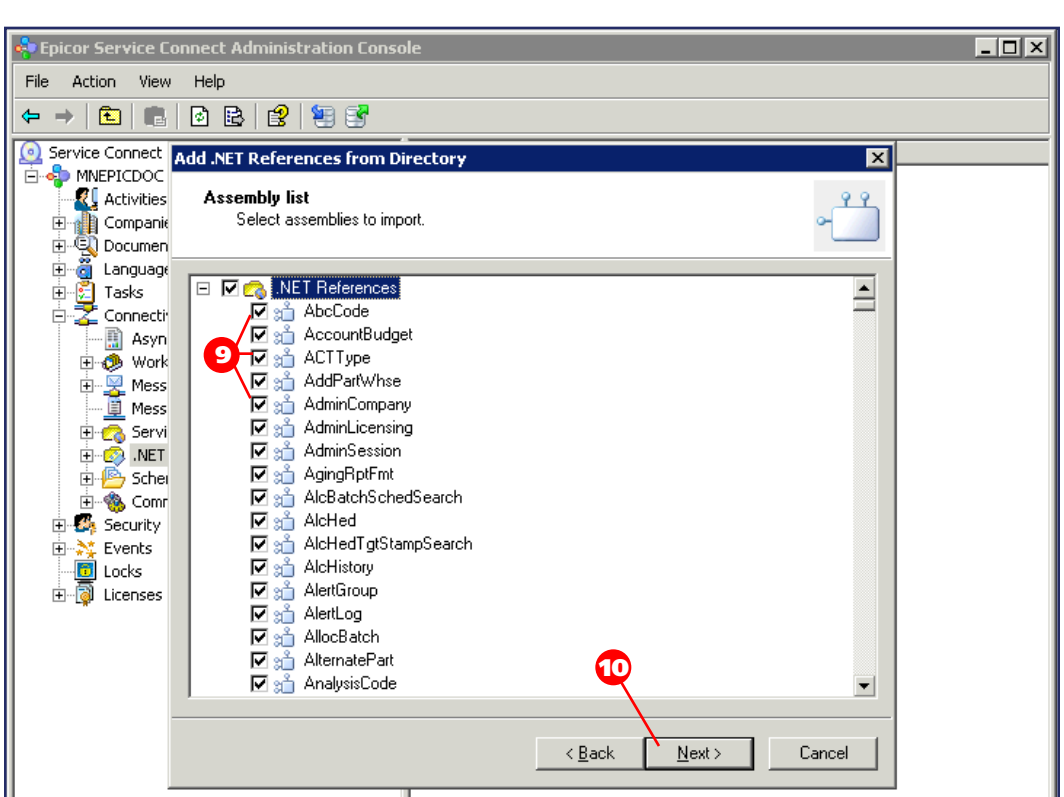
   If you do not select the **Import all methods** check box, you can add the methods one by one later and configure each method individually.
6. Click the **Assenbly type** drop-down list to select the .NET reference type. You can select Generic assembly or Epicor assembly.
7. Click the **File type** dropdown list and select one of the following options:



   - All Files (*.*)
   - NET Assemblies (*.dll)
   - Epicor .NET Assemblies (Epicor.Mfg.BO.*.dll)
   - Epicor .NET Assemblies (Epicor.Mfg.Proc.*.dll)
   - Epicor .NET Assemblies (Epicor.Mfg.Rpt.*.dll)
8. Click **Next**.

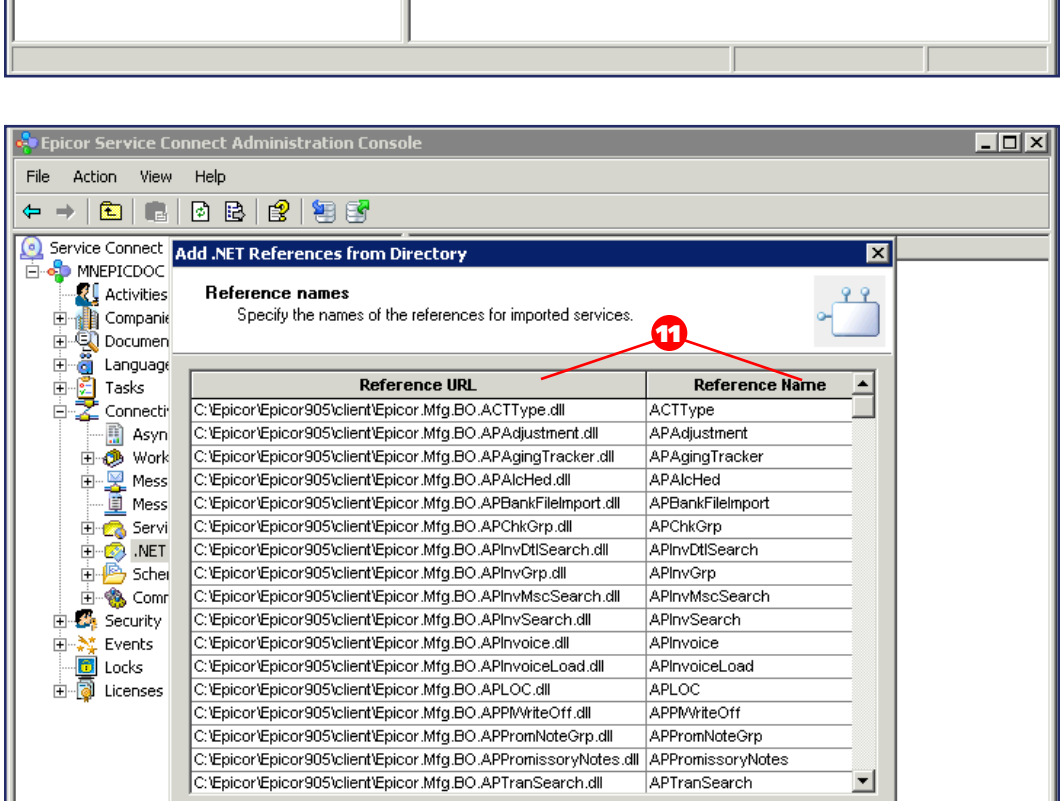
9. In the **Assembly List**, select the check box next to each reference you want to import.

10. Click **Next**.



11. Review the **Reference URLs** and optionally modify the **Reference Names**.

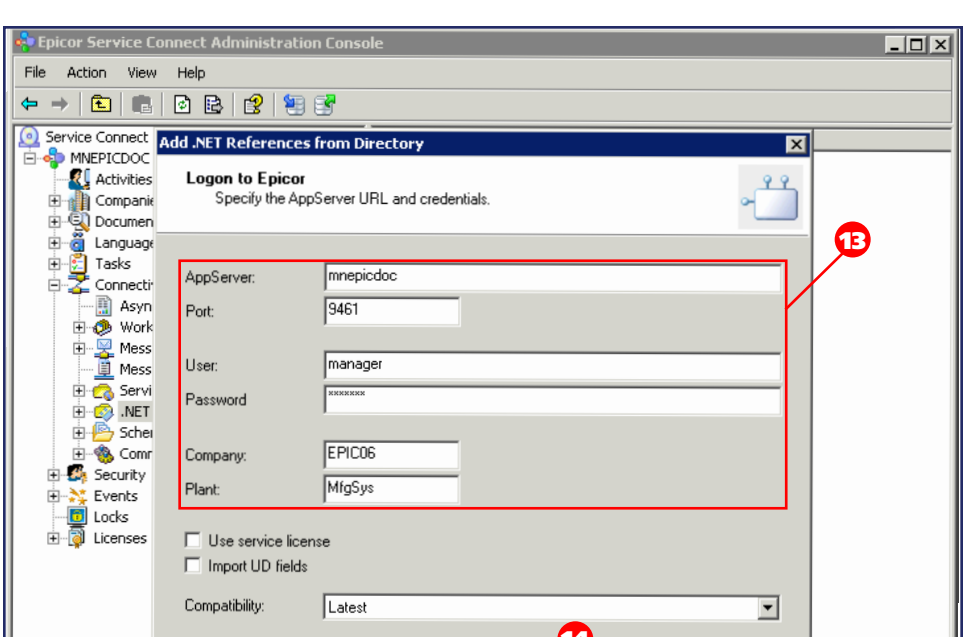
12. Click **Next**.



13. If the selected **Assembly type** is **Epicor assembly**, enter the following Epicor logon settings:

    - AppServer
    - Port
    - User
    - Password
    - Company
    - Plant

    Optionally, use Import UD fields, Use service license and Compatibility options as described before.



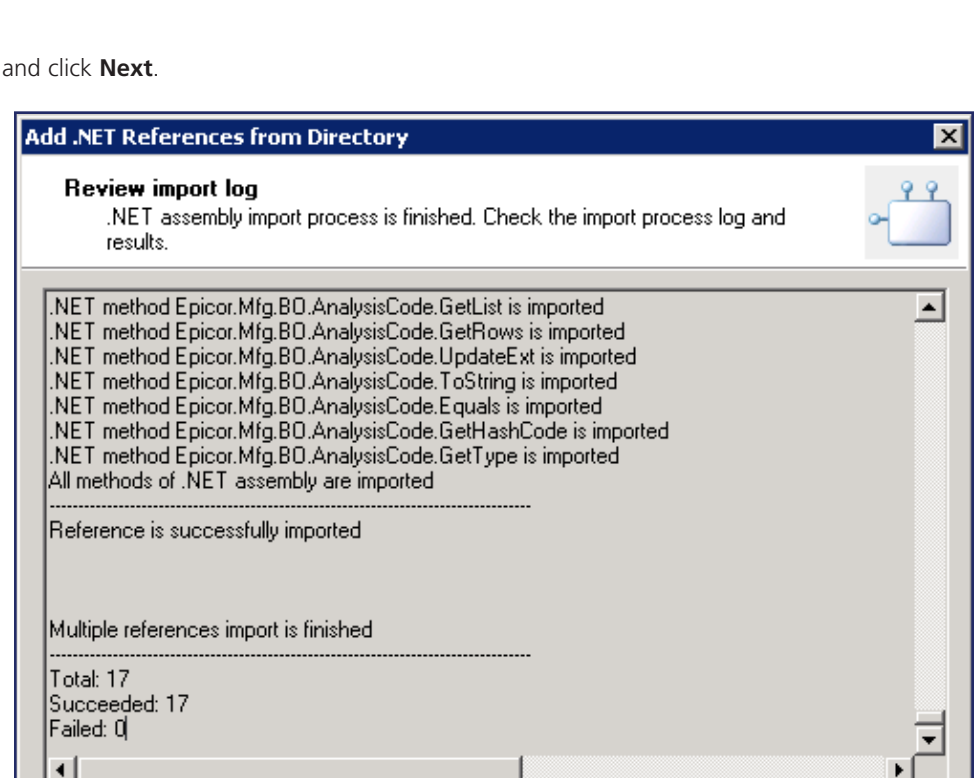
14. Click **Next**.

15. Review the .NET Reference Import information and click **Next**.

16. Review the completed information and click **Next**.

17. On the **Review import log** screen, you can use the **Copy** button to save the import log to the clipboard.

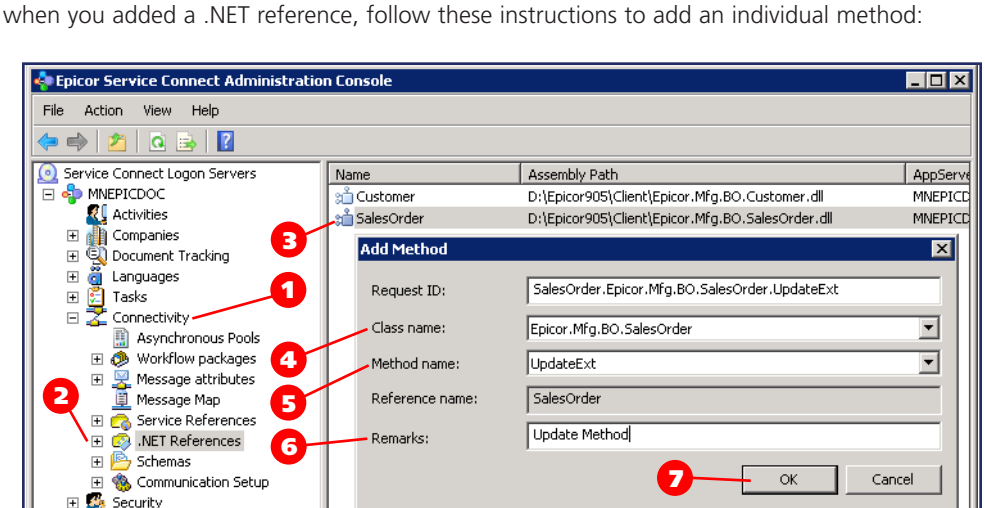
18. Click **Finish**.



Add .NET Methods

If you did not select Import all methods when you added a .NET reference, follow these instructions to add an individual method:

1. In the **Tree View**, expand the **Connectivity** node.
2. Click **.NET References**.
3. In the right pane, right-click the .NET reference to which to add a method and select **Add Method**.
4. The **Add Method** window displays. Select a **Class Name**.
5. Select a **Method Name**.
6. In the **Remarks** field, you can enter a description of the method.
7. Click **OK**.



When you add a new .NET method, a default Request ID is created from the .NET reference name. The system ensures the .NET Method Request ID is unique and does not coincide with the Request ID of another .NET Method or another plug-in. The registration information is stored in the system database.

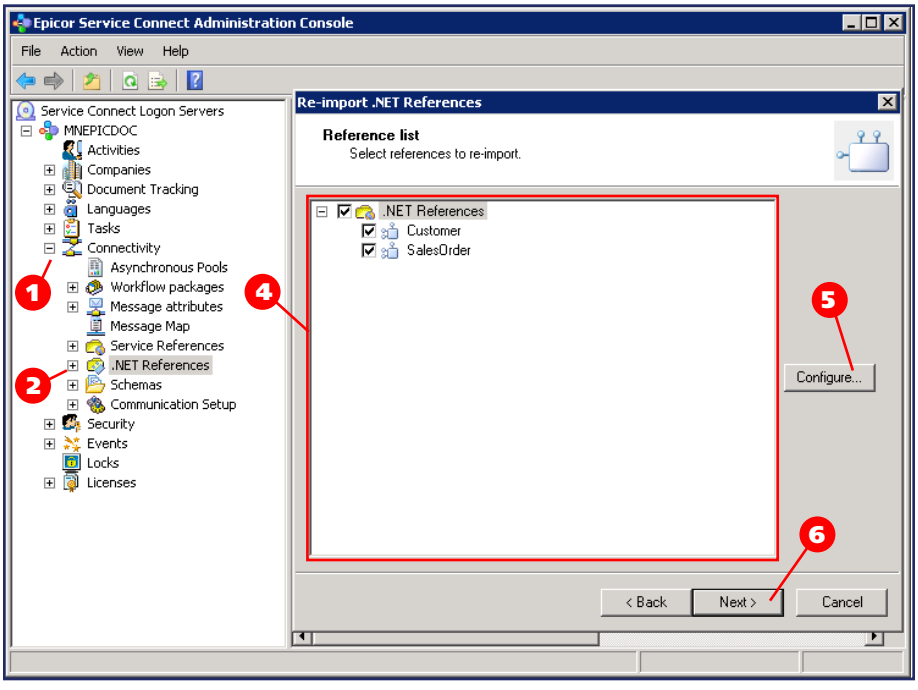
Re-Import .NET References

If you update the .NET assemblies, or if you install a newer version of Service Connect, follow these instructions to re-import the .NET references into the ESC Administration Console.

1. In the **Tree View**, expand the **Connectivity** node.
2. Right-click **.NET References** and select **Re-import References**.
3. On the **Re-import .NET References** welcome window, click **Next.**
4. From the **Reference** list, select the references to re-import.

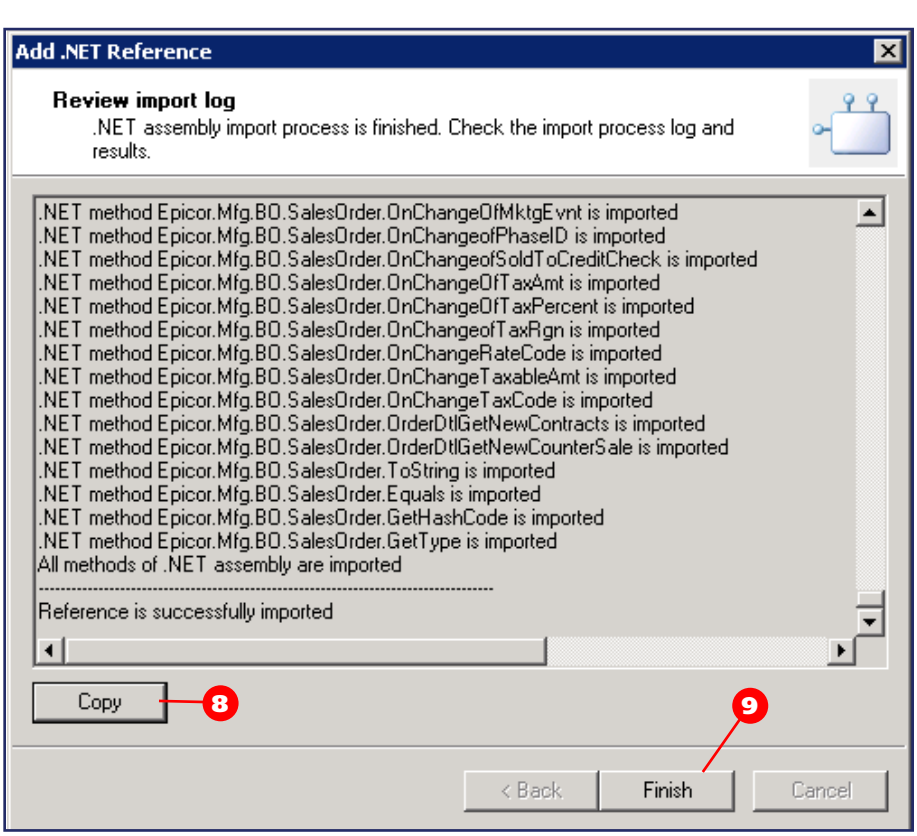
   All references are selected by default.
5. During reference re-import, Service Connect automatically uses the settings originally used to import the .NET reference. If you need to adjust the re-import settings, select the .NET reference, click **Configure** and change the settings.
6. Click **Next**.
7. Review the re-import information and click **Next**.

   Depending on the number of .NET references you re-import, the process can take several minutes.



8. When the process is complete, the application displays a dialog box that shows the settings used to re-import each .NET reference plus a summary that shows how many references were successfully and unsuccessfully re-imported. You can use the **Copy** button to copy the re-import log to the clipboard.

9. Click **Finish**.



You can also re-import an individual .NET reference.

> You must have security permissions to add or re-import a .NET reference. The .NET assembly must be located on the Service Connect server. If the .NET reference fails to import, verify the Assembly path and the Assembly type were correctly selected. If the .NET reference uses the Epicor assembly type, verify the logon information to the server is correct. If necessary, contact your system administrator

Test Methods

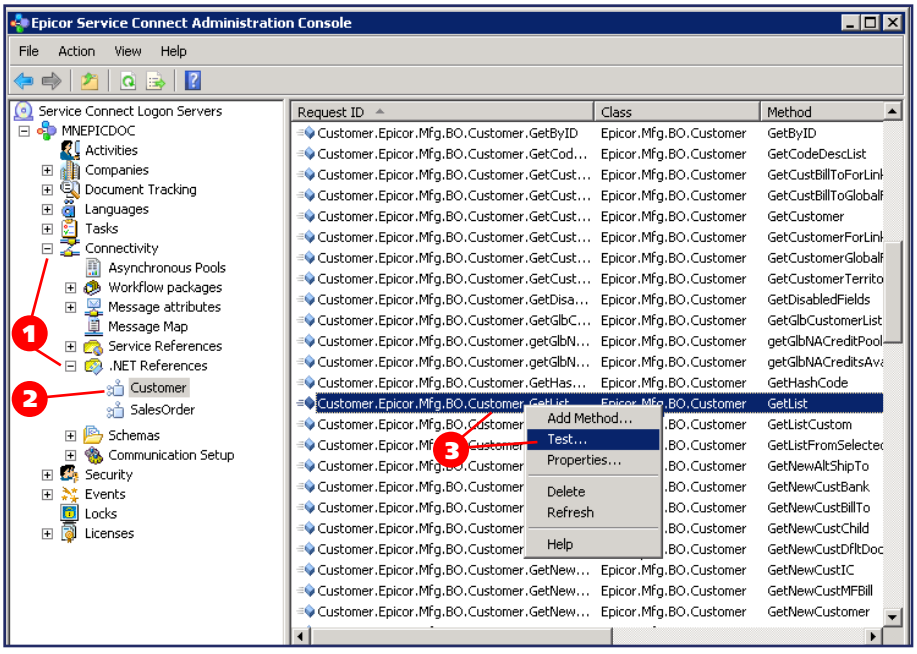
After you add a .NET reference, you can perform a connectivity test of the imported methods. The test method shows the response in xml format.

To test the connectivity of imported methods:

1. In the **Tree View**, expand the **Connectivity > .NET References** node.

2. Select a .NET Reference. In this example, select Customer.

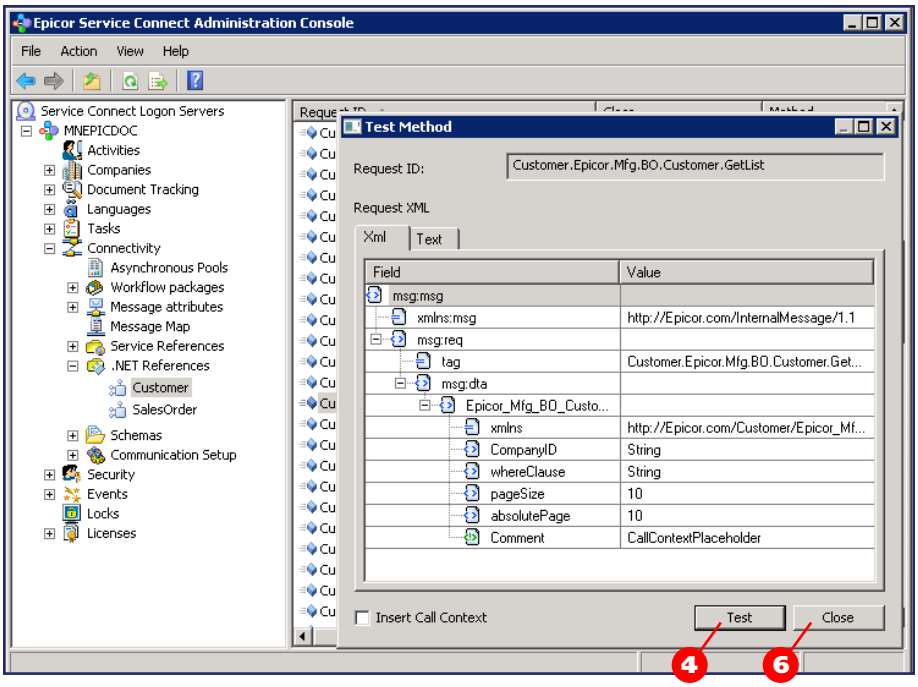
3. In the right pane within the methods list, right-click the method and select **Test**.



4. The **Test Method** dialog opens and displays the request XML structure (XML tab) and the xml text (Text tab). Enter the request criteria and click **Test**.

5. The connectivity check is performed and the response message displays. When the test is complete, you can change the request xml and test it again. To do this, click the **Back** button, edit the xml, and click **Test** again.

6. Click **Close**.



Epicor Session Usage Policy

Epicor sessions are re-used by different .NET references within the same workflow. If a workflow includes different .NET references that use the same credentials to access the same Epicor server and the same company, only the first .NET reference creates an Epicor session in the Epicor Administration Console. Other .NET references within the workflow use this cached Epicor session.

The session policy is applied to the sub-workflows as follows:

- Synchronous sub-workflows share the Epicor session created by the main workflow.
- Asynchronous sub-workflows do not share the Epicor session created by the main workflow.

Session is closed as soon as the main workflow completes or after the main workflow or one of its sub-workflows pauses on a Task element. Epicor references launched outside of workflow (from DES poster or input channel), use session caching mechanism.

If the main workflow did not reach a Finish element as a result of an error or it was aborted by document response processing, the session is closed within 10 minutes.

Schemas

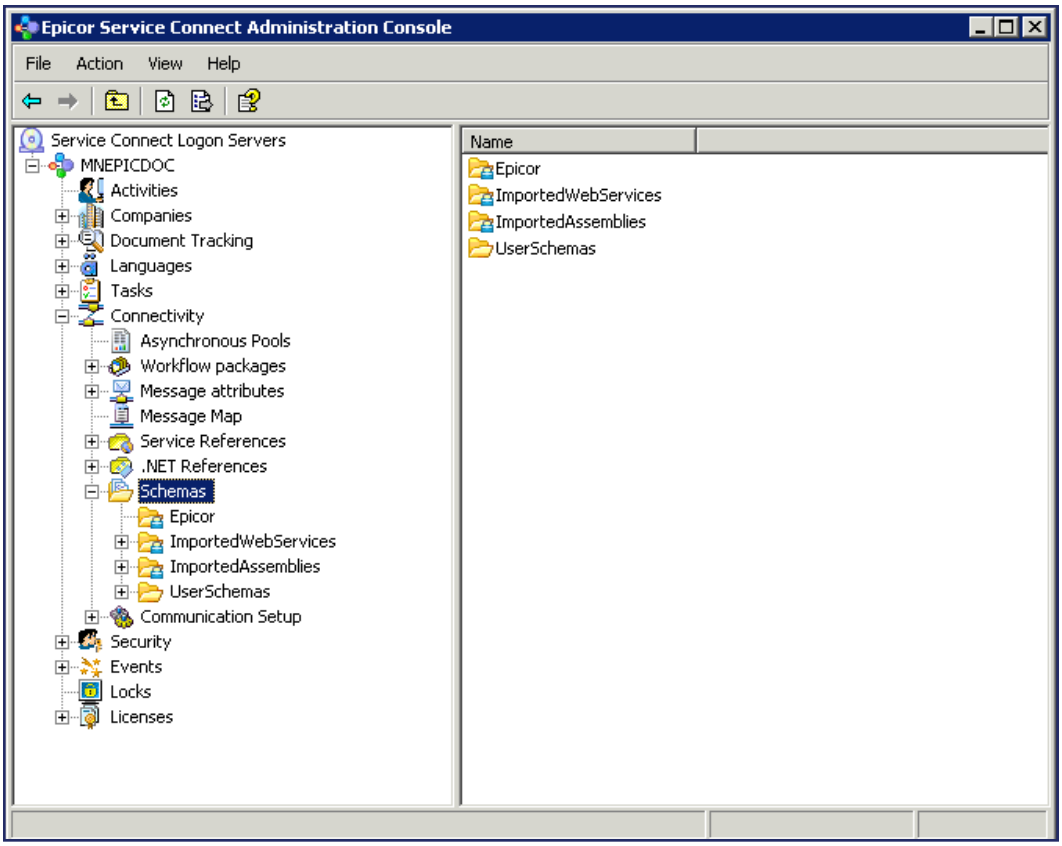
Use the Schemas node in the Epicor Service Connect Administration Console to perform schema-related tasks.

You can perform the following tasks:

- Browse through XSD schema files and folders.
- Create or delete schema files and folders.
- Rename schema files and folders.
- Generate an XSD schema based on the XML provided.
- Generate an XML based on the selected schema, registered in the system

For more information on schemas, review Chapter 04: Workflow Designer and Managing Schemas topic in the Service Connect application help.

Schema snap-in supports multi-site ESC installations.



Channels

Channels are the entry and exit points for documents going into and out of Service Connect. If you choose not to expose a workflow as a web service so it can be called directly or called using an Epicor application, you need to set up channels to get your documents into Service Connect and into the required internal message format. Depending on your workflow, you may not need an output channel. For example, if you complete the required work using a web service inside a workflow, you do not need to output the document.

Channels support various types of listeners for input and speakers for output. Listeners and speakers are either file-based or message-based. File-based listeners watch directories for files and message-based listeners watch queues or other protocol stacks for messages.

The four main parts of channel configuration include the following:

- Select a listener or speaker type and configure the appropriate settings, such as file directory path or queue path.
- Select a scan interval or schedule for the listener.
- Set the message attributes to use when the document is wrapped into the internal message format.
- Select a conversion plug-in to convert the document into or out of the internal message format.

The following fields display for each channel:

- **Name** – This is the channel name.
- **Thread** – This is an index of the thread to which the channel is assigned.
- **Busy** – This field indicates if the channel source has more messages to be processed.
- **Processed messages** – This is the number of successfully processed messages. Received messages are counted for input channels, sent messages are counted for output channels.
- **Messages to send** – This is the number of messages to be sent for output channels.
- **Size** – This field displays the average size of processed messages.
- **Communicator time** – This field displays the average communicator working time.
- **Conversion time** – This field displays the average time of converter.
- **Processing time** – For synchronous input channels , this field displays the workflow processing time. For asynchronous input channels, this field displays the time of posting to asynchronous queue. This field is not calculated for output channels.

Add a Channel

The first step to add a channel is to decide which listener or speaker type to use, based on technology, and then set up the required parts. For example, if you are going to use Microsoft Message Queuing (MSMQ), you need MSMQ enabled on the Service Connect server and other relevant servers; you also need at least one queue configured. Or, if you are going to use a file listener, create the directory on the file system the listener will access.

- **Listener types** - Include COM, File, FTP, IBMMQ, IBMMQ (.NET), POP3, IMAP and Schedule types, MSMQ, and SonicMQ
- **Speaker types -** Include COM, File, FTP, HTTP, IBMMQ, IBMMQ (.NET), SMTP, MSMQ, and SonicMQ

If the data inside a workflow should be posted to a specified destination, the message can be routed to a Poster activity. In the Poster properties sheet, one or more output channels can be specified.
The decision of choosing a technology is yours. You should have some understanding of the technology and be able to attach the documents you want to send to Service Connect in the selected technology. For example, if you are using MSMQ, your sending application must be able to wrap a document in the format the MSMQ requires and send it to a queue. Use a technology you are familiar or are willing to learn. Some are less complex to use but lack features that may be helpful. For example, MSMQ has transactional support that can be configured to roll back unsuccessful transactions.

To set up an input channel:

1. In the **Tree View**, expand the **Connectivity > Communication Setup >** <machine name> **> Channels** node.

2. Right-click **Input Channels** and select **Add New**.

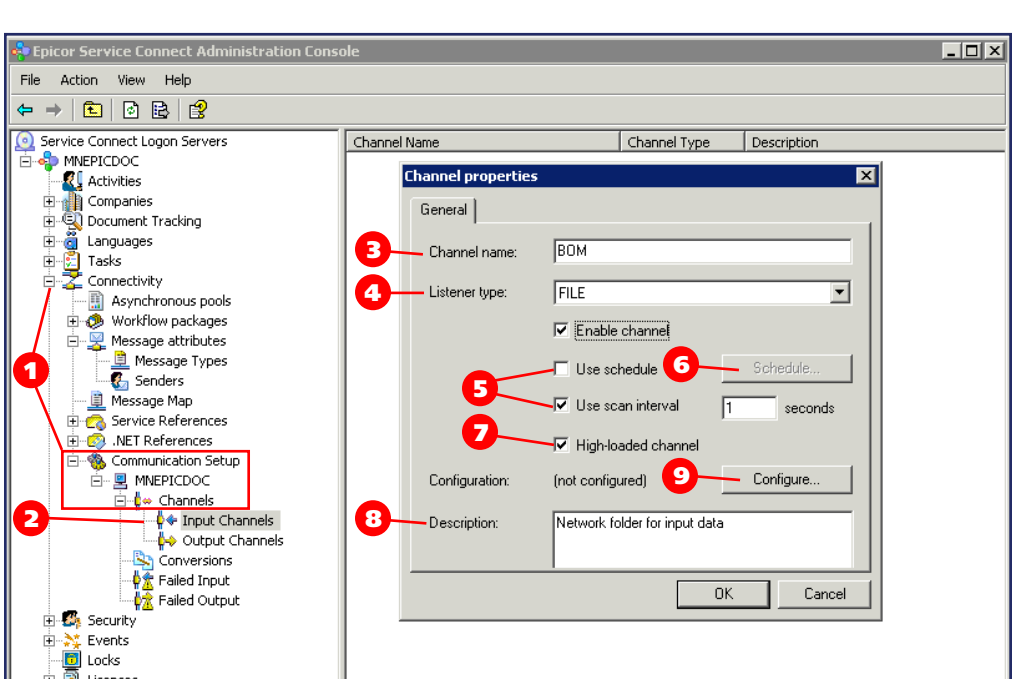
3. The **Channel properties** window displays. Enter a **Channel name**.

4. Select a **Listener type**.

5. Select **Use schedule** or **Use scan interval**.

6. If you select Use schedule, click **Schedule** to set a schedule. This will determine how often the listener is checked for documents.

7. You can configure the channel as a **High-loaded channel**.



When this option is selected, the system tries to allocate a separate thread for the channel. If it fails (no threads available), a warning displays in the event log (logged every five minutes) and the check box clears again. In this case, Epicor recommends you set up the system again so there are more threads available for the channels.

The processing time, in milliseconds, spent to get a response from a channel is measured using the Windows API function.

The High-loaded channel setting works together with the thread count setting; this means a channel will be immediately moved to a free thread to avoid interference. This occurs when two channels work on the same thread. When one channel reads a message and waits for the processing time, the other channel waits for the first one to finish reading. For example, when the first channel reads Files (which is relatively quick, about one second per message) and the second channel is POP3 (which is slow, about 60 seconds per message), then the File channel reads messages approximately every 61 seconds.

The same case occurs when both channels are quick (File or MSMQ) but synchronous. For example, the first channel starts a quick workflow, about 10 seconds, and the second channel starts a slow workflow, about 5 minutes. The first channel will spend time waiting for the second one instead of processing.

Service Connect moves channels between available threads, but it is not always efficient. In the case of a large input load, the time spent moving channels may take up to 20 minutes. This feature has been developed to avoid processing delays.

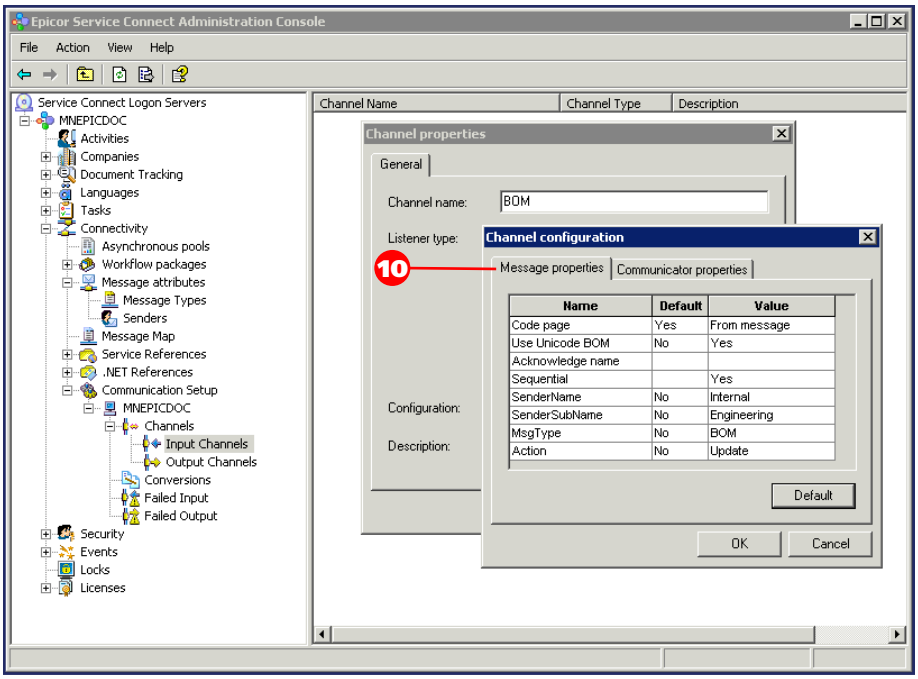
If you need quick channel reaction and need processing to start immediately, use the High-loaded channel option.

To set a maximum number of threads for a server, expand the Communication Setup node. Right-click the server and select Properties from the context menu. On the General tab, the Number of Messenger threads can be selected based on the number of channels and their sequential property. It should be set to at least the number of High-loaded channels in addition to the number of all the sequential channels. If there are slow channels in the system, such as POP3 or SMTP, the number of threads should be increased as well.

8. Enter a **Description**.

9. Click **Configure**.

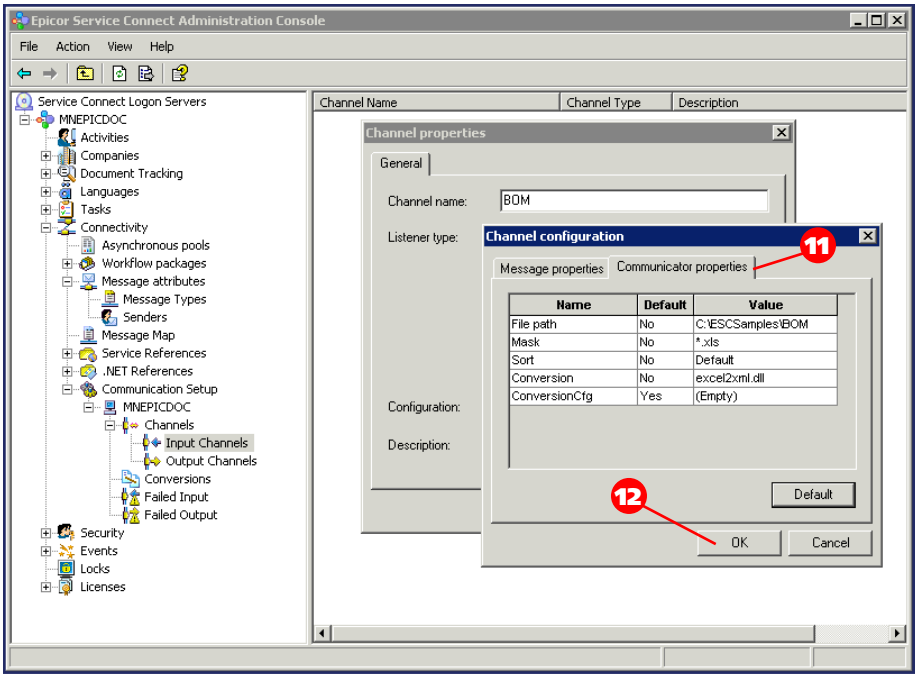
10. In the **Channel configuration** window, click the **Message Properties** tab to display message attributes that will be added to the incoming document. The message attributes are checked against the attributes set up for message maps. If the message attributes match the map attributes, the document is routed to the Request ID on the map. Review the Message Attributes and Message Maps section in this chapter for more information on how to use these properties. The properties most commonly set are SenderName, SenderSubName, MsgType, and Action.



11. Click the **Communicator properties** tab.

The properties listed first are based on listener type. All listeners have the Conversion property which you use to select the conversion tool that can convert your incoming document into the XML internal message format. Use the **ConversionCfg** property to enter a custom plug-in configuration. If you selected a .NET conversion, you may use the Validate button to verify your configuration. In case of an error, a warning message displays with detailed information about an error. You may also use the Default button to add the default configuration automatically.

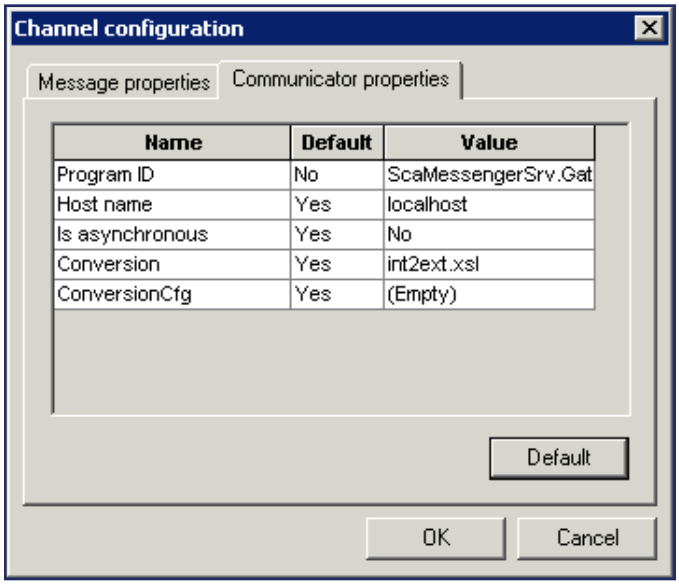
Listener and speaker types are explained in detail in the following sections.



12. Click **OK** until you exit all dialog boxes.

Configure COM Channel Options

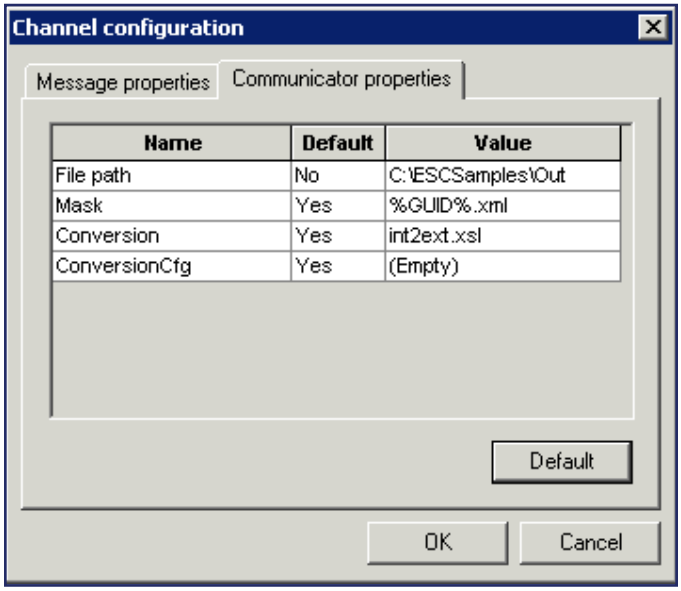
A COM communication channel can be both a listener and a speaker. As a listener, it is used exclusively to receive messages from a SharePoint workflow when Windows SharePoint Services is integrated with Service Connect. As a speaker, it is used to send messages to reusable Windows-based software components based on the Component Object Model.



Property	Description
Program ID	The ID of the program sending the message.
Host name	The name of the local server.
Is asynchronous	Indicates whether the message posts immediately or if it is queued.
Conversion	The conversion plug-in used before the message is sent.
ConversionCfg	Enter one or more configuration parameters. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

Configure FILE Channel Options

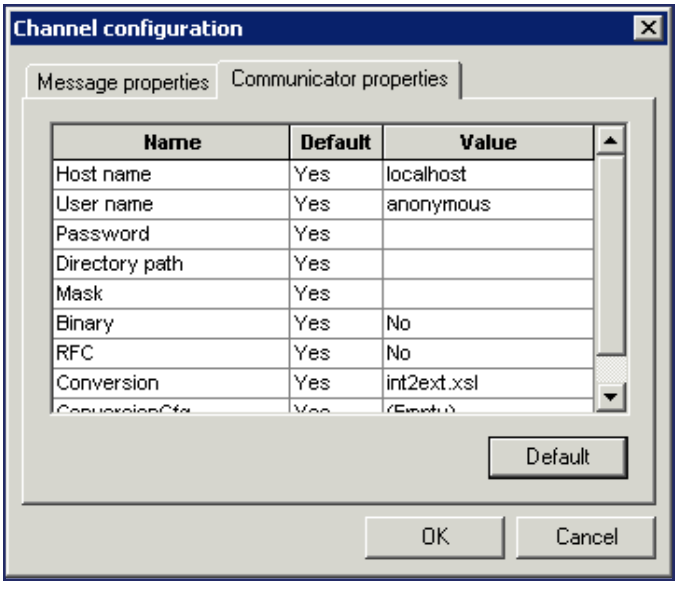
A FILE communication channel can be both a listener and a speaker. As a listener, it is used to consume files placed in a computer or network folder. As a speaker, it is used to send a message as a file to a folder on a computer or to a network folder where the file can be used by another application or even another workflow.



Property	Description
File path	The folder or network directory Service Connect will use as an input or output channel. You must create the folder before you enter the path into this properties sheet. If the folder does not exist, Service Connect displays an error. To use a network folder, use the Universal Naming Convention (UNC), which for Windows, begins with a double backslash; for example, \\server\folder. You cannot use a mapped network drive. To use an environmental variable, insert the variable name, surrounded by percent signs; for example, %sc_input_folder%. If you use a folder on a remote server, the Local System Account must be accepted by the server. If the Local System Account is rejected, you can change the ScsMessengerSrv.exe logon account to one with appropriate credentials.
Mask	For input channels, this is the type of file that will be accepted by Service Connect. To allow only CSV files, for example, set this field to *.csv. The field accepts both "*" and "?" as wildcard characters. For output channels, this indicates the format and name of the file that will be posted by Service Connect. The field accepts macros embedded within % signs. For detailed rules on filename templates and macros, refer to the FILE options topic in the Administration Console application help. The following macros are available: • %GUID% • %DATE(FORMAT)% • %TIME% • %DATETIME(FORMAT)% • %COUNTER% In the example above, the filename will be a globally unique identifier (GUID).
Conversion	The conversion plug-in used when consuming or posting a file.
ConversionCfg	Enter one or more configuration parameters. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

Configure FTP Channel Options

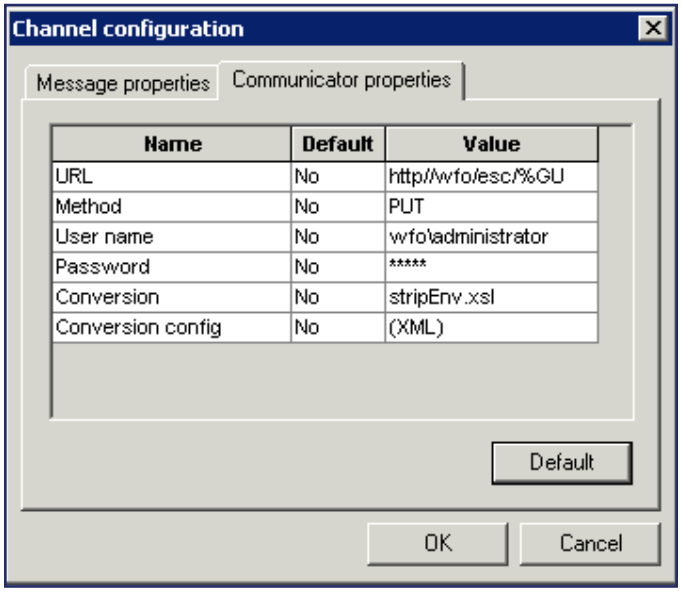
An FTP communication channel can be used as both a listener and a speaker. Use it as a listener to consume files placed in a specific FTP directory. Use it as a speaker to send a message as a file to an FTP directory where another application or even another workflow can use the file.



Property	Description
Host name	The name of the FTP server.
User name	The user account name that will log onto the server.
Password	The user account password.
Directory path	The path to the folder on the FTP server where the file will be posted; for example, pub/sc/inchannel.
Mask	The mask that determines the file types that will be consumed or posted by Service Connect. To allow only CSV files, for example, set this field to *.csv.
Binary	The method used to transfer the files, either binary or ASCII.
RFC	A switch between transfer modes. When set to Yes, communication is based on a custom implementation of RFC 959 to support compatibility with AS400. The default value of No uses communication on WINAPI interfaces.
Conversion	The conversion plug-in used when consuming or posting the file.
ConversionCfg	Enter one or more configuration parameters. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

Configure HTTP Channel Options

An HTTP communication channel can be used as a speaker that sends a message to the HTTP protocol using the PUT or POST method. The PUT method is used when sending a message to a SharePoint document library. The POST method is used for all other HTTP communications.



Property	Description

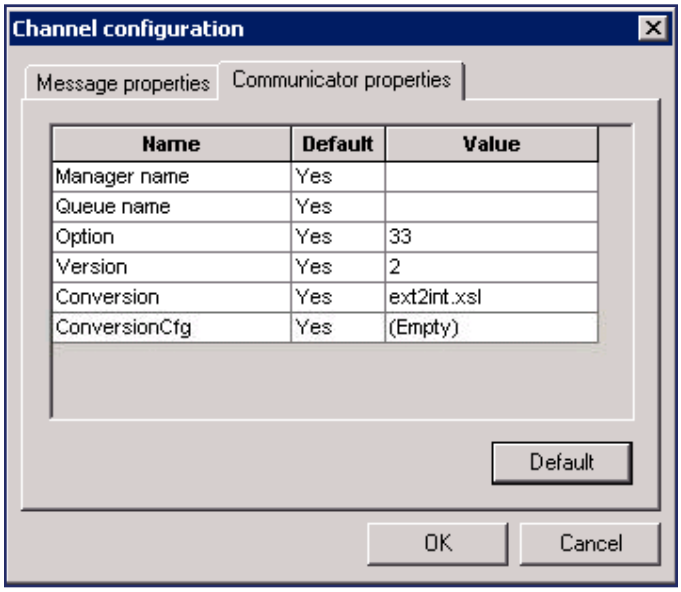
URL	The URL of the ASP page to which the file should post.
	
	The following is a sample ASP page.
	
	`<%@ Page language="c#" AutoEventWireup="true" %>`
	
	`<script runat="server">`
	
	`public void Page_Load()`
	
	`{`
	
	`Request.SaveAs(@"c:\UploadMessage\MessageTrace.xml", false);`
	
	`byte[] content = new byte[Request.InputStream.Length];`
	
	`Request.InputStream.Read(content, 0, (int)Request.InputStream.Length);`
	
	`Response.OutputStream.Write(content, 0, content.Length);`
	
	`}`
	
	`</script>`

Conversion	The conversion plug-in used when posting a file.
ConversionCfg	Enter one or more configuration parameters. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

Configure IBM MQ and IBM MQ (.NET) Channel Options

An IBM Message Queue (MQ) communication channel can be both a listener and a speaker. As a listener, it is used to consume messages placed in a named IBM WebSphere Message Queue. As a speaker, it is used to send a message as a file to a named queue where it can be used by another application or even another workflow.

IBM MQ channel is used for 32-bit operating systems. IBM MQ (.NET) channel is used for 64-bit operating systems.



IBM MQ channel has the following properties:

Property	Description
Manager name	The name of the queue manager.
Queue name	The name of the queue.
Option	This value is transmitted to the MQOPEN function of WebSphere MQ API. For more information, read the Option parameter description for the MQOPEN function in the IBM WebSphere MQ Series documentation. Epicor strongly recommends you keep the default value.
Version	Version of incoming or outgoing messages, which can be set either to 1 or 2. Message splitting is not supported for version 1. To allow processing of large messages, set a larger message size in the IBM WebSphere MQ manager or in the output queue, or use protocol version 2.
Conversion	The conversion plug-in used when consuming or posting the file.
ConversionCfg	Enter one or more configuration parameters. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

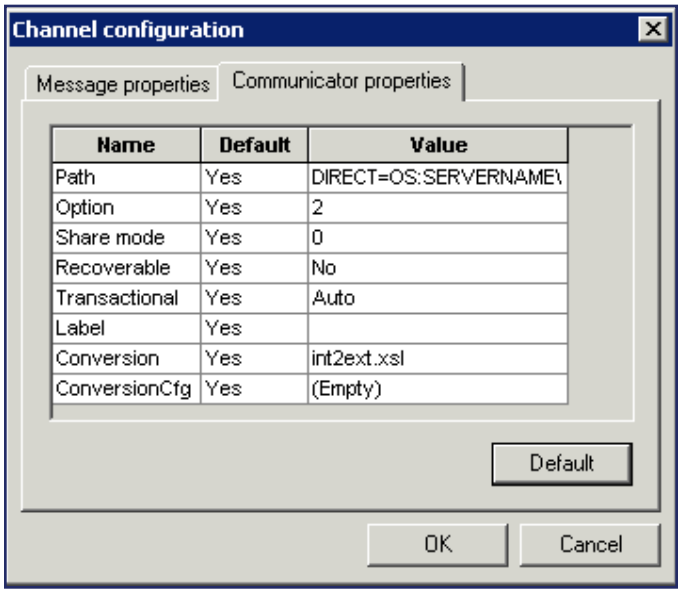
IBM MQ (.NET) channel has the following properties:

Property	Description
Server name	The name of the WebSphere MQ server.
Server Port	The port number of the queue manager's listener.
Channel Name	The name of the server-connection channel on the queue manager.
Manager name	The name of the queue manager.
Queue name	The name of the queue.
Option	The value transmitted to the MQOPEN function of WebSphere MQ API. For more information, review the Option parameter description for the MQOPEN function in the IBM WebSphere MQ Series documentation. Epicor strongly recommends you keep the default value.
Version	Version of incoming or outgoing messages, which can be set either to 1 or 2. Message splitting is not supported for version 1. To allow processing of large messages, set a larger message size in the IBM WebSphere MQ manager or in the output queue, or use protocol version 2.
Conversion	The conversion plug-in used when consuming or posting the file.
ConversionCfg	Enter one or more configuration parameters. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

If WebSphere MQ and Epicor Service Connect are installed on the same computer, it is not necessary to enter Server name and Channel name properties.

Configure MSMQ Channel Options

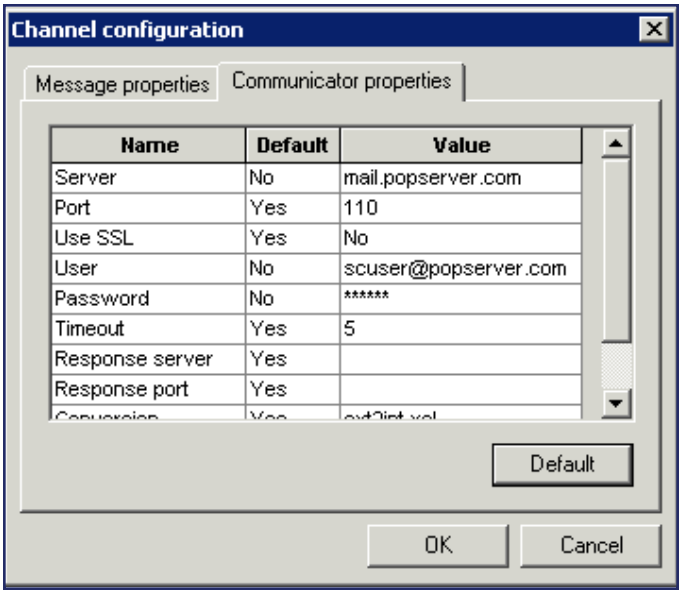
A Microsoft Message Queue (MSMQ) communication channel can be both a listener and a speaker. Use it as a listener to consume messages placed in a named queue. Use it as a speaker to send a message as a file to a named queue where it can be used by another application or even another workflow.



Property	Description
Path	The path to the message queue.
Option	One of the following: • **32** – You can view messages but cannot remove them from the queue. • **2** – You can send messages to the queue. • **1** – You can view and remove messages from the queue. Epicor strongly recommends you keep the default value.
Share mode	One of the following: • **0** – The queue is available to all users. • **1** – This limits the number of users available to receive messages from the queue to the current workflow. Once a workflow opens a queue with this share mode, no one, including the workflow that opened the queue, can open it again to read messages, which includes attempting to open the queue with multiple threads within the same workflow, until the original caller closes the queue.
Recoverable	By default, MSMQ stores all messages in RAM memory during routing and delivery, providing fast processing. This leads to the risk of possibly losing the messages if the machine is rebooted or if the MSMQ service is restarted. If you set Recoverable to Yes, all messages will be written to disk.
Transactional	Sending or retrieving messages can be performed within the context of a transaction, known as transactional messaging. The available transactional messaging options include Auto, Yes, and No.
Label	Label property of an MSMQ message and can be used as a message subject.
Conversion	The conversion plug-in used when consuming or posting a file.
ConversionCfg	Enter one or more configuration parameters. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

Configure POP3 Channel Options

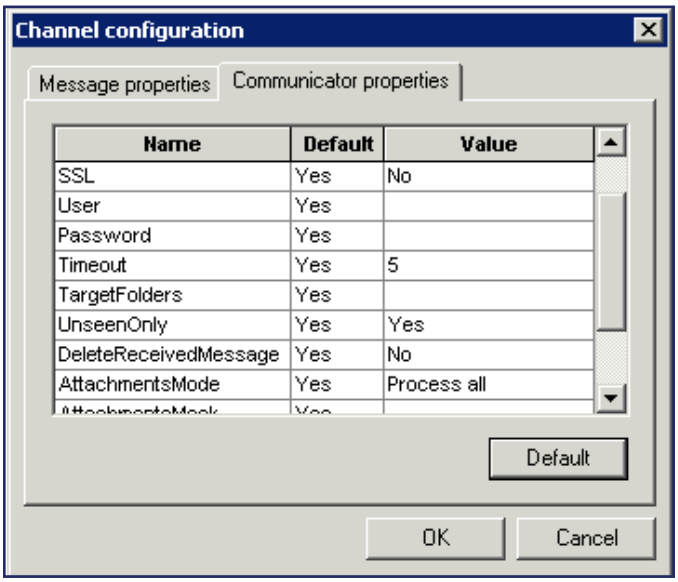
A POP3 communication channel is a listener that can be used to monitor an e-mail inbox at a POP3 server. When an e-mail with an attachment arrives in the POP3 inbox, Service Connect applies the selected conversion to the attachment and routes the attachment to a workflow.



Property	Description
Server	The POP3 server name. If the server is within a trusted domain, you can enter just the server name. If the server is outside a trusted domain, enter the Fully Qualified Domain Name (FQDN). For example, if mycompany.pop3server.net is within a trusted domain, you can enter pop3server.
Port	The server port number.
Use SSL	Select whether to use the Secure Sockets Layer (SSL) protocol to manage message transmission security on the Internet.
User	The user name for an account used to access the server. You will enter a user name and password when the POP3 server is password secured or if it is outside a trusted domain.
Password	The user account password.
Timeout	The amount of time before the server connection is closed due to inactivity. The receipt of any command during the timeout interval resets the timer.
Response server	An SMTP server that can be used for message delivery confirmation. Whenever an input channel receives a message from the POP3 server, it will send a message to this SMTP server.
Response port	The response server port number.
Conversion	The port of the SMTP server.
ConversionCfg	The conversion plug-in used when consuming or posting the file. You can enter one or more configuration parameters in this field. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

Configure IMAP Channel Options

All the settings for POP3 communicator are valid for IMAP communicator. Following are specific IMAP settings.

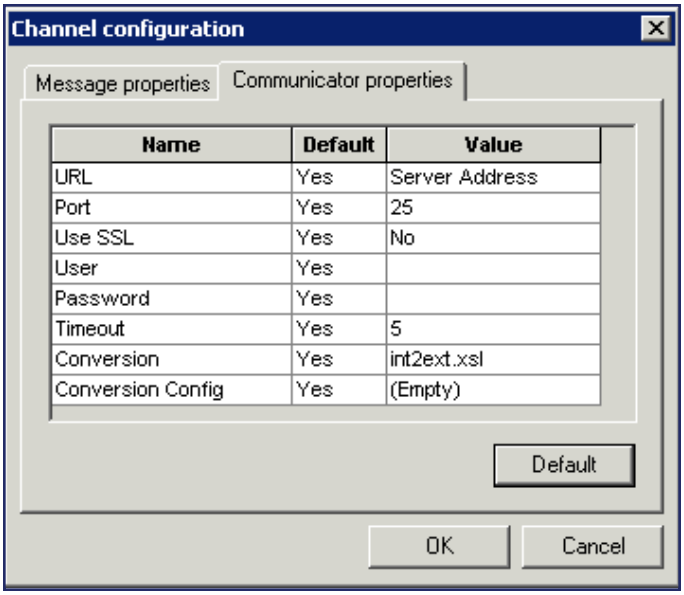


Property	Description
TargetFolders	A list of IMAP folders from which messages should be read. You can delimit folders with ',' or ';'. If no folders are specified, messages are read from Inbox.
UnseenOnly	When set to Yes, only unseen messages are received. When set to No, all messages are received.
DeleteReceivedMessages	When set to No, received messages are not deleted from the server. When set to Yes, received messages are deleted from the server.

To avoid multiple receipt of the same messages, use DeleteRecievedMessages set to No with UnseenOnly set to Yes.

Configure SMTP Channel Options

An SMTP communication channel is a speaker that can send an e-mail message from an SMTP server. A Poster can be used to create an e-mail template. The e-mail template can contain default text plus information from the document nodes in the workflow. Review Chapter 4: Workflow Designer for an example of how to use a Poster activity to send an e-mail using an SMTP channel.

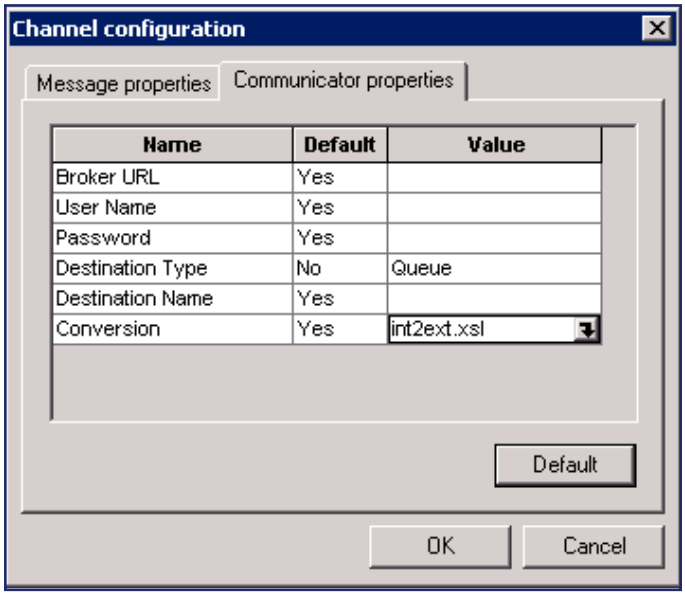


Property	Description
URL	The URL of the SMTP server. If the server is within a trusted domain, you can enter just the server name. If the server is outside a trusted domain, enter the Fully Qualified Domain Name (FQDN). For example, if mycompany.smtpserver.net is within a trusted domain, you can enter smtpserver.
Port	The server port number, usually port 25. The default settings for some virus protection programs block port 25 to prevent mass mailing worms from sending mail. If a workflow does not send an e-mail as designed, check your virus protection program's port blocking rules.
Use SSL	Select whether to use the Secure Sockets Layer (SSL) protocol to manage message transmission security on the Internet.
User	The user name for an account used to access the server. You will enter a user name and password when the SMTP server is password secured or if it is outside a trusted domain.
Password	The user account password.
Timeout	The amount of time before the server connection is closed due to inactivity. The receipt of any command during the timeout interval resets the timer.
Conversion	The conversion plug-in used when consuming or posting the file.
ConversionCfg	Enter one or more configuration parameters. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

Configure SonicMQ Channel Options

A SonicMQ communication channel can be used as both a listener and a speaker. Use it to receive and send messages over a Java Messaging Server that supports asynchronous messaging, guaranteed message delivery, security, transaction capabilities, and XML messages.

With SonicMQ, an application creates a connection and then establishes one or more sessions on that connection. The message sender is referred to as the Producer. The Producer sends a message to a Destination. The message receivers are referred to as Subscribers. Subscribers receive messages from Destinations.



Property	Description
Broker URL	The URL and port number of the server where JMS SonicMQ is installed. Possible connections include: • TCP (tcp://localhost:2506) • SSL (ssl://localhost:2506 –u UserName –p Password) • HTTP • HTTPS Service Connect uses a TCP connection by default.
User name	The user name for an account used to access the server. You will enter a user name and password when the server is password-secured.
Password	The user account password.
Destination Type	One of the following: • **Queue** (point-to-point) - One-to-one communication. A producer sends a message to the queue. A single consumer receives the message, no matter how many consumers are listening to the queue. • **Topic** (publish/subscribe) - One-to-many communication. A producer publishes a message to a topic. All consumers who subscribe to the topic receive the message.
Destination Name	The name of the destination on the Java Messaging Server.
Conversion	The conversion plug-in used when consuming or posting the file.
ConversionCfg	Enter one or more configuration parameters. The configuration parameters you can use are based on the type of conversion plug-in. Read the Conversion Plug-In Configuration Options section later in this chapter for more information.

Conversions

A conversion is an operation on business, and sometimes system, data, which transforms full or partial sets of data from a specific source format to a specific target format of the document. The transformation can change the document structure and perform simple data manipulation, such as string and number formatting, basic arithmetic operations, and some other functions. In other words, the data in both the source and the target document is basically the same; only the format differs. That is why conversion is not a business operation. To specify a business operation, use a registered web-service method workflow or call external components via the Requester element in workflows.

The two scenarios where conversions take place include:

- **Conversions between internal XML documents** - Read Chapter 4: Workflow Designer for more details on how to use conversions in workflows and the XML Mapper tool.
- **Conversions between internal and external document types, which happen in channels** - A communication channel can accept any binary or text data, which are not necessary XML documents, or it may have a structure unknown to web services or workflows. To transform incoming data to one of the internal documents, you may need to apply a conversion to the communication channel before the incoming document is passed to the workflow.

The items under the Conversions node - Connectivity, Communication Setup, server name, and Conversions - represent the conversion plug-ins that can be used by input and output channels. You can add your own plug-ins if the standard plug-ins that come with Service Connect do not meet your needs.

Set up Conversion Plug-In Configuration Options

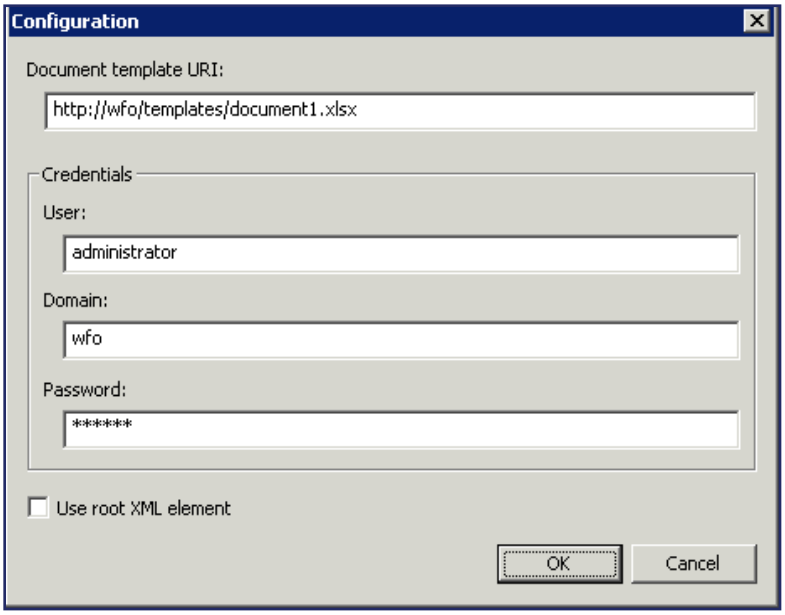
Conversion plug-ins that use the .NET interface support configuration parameters. Each parameter, or option, is an XML fragment that controls how the plug-in functions.

The various .NET plug-ins, and the configuration options each plug-in supports, are:

- **csv2xml.dll** - This supports DefaultNamespace, Separator, FirstRowContainsFieldNames, TextQualifier, and AlwaysAddText Qualifier.
- **CsvAppendConverter.dll** - This supports the Separator, FirstRowContainsFieldNames, AllowMultipleTables, TextQualifier, AlwaysAddTextQualifier, OutFileEncoding and OutFilePath settings.
- **excel2xml.dll** - This supports DefaultNamespace.
- **anyxml2xml.dll** - This supports DefaultNamespace.
- **Office2007Converter.dll -** This supports TemplateDocument.
- **CryptoConverter.dll** - This supports signing and encryption algorithms.
- **fixedwidth2xml.dll** - This supports incoming file parameters (such as Field width, Field name, and CheckLength) and output XML message parameters (such as DefaultNamespace, DontTrimValues, DontTrimNames, Table, and Row).

The table below describes each configuration option. Each configuration option must be written as a well-formed XML fragment as shown in the examples.

Configuration Option	Description
DefaultNamespace	The XML namespace used for all the data nodes the plug-in generates. If this option is absent, Service Connect creates a namespace using the http://Epicor.com/SC/UserSchema/Csv URI and a hash value based on the column names in the incoming document. This option affects only input channels. **Example:** <DefaultNamespace>http://myorgnamespace</DefaultNamespace>
Separator	The field delimiter for CSV files. When present, the value should be a single character or empty. An empty separator indicates the entire string should be considered a single column. If this option is absent, a comma is used as the separator. This option affects both input and output channels. **Example:** <Separator>,</Separator>
FirstRowContainsFieldNames	Indicates whether the first line in the input file should be interpreted as column headings. You can set the value to true or false. When absent, the system uses true. This option affects both input and output channels. **Example:** <FirstRowContainsFieldNames>false</FirstRowContainsFieldNames>
TextQualifier	Indicates which character is used as a text qualifier. All text between text qualifiers is considered a single column, even if the text contains a delimiter character. When present, the value should be a single character. When absent, the system uses a double quote. This setting affects both input and output channels. **Example:** <TextQualifier>"</TextQualifier>
AlwaysAddTextQualifier	This node may be used to disable quote marks around all the data. **Example:** <AlwaysAddTextQualifier>"</AlwaysAddTextQualifier>
GroupByFields	Defines which fields are used to group nodes. **Example:** <GroupByFields>Field1,Field3</GroupByFields> **Example:** <GroupByFields>1,3</GroupByFields> **Example:** <GroupByFields>Field1,3</GroupByFields> The GroupByFields is parsed the same way as any CSV line – settings like Separator and Text Qualifier are applicable. Fox example, if you set Separator to ; : <Separator>;</Separator> You must then use the following format for the GroupByFields: <GroupByFields>2;3;25;26;27;28;29</GroupByFields> If you have FirstRowContainsFieldNames off, you can specify only field indexes. If you have FirstRowContainsFieldNames on, you can specify field names or indexes. Field names to group by must be valid.
GroupByCount	Limits the number of nodes in the group. **Example:** <GroupByCount>30</GroupByCount> limits the number of nodes in the group to 30.
OutFileEncoding	The encoding of the output file. The following values are accepted: 'UTF8' 'UTF16' 'ASCII' 'Unicode'
OutFilePath	Output file with full path.

Configuration Option	Description
TemplateDocument	The path and filename of a Microsoft Office 2007 document that can be used as a template for a SharePoint document library. This example shows a template stored in a SharePoint document library on another server. 
Signing and Encryption algorithms	You can set Service Connect to use a digital signature or encryption to protect Service Connect documents sent electronically. These digital methods can be applied to the XML messages that pass through the Input/Output Channels of the Messenger service. Service Connect can digitally sign xml documents and read digital signatures in incoming messages using the W3C xml signature standard. Cryptographic tasks are performed when messages arrive to a channel or leave a channel. There are completed messages in the external envelope. Operations to sign and encrypt messages are performed before conversion of an input message and directly after conversion of an output message. Conversions are performed during ext2int.xstl or int2ext.xslt transformation. So, all transformation changes affect CryptoConverter plug-in workability.
Field width and name	Field width is a mandatory parameter of an incoming file. Field name is an optional parameter of an incoming file. **Example:** <Fields> <Field width =”number” name = “fieldname” /> </Fields>
CheckLength	This flag enables line length checking; it is an optional incoming file parameter. **Example:** <CheckLength/>
DontTrimValues	This flag disables the trimming of ending spaces in table values. **Example:** <DontTrimValues/>
DontTrimNames	This flag disables the trimming of ending spaces in field names. **Example:** <DontTrimNames/>
Table and Row	These optional parameters set custom names for tables and lines. **Example:** <Table>table_title</Table> <Row>row_title</Row>

Manage Custom Conversions

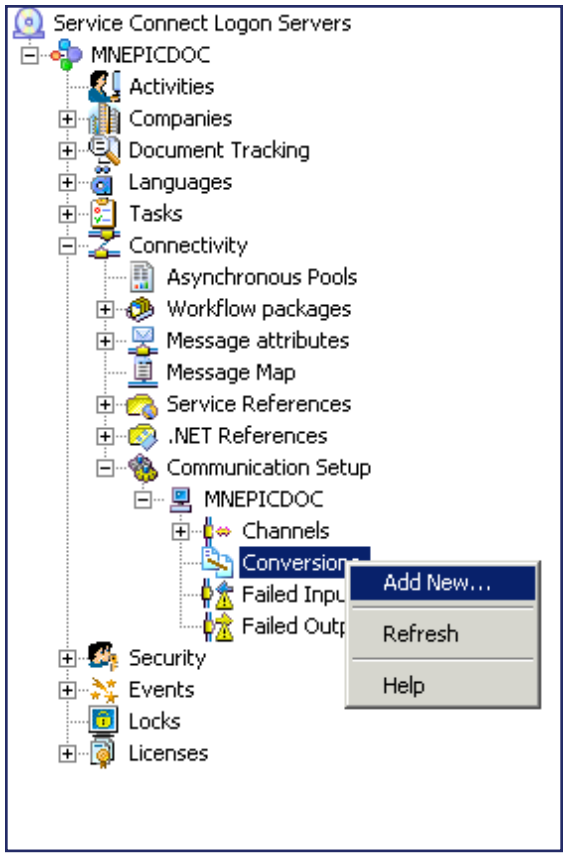
In ESC Administration Console, you can register a new conversion.

To create a new conversion, right-click Conversions and select Add New.

There are three conversion types available for creation: **XSLT Transformation**, **COM Object** or **.NET Class**. You can assign the customer conversion to an input channel, output channel, or both. Once created, it is possible to edit a custom conversion.

A programming sample for Conversion Plugin source code is available at <Epicor Service Connect Client installation Folder>\Tutorials\SCSSamples\CustomCsv2Xml>.

For more information on how to create a custom conversion, review the Epicor Service Connect Help.



Failed Input

If an input communication channel encounters a message it cannot handle, the message is stored in the Failed Input node. Use the document tracking feature to trace workflow execution. Related document tracking activity is marked as an error and the message is moved to Failed Input. Examine the Event log and Failed Input to analyze the issue.

In the Activity Progress window, check if any errors point to an input channel and double-click the error to view the details. A message is moved to Failed Input only if an error was caused by the message itself; for example, if the Conversion Layer transformation failed or the message could not be parsed. Once a message is moved into Failed Input, it stays there until you manually remove it or return it to the normal queue. The error information is also logged in the list of events.

Use the Failed Input node to:

- View all the Failed Input messages.
- Filter Failed Input entries by channel and date.
- Delete a message from Failed Input.
- Export a message into the file or move it back to the channel queue.

Failed Output

If an output communication channel encounters a message it cannot handle, after a certain number of attempts, the message is stored in the Failed Output node. Use the document tracking feature to trace workflow execution. Related document tracking activity is marked as an error and the message is moved to Failed Output.

In the Activity Progress window, check if any errors point to an output channel and double-click the error to view the details. Once a message is moved into Failed Output, it stays there until you manually remove it or return it to the normal queue.

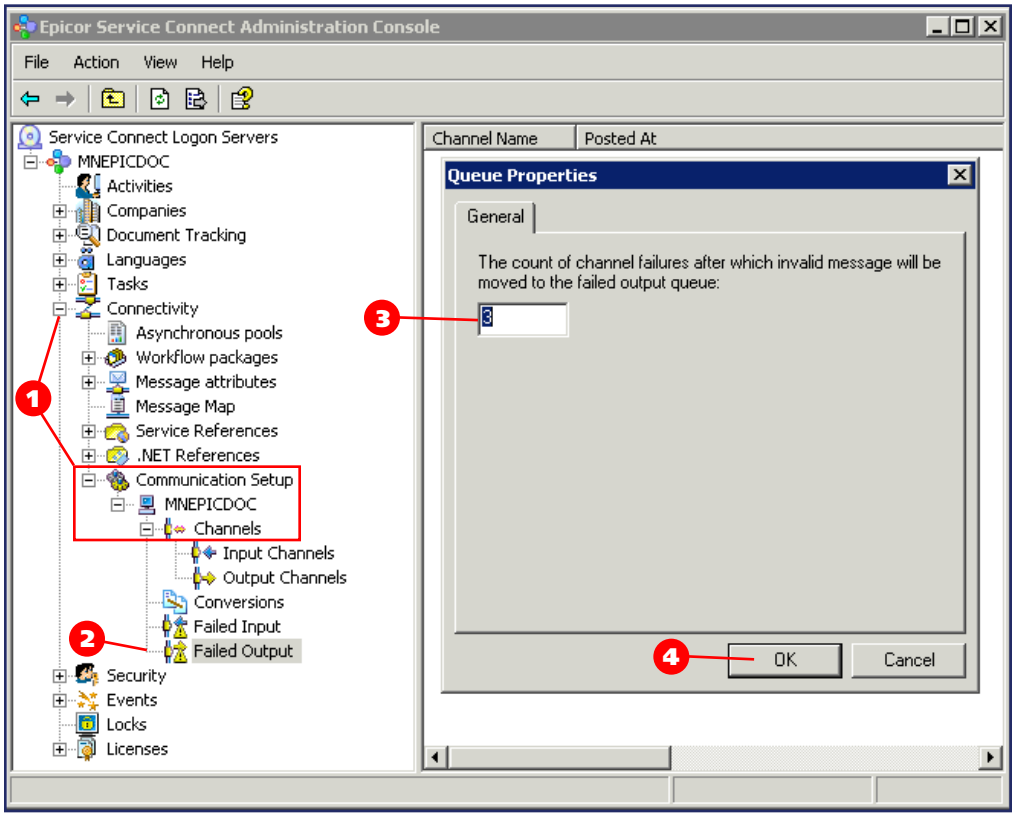
To resolve the issue, change the channel or workflow settings, delete the Poster trace in the document tracking, resume activity, or re-send a message from Failed Output. When a communication channel is deleted, all corresponding Failed Output entries are deleted automatically. A corresponding warning message displays, informing you of the Failed Output location to give you an opportunity to preserve the messages. The error information is also logged in the list of events.

Use the Failed Output node to:

- View all the Failed Output messages.
- Filter Failed Output entries by channel and date.
- View message details.
- Delete a message from Failed Output.
- Export a message into the file or move it back to the channel queue.

To set the number of message processing attempts for Failed Output:

1. In the **Tree View**, expand the **Connectivity > Communication Setup >** <machine name> **> Channels** node.
2. Right-click **Failed Output** and select **Properties**.
3. In the **Queue Properties** window, set the number of failures after which the invalid message will be moved to the failed output queue.
4. Click **OK**.



Web Methods to .NET Calls Converter

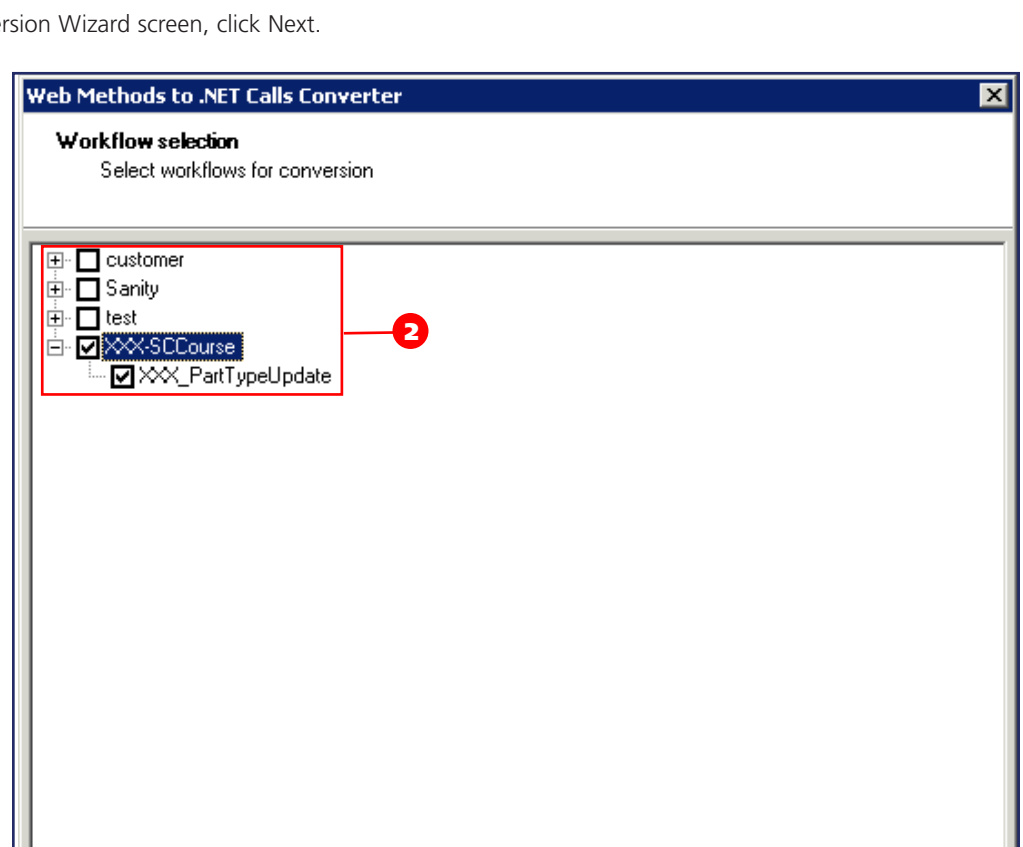
Use the Web2Net utility to convert Web Methods within workflows to .NET Calls. Web2Net Converter is a part of the standard Service Connect installation package.

Before you run the process on the Service Connect server, import the .NET references to replace the Web Methods. Ensure the .NET references have exactly the same names as the Web Methods.

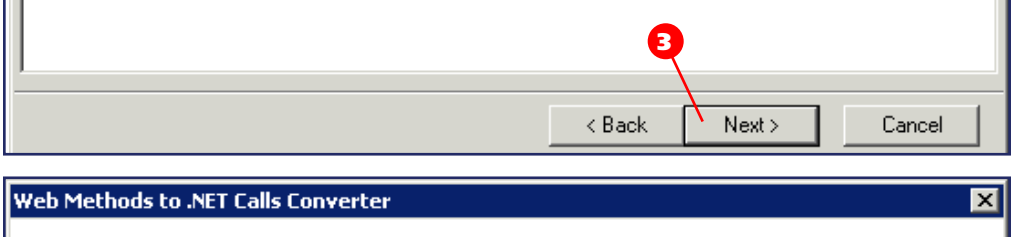
Start menu > All Programs > Epicor Software > Epicor Service Connect > Web2NETConverter

To convert a workflow:

1. On the Welcome to the Conversion Wizard screen, click Next.
2. On the Workflow selection screen, select a workflow package, or expand one of the workflow packages and select a workflow.
3. Click Next.



4. On the References mapping screen, in the **WebRef** column, all the Web Service References existing in ESC display. For each Web Method, a replacing .NET reference displays in the **NetRef** column.

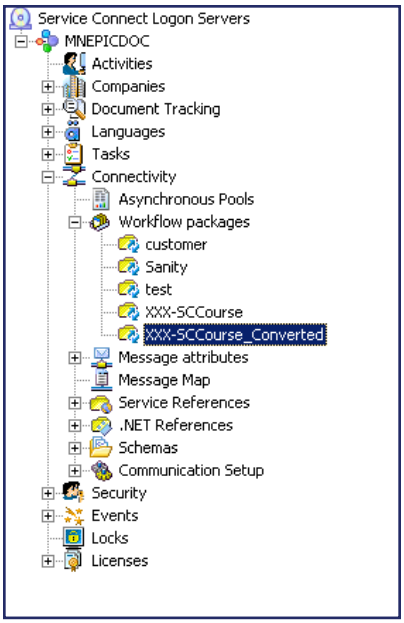


5. Verify the mapping is accurate. If the mapping is not found by the Converter, the first .NET Reference in the list is used.
6. Click **Next**.
7. To the confirmation message on the Conversion progress window, click **OK**.

9. On the **Conversion complete** window, view the conversion log.

10. Click **Finish**.

The Web2NET Converter creates a new workflow package named <SourceWorkflowPackage>_Converted. The resulting converted workflow is located within this workflow package and has the same name as the source workflow.



SharePoint Integration

Use the Service Connect SharePoint integration component to perform the following tasks:

- Extend Windows SharePoint Services workflows.
- Use a SharePoint document library as an input channel.
- Publish documents to SharePoint document libraries using an output channel.

Unlike other listeners, the SharePoint input channel does not always consume a document when it is added to a library. Instead, you have the option to leave the document in the library and initiate a workflow when you add or modify a document.

Refer to the Service Connect Installation Guide for system requirements and how to install the Service Connect SharePoint Integration component. After the integration is installed, you follow these steps to consume documents from a SharePoint document library and use them in a workflow:

1. Set up an input channel to monitor the SharePoint document library.
2. Create a Service Connect workflow.
3. Add a message map to direct documents the input channel receives to the workflow.
4. Define a SharePoint workflow to use the integration.
5. Optionally define an output channel to publish documents from the Service Connect workflow to a SharePoint document library.
6. If you perform step 5, use the output channel with a Poster activity in the Service Connect workflow.

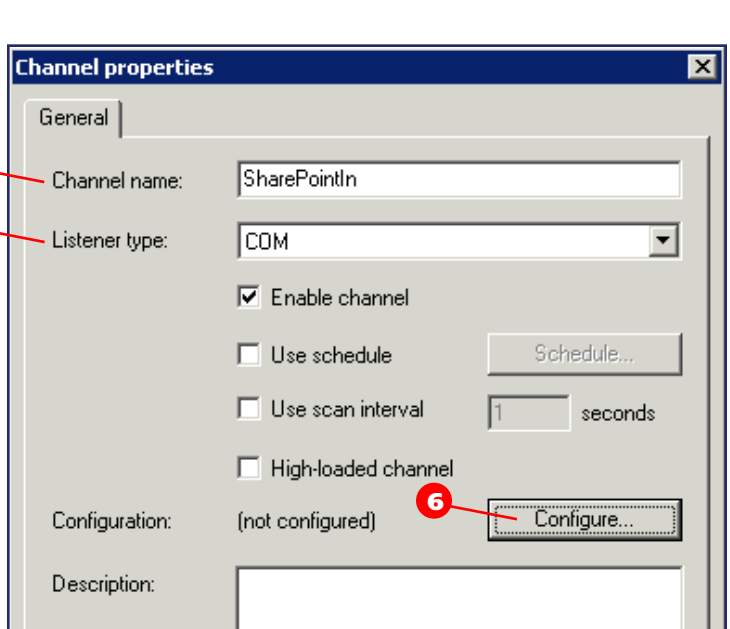
An example of how to perform each of these tasks is shown next.

Set Up the Input Channel

SharePoint integration uses a COM input channel. More information about COM channels is found earlier in this chapter.

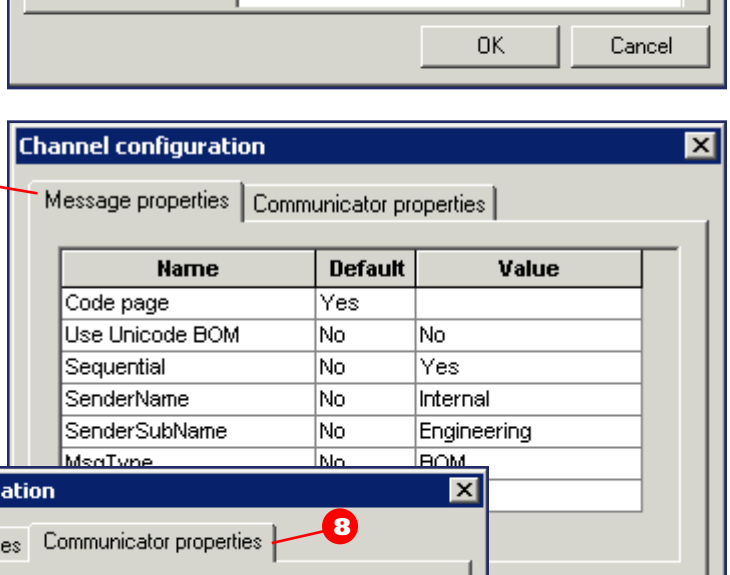
To add an input channel:

1. Log into the Epicor Service Connect Administration Console using the menu path **Start > All Programs > Epicor Service Connect > Service Connect Administration Console**.
2. In the **Tree View**, expand the **Connectivity > Communication Setup >** <machine name> **> Channels** node.
3. Right-click **Input Channels** and select **Add New**.
4. The **Channel properties** window displays. Enter a **Channel name**.
5. Click the **Listener type** drop-down list and select **COM**.
6. Click **Configure**.

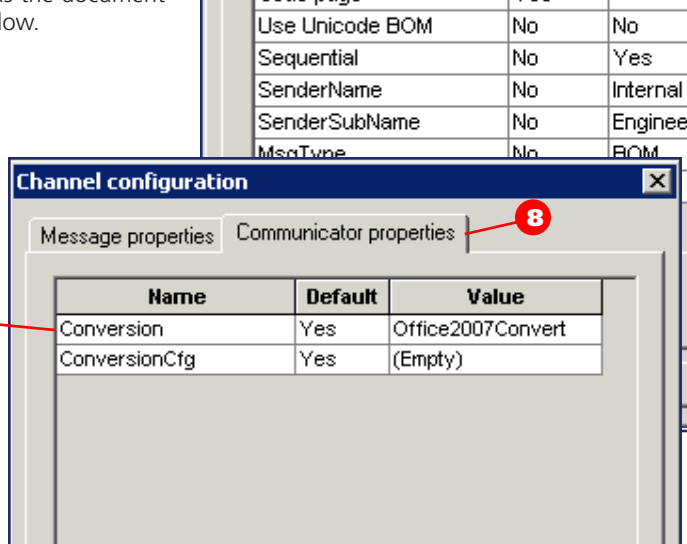


7. The **Channel configuration** window displays. On the **Message properties** tab, enter the appropriate message attributes.

   Message attributes are added to the document when it enters the input channel. Service Connect looks for a message map with the same attributes as the document to determine how to route it to a workflow.



8. Click the **Communicator properties** tab.
9. In the **Conversion** field, select the appropriate conversion.
10. Click **OK** until you exit all dialog boxes.

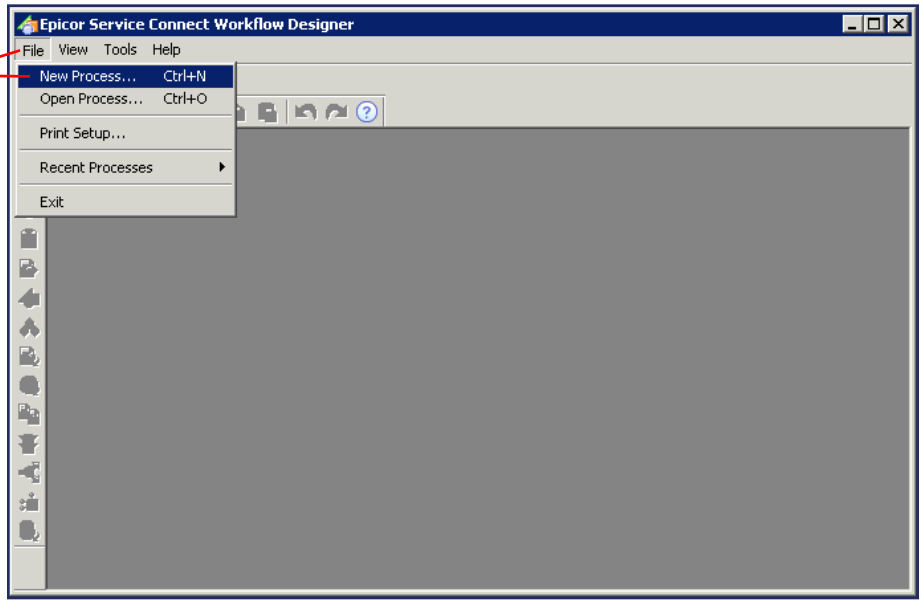


Create a Workflow

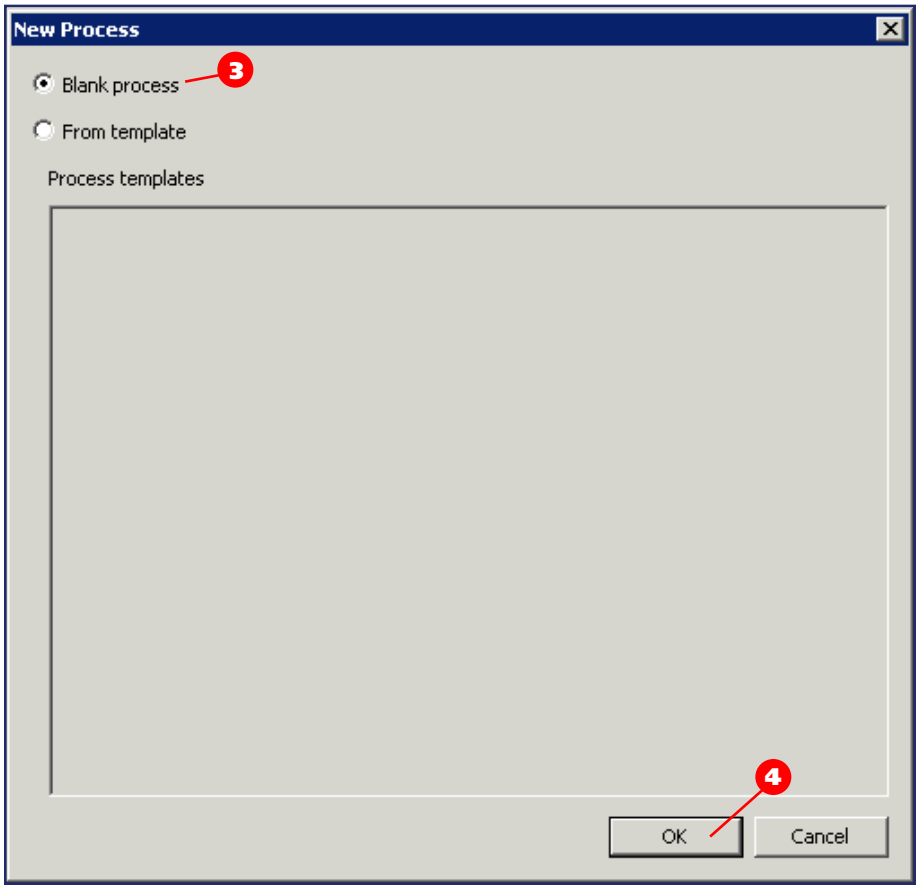
You must create a workflow that can receive incoming documents from the SharePoint document library.

To create a new workflow:

1. Log into the Workflow Designer using the menu path **Start > All Programs > Epicor Service Connect > Workflow Designer.**
2. From the **File** menu, select **New Process**.

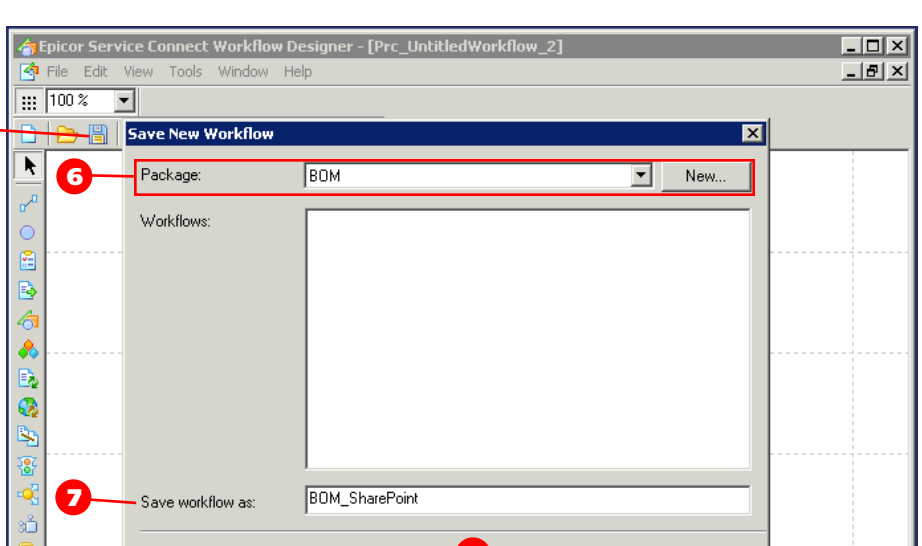


3. The **New Process** window displays. Leave the **Blank process** option selected.
4. Click **OK**.



5. Click **Save**.
6. Click the **Package** drop-down list to select a workflow package or click **New** to create a new one.
7. In the **Save workflow as** field, enter a workflow name.
8. Click **Save**.
9. Click **Yes** when prompted to save the schema.

You can add activities to the workflow as needed. Review Chapter 4: Workflow Designer for detailed descriptions of each workflow activity.



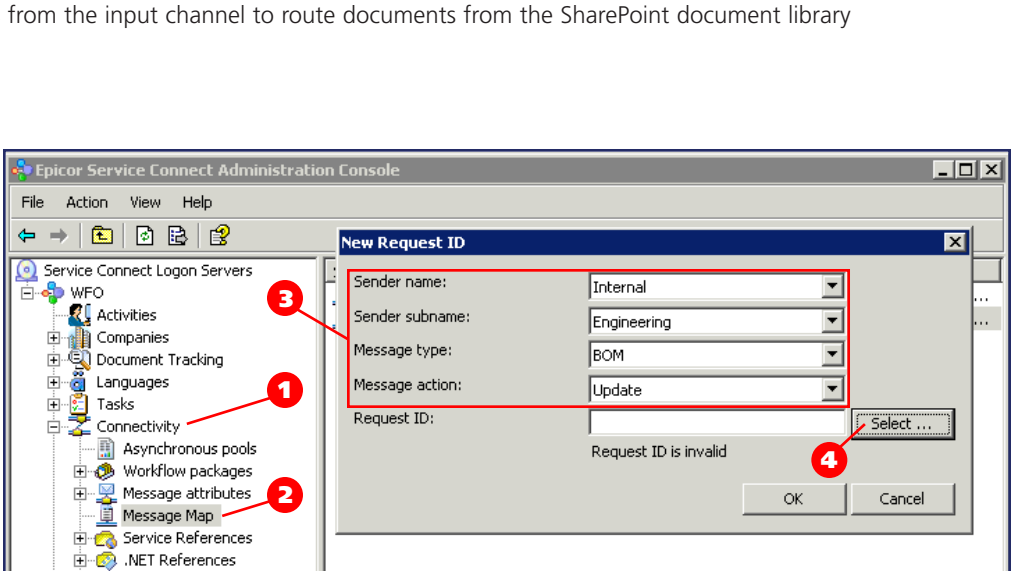
To use the incoming document in the workflow, you must create a schema. This procedure is explained in Chapter 4: Workflow Designer in the Schema Utility section.

Add a Message Map

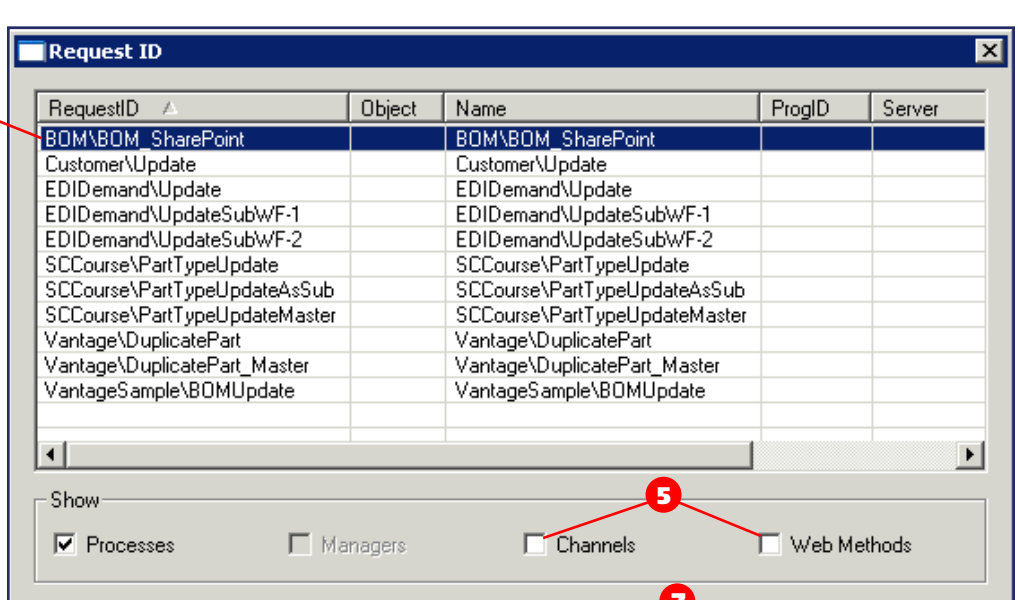
The message map uses the attributes from the input channel to route documents from the SharePoint document library to the workflow.

To add a new message map:

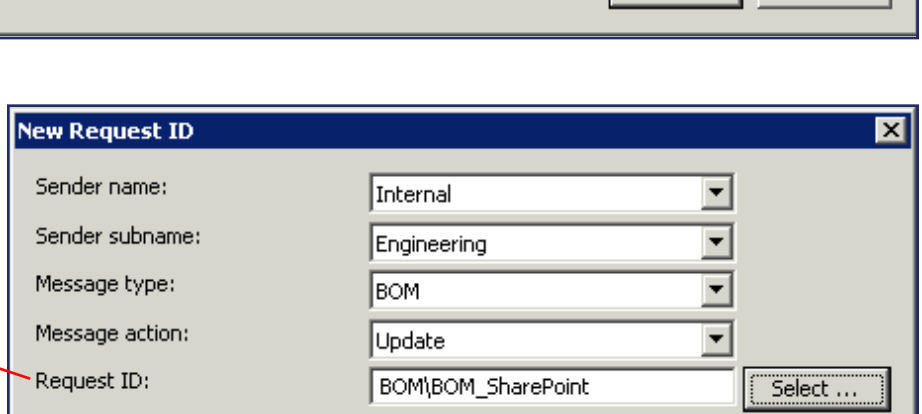
1. In the **Tree View** of the **Epicor Service Connect Administration Console**, expand the **Connectivity** node.
2. Right-click **Message Map** and select **Add new Request**.
3. The **New Request ID** window displays. Select the **Sender name**, **Sender subname**, **Message type**, and **Message action** information used for the input channel.
4. Click **Select**.



5. The **Request ID** window displays. Clear the **Channels** and **Web Methods** check boxes to hide these items.
6. In the **RequestID** column, select the row that contains the workflow you created.
7. Click **OK**.



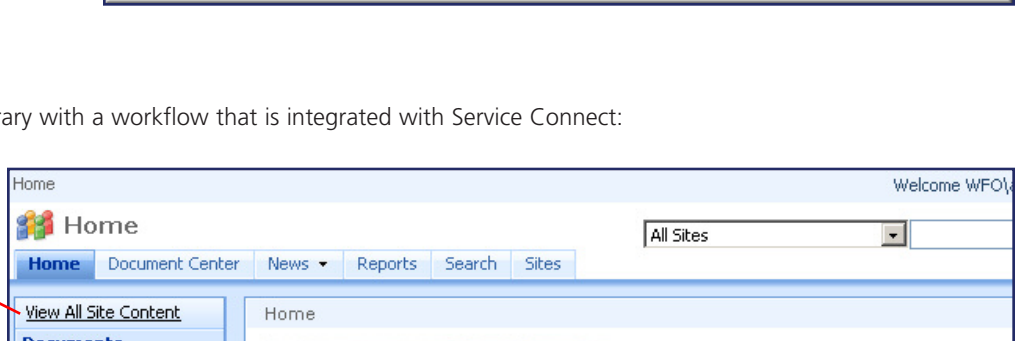
8. Verify the **Request ID** is valid.
9. Click **OK**.



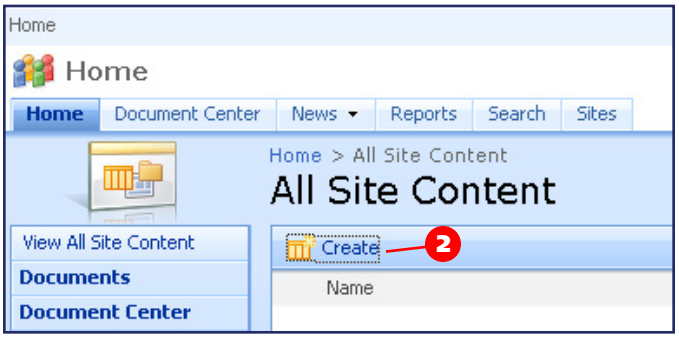
Define a SharePoint Workflow

To create a SharePoint document library with a workflow that is integrated with Service Connect:

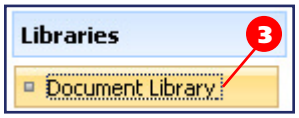
1. From your SharePoint site's home page, click **View All Site Content**.



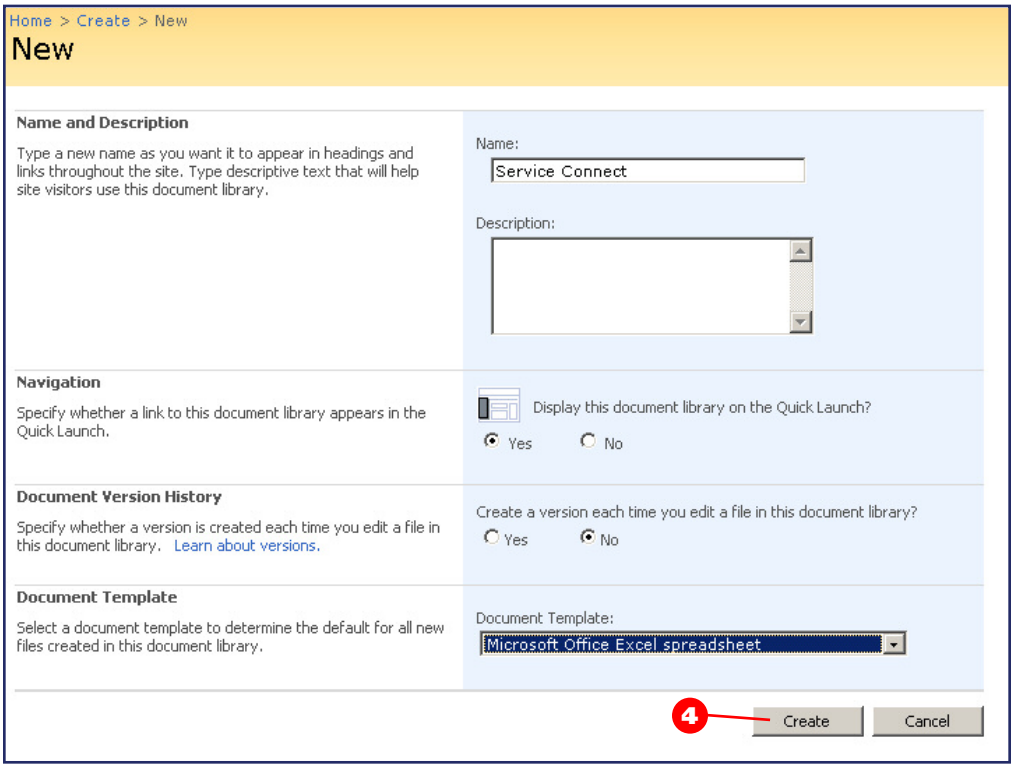
2. Click **Create**.



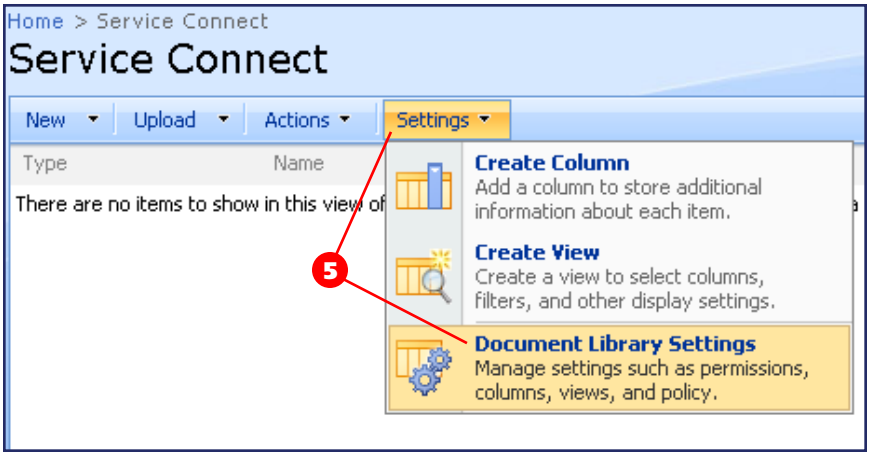
3. Click **Document Library**.



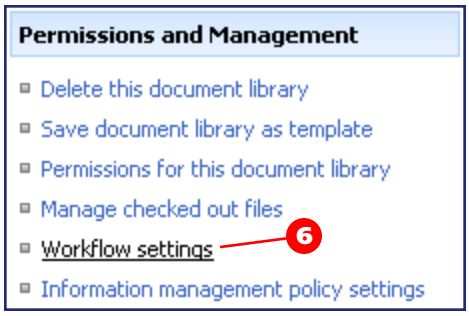
4. Enter and select the appropriate values and click **Create**.



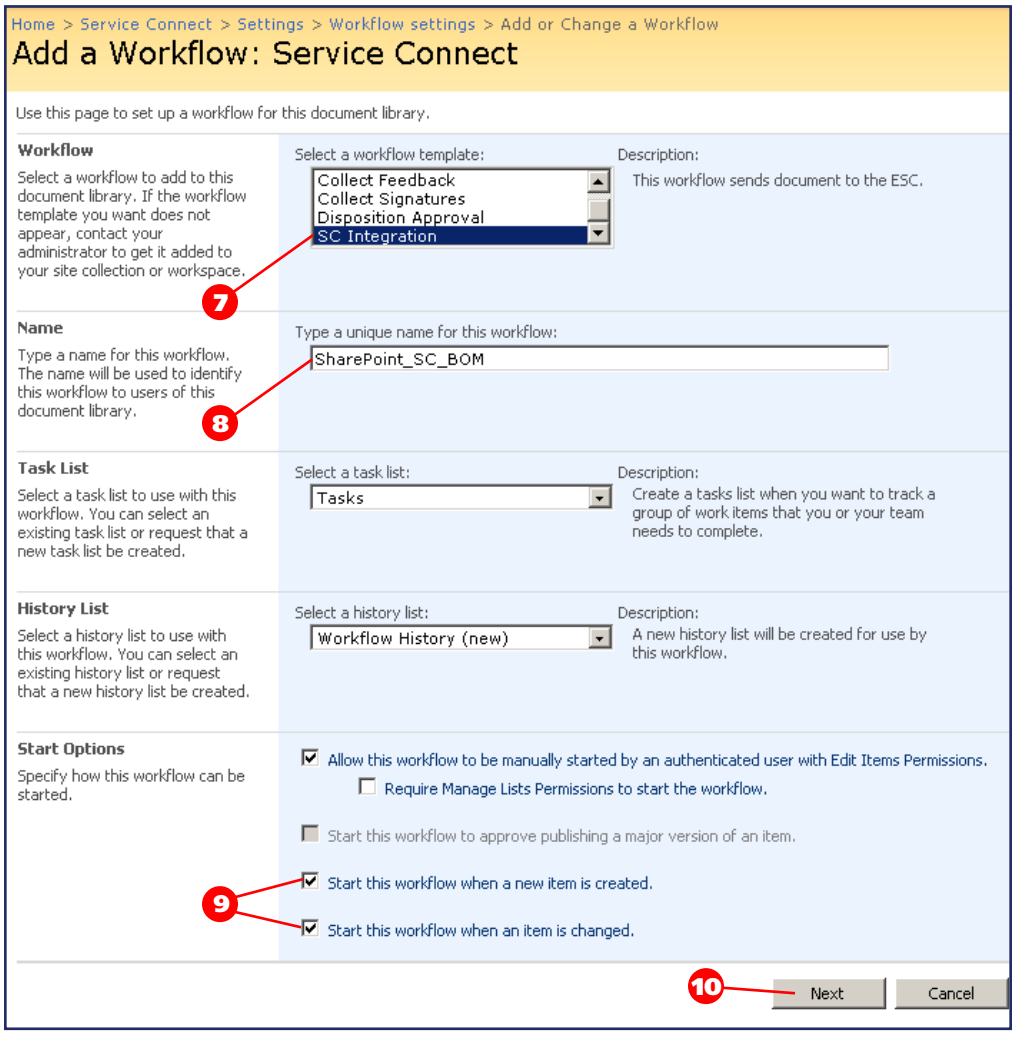
5. Click the **Settings** drop-down list and select **Document Library Settings**.



6. Under **Permissions and Management**, click **Workflow settings**.



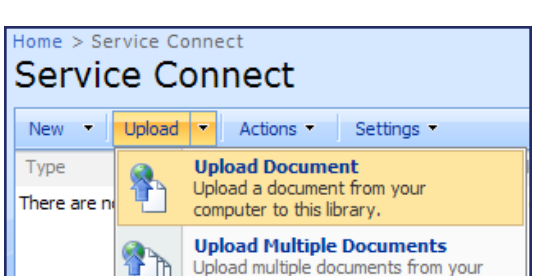
7. In the **Workflow** field, select **SC Integration**.
8. In the **Name** field, enter a name for the SharePoint workflow.
9. In the **Start Options** section, you can select the **Start this workflow when a new item is created** check box, the **Start this workflow when an item is changed** check box, or both.
10. Click **Next**.



11. Click the **ESC channels** drop-down list to select an input channel.
12. Select the **Delete document after sending** check box if you want to delete the document after it is sent to Service Connect.
13. Click **OK**.

If you receive an error that starts with "Guid should contain", refer to the Troubleshooting topic in the Service Connect application help to resolve the issue.

The SharePoint workflow and integration to Service Connect is now complete. When you upload a document to the SharePoint document library, it will be sent to a Service Connect workflow based on the message map set up in the Administration Console.

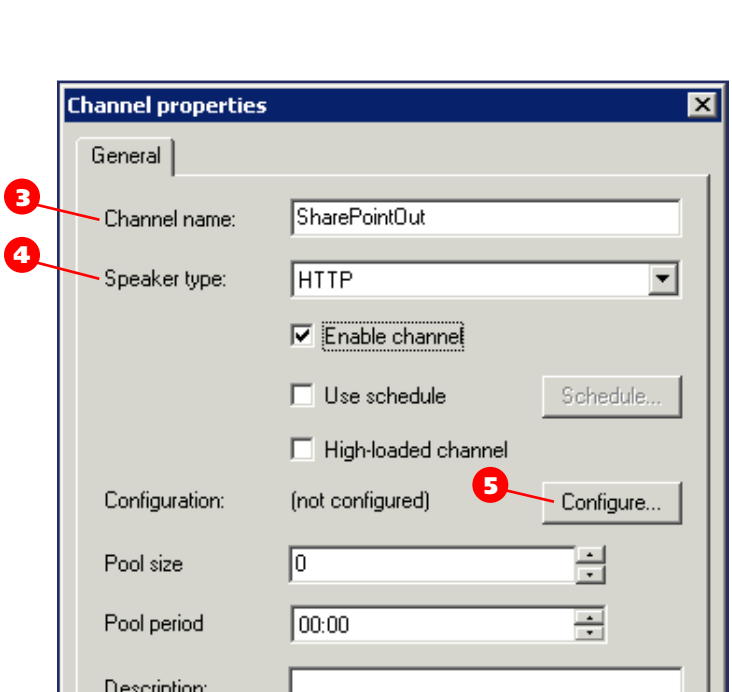


Add an Output Channel

Use an HTTP output channel to send documents to a SharePoint library from a Service Connect workflow.

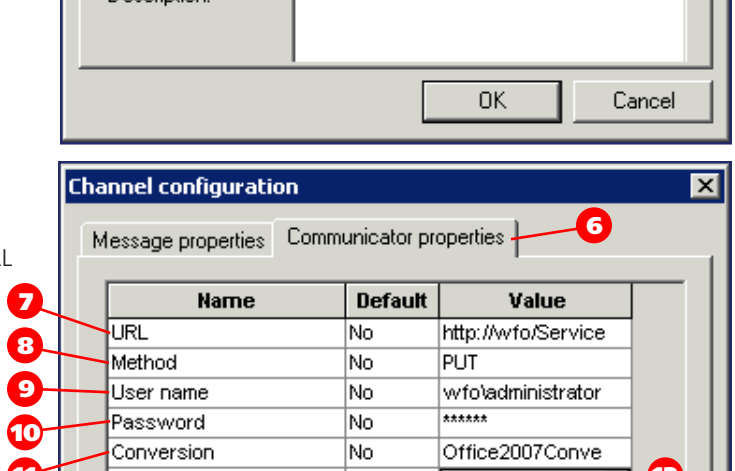
To add a new output channel:

1. In the **Tree View** of the **Epicor Service Connect Administration Console**, expand the **Connectivity > Communication Setup >** <Machine Name> **> Channels** node.
2. Right-click **Output Channels** and select **Add New**.
3. The **Channel properties** window displays. Enter a **Channel name**.
4. Click the **Speaker type** drop-down list and select **HTTP**.
5. Click **Configure**.



6. The **Channel configuration** window displays. Click the **Communicator properties** tab.
7. In the **URL** field, enter the SharePoint document library URL and the output filename.

   You can use filename templates and macros, as described earlier in this chapter, for the FILE channel type to define the filename. However, you cannot use the %GUID% macro because SharePoint document libraries cannot accept files with curly braces ({ }) in their names.
8. In the **Method** field, select **PUT**.
9. In the **User name** field, enter the domain and user name of a user account with at least Contribute privileges to the SharePoint document library.



10. In the **Password** field, enter the account password.

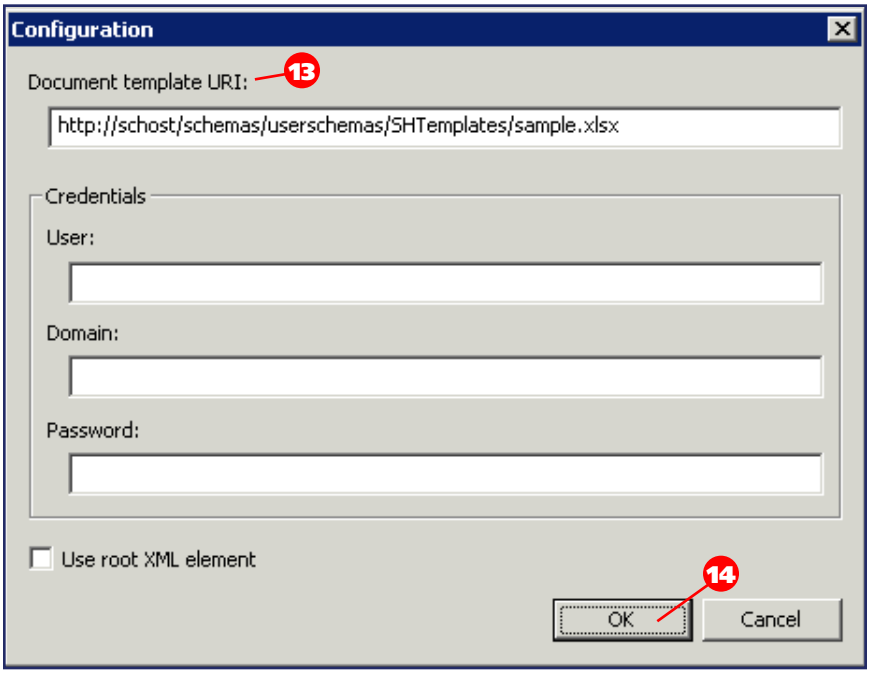
11. In the **Conversion** field, select the appropriate conversion for the document you want to send from Service Connect.

12. Click the button on the **Conversion config** field.

13. The **Configuration** window displays. In the **Document template URI** field, enter the path to a file that can serve as a template for the output channel.

    If the document is on another server, you can enter account credentials to access the file.

14. Click **OK** until you exit all dialog boxes.

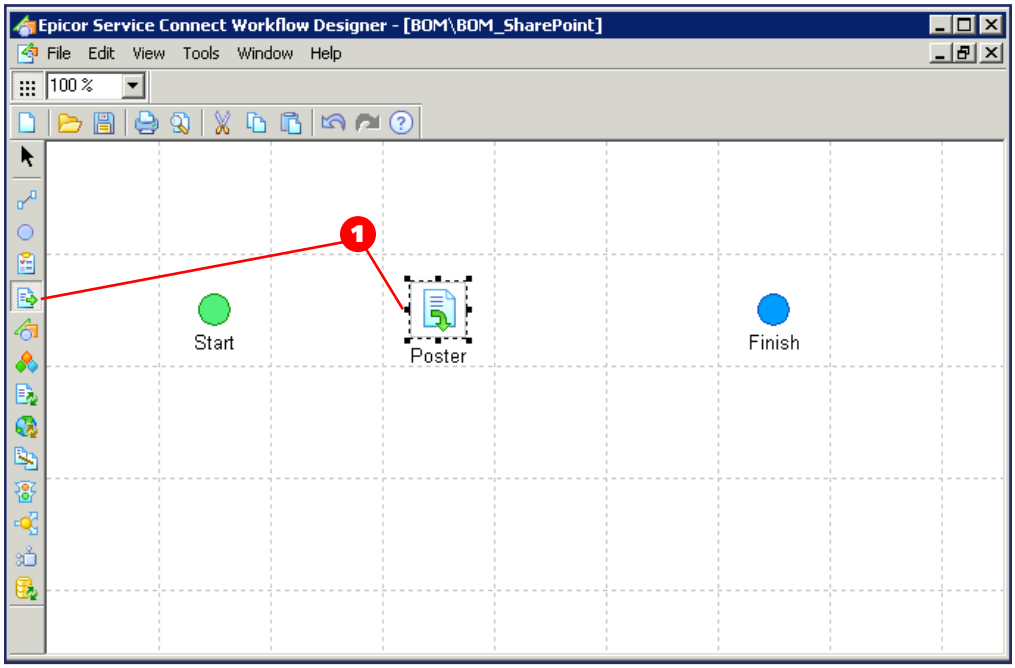


Use the Output Channel from a Poster

A Poster activity uses the output channel from within the workflow.

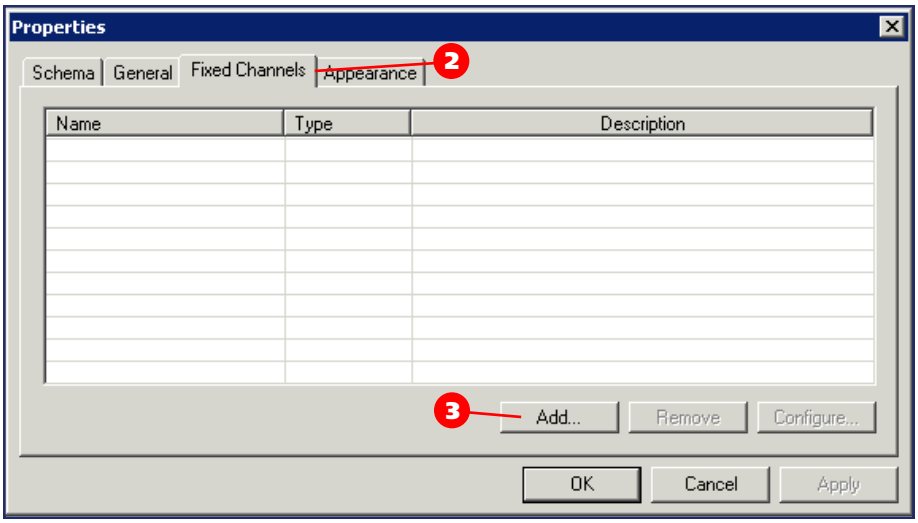
To use the output channel:

1. Click the **Poster** button on the Items toolbar. Click the design area to add a Poster activity.



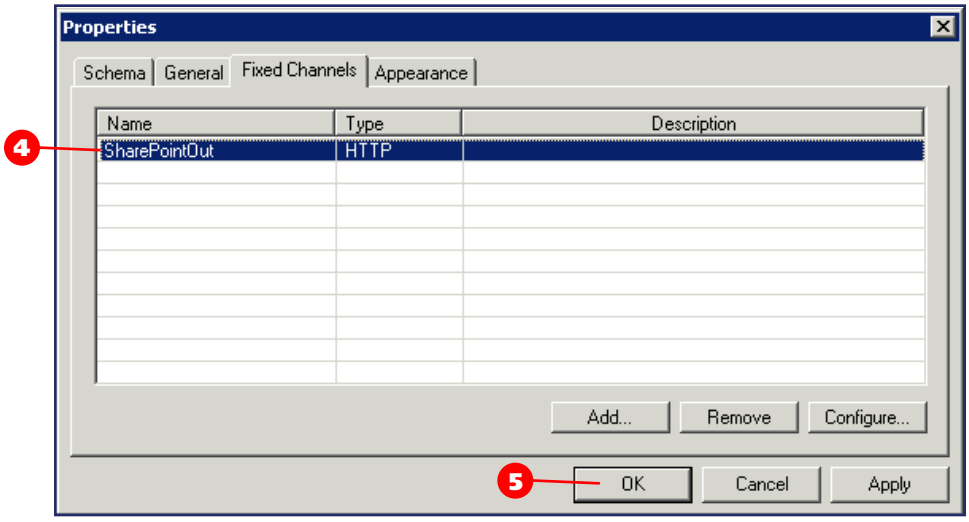
2. The **Properties** window displays. Click the **Fixed Channels** tab.

3. Click **Add**.



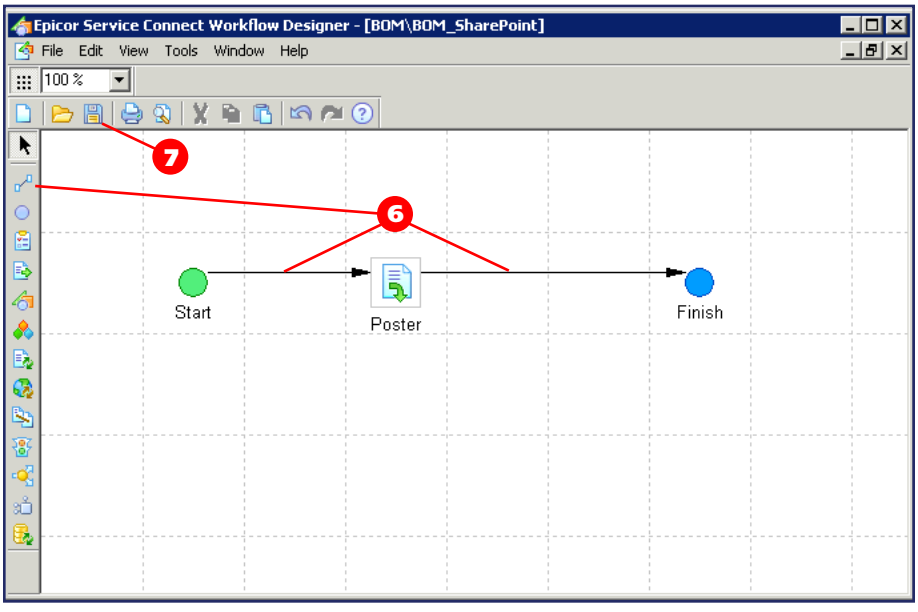
4. Select the output channel created for SharePoint.

5. Click **OK** until you exit all dialog boxes.



6. Click the **Connection** tool on the **Items** toolbar and connect the Start to the Poster and the Poster to the Finish.

7. Click **Save**.



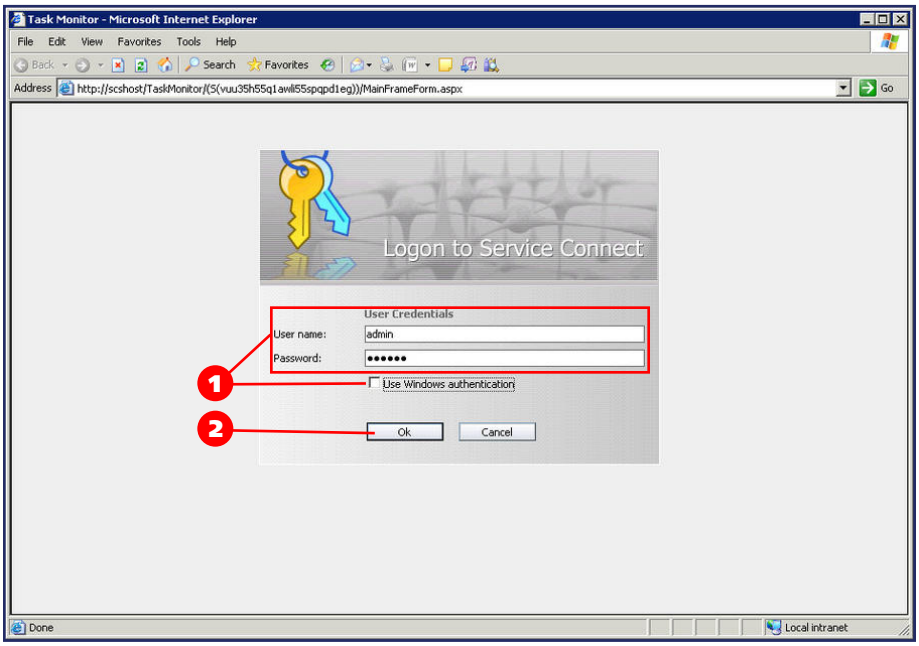
Tasks

Tasks are workflow activities that can halt a workflow under certain circumstances and then send data to the Task Monitor web site where an authorized user can take action and possibly choose how to route the information. Two typical uses of the Task activity are in authorization scenarios (for example, credit limit), or in error handling, (for example, continue processing or halt processing). Review Chapter 4: Workflow Designer for information about how to use the Task activity in a workflow.

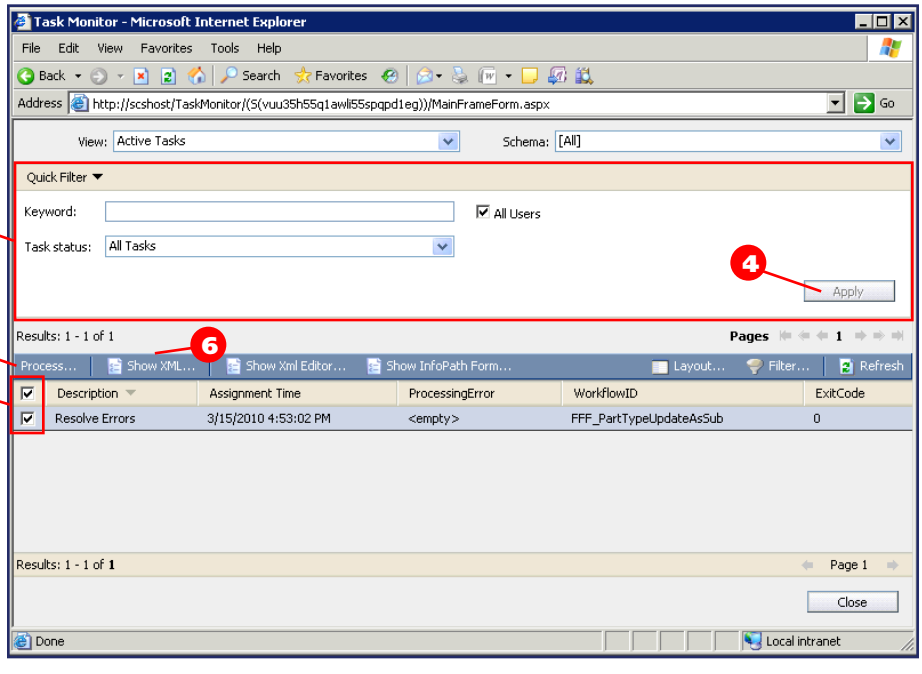
Process a Task

To process a task:

1. In the **Logon to Service Connect** page, enter your Service Connect credentials in the **User name** and the **Password** fields. You can also select the **Use Windows authentication** check box if Service Connect recognizes your Windows account.
2. Click **OK**.



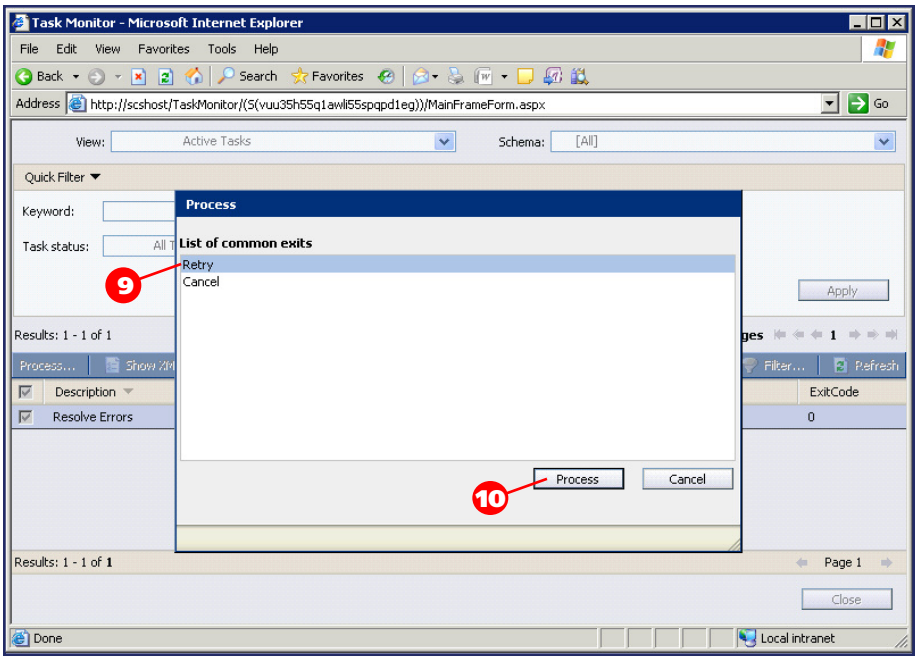
3. The **Task Monitor** displays all tasks. Use the **Quick Filter** options at the top of the form to filter the list of tasks
4. Click **Apply**.
5. In the **Results** grid, select the check box next to the tasks you to process.
6. Click **Show XML** to edit or process the XML message sent to the task.
7. The message data displays in an editable format. To process the message as it is, click **Process**. Otherwise, edit the message data, click **Save,** and click **Process**.



9. The **Process** window displays the **List of common exits**. Select an exit.

10. Click **Process**.

The message is submitted according to the selected exit and the workflow design.



Document Tracking

Use document tracking to trace Service Connect workflow progress. Document tracking is done in the Epicor Service Connect Administration Console. You can view certain document metadata and the message data at various execution points, including each activity within a workflow. This can be valuable for troubleshooting during workflow development and also in your production environment.

In Document tracking you can also resume incomplete activities, search on particular message attributes, and tally actual and expected activities, which is typically the same number.

Set Up Document Tracking

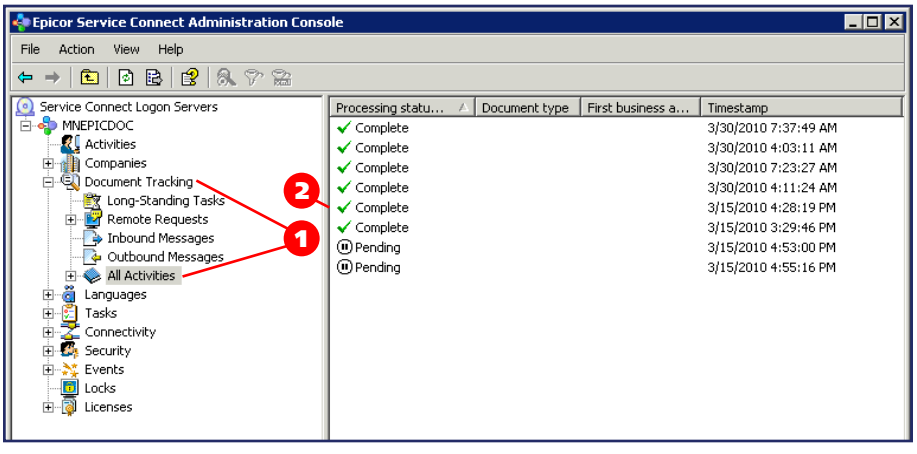
Various Service Connect components, such as message attributes, have an Enable Document Tracking option. Select this option to maximize your ability to view workflow activities.

Document tracking information accumulates on the server until you manually delete it from the Administration Console or until you turn it off. You can turn off document tracking by adjusting the Message Types and Senders options.

To view activities:

1. In the **Epicor Service Connect Administration Console**, expand the **Document Tracking** node and select **All Activities**.

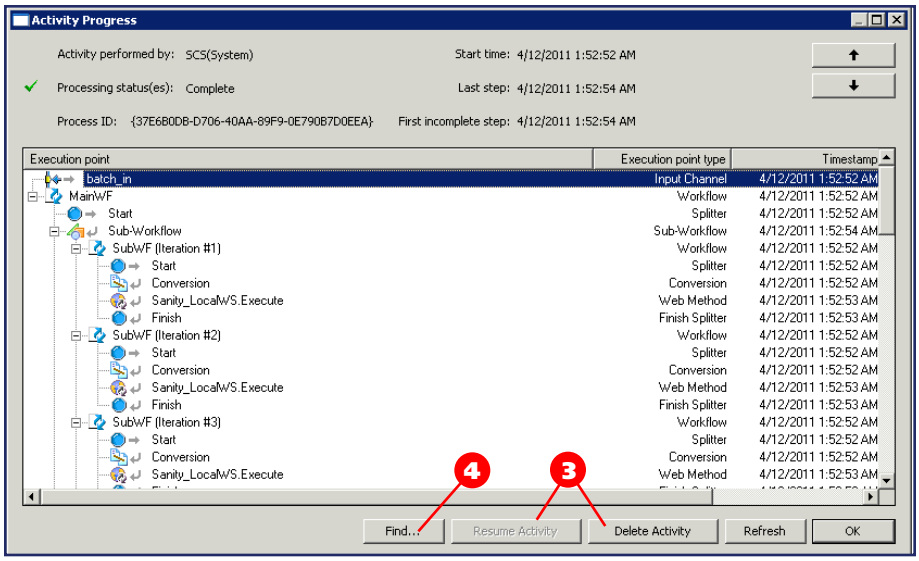
2. All activities display in a list. Double-click an activity to view the details.



3. The **Activity Progress** window displays, with each execution point listed. If the activity is paused, you can select an execution point and click **Resume Activity**. You can also click **Delete Activity** to delete the whole activity.

   Incomplete activities display in red.

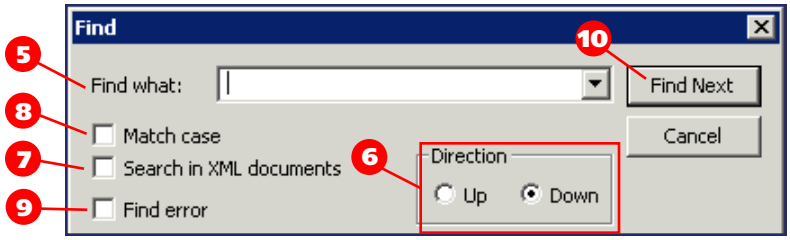
4. To quickly identify a text or an error and examine a complex workflow, use the **Find** button.



5. In the **Find** window, enter the text string you are looking for in the **Find what** field.

6. Select the **Direction** of the search using the **Up** and **Down** options.

7. You can include the xml documents in the search by selecting the **Search in XML documents** check box.



8. You can also perform a case-sensitive search by selecting the **Match case** check box.

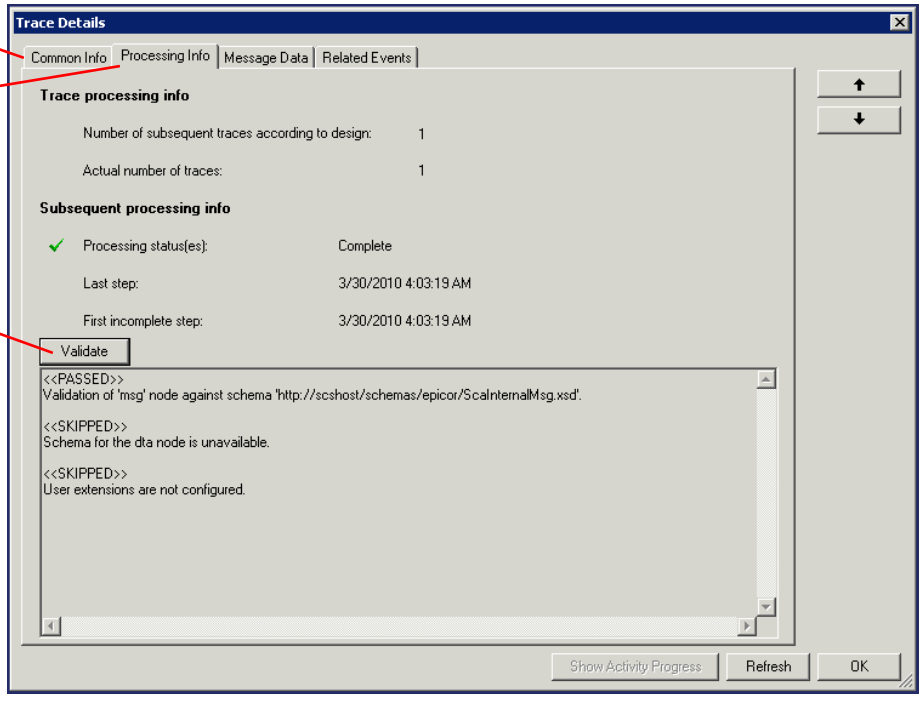
9. To find errors, select the **Find Error** check box.

10. Click **Find Next** and double-click an execution point to view the details.

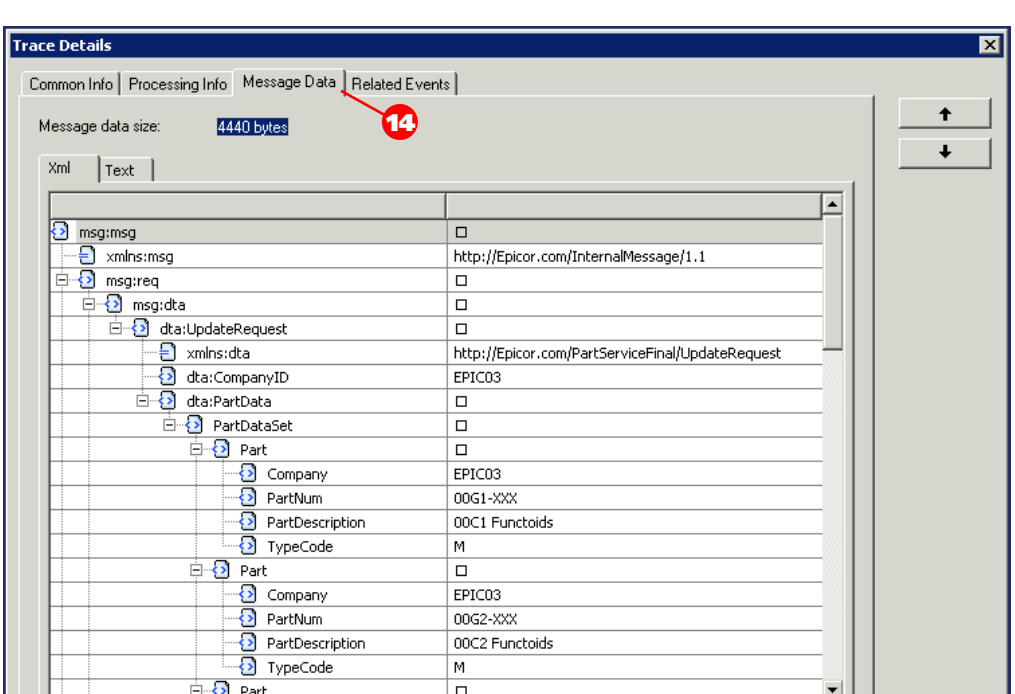
11. Detailed data displays on four tabs. Click the **Common Info** tab to display the common processed document attributes.

12. Click the **Processing Info** tab to display information used to designate whether the document has passed through all the processing stages properly.

13. To check the data that displays in the Message Data tab against the schemas that correspond to the trace element, click the **Validate** button. The validation results display in the lower pane.



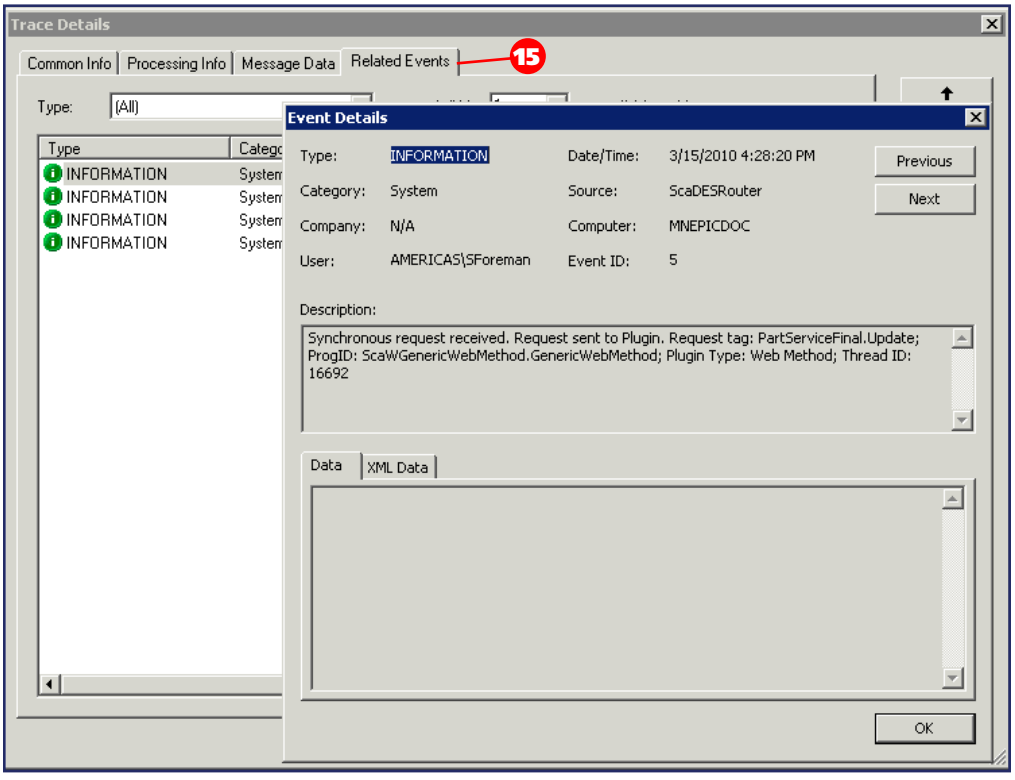
14. Click the **Message Data** tab which contains the XML data of the document at that execution point. If the message data is too large to display in the dialog box, the system offers access to a file that contains the data. This can be valuable information during workflow development and error resolution.



Service Connect automatically detects the Initial encoding based on unicode BOM signature. The message text displays using the selected encoding. If the inbound message can be loaded to msxml then you can view the message as an xml or as text on the Xml or Text tabs. Otherwise the message displays in text and in HEX forms.

To view the statistic information, right-click anywhere on the Xml tab and select **Show statistics**. Additional **Statistics** column displays the number of child nodes and the Xml size of the node (including its child nodes) for each xml node in the message.

15. Click the **Related Events** tab which provides direct access to event logs created for the execution point. These are the same logs that can be viewed in the Events node of the ESC Administration Console. Event logs can provide additional troubleshooting information during workflow develop ment.



Activity Views

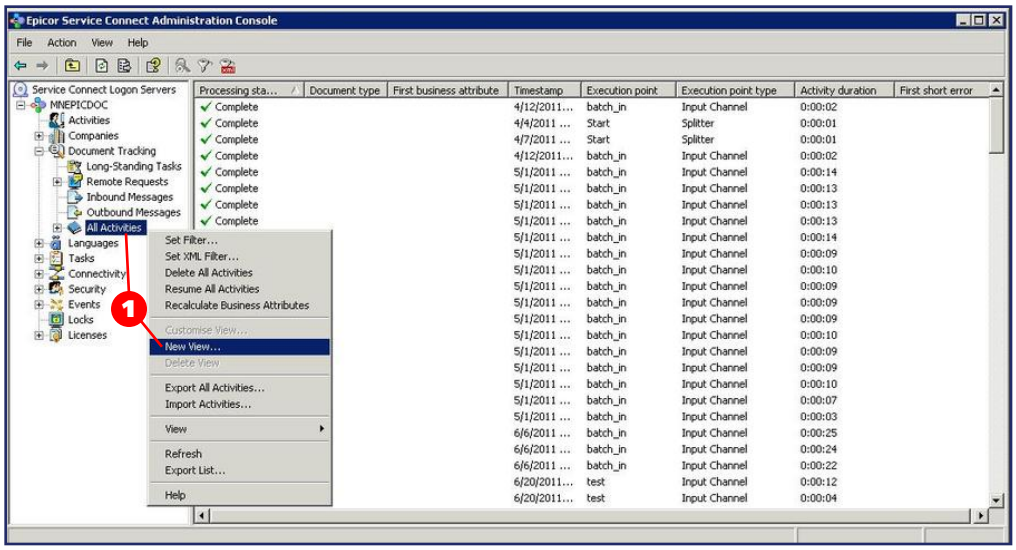
The nodes underneath Document Tracking are views with a defined set of filters. For example, the Pending node, under Incomplete Activities, shows activities whose processing status is not complete and is set to Pending. You can create your own filtering views by creating a new view and defining filters for it or by adding filters to the views included with the product.

You can also define document types and business attributes to create views that display workflow information based on processed business data. For example, you can define a document type called Customer and the view will display the status of workflows that handle customer information. If you add a business attribute called customer region, the view can be limited to workflow processing for customer documents where the customer is located in a particular area.

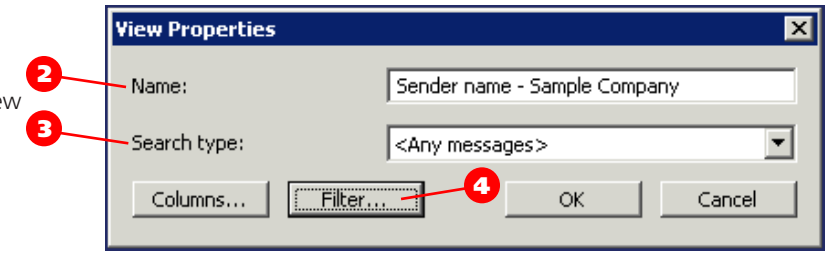
Add a New View

To create a new view:

1. In the **Tree View**, expand the **Document Tracking** node, right-click the view you want to use as your starting point and select **New View**. Or, use **All Activities** node as the starting view to begin with an unfiltered view.



2. The **View Properties** window displays. Enter a view **Name**.
3. Optionally, select a **Search type** to restrict the view to certain types of messages.
4. Click **Filter**.



5. The **Filter** window displays. Click the **Advanced** tab.

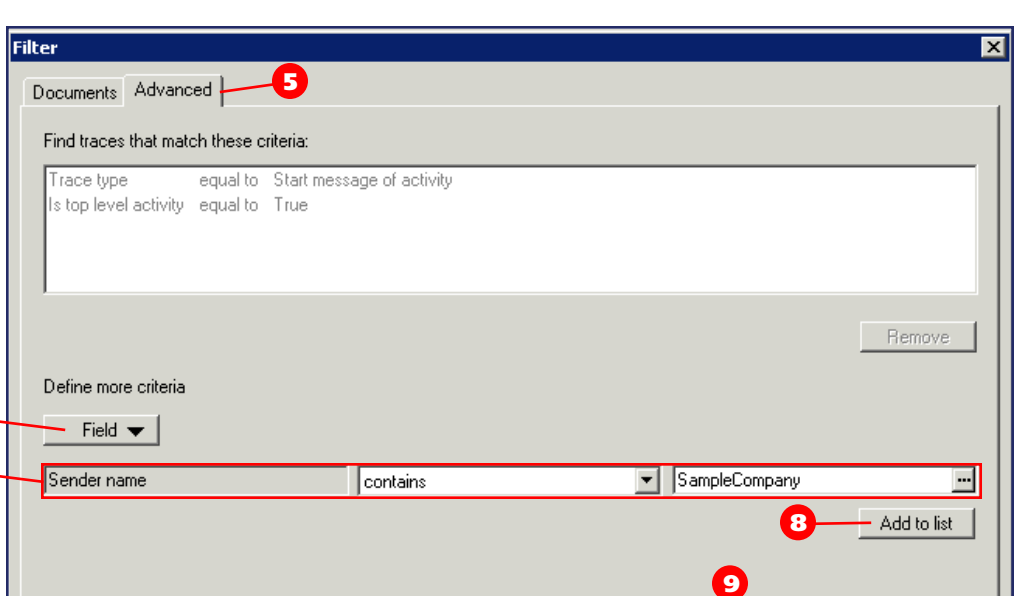
6. Click **Field** and select a message field to use in a criteria statement.

   To view the list of message attributes you can use to set up a filter for the Document Tracking views, see Epicor Service Connect Help.

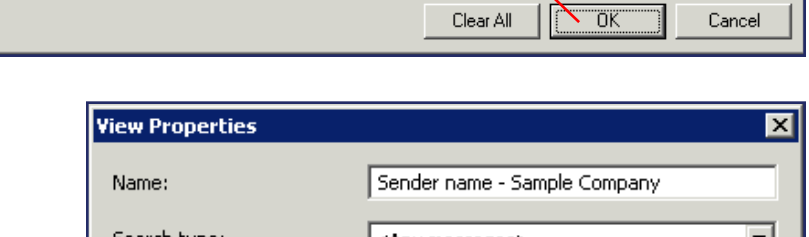
7. Enter filter criteria to complete your statement.

8. Click **Add to list**.

9. Click **OK**.



10. In the **View Properties** window, click **Columns**.

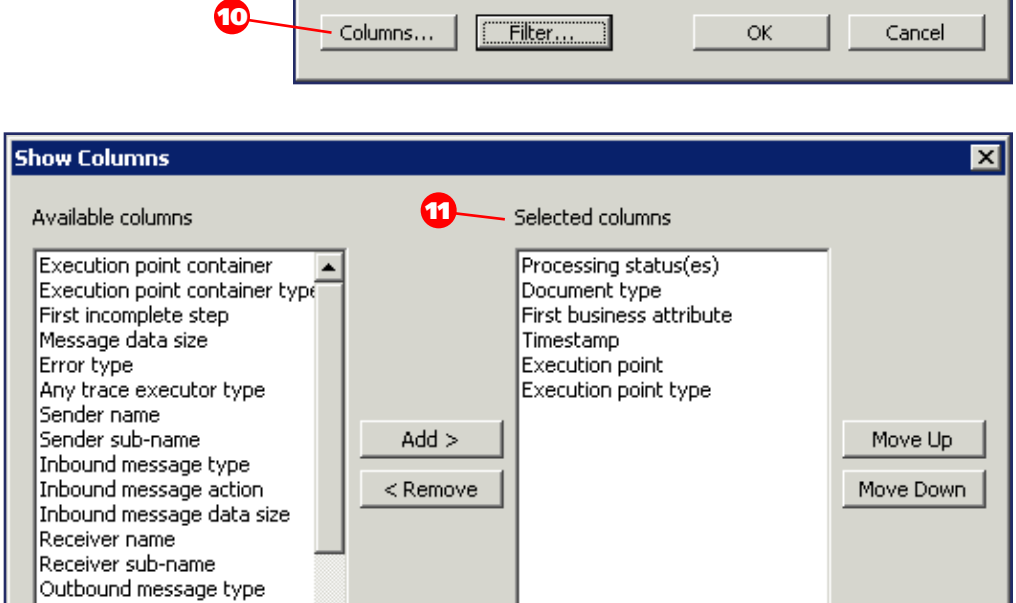


11. The **Show Columns** window displays. Add or remove columns from the **Selected columns** list and adjust the order if needed.

    You specify the columns that appear in the display pane of the ESC Administration Console when you select this view.

12. Click **OK** until you exit all dialog boxes.

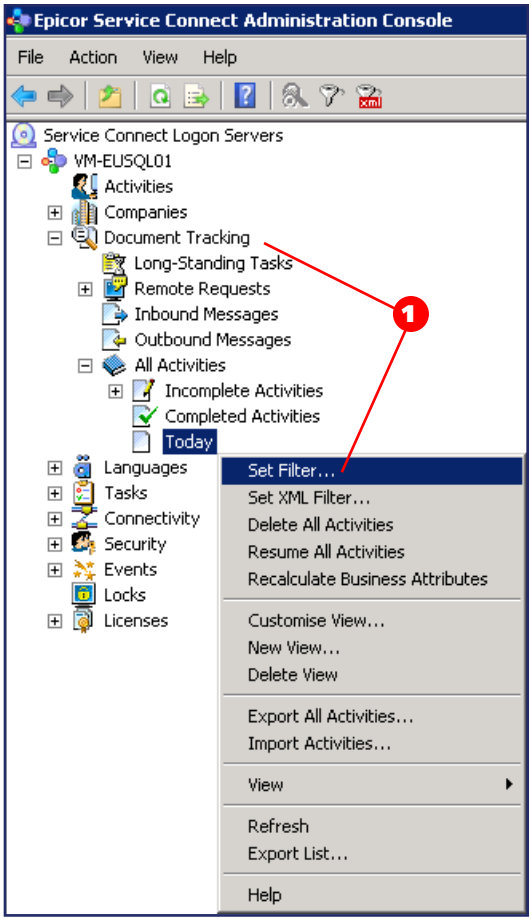
Click your new view to see the results filtered according to your criteria.



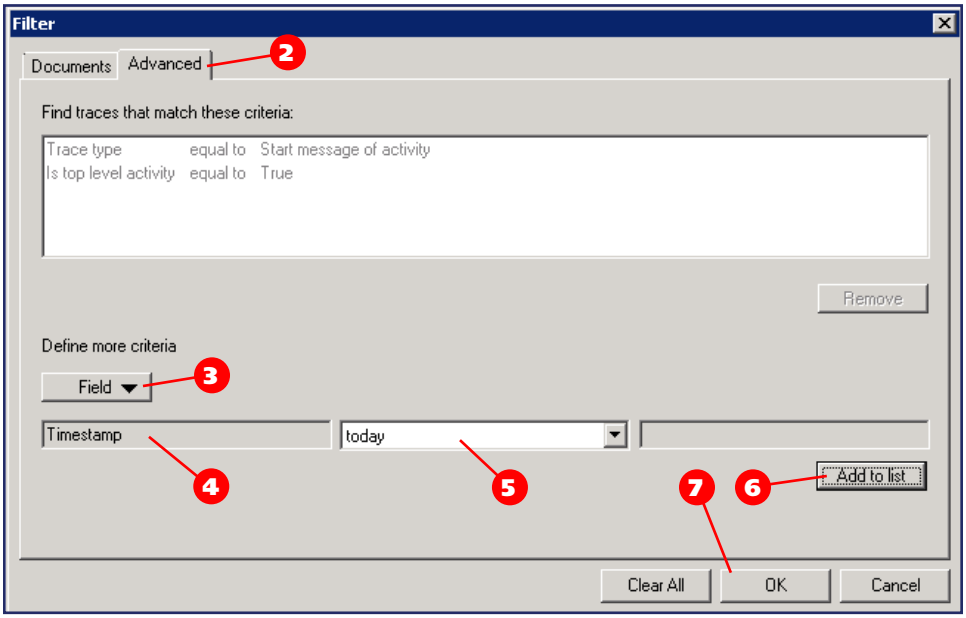
Filter an Existing View

To change the filter for an existing view:

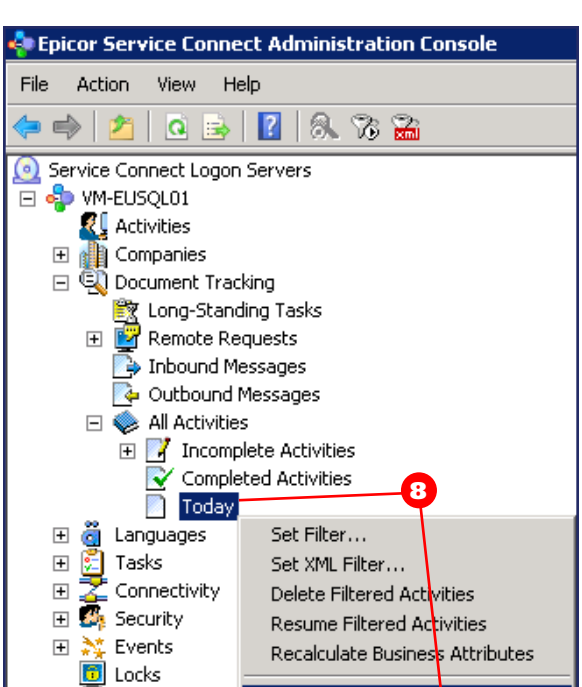
1. In the **Tree View**, expand the **Document Tracking** node, right-click the view you want to use as your starting point and select **Set Filter**.



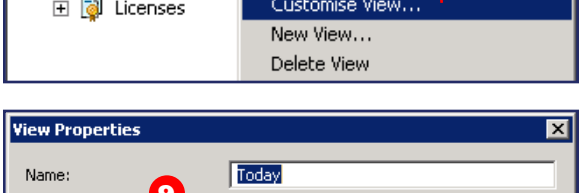
2. The **Filter** window displays. Click the **Advanced** tab.
3. Click **Field.**
4. Select a message field to use in a criteria statement.
5. Enter filter criteria to complete your statement.
6. Click **Add to list**.
7. Click **OK**.



8. Right-click the view and select **Customize View**.



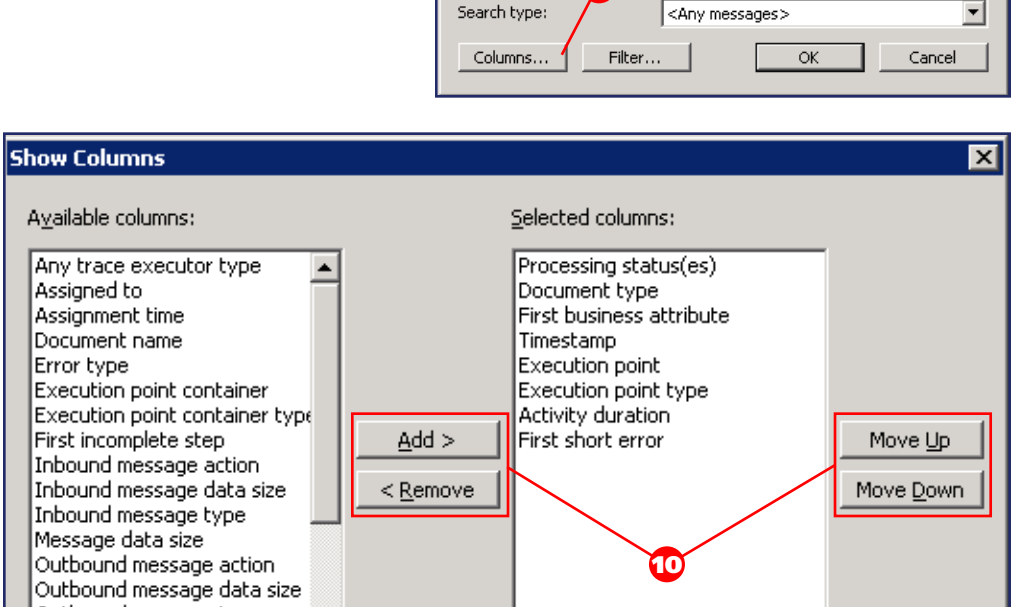
9. In the **View Properties** window, click **Columns**.



10. Add or remove columns from the **Selected columns** list and adjust the order if needed.

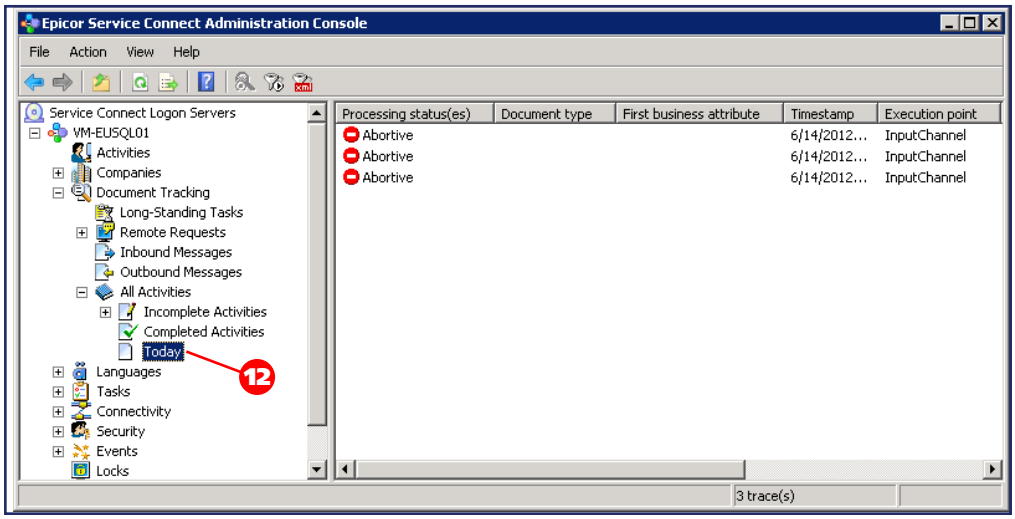
    You specify the columns that appear in the display pane of the ESC Administration Console when you select this view.

11. Click **OK** until you exit all dialog boxes.



12. Click the updated view to see the results filtered according to your criteria.

The view changes remain between ESC Administration Console sessions. To reverse your changes, use the same steps except select your added criteria and select Remove to restore the original view.



Use a Document Type to Filter a View

You can define document types that are assigned to messages that enter Service Connect. In turn, you can use the document type as a view filter, so the view only displays the processing status of documents of a specific type, such as an Engineering Change Order or a Customer.

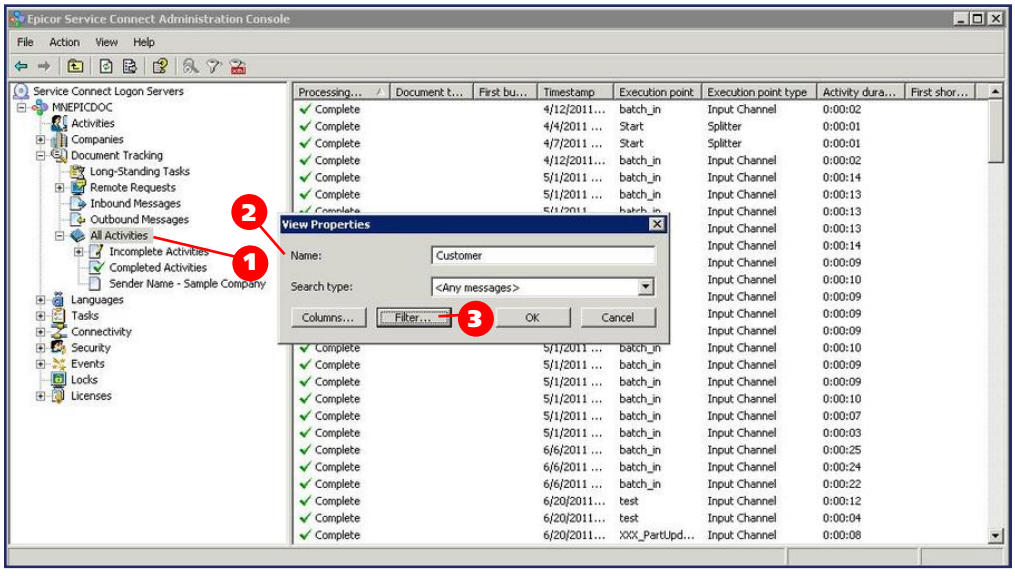
Before you perform this procedure, the schema used to define the document type must be registered in Service Connect. You can register schemas by importing a service reference, discussed in Chapter 2: Epicor Service Connect Administration, or by using the Schema Utility, discussed in Chapter 4: Workflow Designer.

To use a document type as a filter:

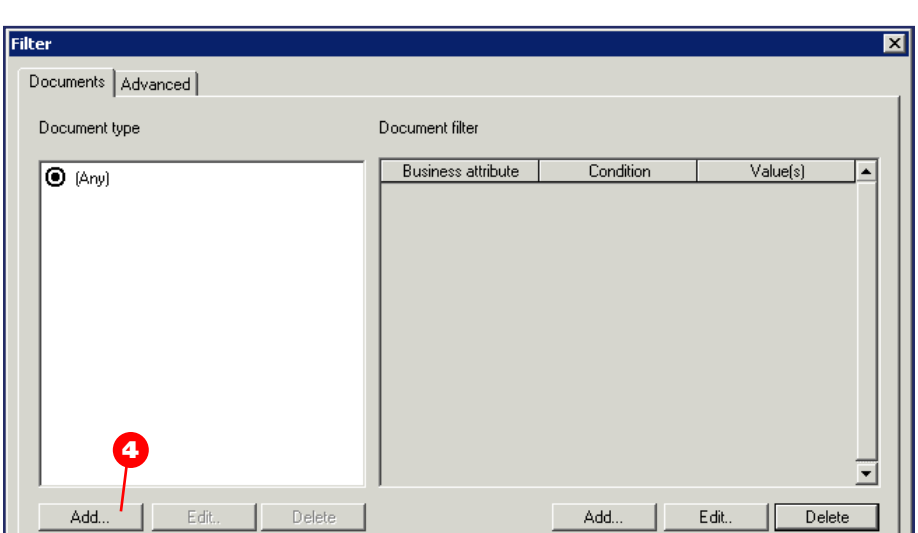
1. In the **Tree View**, expand the **Document Tracking** node, right-click the view you want to use as your starting point and select **New View**. Use **All Activities** as the starting view to begin with an unfiltered view.

2. The **View Properties** window displays. Enter a view **Name**.

3. Click **Filter**.

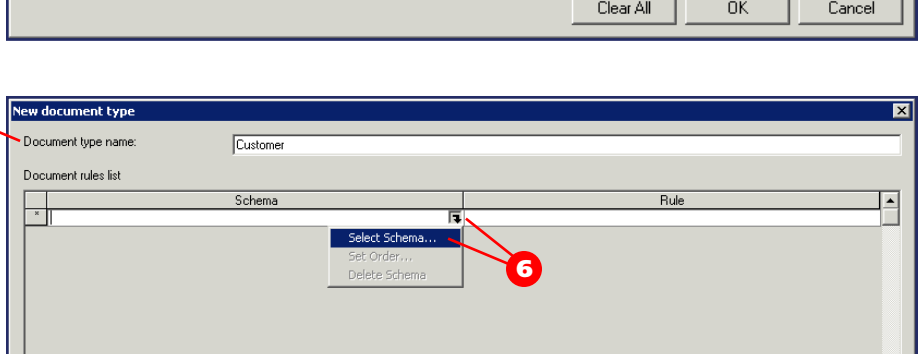


4. The **Filter** window displays. Click the **Add** button under the Document type field.



5. The **New document type** window displays. Enter a **Document type name**.

6. Click the button in the **Schema** field and click **Select Schema**.

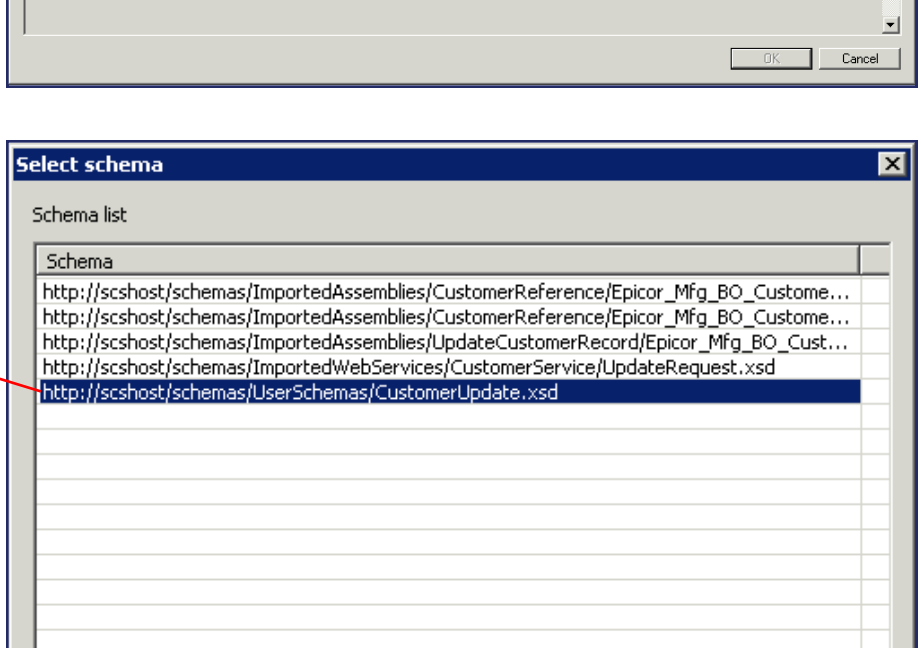


7. The **Select schema** window displays. Select a **Schema**.

8. If a schema is not defined for the document type, you can add it by clicking **Add** and browsing the service references or user-defined schemas.

   Service Connect evaluates documents that enter the system and assigns a document type to any messages that use the selected schema. The messages that use the selected schema display in the view. Optionally, you can define a rule for the schema that must also be met in order for a message to display in the view.

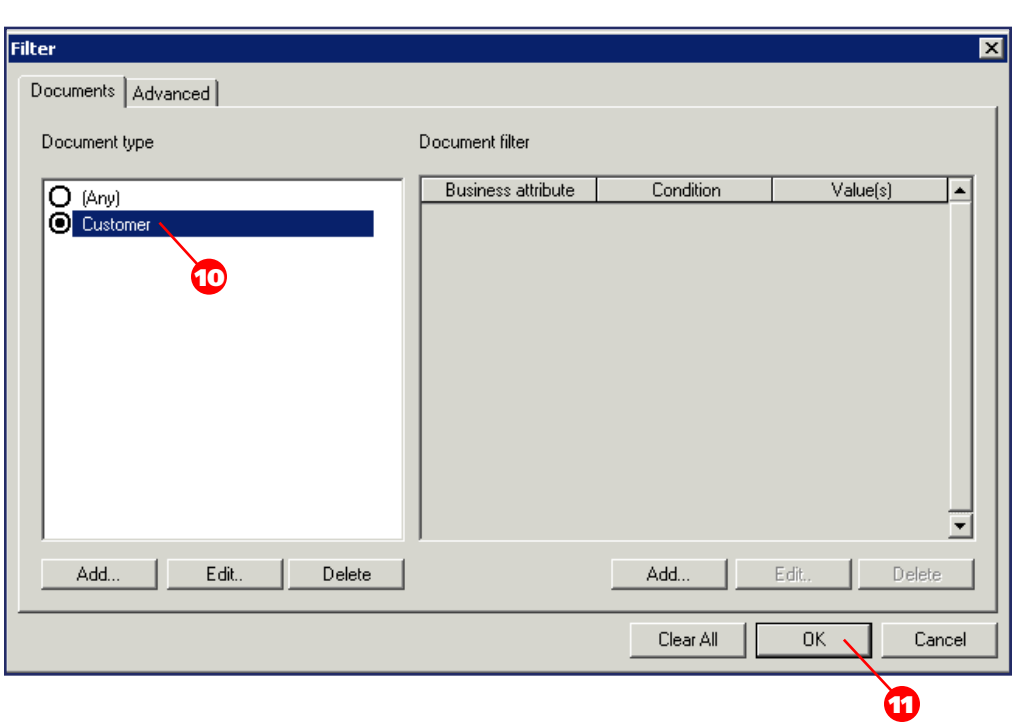
9. Click **OK** twice to return to the **Filter** dialog box.



10. Select the **Document type**.

11. Click **OK** until you exit all dialog boxes.

The view displays only the messages assigned the document type
you defined.

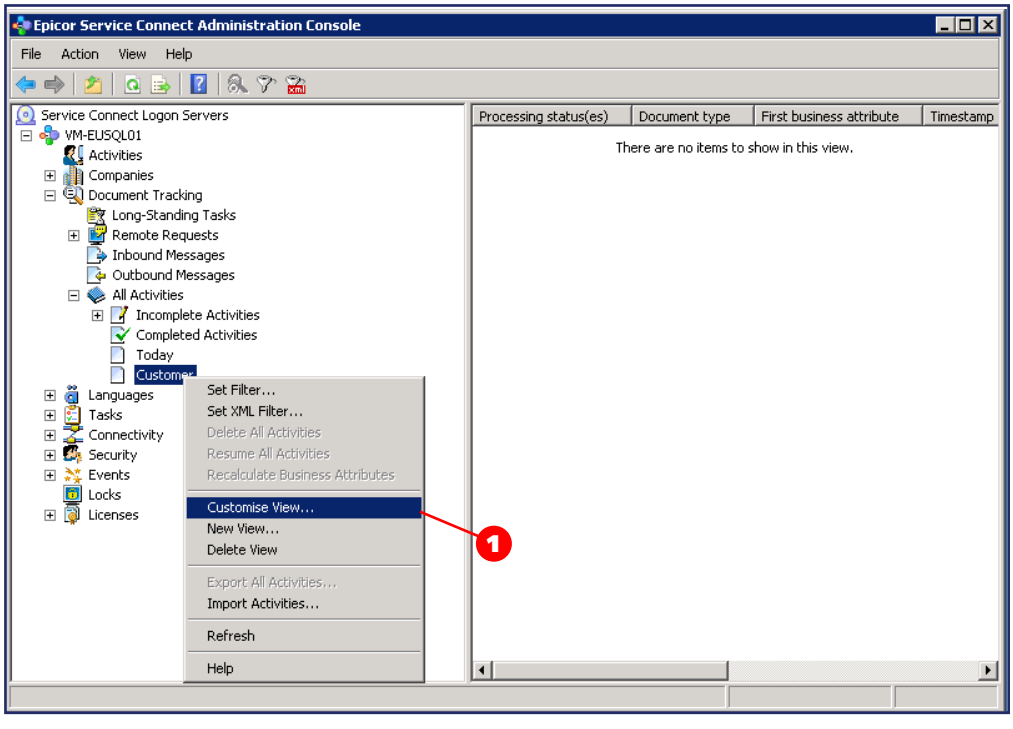


Filter a View Using Business Attributes

You can define business attributes that read the business data in Service Connect messages and use those attributes to further control the number of records that display in a Document Tracking view. For example, if you have a view that displays the status of Customer documents, you can add a business attribute to the view so Service Connect displays only the documents where the Customer ID starts with D.

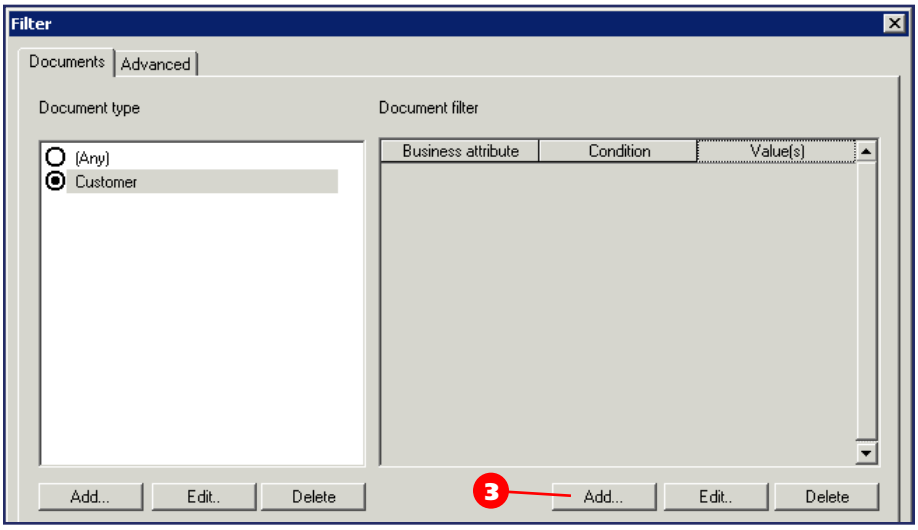
To use business attributes to filter a view:

1. In the **Tree View**, expand the **Document Tracking** node, right-click the view you want to use as your starting point and select **Customize View**.



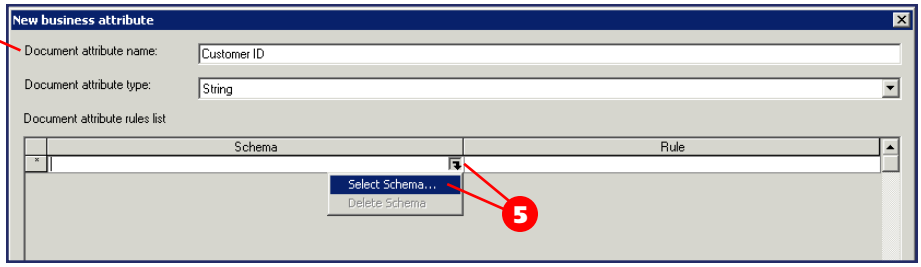
2. The **View Properties** window displays. Click **Filter**.

3. The **Filter** window displays. Under the **Document filter** grid, click **Add**.



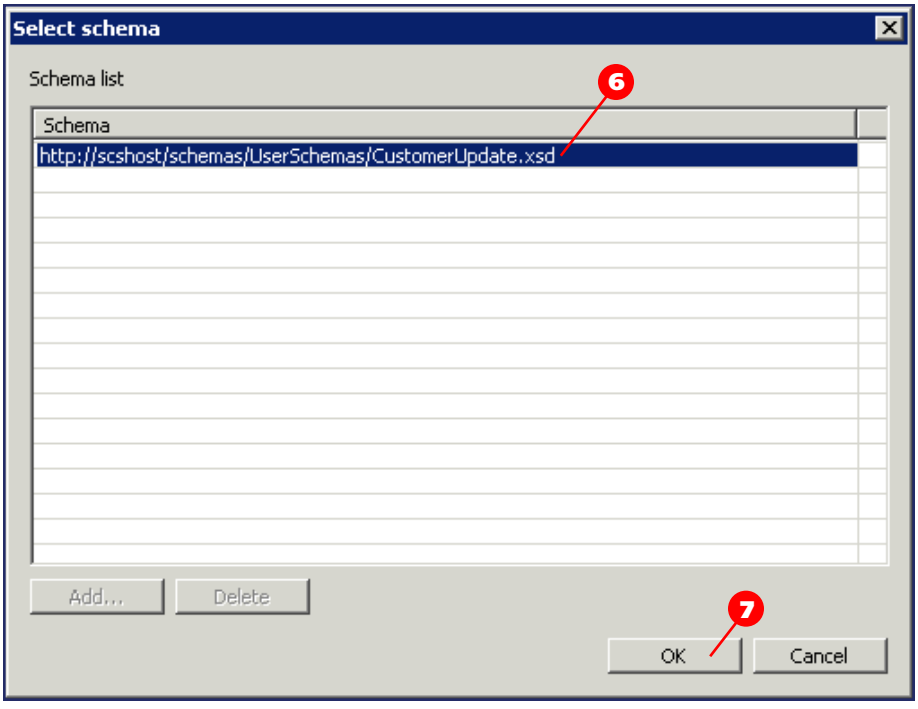
4. The **New business attribute** window displays. Enter a **Document attribute name**.

5. Click the button in the **Schema** field and click **Select Schema**.

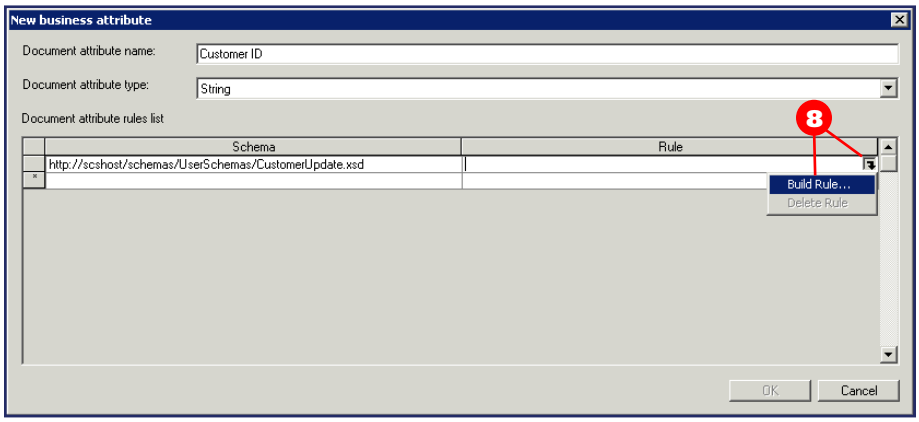


6. The **Select schema** window displays. Select a **Schema.**

7. Click **OK**.



8. The dialog box displays all the schemas associated with the document type. Click the button in the **Rule** field and select **Build Rule**.

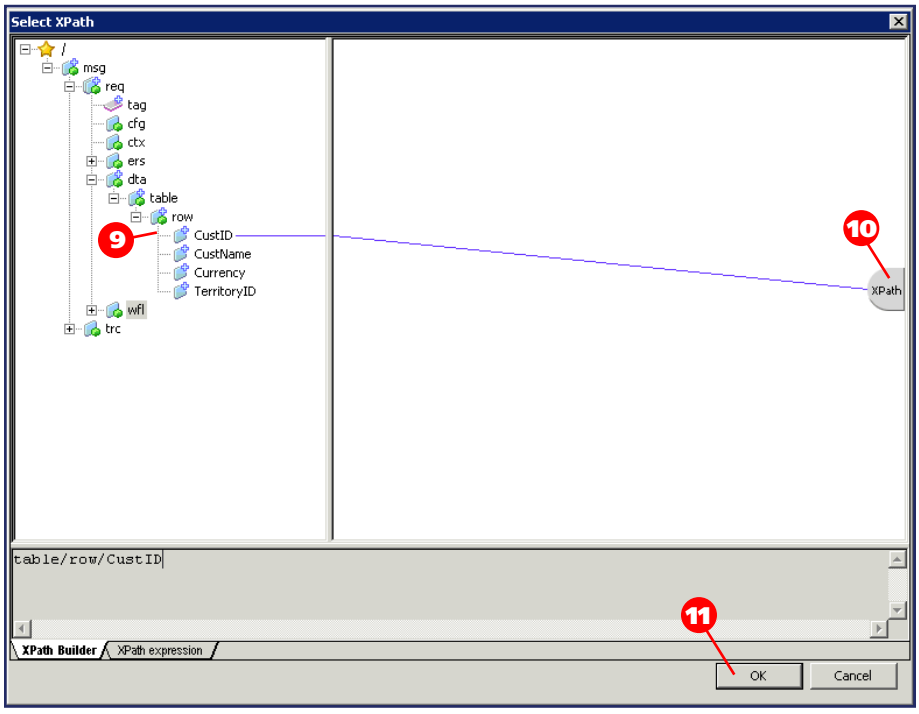


9. The **Select XPath** window displays. Expand the schema nodes to locate the node that contains the business information you want to use as the attribute.

10. Drag the node you want to use and drop it in the **XPath** marker on the right pane. Performing this action indicates the value of this node will be part of the rule.

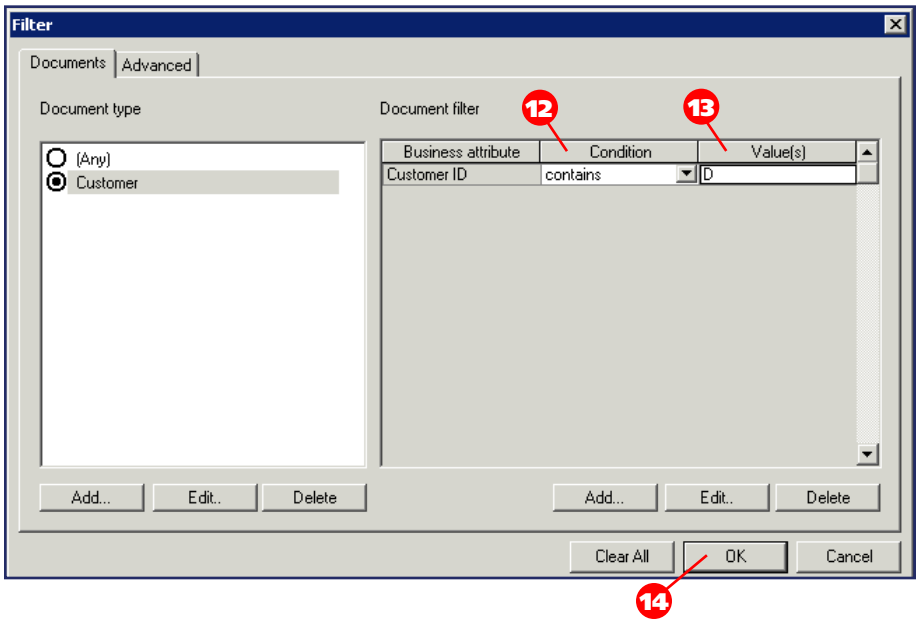
    In this example, the mapping indicates the value of the CustID node will be used to filter the document tracking view.

11. Click **OK** twice to return to the **Filter** dialog box.



12. Select a **Condition** for the rule. Available options:

    - Any
    - Contains
    - Equal to
    - Doesn't contain
    - Is empty
    - Is not empty
    - Is greater than
    - Is less than
    - Is between



13. Enter a **Value(s)** to evaluate against the message data.

    Not all conditions accept values, such as Is empty or Is not empty. The Is between condition requires two comma-separated values.

14. Click **OK** until you exit all dialog boxes.

The resulting view is filtered by the rule according to the data in the message.

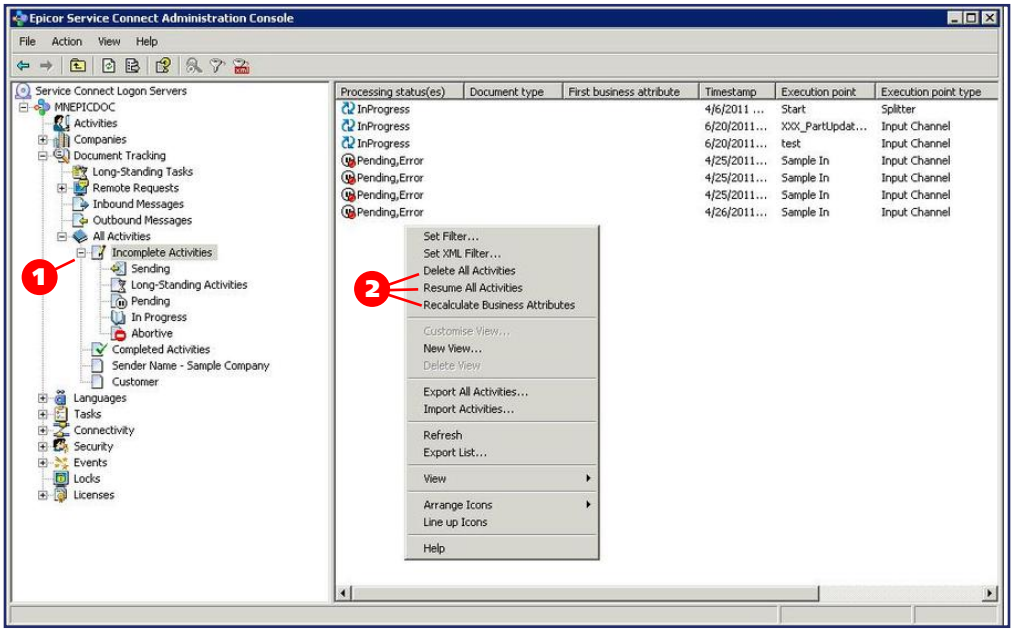
Apply an Action to All Activities in a View

A benefit to using views in Document Tracking is the ability to use a single command to apply actions to all the activities that display in the view.

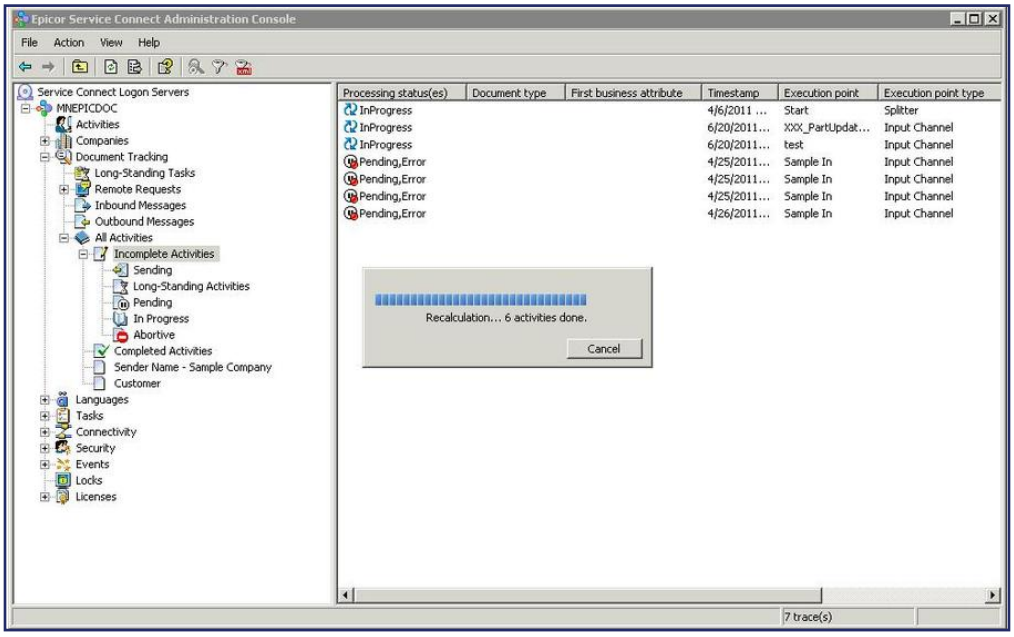
To apply an action to all activities:

1. In the **Tree View**, expand the **Document Tracking** node and click any view.

2. Right-click the display pane where the activities are listed. Select **Delete All Activities, Resume All Activities, or Recalculate Business Attributes**.



Document Tracking view can take some time to populate. If you select a context menu option while the view is still being populated, the wait dialog displays. Wait till the view is populated or click Cancel to terminate the menu operation. The wait dialog can display when the following contect menu items are selected: Delete All Activities, Resume All Activities, Recalculate Business Attributes, Export All Activities and Import Activities.

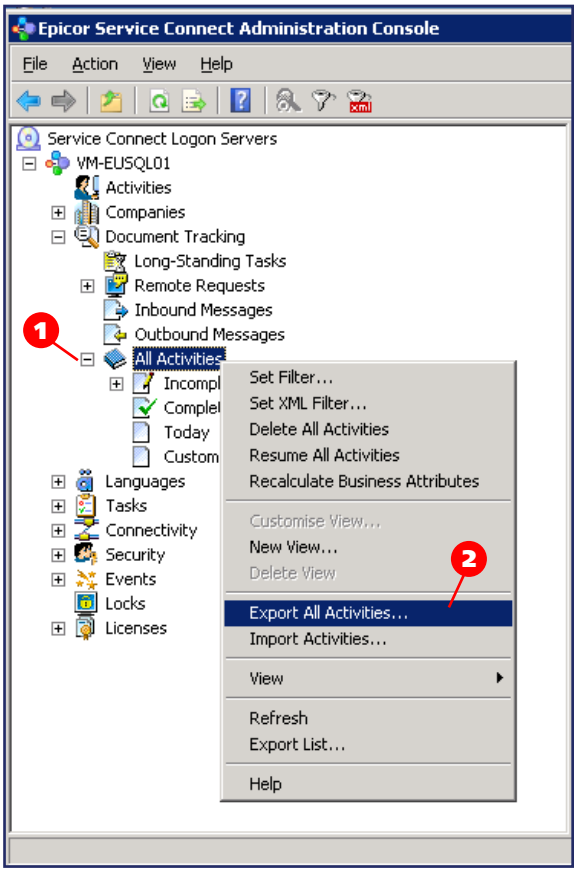


Import to and Export from Document Tracking

The Document Tracking system supports the export and import functionality, which benefits support teams that evaluate issues and share information.

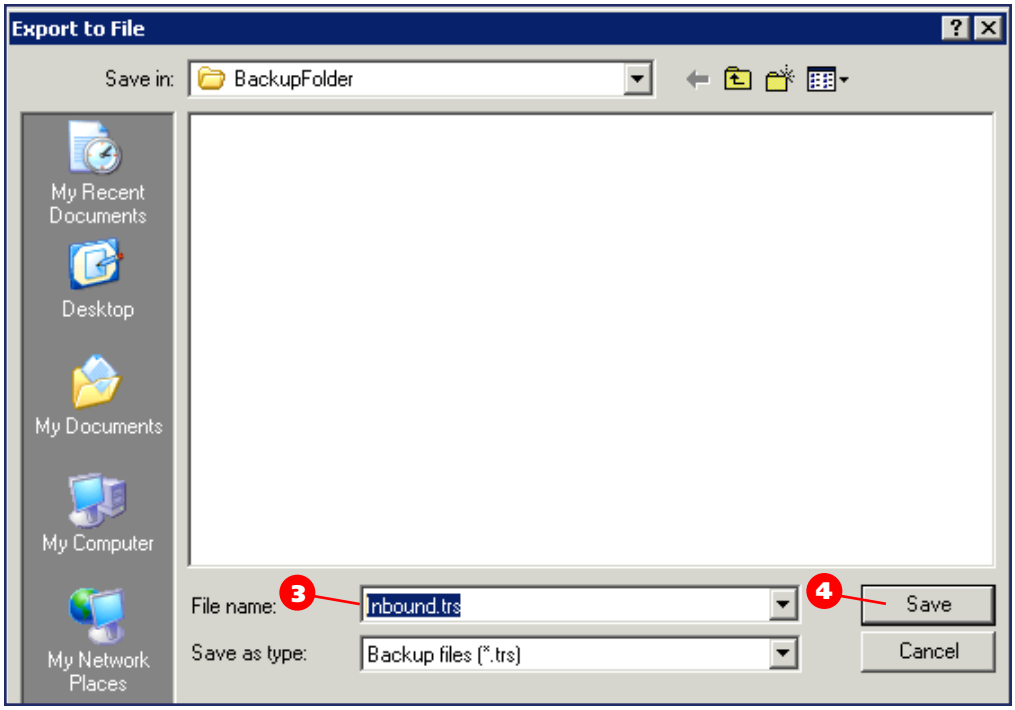
To export activities from document tracking:

1. In the **Tree View**, expand the **Document Tracking** node and click any view.
2. To export all activities within a view, right-click a view and select **Export All Activities**.

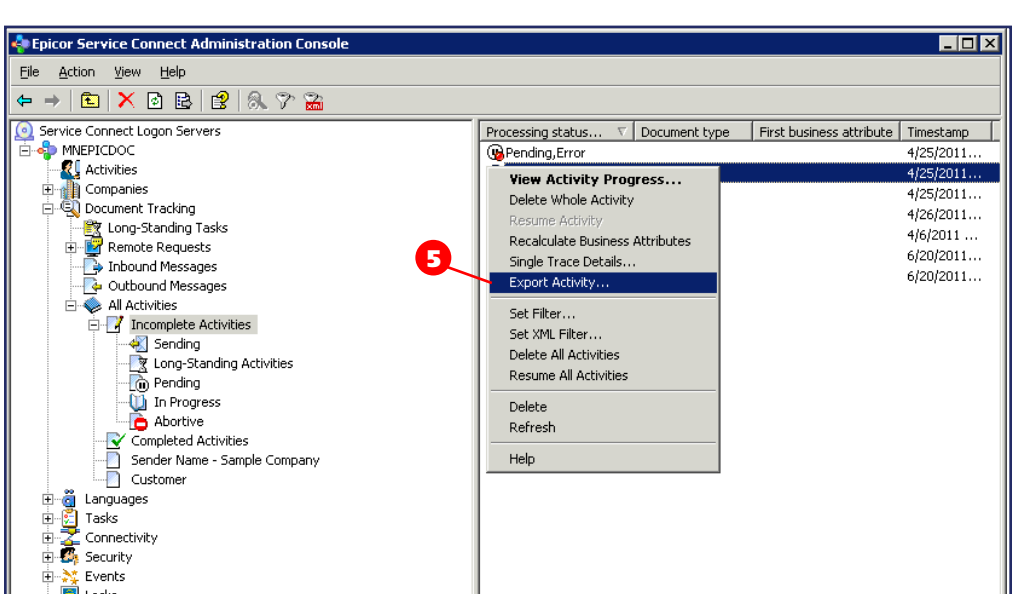


3. Enter a **File name** for the backup file.
4. Click **Save**.

   When you export incomplete activities, you must confirm your choice.

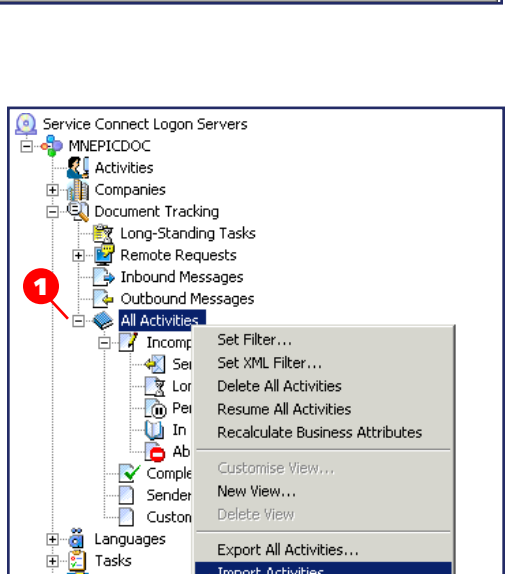


5. To export a single activity or a selection of activities, right-click the activity or activities and select **Export Activity**.



To import activities into the ESC Administration Console:

1. Under the **Document Tracking** node, click any view.
2. Right-click the view and select **Import Activities**.



Summary

This chapter introduced several main application components that make up Service Connect and outlined the basics for configuring a Service Connect solution. The next chapter describes the Workflow Designer tool and the workflow development process.

Chapter 4
Workflow Designer

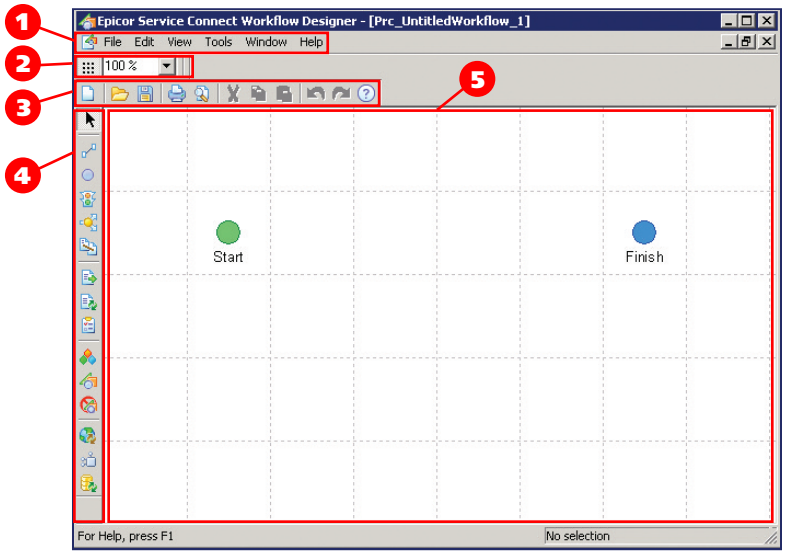
Workflows are graphical representations of a set of interconnected data operations, known as activities or workflow elements. Each workflow models and supports a business process. Each activity is represented by an icon in the diagram. Activities are linked together with connections that show the operation sequence. This chapter explains key Workflow Designer features.

Several activity explanations include descriptions of common data processing techniques. More attention is given to parts of the Workflow Designer that relate to data processing and special techniques. Common functions, such as Save and Open, are explained in less detail.

Workflow Designer Window

The Workflow Designer window consists of the following:

1. Menu bar
2. Commands toolbar
3. Standard toolbar
4. Items toolbar
5. Workflow design area



Menu Bar

The Menu Bar consists of the following menus.

File Menu

Select	To
New Process	Create a new workflow. You can open multiple workflows in the Workflow Designer at the same time.
Open Process	Open an existing workflow.
Close	Close the active workflow.
Save Process	Save the active workflow.
Save Process As	Save the active workflow with another name.
Process Properties	Open the Process properties dialog box. Properties you can set include company information used for the workflow, the workflow version, the default Asynchronous Pool for concurrent execution, document tracking options, message extensions and process variables. Message extensions and process variables are discussed later in this chapter.
Page Setup	Open the Page Setup dialog box where you can set the print settings for the active workflow.
Print Preview	Open a print preview of the active workflow diagram.
Print	Print the active workflow diagram.
Recent Processes	Select from a list of recently opened workflows.
Exit	Exit the Workflow Designer.

Edit Menu

Select	To
Undo	Undo the last action.
Redo	Redo the last undone action.
Cut	Cut a selected item from a workflow and place it in the clipboard.
Copy	Copy a selected item from a workflow and place it in the clipboard.
Paste	Paste an item from the clipboard into a workflow diagram.
Delete	Delete a selected item from a workflow.
Select All	Select all items in the workflow diagram, including connections.

Tools Menu

Select	To
Settings	Open the Settings dialog box, where you can change the logon user.
Schema Utility	Open the Schema Utility. Use this tool to create schemas so you can import data into Service Connect from outside sources. You can also use the tool to create a sample XML document based on a schema. You can alternatively generate a schema in the Schemas node in Epicor Service Connect Administration Console.
Epicor Log Converter	Open the tool that converts standard Epicor logs into Service Connect workflows. The Converter automates the manual routine of extracting business objects and methods from Epicor log files and creating Service Connect workflows.
Generate Schema from Sample Data	Open Generate Schema wizard that helps you to create conversion schema for a sample data file. You use this schema to import data into Service Connect from outside sources. This functionality is discussed later in this chapter.

View Menu

Select	To
Properties	Open the Properties sheet for the active workflow activity. You can set all properties that relate to an activity in the Properties sheet, including details required for the activity to function and the activity's appearance.
Toolbars	Open the Toolbars dialog box, where you can show or hide the various toolbars.
Status Bar	Show or hide the Status Bar at the bottom of the window.
Options	Open the Appearance Options dialog box where you can customize the workflow diagram appearance. Available options include grid line styles and grid size, as well as the fill color and line color of the initial node and connections.

Window Menu

Select	To
Cascade	Arrange open workflow diagrams so they overlap each other, with the title bars visible.
Tile	Arrange open workflow diagrams so they are all visible.
Close All	Close all workflow diagrams.
<Workflow Name>	Bring a workflow diagram window to the front.

Help Menu

Select	To
Help Topics	Open the application help topic for the selected workflow element.
About Workflow Designer	Open the About Workflow Designer window to display the program version number.

Commands Toolbar

The Commands toolbar contains the following options.

Click	To
	Show or hide the grid lines in the workflow design area.
	Set the magnification level in the design area. Select a higher number to make activities appear larger and a smaller number to make activities appear smaller.

Standard Toolbar

The Standard toolbar contains the following commands:

Click	To
	Create a new workflow. Multiple workflows can be open in the Workflow Designer at the same time.
	Open an existing workflow.
	Save changes to the active workflow.
	Print the active workflow diagram.
	Open a print preview of the active workflow diagram.
	Cut a selected item from a workflow and place it in the clipboard.
	Copy a selected item from a workflow and place it in the clipboard.
	Paste an item from the clipboard into a workflow diagram.
	Undo the last action.
	Redo the last undone action.
	Open the application help topic for the selected workflow element..

Items Toolbar

The Items toolbar contains the Selector tool and all the activities you can use in a workflow. The following is the complete list of workflow activities in the order they appear on the toolbar.

Selector

Use this tool to select components that have already been added to a workflow. To select more than one item, hold the Ctrl key while clicking items.

Connection

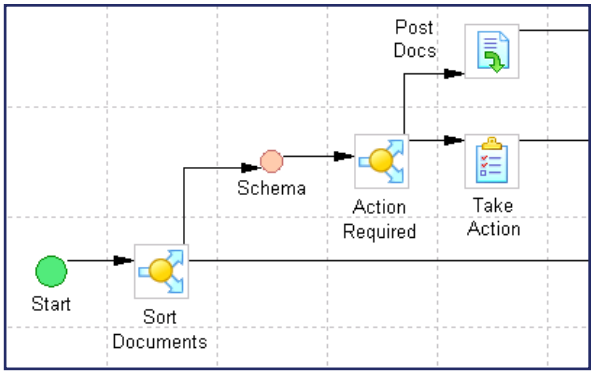
Use this item to connect two workflow activities. All the activities in a workflow must be connected, and at least one activity must be connected to the Finish, before a workflow will function. Connections can have a Caption. The Caption is used by Choice and Task activities, which can have more than one outbound Connection. The Caption appears as the name of the Connection when you define rules for Task and Choice activities.

Splitter

Use this activity to simplify and clarify the workflow layout or to apply a schema to a document on this processing point. Splitters can accept more than one inbound Connection and more than one outbound Connection. The only functional property of a Splitter is the schema that provides context for the following workflow activity.

Example

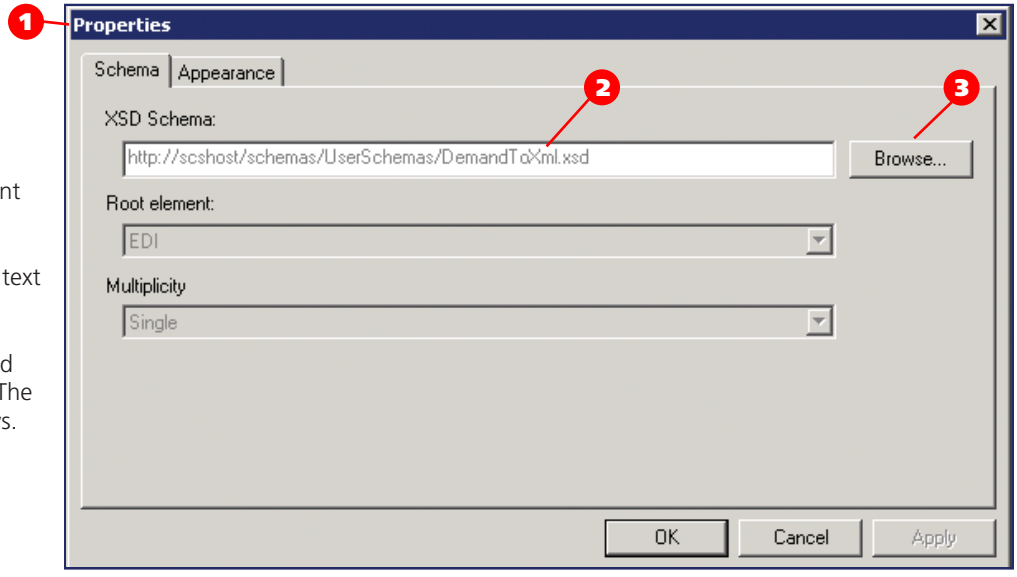
A workflow accepts documents in two different formats.
A Choice activity routes incoming documents based on type. You can use a Splitter after the Choice to apply a schema to the document.



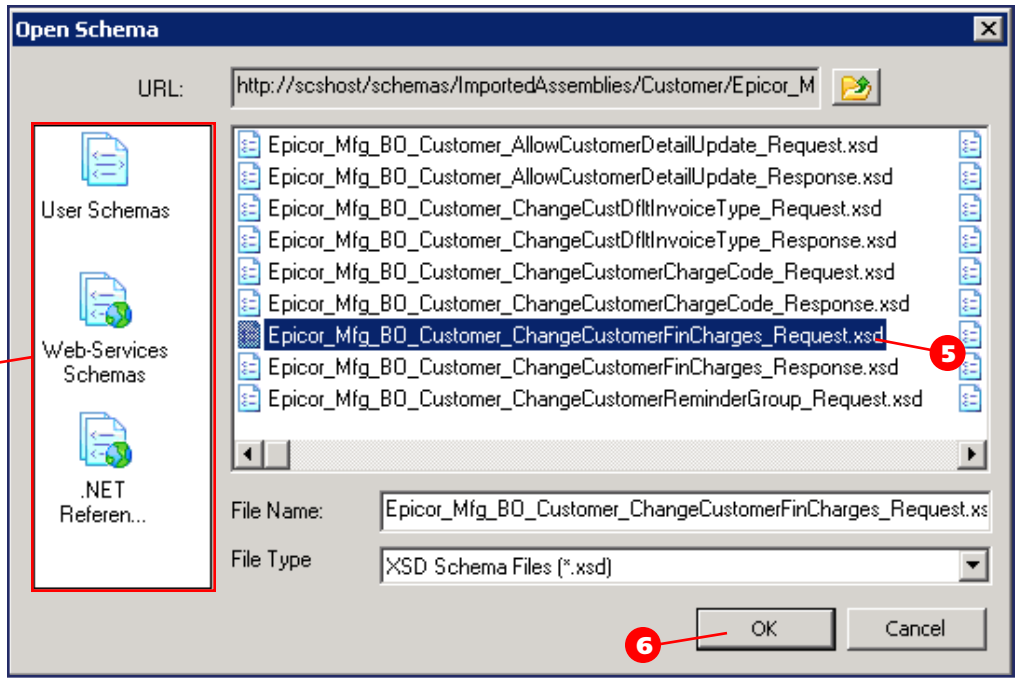
Set Up a Splitter

To set up a Splitter activity:

1. In the **Workflow design area**, add the **Connections** to and from the **Splitter.** Double-click the **Splitter** to open the **Properties** window.
2. If the previous workflow element already has the outbound schema set, this schema is automatically displayed in gray text as the Splitter **XSD Schema**.
3. Click the **Browse** button to find and select a different schema. The **Open Schema** window displays.

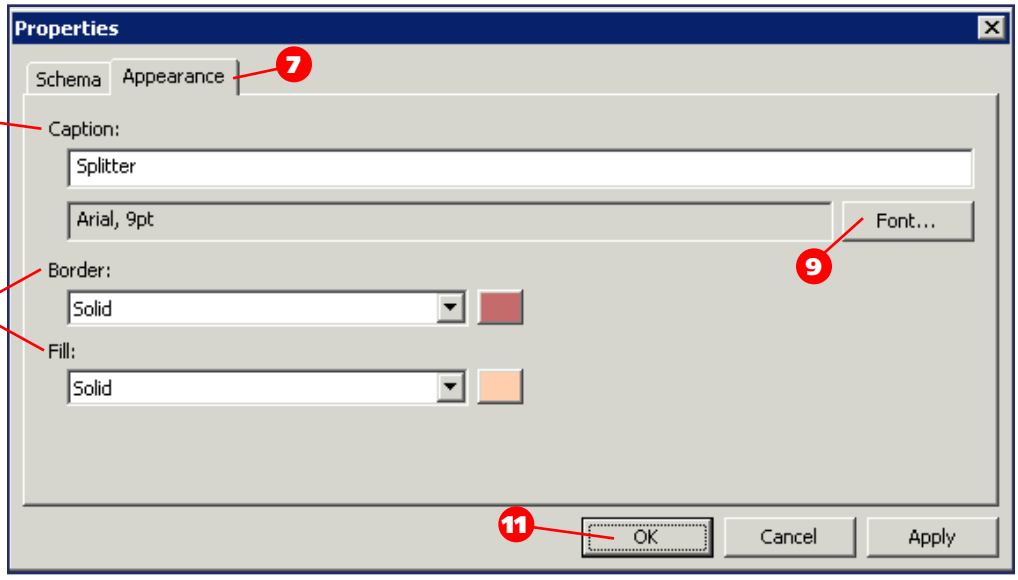


4. In the left pane, select a schema type. Available options:
   - Web-Services Schemas
   - NET Reference Schemas
   - User Schemas
5. In the right pane, select a schema.
6. Click **OK**.



7. Click the **Appearance** tab.
8. In the **Caption** field, enter the Splitter caption to display in the workflow schema.
9. Click the **Font** button to adjust the font settings (style, size, and so on) for the Splitter caption.
10. Click the **Border** and **Fill** drop-down lists to select border and fill styles.

    Use color buttons to the right of the lists to set the border and fill colors.
11. Click **OK**.



Task

Use this activity to issue an assignment to a user in the Service Connect Task Monitor. In the Task Monitor, a user can review the incoming document and edit it if necessary. Afterwards, the user can route the document to one or multiple outbound Connections. You can find examples of how to use the Task Monitor in separate publications that explain how to integrate Service Connect with Epicor ERP, Epicor for Service Enterprises, and Integration Hub. These publications are available for download from EPICweb.

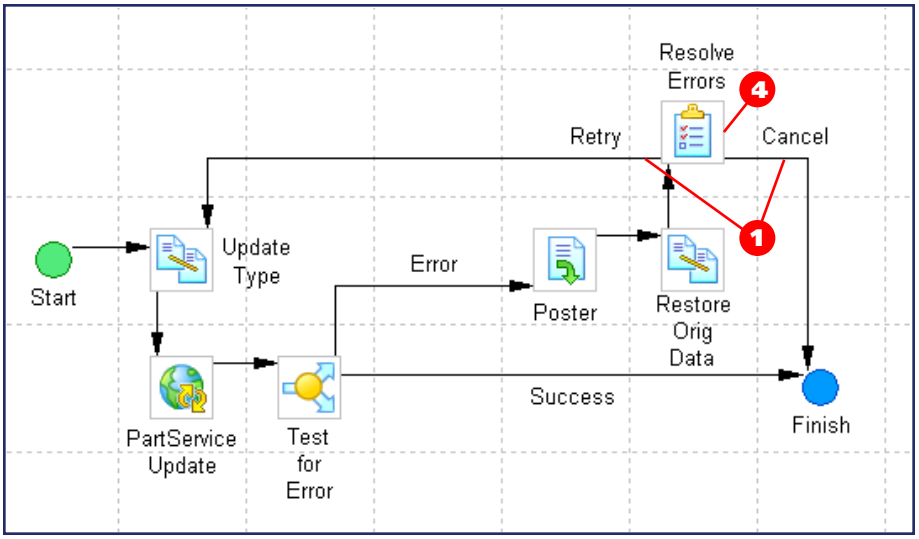
Example

A document is routed along different paths in a workflow based on automatic processing rules and to test for web service errors. If a web service returns an error, you can route the document to a Task activity. The Task pauses the workflow so you can decide how the workflow should proceed. The user you assign to the task can log in to the web-based Task Monitor to make the decision. In Task Monitor, you can edit the document and resubmit it for a database update if necessary.

Set Up a Task

To set up a Task activity:

1. In the **Workflow design area**, add the inbound and outbound Connections to the Task.

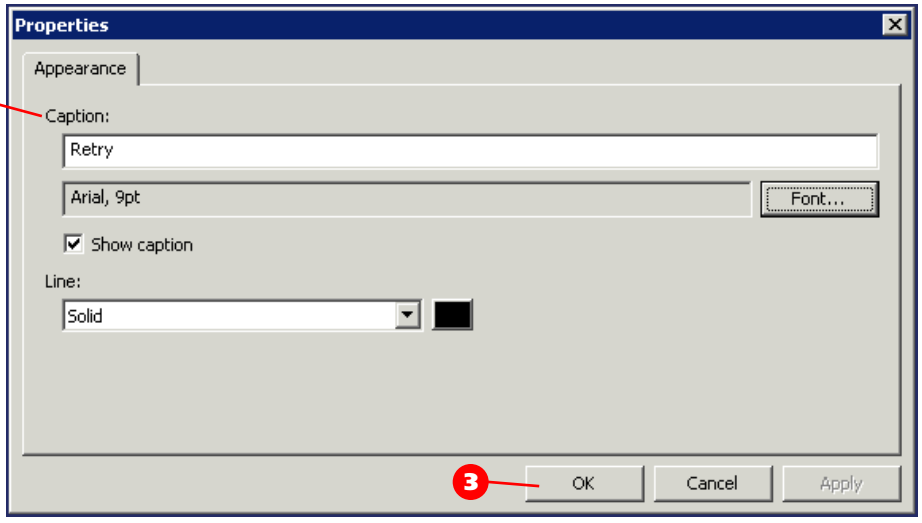


2. Double-click each Connection to define the **Caption** in the **Properties** dialog box.

   Enter a logical Caption for each outbound Connection, to easily identify them when you configure the Tasks exits.

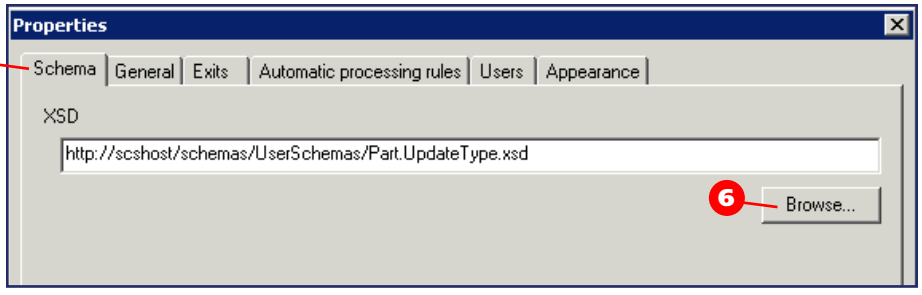
3. Click **OK**.

4. In the **Workflow design area**, double-click the **Task** to modify its properties.



5. The **Properties** window displays. Click the **Schema** tab to select the XSD schema appropriate for the document that will be sent into the Task.

   If the previous workflow element already has the outbound schema set, this schema is automatically displayed in gray text as the Task XSD Schema.



The XML Editor in the Task Monitor uses the XSD schema to visualize task data. If a schema is not assigned to the task, the XML Editor will not be available.

6. Click the **Browse** button to find and select a different schema.

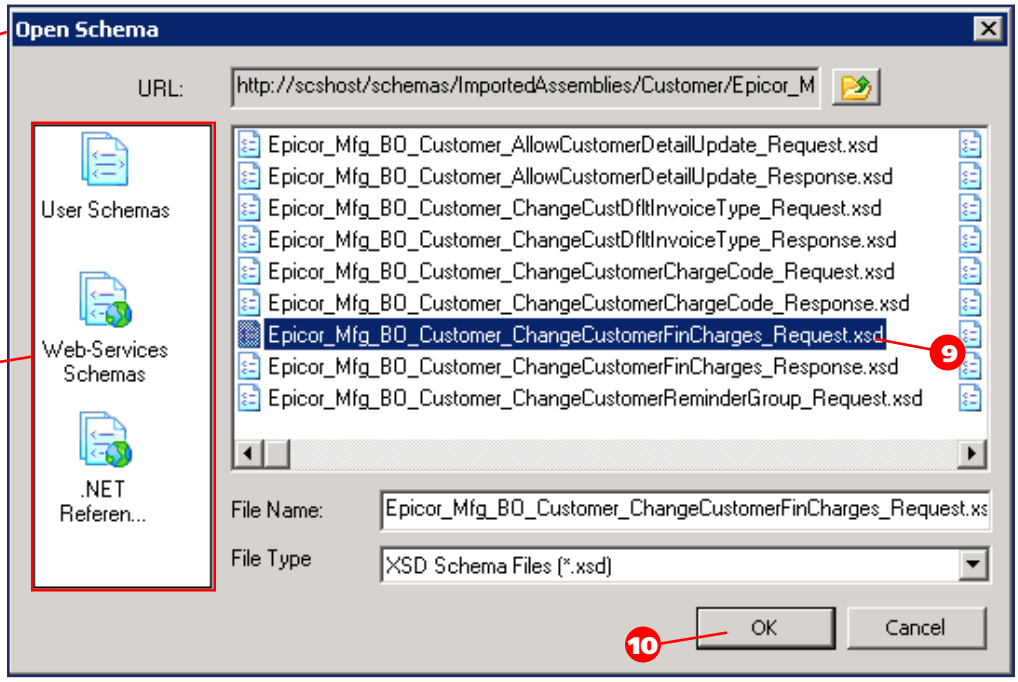
7. **Open Schema** window displays.

8. In the left pane, select a schema type. Available options:

   - Web-Services Schemas
   - NET Reference Schemas
   - User Schemas

9. In the right pane, select a schema.

10. Click **OK**.



11. Click the **General** tab.

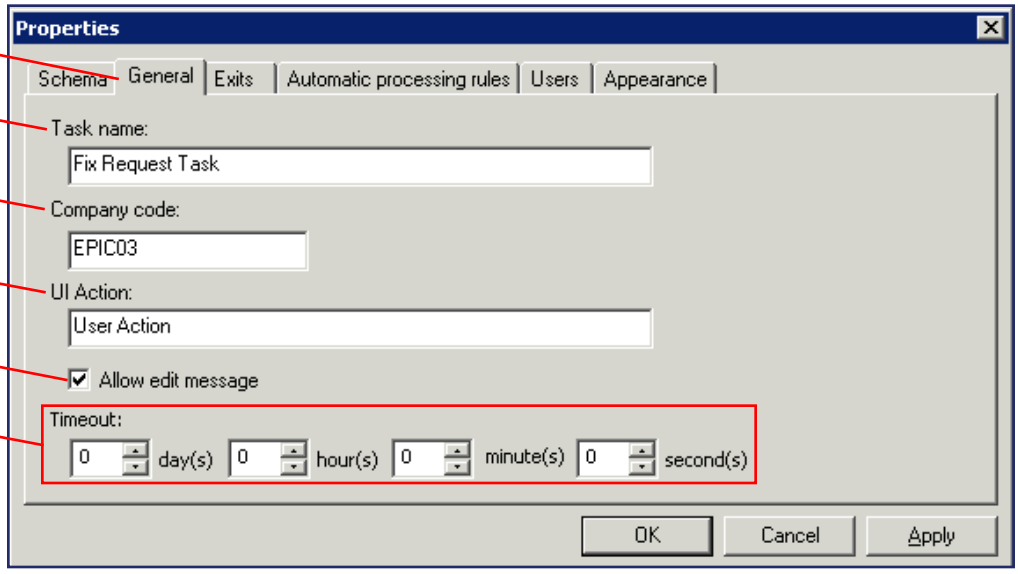
12. Enter the **Task name**.

13. In the **Company code** field, enter the company code.

14. In the **UI Action** field, enter the user action associated with the Task.

15. If you want the Task outgoing XML message available for editing in the Task Monitor, select the **Allow edit message** check box.

16. Use the **Timeout** fields to set the time period (days and/or hour(s), minute(s), and second(s)) during which the Task must be completed.



When the specified period elapses, the Task is considered expired, and the Process is resumed automatically using the default option for further action.

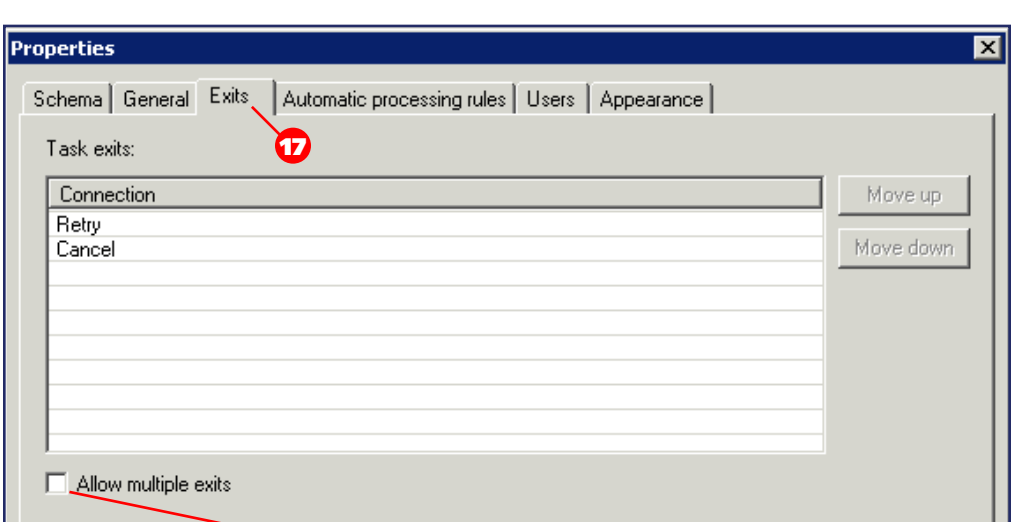
Use timeout feature to avoid deadlocks in data processing in case the decision-maker is unable to complete the Task.

17. Click the **Exits** tab to set how the Task exits rules. .

This example shows two possible outbound Connections: Retry and Cancel.

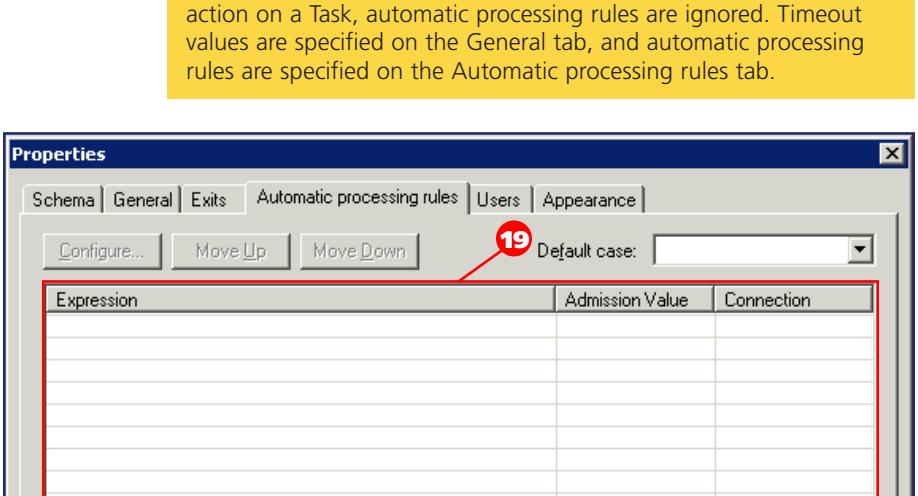
18. Clear the **Allow multiple exits** check box to ensure the user who processes the Task in the Task Monitor can select only one of the outbound Connections.

Each automatic rule is an XPath expression evaluated against the incoming document as either true or false. You can enter one rule for each outbound Connection leaving the Task activity. Documents can progress through each outbound Connection for which they satisfy the rule.



You can also define automatic processing rules for a Task to allow the workflow to continue after the Timeout period for the Task expires. If a Timeout is not set in the Task Properties sheet, the workflow pauses until a user takes action in the Task Monitor. When a user takes action on a Task, automatic processing rules are ignored. Timeout values are specified on the General tab, and automatic processing rules are specified on the Automatic processing rules tab.

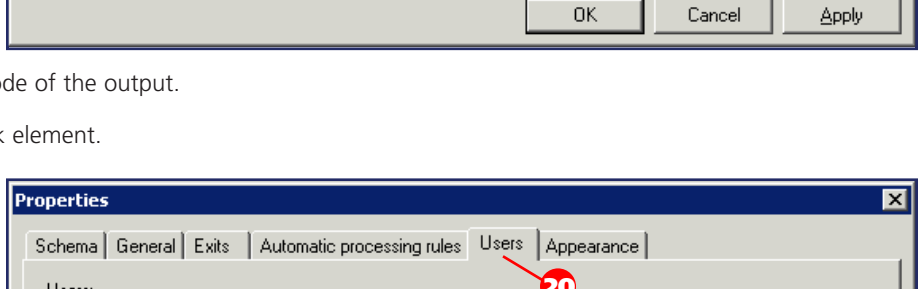
19. On the **Automatic processing rules** tab, at the beginning of the step, enter the following information in the grid columns:



- **Expression** - The XPath statement that is evaluated as the logical expression.
- **Admission Value** - The value the rule expression evaluated value is compared with. If the two values are equal, the Connection specified in the Connection column and the element to which it is connected are selected for further processing.
- **Connection** - The value of <Caption> node, or, if the Caption is empty, the value of <NativeName> node of the output.

  Connection element linked to the Task element.

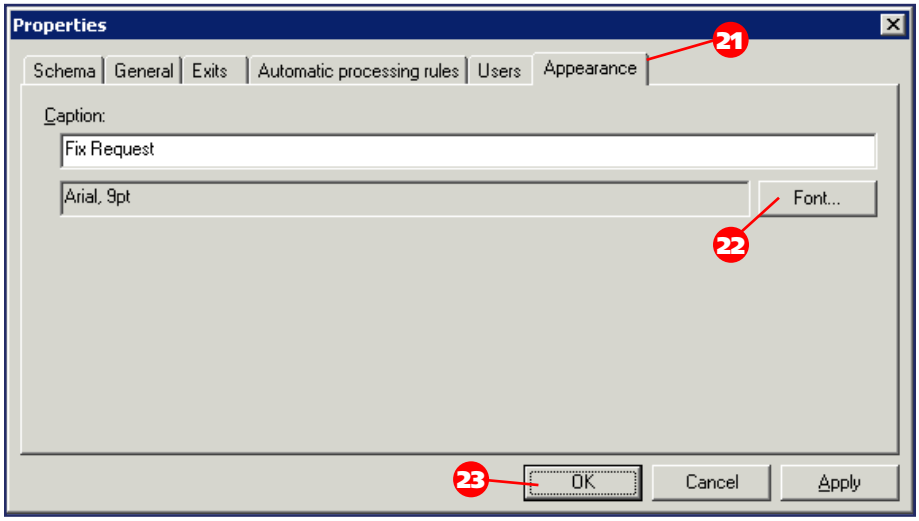
20. Click the **Users** tab to set the user or user group to which the Task will be assigned. The Task will display in the Task Monitor queue for the designated user or for a member of the designated group.



21. Click the **Appearance** tab to enter the **Caption.**

22. You can click the **Font** button to customize the font.

23. Click **OK**.



Poster

Use this activity to publish XML documents from a workflow. The document can be posted through several channels such as a message queue, file system, FTP site, or e-mail based on the Output Channel(s) assigned to the Poster. If the document is posted to a location that a Service Connect Input Channel monitors, you can use the document to trigger a separate workflow. If an Output Channel assigned to the Poster is an SMTP channel, you can use the document as part of an e-mail to notify users about events that occurred within a workflow. You can also use Posters to publish XML documents to the file system to help with troubleshooting. As an alternative to using Document Tracking in the ESC Administration Console, a user designing a workflow can review posted documents to see if Conversions and other activities are functioning as intended.

Example

An incoming document contains a new customer record, which is added to the target database using a Web Method call. After the customer record is added to the database, you can send an e-mail notification to a Sales Manager or Customer Support Manager.

Set Up a Poster

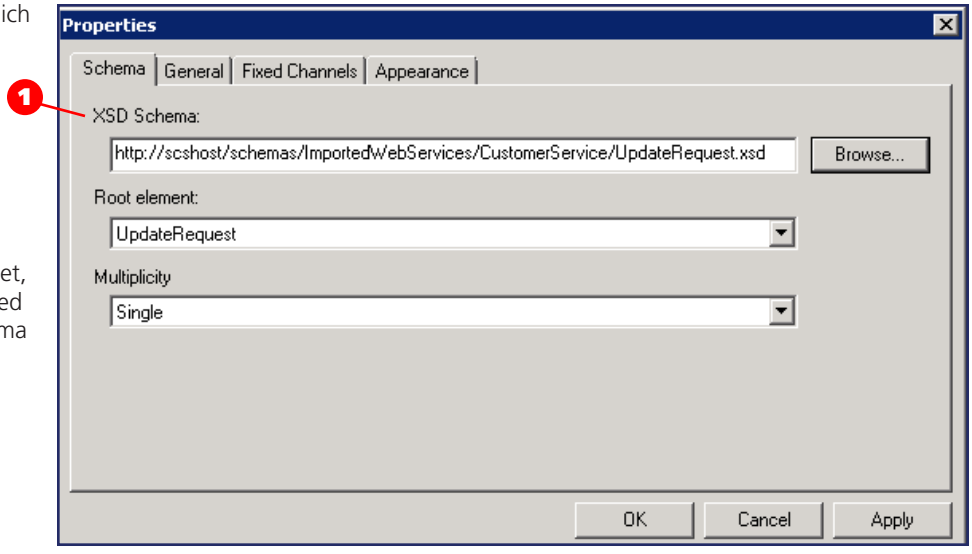
Before you set up the Poster, define an appropriate Output Channel. For this example, configure the Output Channel to use an SMTP server. Review the Channels section in Chapter 3: Connectivity Components for more information.

To set up a Poster activity:

1. Set the **XSD** schema to indicate which document can be used as part of the email.

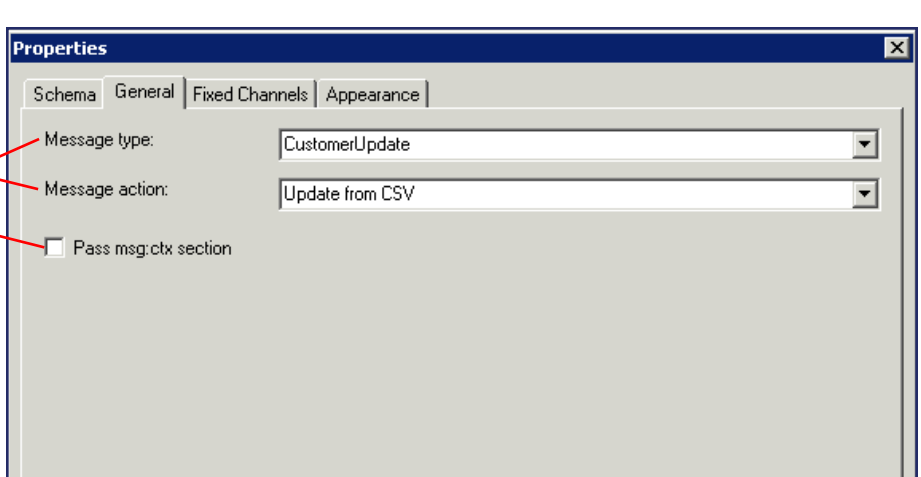
   In this example, the XSD schema corresponds to the new customer information.

   If the previous workflow element already has the outbound schema set, this schema is automatically displayed in gray text as the Poster XSD Schema



2. On the **General** tab, select the **Message type** and **Message action** for the corresponding envelope fields of the message.

3. Select the **Pass msg:ctx section** check box to send the Poster configuration to the workflow. Settings in this node take precedence over the other workflow element configuration settings.

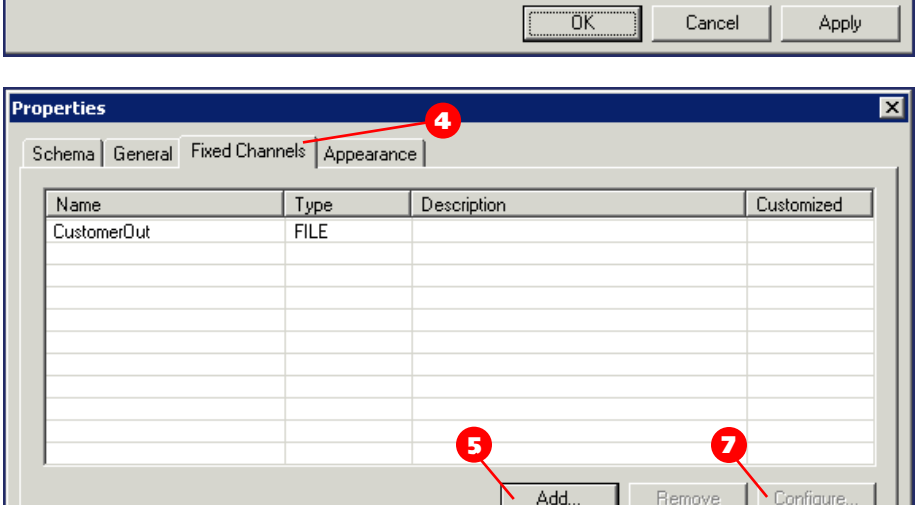


4. Click the **Fixed Channels** tab.

5. Click **Add** to add an Output Channel that is configured to use SMTP.

6. In the **Select Channels** window, select the channel and click **OK**.

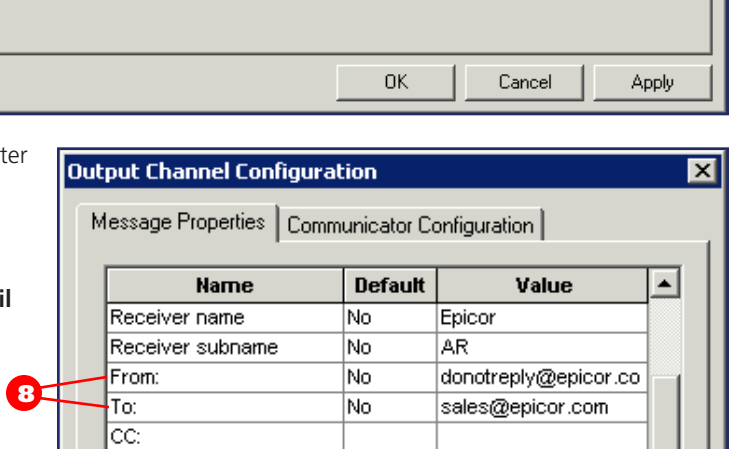
7. Select the channel and click **Configure**.



8. The **Output Channel Configuration** window displays. Enter valid **From**: and **To:** email addresses.

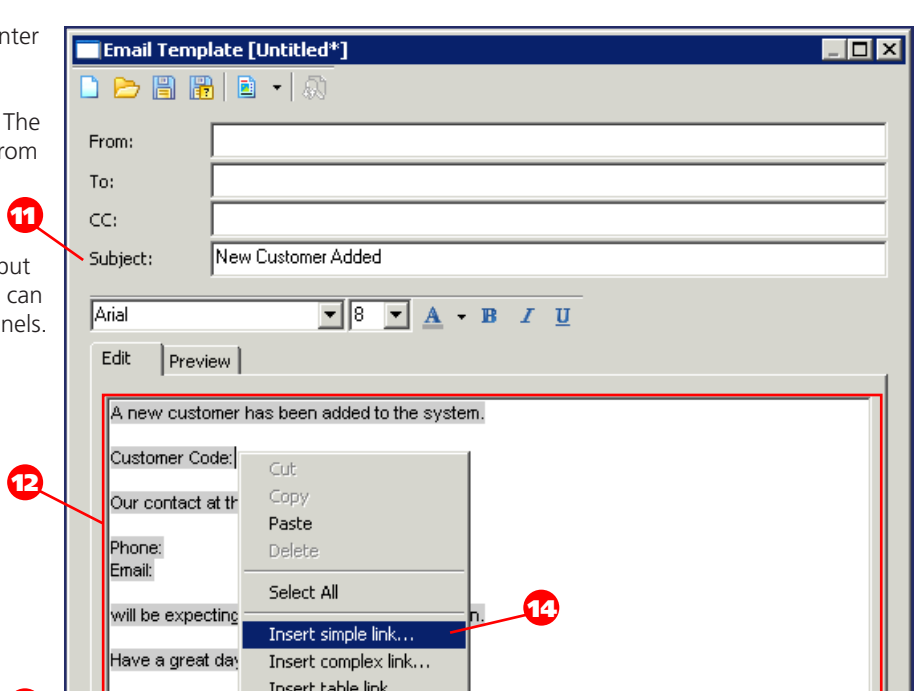
9. Enter an **Email subject**.

10. Click the **down arrow** button on the far right of the **Email template name** field to open the Email Template builder.

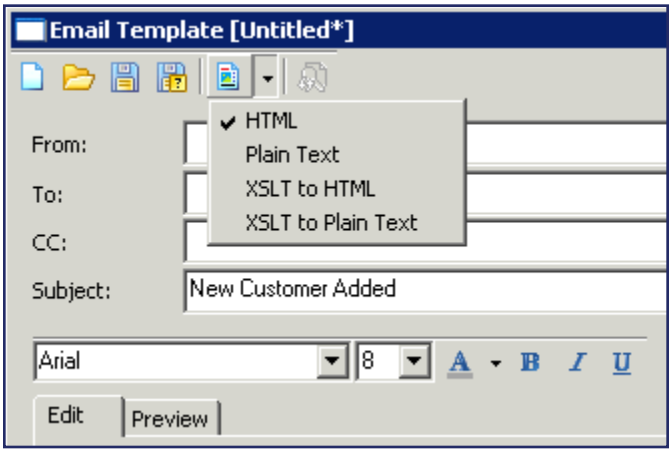


11. The **Email Template** window displays. Enter a **Subject**.

Notice the From and To fields are empty. The information for those fields will default from the message properties entered in the Output Channel Configuration dialog box. The message is set up so the From and To fields are associated with the output channel and not the template. Thus, you can use the template with other output channels.



In the Email Template Builder, you can create templates in several editing modes. Available options:



- **HTML** – In this mode you can build email templates that include tables. Email templates built in this mode result in an outgoing email with HTML body section.
- **Plain Text** – In this mode, you can build email templates with portions of static and dynamic (message-dependent) text; the latter is added via links.
- **XSLT to Plain Text** – In this advanced mode you can build the whole content of the corresponding email message text section by the single XSLT transformation.

  You can load XSLT from an external file, paste XSLT code or enter it directly in the Edit tab of the message body editor.
- **XSLT to HTML** – In this advanced mode you can build the whole content of the corresponding email message HTML section by the single XSLT transformation.You can load XSLT from an external file, paste XSLT code or enter it directly in the Edit tab of the message body editor. Use the editing mode button to switch between modes.

12. In the **Edit** field, enter the body of the email.

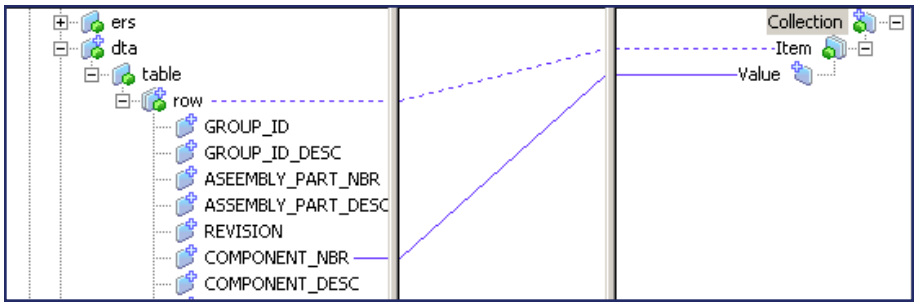
Use the toolbar above to select formatting options, such as font, size, and color.

13. Select the **Attach outgoing message** check box to attach the XML document that contains the customer record to the email.

14. Right-click the message where you want to include information from the workflow and select **Insert simple link**.

In HTML mode, the **Insert table link** option is available. Table links produce formatted tables in outgoing email message and display as a link with a name that consists of column names delimited by pipe characters. For more information on creating a table link, see the Epicor Service Connect Help.

Complex links are useful for documents that contain a collection of items. In this example, the document on the left represents a bill of materials. The row node is a collection, meaning that one or more rows may be present in the document. To create a complex link, map the row node to the Item node on the right. This mapping, shown as a dashed line, indicates that for each row, an item will be created in the email template. The second mapping indicates the value of the COMPONENT_NBR node will appear in the email message. In this instance, select a separator to appear between the items in the output. You can select either TAB or CRLF (Carriage Return/Line Feed).

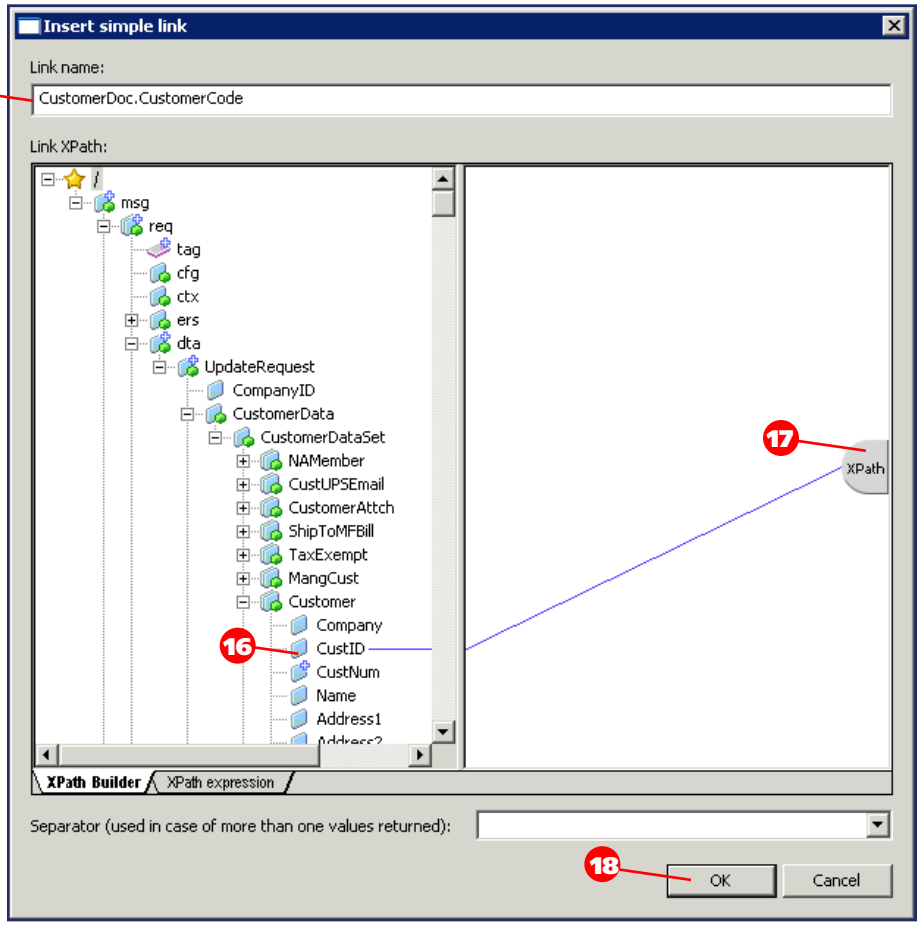


15. Enter a **Link name**.

16. In the **Tree View**, expand the schema nodes until you locate the value you want to insert into the body of the email.

17. Drag the node and drop it in the **XPath** marker on the right pane.

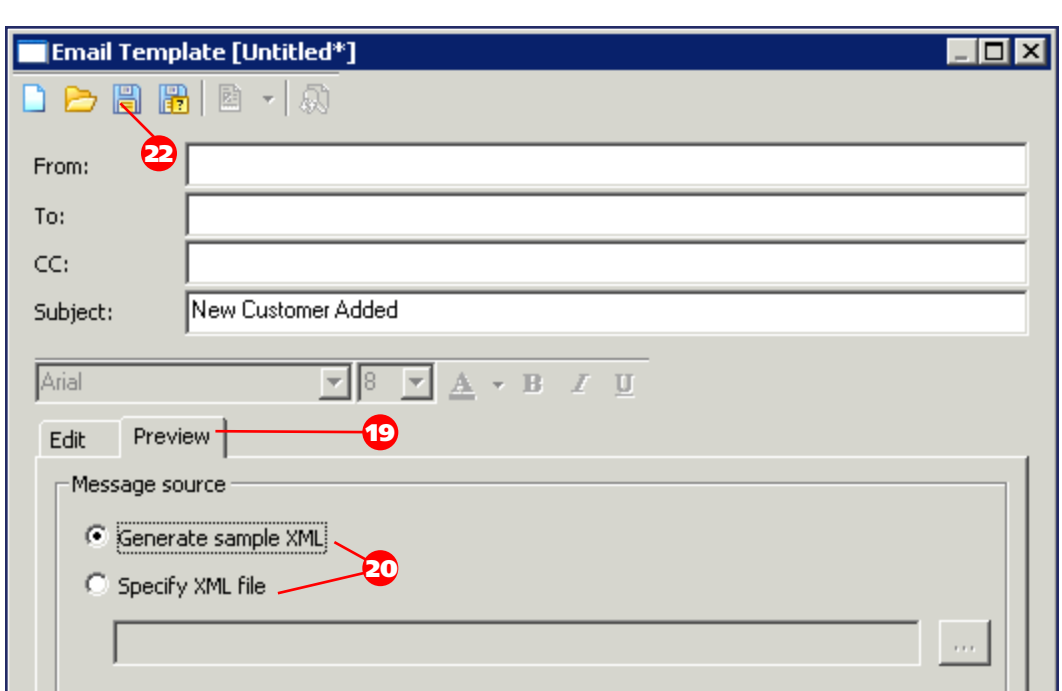
18. Click **OK**.



19. To preview the template as it displays in the email client application, click the **Preview** tab.

20. Select **Generate sample XML** to automatically generate sample XML or select **Specify XML file** to find and select an external XML file.

21. Click **Show preview** to view the results.



If you select the **Specify XML file** option, it is validated against the internal envelope schema. In case the provided XML does not conform to the internal envelope schema, an error is reported.
If you select the **Generate sample XML** option, the preview is generated with all process variables and message extensions defined in the current workflow.

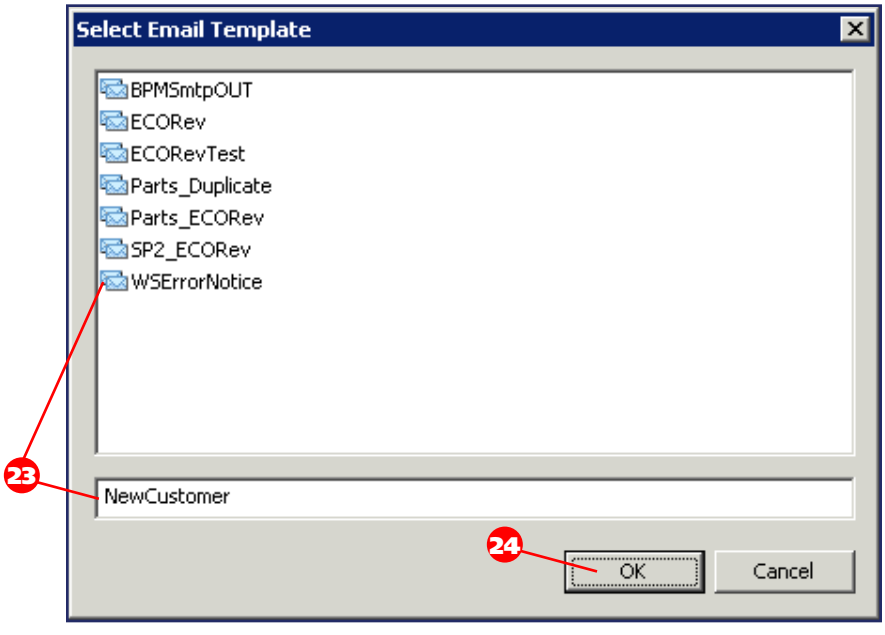
In message preview, all links are replaced with their values, extracted from the source XML provided. In HTML mode, any table links are rendered as tables as well. In XSLT editing modes, preview contains the result of the specified XSLT transformation applied to the source XML provided, either in plain text or HTML depending on editing mode selected. In case links or XSLT transformation processing results in error, empty document displays. The error message provides error description.

22. Continue to add information from the workflow to the Email Template and click **Save**.

You can add information from the workflow to the From, To, CC, and Subject fields.

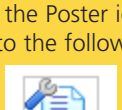
23. In the **Select Email Template** window, enter an email template name or select one from the list..

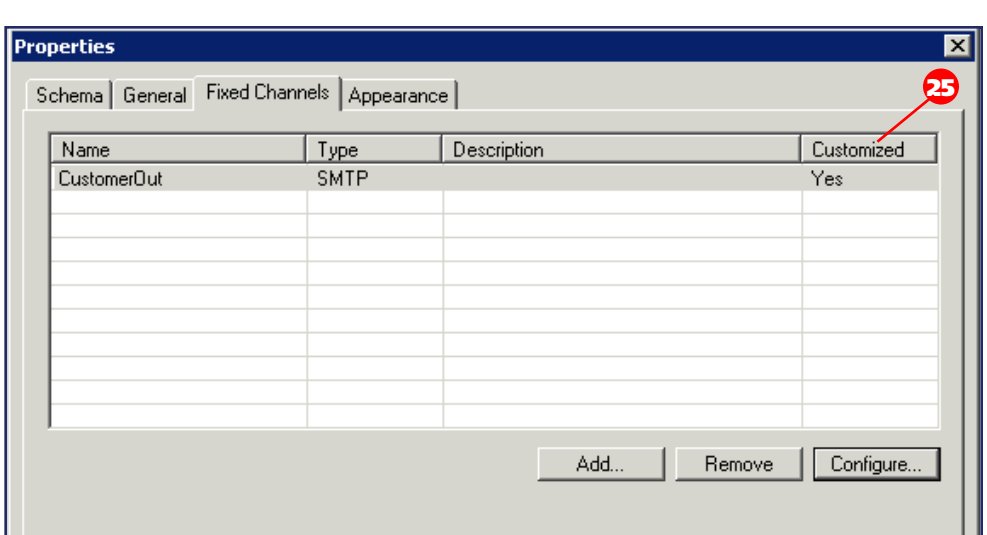
24. Click **OK** until you exit all dialog boxes.



25. Notice the Customized column on the Fixed Channels tab indicates the channel configuration was modified.

If you modify the channel configuration in the Workflow Designer, the Poster icon changes to the following:

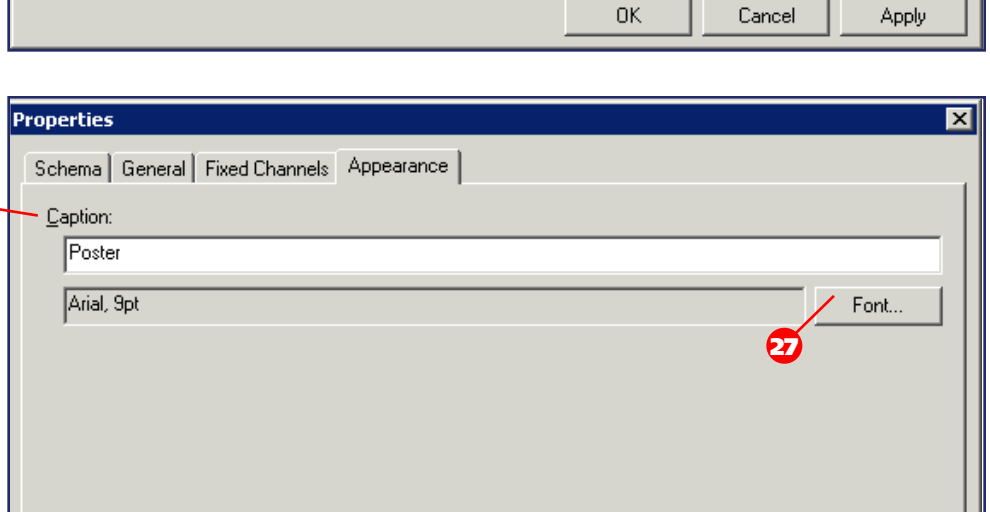




26. Click the **Appearance** tab to enter the **Caption**.

27. You can click the **Font** button to customize the font.

28. Click **OK**.



Sub-Workflow

Use this activity to call a workflow as a subroutine for another workflow. You can set the Sub-workflow to run asynchronously (the main workflow continues to execute) or synchronously (the main workflow pauses until the Sub- workflow finishes). When a Sub-workflow is set to execute synchronously, the results of the Sub-workflow will be available for use in the following activity of the main workflow.

In addition, you can set the Sub-workflow to execute once or to cycle through specific nodes in a document. For example, if a document contains a sales order, you can set up a Sub-workflow to cycle through each sales order line item. If necessary, you can also send the data stored in message extensions for use in the Sub-workflow. Review the Message Extensions section later in this chapter for more information about that feature.

You can use Sub-workflows to validate information sent into a workflow or to retrieve information required for future workflow operations. The following shows a Sub-workflow activity as part of a workflow.

Example

A master workflow calls another workflow as a subroutine which updates part records. Instead of attempting to update three part records at once, the Sub-workflow cycles through records and updates a single record at a time. Thus, if the incoming document has one bad row of data, the two good rows are updated and only the bad row fails.

An incoming document contains expenses that are logged against projects in a project management application. The document contains the project code and task name for each expense, but to log the expense against the correct project task, the system requires the task's universal identifier (TaskUID). A Sub-workflow can cycle through the expenses, retrieve the TaskUID for each expense, and return the TaskUIDs to the main workflow where they can be merged with the original expense information.

Set Up a Sub-Workflow

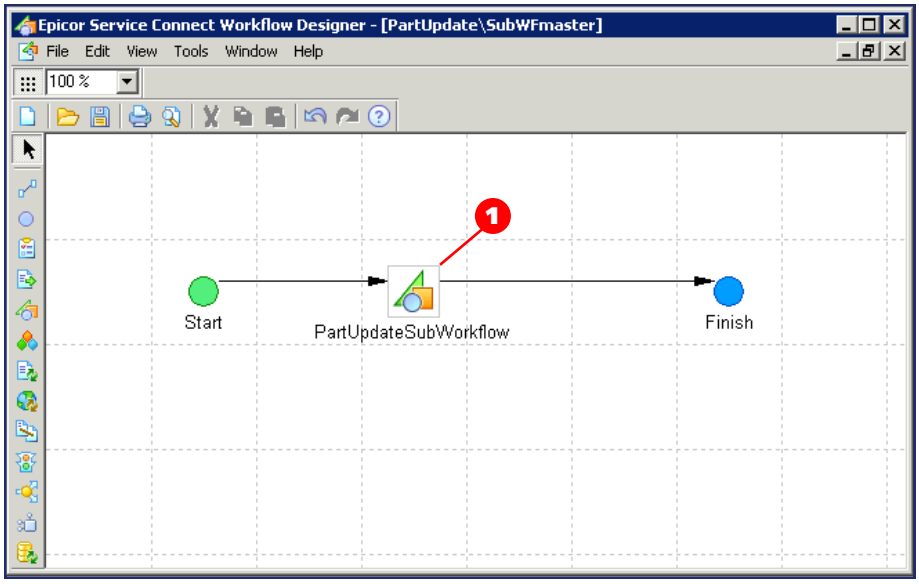
Sub-workflow setup involves the following tasks:

Create a workflow to serve as a subroutine to the main workflow. Usually, a Sub-workflow is saved to the same workflow package as the main workflow.

Add the Sub-workflow activity to the main workflow.

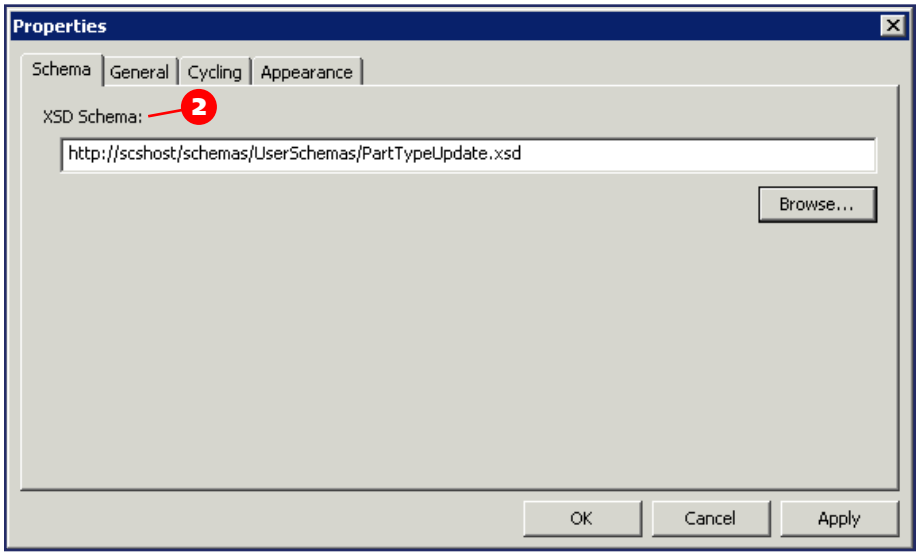
To set up a Sub-workflow activity:

1. In the **Workflow design area**, add the **Sub-workflow** activity.



2. The **Properties** window displays. Click the **Schema** tab to set the **XSD Schema** to the document that comes to the Sub-workflow element. You can send the document to the Sub-workflow fully or send only selected nodes.

   For Sub-workflows that cycle through nodes, the schema sets the document structure in the same way as cycling.

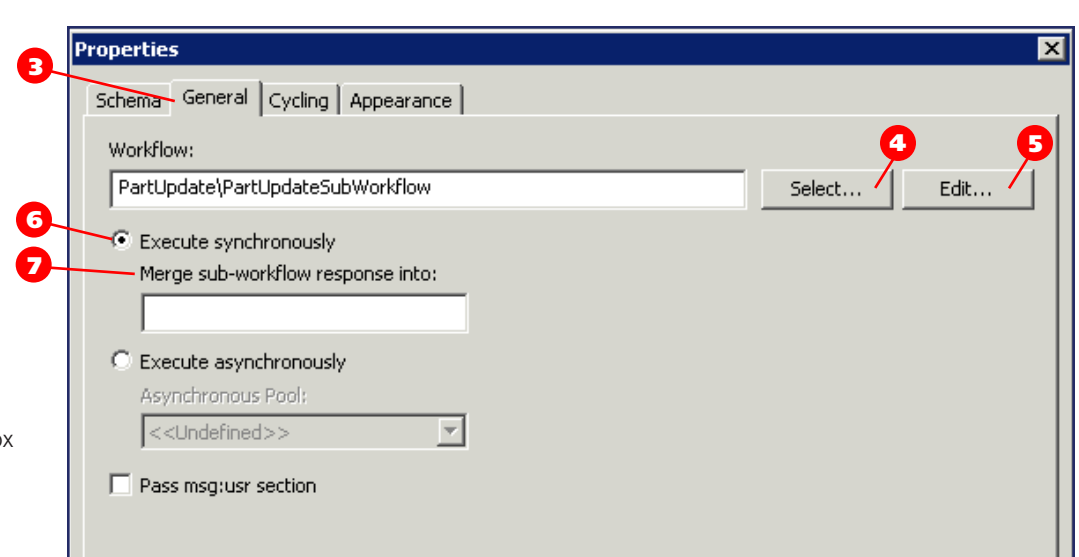


3. Click the **General** tab.

4. Click **Select** to choose a workflow to serve as the Sub-workflow.

5. You can click **Edit** to launch the second instance of Service Connect Workflow Designer and edit the selected workflow.

6. Select the **Execute synchronously** check box to pause the main workflow while the Sub-workflow executes.

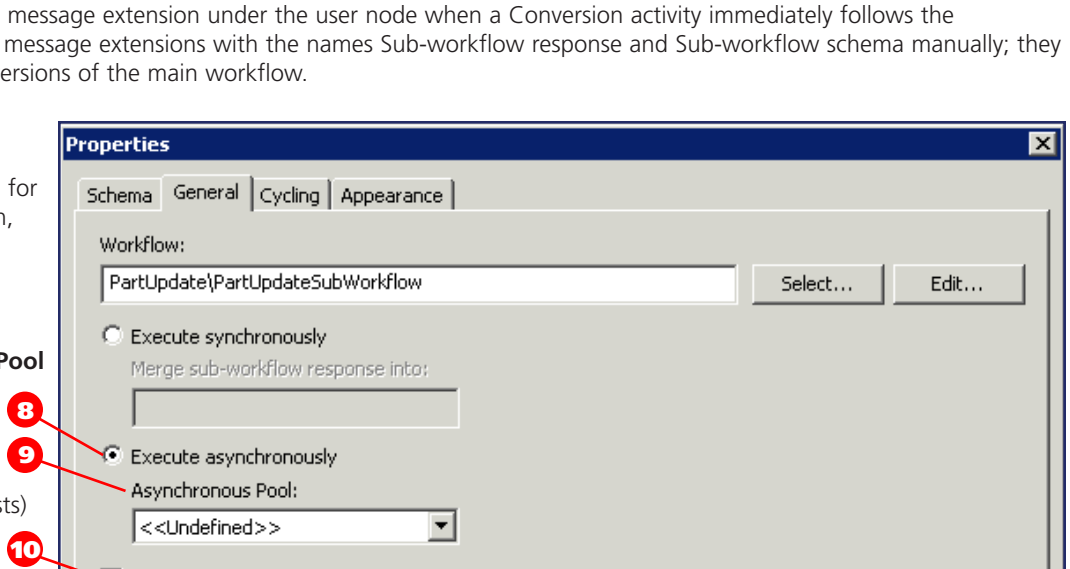


7. In the **Merge sub-workflow response into** field, specify the target message extension in which to merge the Sub-workflow responses.

   The name will display as a message extension under the user node when a Conversion activity immediately follows the Sub-workflow. Create the message extensions with the names Sub-workflow response and Sub-workflow schema manually; they will display in all the Conversions of the main workflow.

8. If you want the main workflow to continue execution without waiting for the Sub-workflow to finish, select the **Execute asynchronously** option.

9. You can select the pool from the **Asynchronous Pool** drop-down list.



   Use an asynchronous pool to handle incoming asynchronous calls (requests) for message processing. It is defined with the pool name and thread number. The thread number can have a value from 1 to 20. The number of active threads cannot be more than the number of available threads. The thread number limits the maximum number of concurrently executing processes that receive messages from asynchronous channels or asynchronous workflows.

   You should ideally run Service Connect from its own server but the size of the data (xml) being processed is the real factor. The amount of physical memory on the server must be able to handle the data size being processed. The equation of memory is:

   XML data size x 8 x number of thread count = physical memory needed

   Service Connect internal overhead increases when the thread number increases. For example, if 10 threads give 10 seconds for one request, 20 threads could get 20 seconds for the same request. In general, this means you get the same performance with less server overload.

For example, you may process a message from an asynchronous channel as several requests. A message from a simple Start-Finish workflow is processed as one request. For workflows with asynchronous Sub-workflow elements, each Sub-workflow call creates an additional request. The Task element always divides workflows into two requests.

You should inherit the pool property for Sub-workflows in the following order:

- From the configuration of a Sub-workflow element of the main workflow
- From the main workflow in case of a Sub-workflow

If an asynchronous Sub-workflow element is not configured to use a particular Pool, then it should be executed with the same Pool as the parent workflow.

- From workflow configuration
- From the default, which should be the System Pool

Asynchronous pools are set up in the Epicor Service Connect Administration Console. Review Chapter 3: Connectivity Components for more information.

10. If you want to store information in message extensions and process variables to be available in the Sub-workflow, select the **Pass msg:usr section** check box.

For more information about message extensions and process variables, review the Message Extension and the Process Variables sections at the end of this chapter. If message extensions contain a significant amount of data, passing that data to the Sub-workflow can affect performance.

11. Click the **Cycling** tab.

Use this tab when a document contains more than one record and you want the Sub-workflow to process one record at a time.

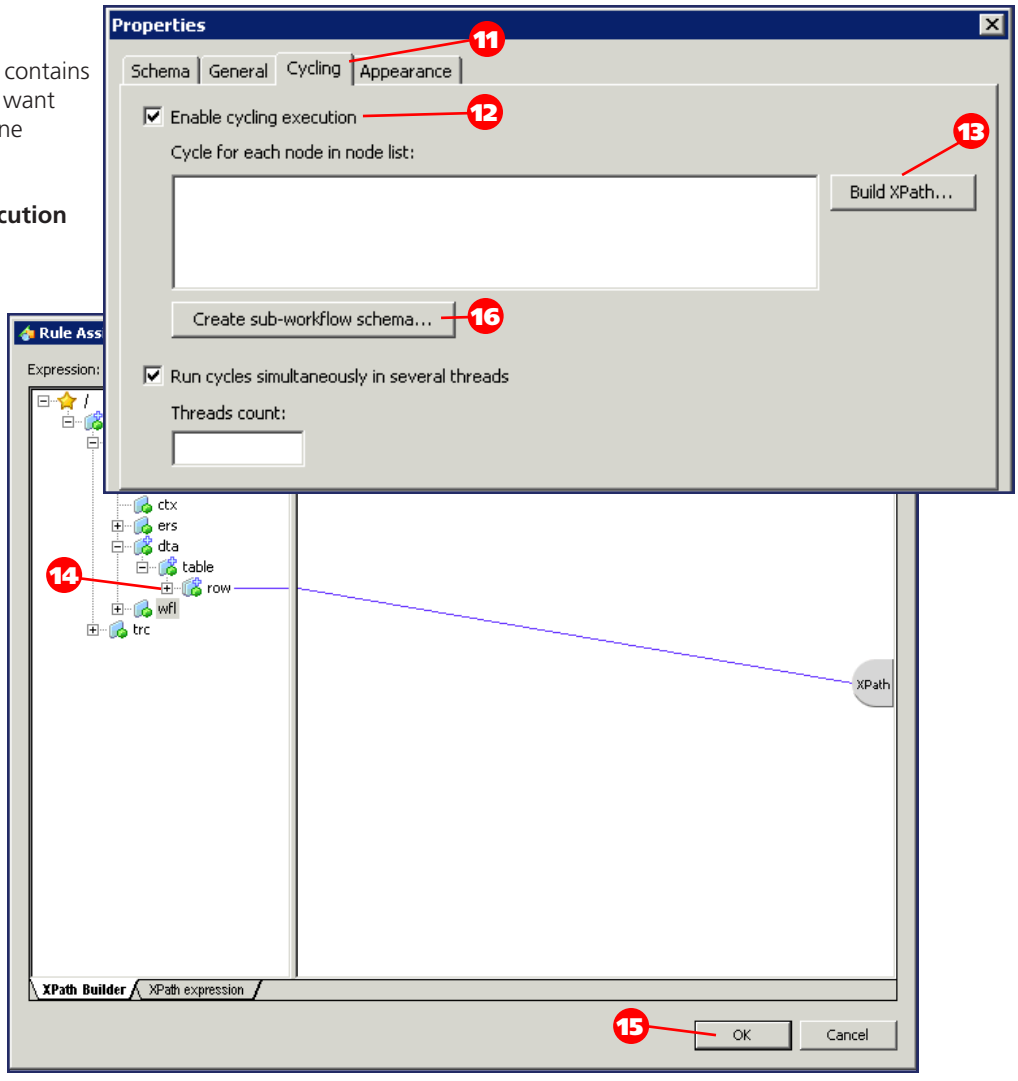
12. Select the **Enable cycling execution** check box.

13. Click **Build XPath** to select the part of the document on which the Sub-workflow will cycle.

14. The **Rule Assistant** displays. Expand the nodes until you locate the collection node to use for cycling.

In this example, a user schema is selected for the Sub-workflow and the row node is used for cycling execution.

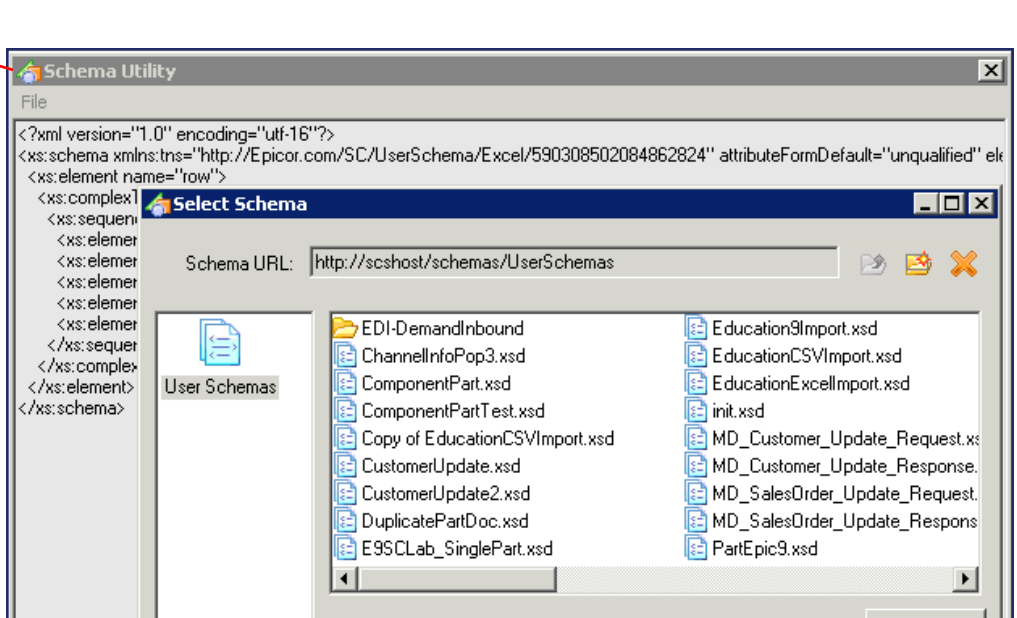
15. Click **OK**.



16. In the **Properties** window, click **Create sub-workflow schema** to create a schema for the Sub-workflow.

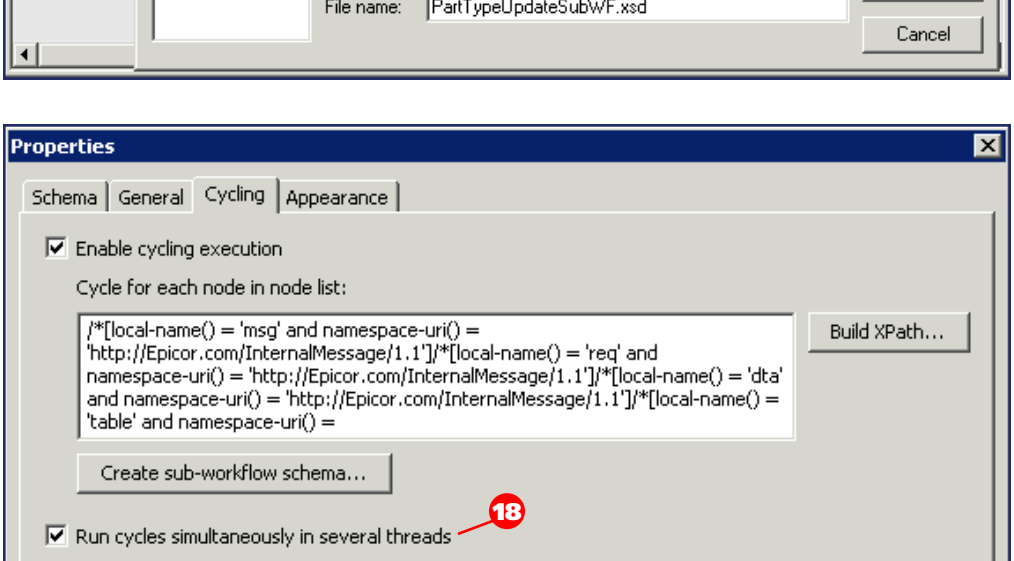
This opens the Schema Utility, which builds a schema based on the cycling node. The schema will be used in a Conversion activity of the Sub-workflow to specify the document format sent to the Sub-workflow. A schema is not always required for cycling execution; it is possible to use the schema for a Web Method request in a Sub-workflow activity or to use only an XPath expression.

17. The **Schema Utility** window displays. Use the Schema Utility to import the schema into Service Connect. Review the Schema Utility section later in this chapter for more information.



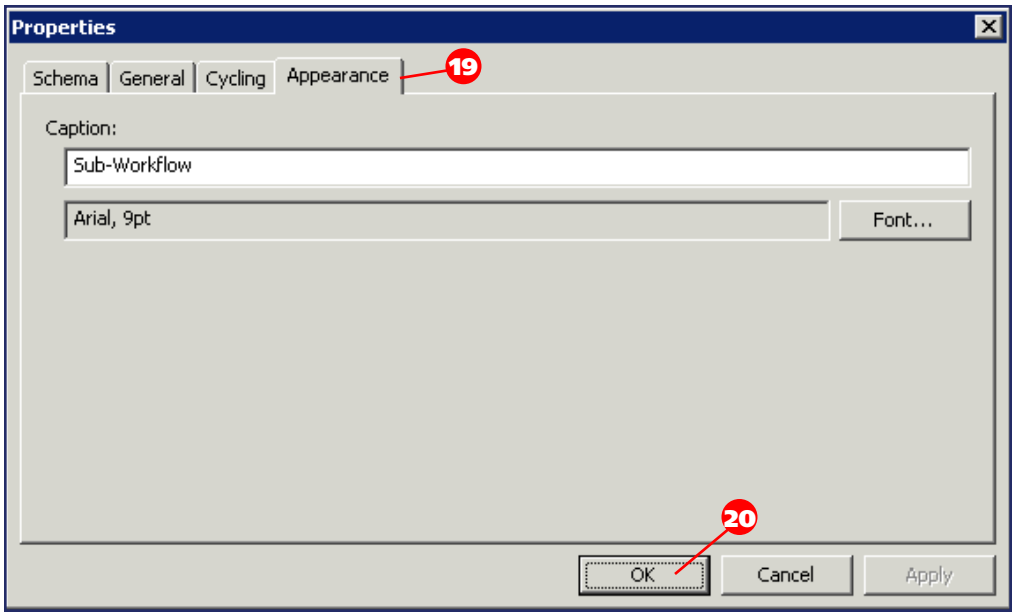
18. You can select the **Run cycles simultaneously in several threads** check box and specify the number of threads.

Use this option when you need Sub-workflows to run in parallel threads. This option works only if you select **Execute synchronously** on the **General** tab. System administrators can elect whether to use this feature as it depends on the underlying business process and hardware configuration.



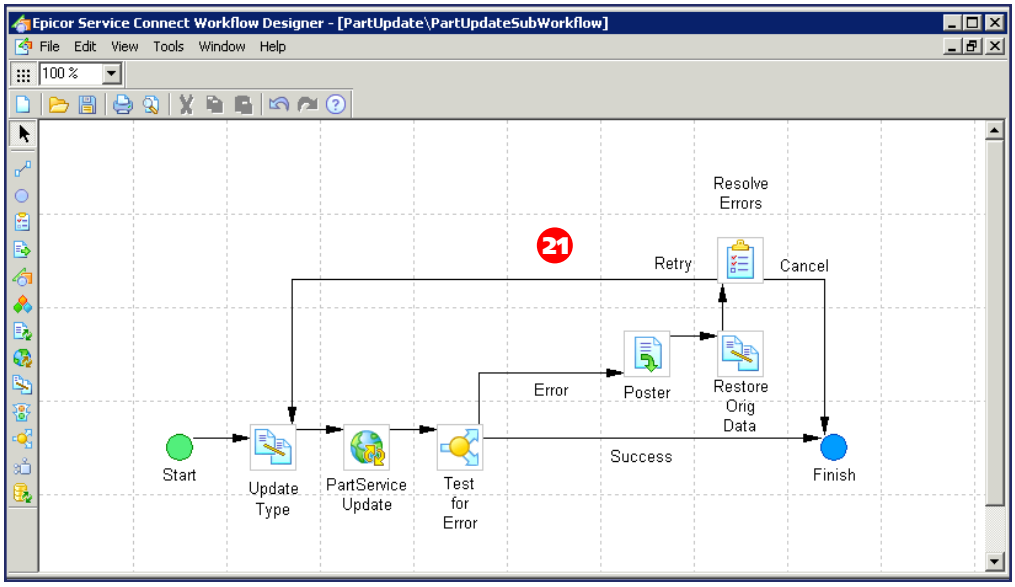
19. Click the **Appearance** tab to enter the **Caption**. If you want to customize the font, click **Font**.

20. Click **OK**.



21. Add **activities** to the workflow to use as the Sub-workflow.

In this example, the Sub-workflow uses two activities: a Conversion and a Web Method to update the part record. The Choice activity that follows verifies record accuracy. If a failure occurs, the Poster activity sends an email to a selected user to announce that the task is waiting in the Task Monitor. The original data is restored; a user can then add the missing data and restart the update process or cancel the operation.



You have the ability to break out of the Sub-workflow process based on an exception raised in one of the looping processes without having to complete all the process loops. Review the Web Method section later in this chapter for more information on response processing functionality.

Windows Workflow Foundation® Call

Use this activity to call a Windows Workflow Foundation (WF) workflow as a subroutine for a Service Connect workflow. WF is a Microsoft technology for defining, executing, and managing workflows. It is part of the .NET Framework 3.0 which is available natively in the Windows Vista® operating system and can be installed on Windows XP SP2, Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2. For details about WF, refer to the Microsoft website (http://msdn.microsoft.com).

Windows Workflow Foundation is the programming model, engine and tools for quickly building workflow enabled applications on Windows. It consists of in-process workflow engine, and designers for **Visual Studio 2010**. WF includes support for both system workflow and human workflow across a wide range of scenarios including: workflow within line of business applications, user interface page-flow, document-centric workflow, human workflow, composite workflow for service oriented applications, business rule driven workflow and workflow for systems management

Important for Windows WF Development on Windows Vista or Windows XP: When you develop Windows WF workflows on Windows XP or Windows Vista machines, you must copy Epicor.ServiceConnect.Core.dll from the Service Connect machine to the local machine. You can find the library at <Service Connect installation folder>\System\Services\DES.

Set Up the Windows WF Integration

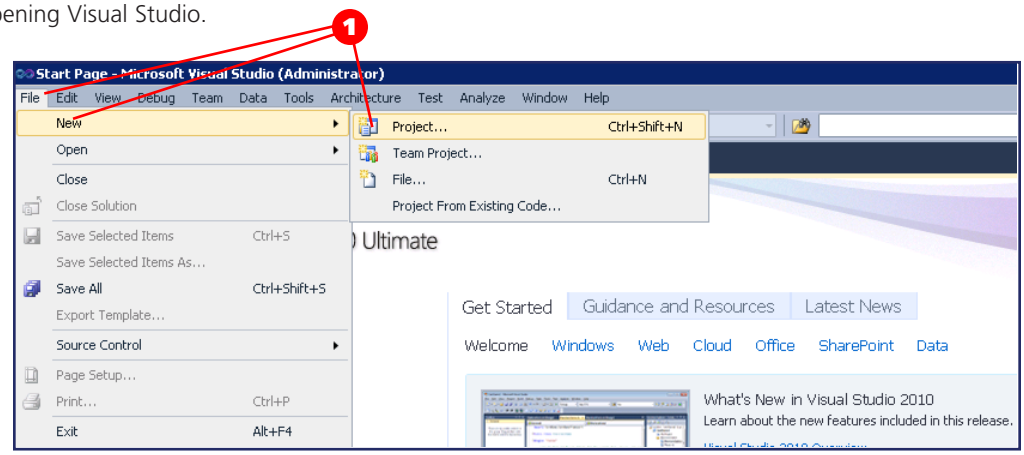
To use Windows WF with Service Connect:

1. Create a Project in Visual Studio.
2. Add Service Connect activities to the Visual Studio Toolbox.
3. Register the WF workflow in Service Connect.
4. Call the WF workflow from a Service Connect workflow.

Create a Project in Visual Studio

Perform the following steps after opening Visual Studio.

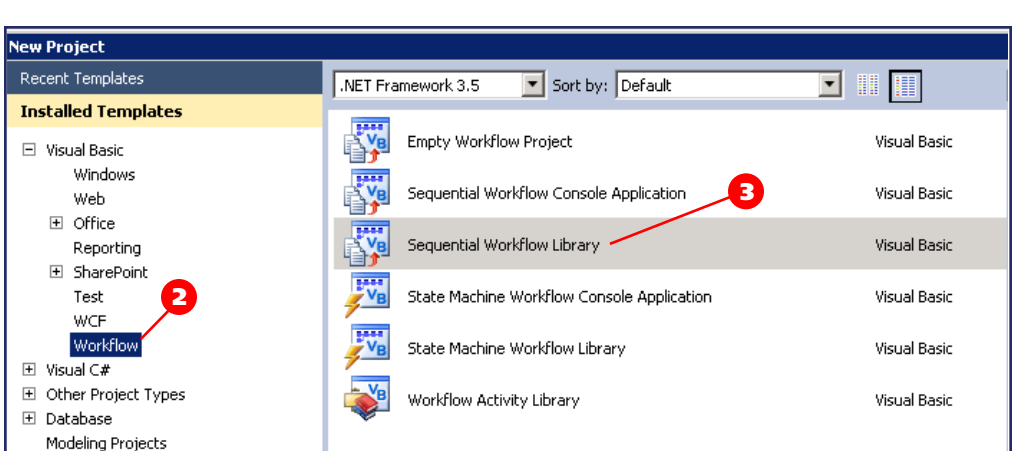
1. From the **File** menu, select **New > Project**



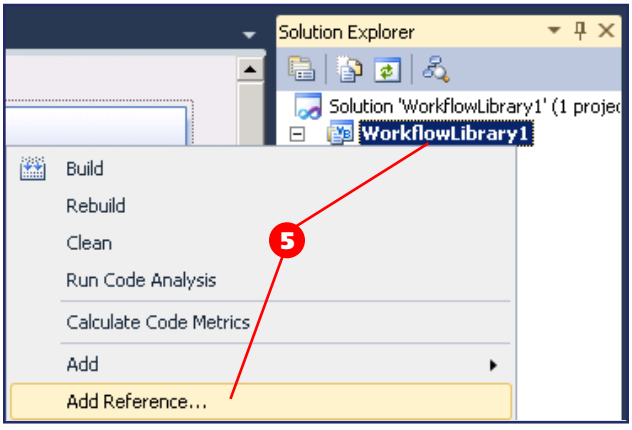
2. The **New Project** window displays. In the **Tree View**, expand the Visual Basic node and select **Workflow**.

3. In the right pane, select **Sequential Workflow Library**.

4. Click **OK**.



5. In the Solution Explorer, Add Reference to the *Epicor.ServiceConnect.Core.dll* core library. In the Solution Explorer, right-click the **WorkflowLibrary1** and select **Add Reference**.



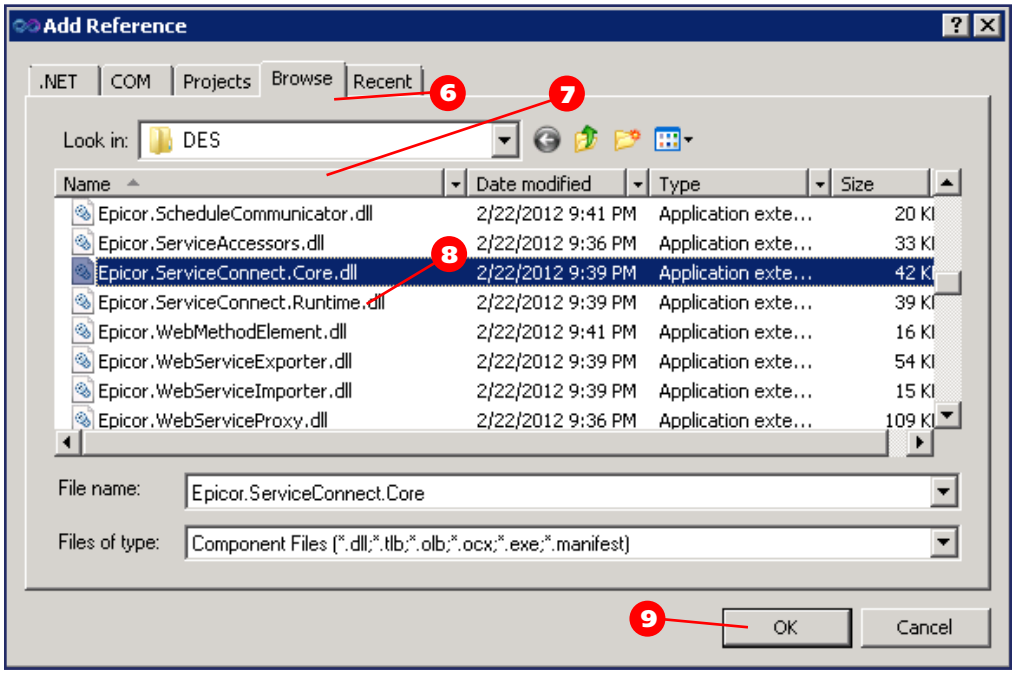
6. The **Add Reference** window displays. Click the **Browse** tab.

7. Navigate to **<Service Connect installation folder>\System\Services\DES**.

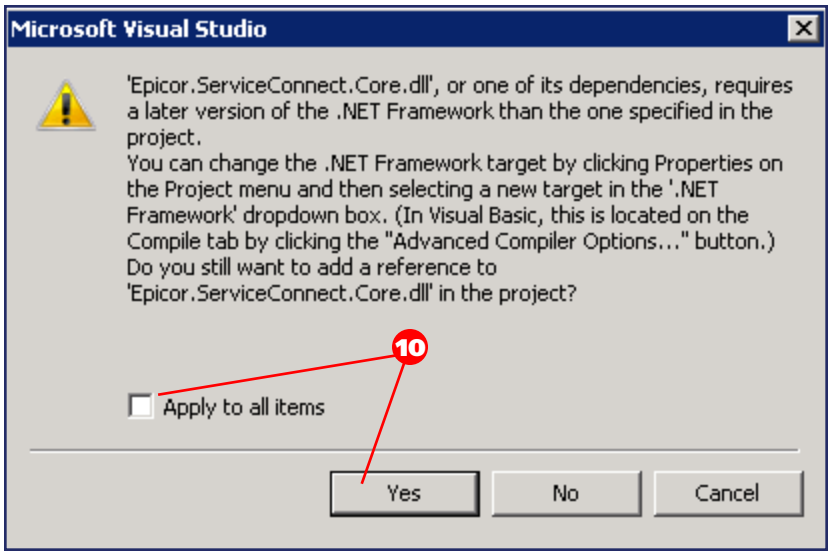
8. Select **Epicor.Service Connect.Core.dll.**

9. Click **OK**.

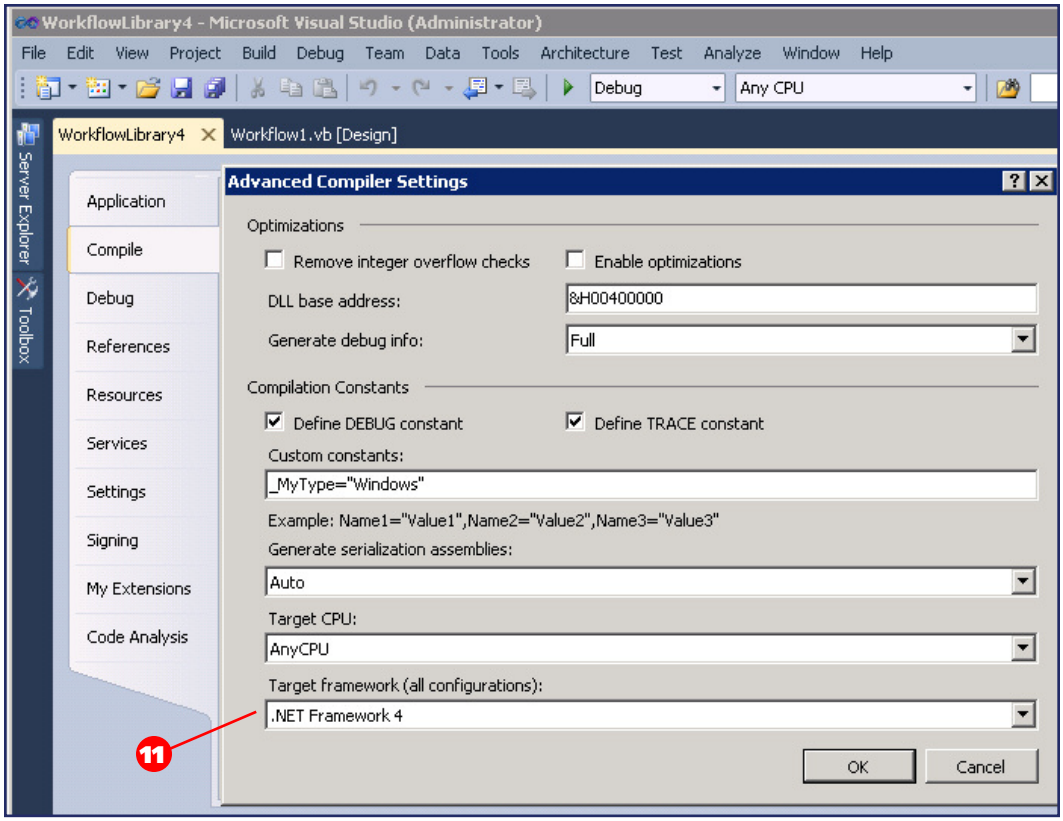
Epicor.ServiceConnect.Core.dll must be on the same computer as Visual Studio.



10. A warning message displays. Select the **Apply to all items** option and click **Yes**.

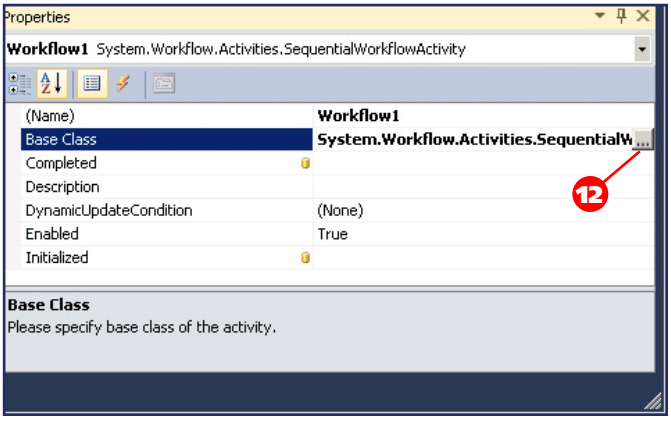


11. Select the **Compile > Advanced Compiler Settings** tab to change the target framework. Click the **Target framework (all configurations)** drop-down list and select **.NET Framework 4**.



12. Click a blank area in the workflow design area and under **Properties**, click the **... (Ellipse)** button in the **Base Class** field.

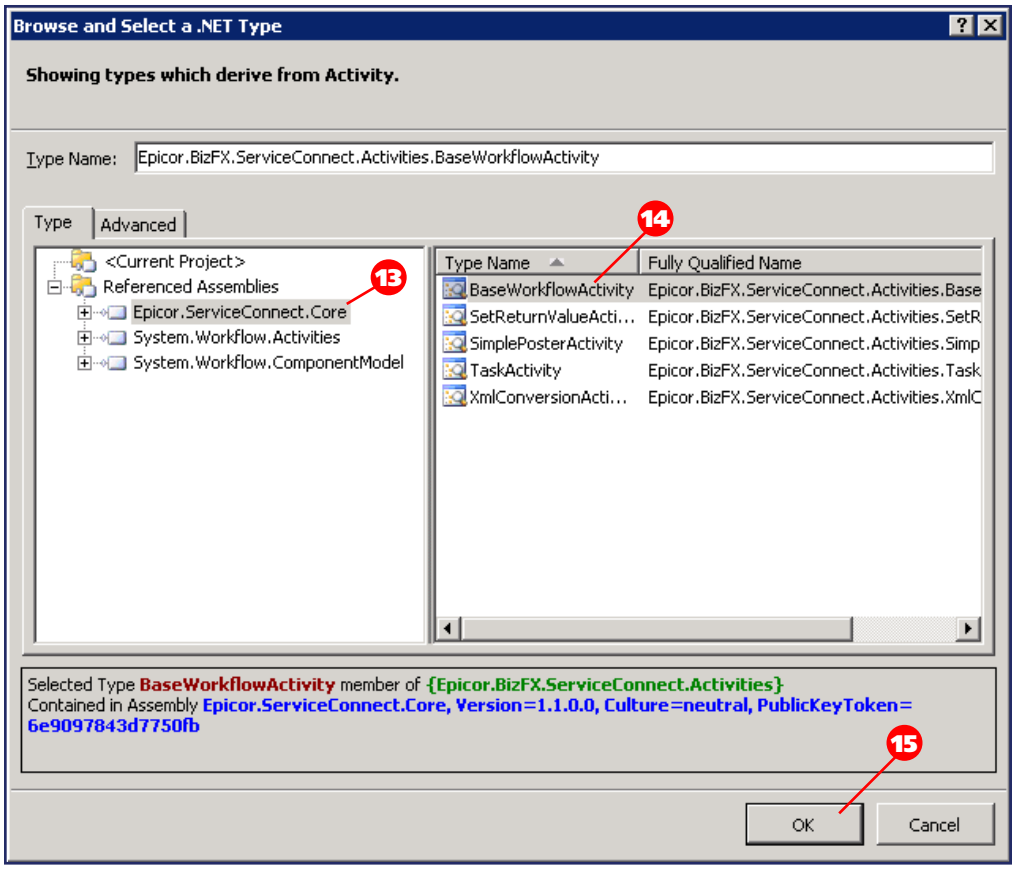
You may have to click the Base Class field to display the **... (Ellipse)** button.



13. The **Browse and Select a .NET Type** window displays. In the **Tree View**, select **Epicor.ServiceConnect.Core**.

14. In the **Type Name** column, select **BaseWorkflowActivity**.

15. Click **OK**.



Two new properties display in the properties sheet:

- **Request Header** – This corresponds to the ctx and cfg sections of the internal message envelope.
- **Request Body** – This corresponds to the dta section if the internal message envelope.

You can now begin to develop the Windows WF workflow.

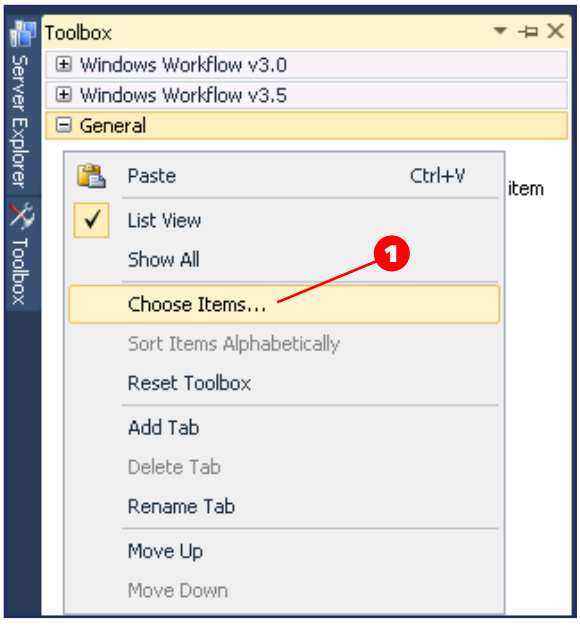
Review the Internal Message Envelope section in Chapter 1: Epicor Service Connect Overview for more information.

Add Service Connect Activities to the Visual Studio Toolbox

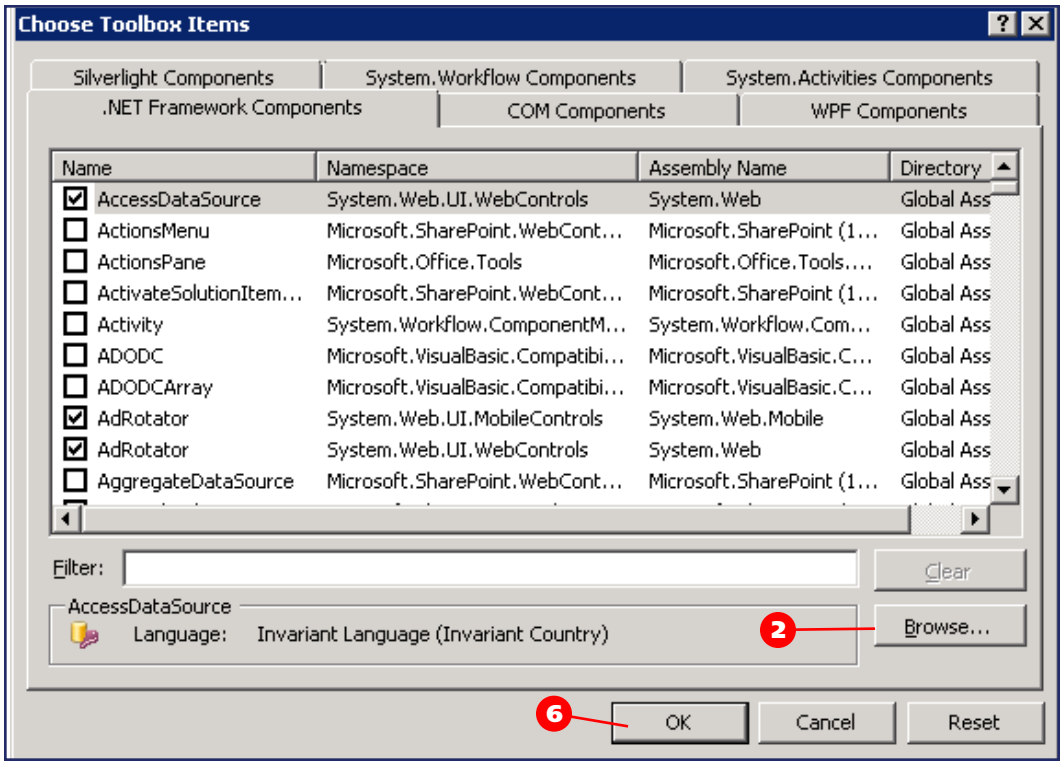
You can add several Service Connect activities to the Visual Studio Toolbox and can insert the activities into a WF workflow.

To add Service Connect activities to the Toolbox:

1. In the **Visual Studio Toolbox**, right-click an open area and select **Choose Items**.



2. The **Choose Toolbox Items** window displays. Click **Browse**.

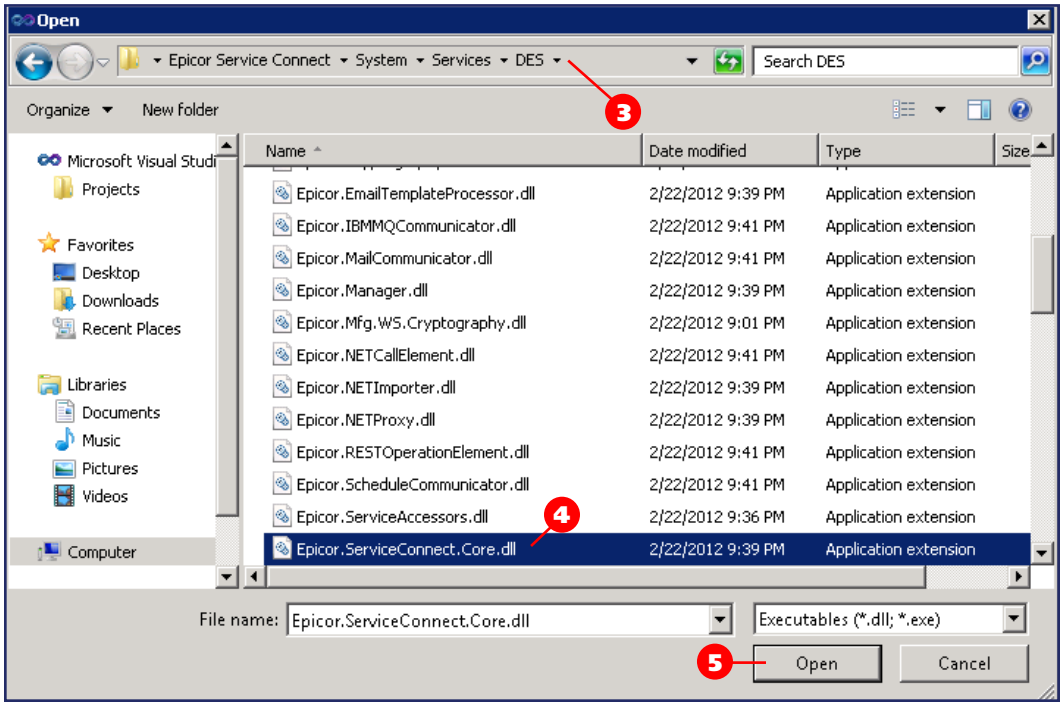


3. The **Open** window displays. Browse to <Service Connect installation folder>\System\Services\DES.

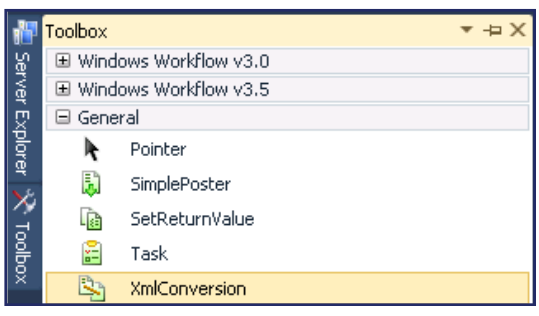
4. Select **Epicor.Service Connect.Core.dll.**

5. Click **Open**.

6. In the **Choose Toolbox Items** window, click **OK**.



The Service Connect workflow activities display in the Visual Studio Toolbox. With these activities, you can now use Service Connect items, such as channels or tasks, from inside the Windows WF workflow. The following describes each activity as it functions in a Windows WF workflow.



The **Task** is similar to a Service Connect Task but has the following properties:

- **Input Document** – This property is bound to the dta section of the internal envelope. You cannot leave this property blank.
- **Output Document** – This property can be left blank or can be set to the same value as the Input Document property in order to use the input document further in the WF workflow.
- **User List** – This property is the list of users or user groups who have the right to execute the task. The User List property is specified in XML format. By default, the user list includes all the users registered in the Administration Console under the Root node.
- **Exit Configuration** – This property provides the list of choices. Each choice corresponds to a task exit option. After task processing, the value of the task exit option is assigned to the Exit Result property. Later, you can use the value of the Exit Result property as a branching condition.

The **SimplePoster** activity is used to post XML messages during workflow execution. The SimplePoster activity is similar to the Service Connect Poster, but unlike the Service Connect Poster, the SimplePoster cannot send information to multiple channels. The SimplePoster activity has the following properties:

- **ChannelName** – This property specifies the channel name. Unlike the Service Connect Poster, the SimplePoster does not allow you to configure its channel; it uses the channel parameters set up in the Service Connect Administration Console for the corresponding channel.
- **Document** – The XML document, which is bound to this property, is mapped to the dta section of the SimplePoster output message.
- **MessageType** and **MessageAction** – These properties specify the outgoing message type and action.

The **XmlConversion** tool provides only predefined conversions for WF workflows. Windows WF does not provide a design-time user interface like the Workflow Designer XML Mapper; however, you can use the Service Connect XML Mapper to create an XSLT file, save the XSLT file to the local machine, and refer to the XSLT file from the XmlConversion WF activity.The XmlConversion element has the following properties:

- **Input Document** – This property is mapped to the dta section of the internal envelope. You cannot leave this property blank.
- **Output Document** – This property can be left blank or can be set to the same value as the Input Document property in order to use the input document further in the WF workflow.
- **Transform** – This property provides the path to the conversion file. This path is absolute; relative paths are not supported.

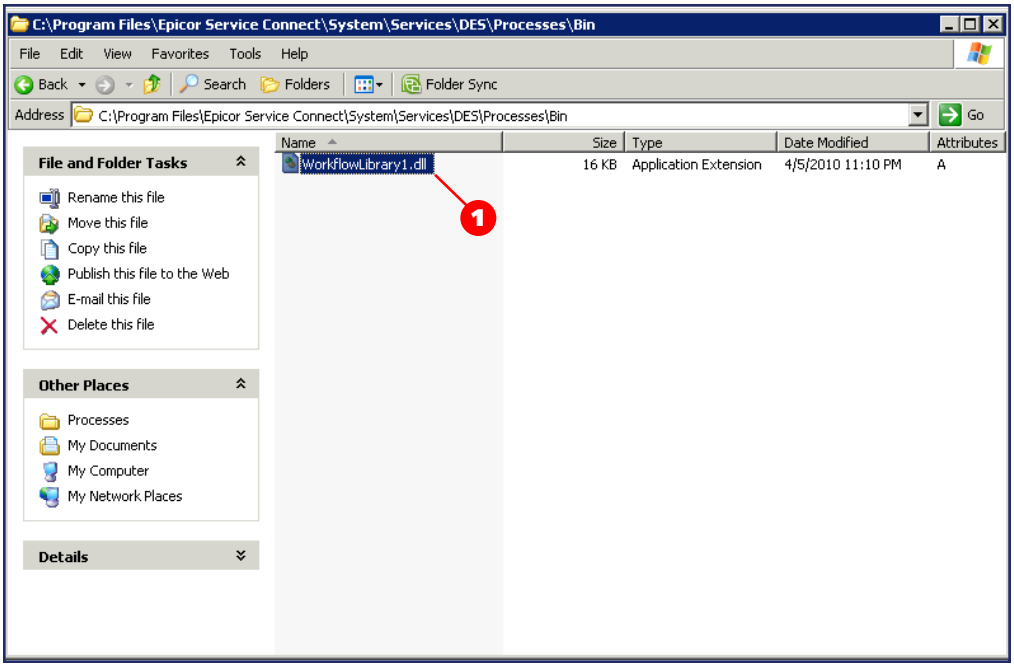
The paths to the conversion file should be the same on the server and on the design machine - the machine on which the project is created in Visual Studio.

The **SetReturnValue** tool is intended for specifying the return value of the Windows WF workflow. This value is then passed to the Service Connect workflow. The WF activity return value is bound to the Return Value property. You can specify the return value on the Windows WF call in the Service Connect workflow so the value(s) the Windows WF workflow returns are available in Service Connect.

Register the WF Workflow in Service Connect

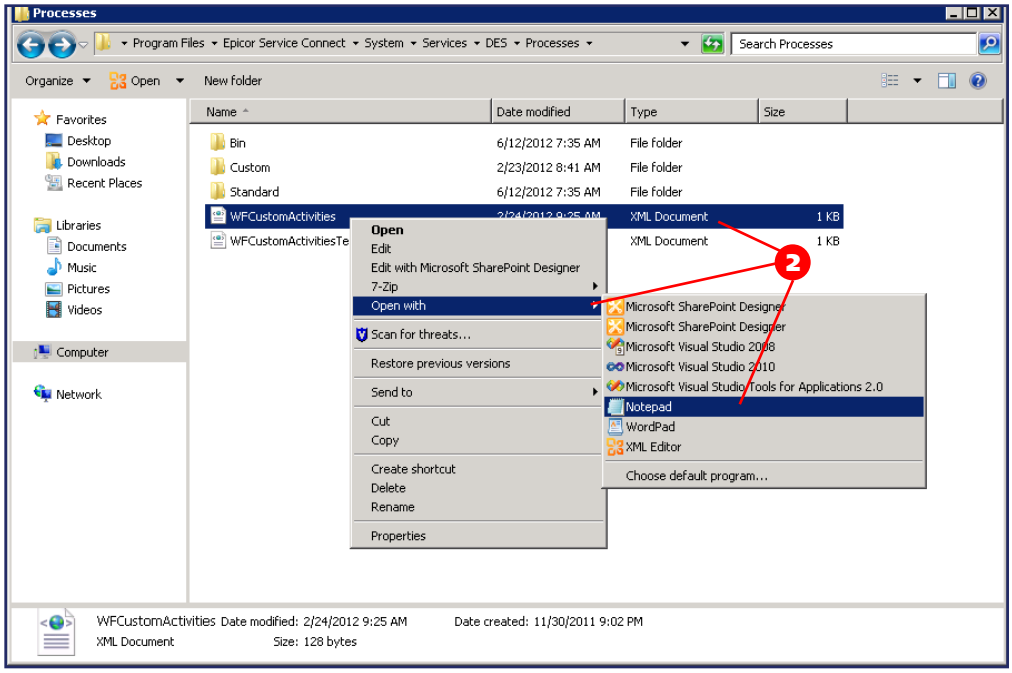
Register the WF workflow to make it available from within a Service Connect workflow. Perform this procedure after you compile the Windows WF project in Visual Studio.

1. Copy the compiled WF workflow, a DLL file, to the following folder on the Service Connect Server: **<Service Connect installation folder>\System\Services\DES\bin**.



2. From the **Processes** folder, open the **WFCustom Activities.xml** file in a text editor.

3. Modify the file so the Package Name is the name of the Windows WF DLL file without the .dll extension. Do this so the Workflow name is one of the workflows in the project.

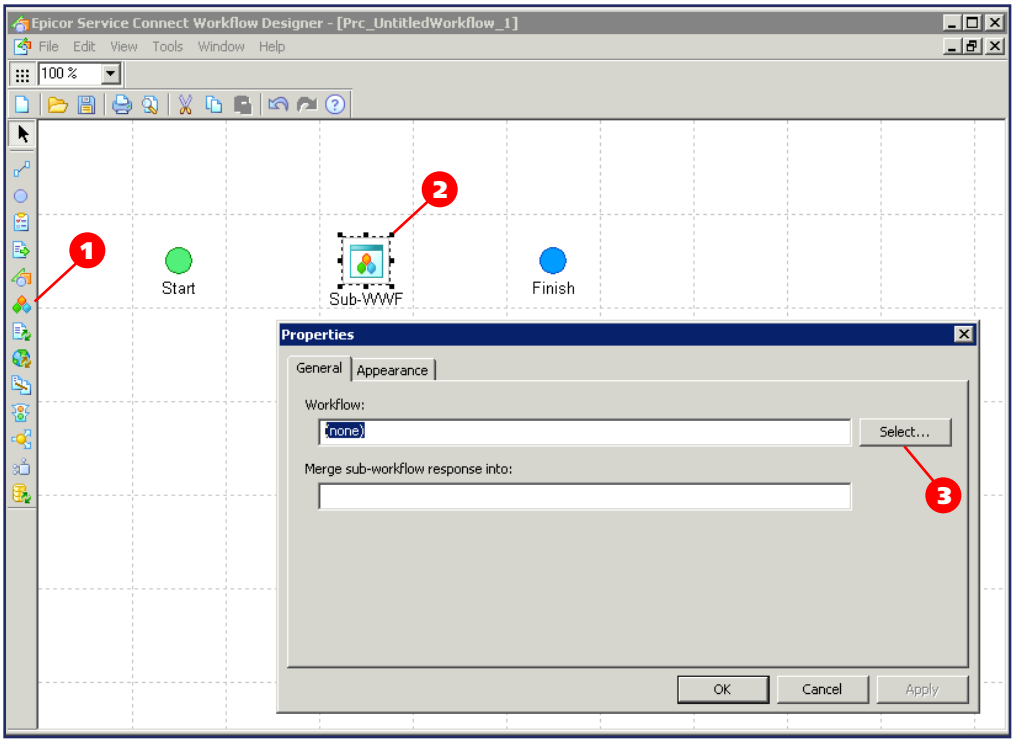



```
<Processes>
<Package Name =
"Workflow
Library1">
<Workflow Name =
"Workflow1" />
</Package>
</Processes>
```

4. **Save** and **close** the file.

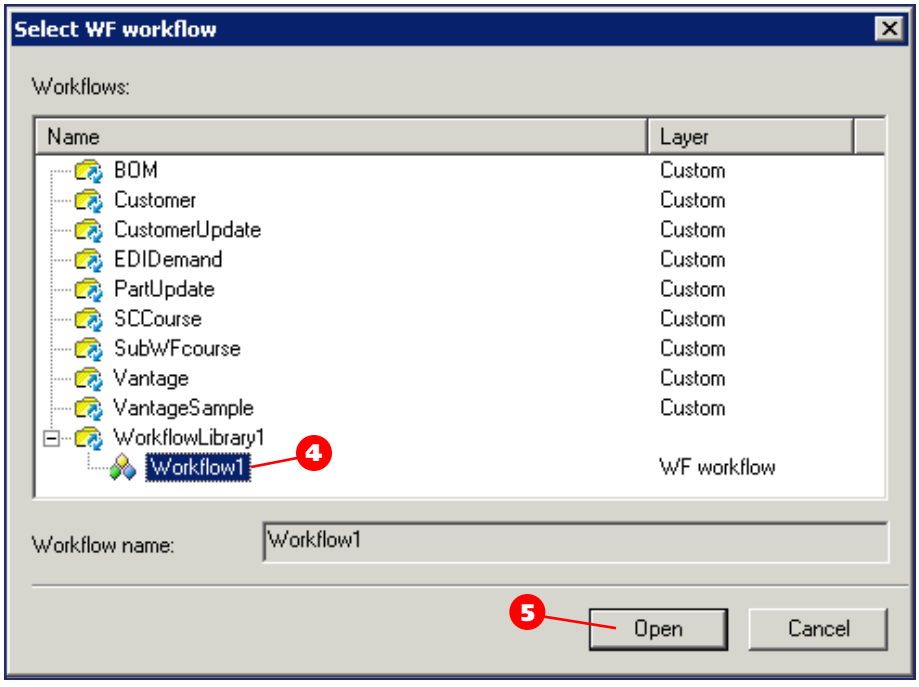
Call the WF Workflow from a Service Connect Workflow

1. In a Service Connect workflow, select the **Windows WF call** button in the **Items** toolbar.
2. Click an open area of the workflow diagram.
3. The **Properties** window displays. Click **Select** to choose the Windows WF workflow you want to call.

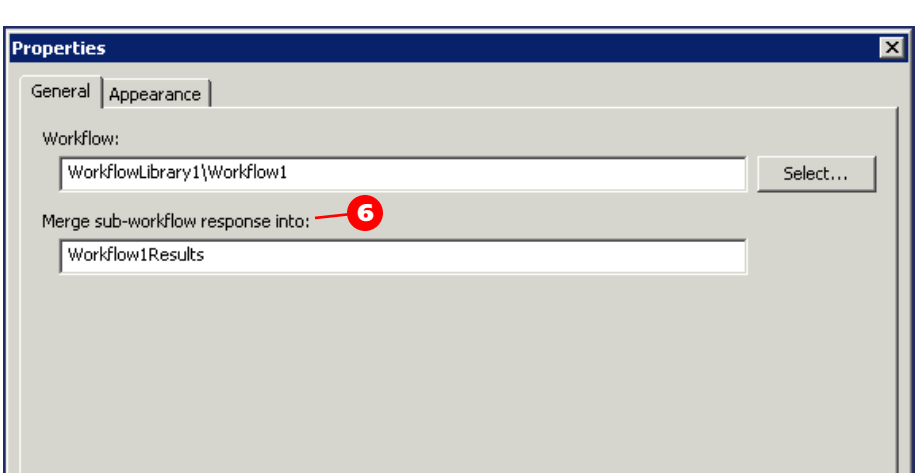


4. The **Select WF workflow** window displays. Select a workflow.
5. Click **Open**.

   If the Windows WF workflow does not display in the Workflow list, close the dialog box and try again.



6. In the **Properties** window, enter a **Merge sub-workflow response into** value to create a temporary message extension that will hold the returned value(s) from the Windows WF workflow. Use this field if you are using the SetReturnValue activity in the Windows WF workflow.



The message extension will display beneath the wfl/usr node in the first Conversion activity that follows the Windows WF call.

If you want to use the Sub-workflow call results later in the main workflow, create a message extension based on the structure of the results you expect. Then, you can enter the message extension name in this field. Review the Message Extensions section later in this chapter for more information.

Requester

Use this activity to support request or response communication with an external system during workflow processing. The external system is usually another application, but it can also be another workflow. The Requester forwards the incoming message to output communication channels in the same way as a Poster and then waits for a response. The response returns to the workflow through an input channel and is routed to the Correlation Manager according to its message map settings. When the response is received or the timeout expires, the suspended workflow resumes.

A Requester relies on two components: the Requester activity and the Correlation Manager. The Requester posts a message similarly to a Poster and adds attributes to the message that identify the Requester and workflow that posted the message. When a response is returned to the system, the input channel that receives the message is not aware that it requires special handling. To route the message back to the waiting Requester, the message is sent to the Correlation Manager. The Correlation Manager analyzes the attributes the Requester added and resumes the workflow from the Requester activity.

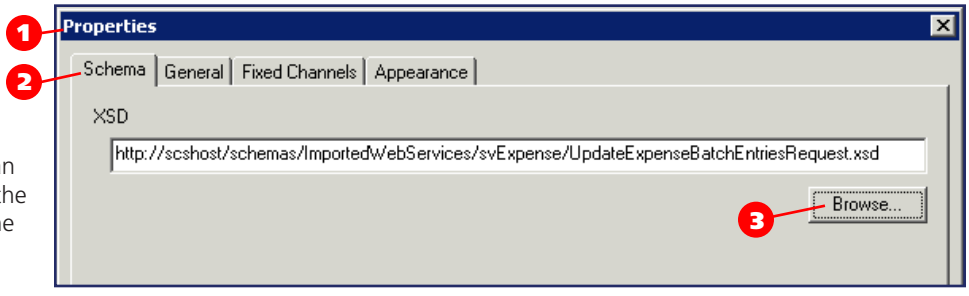
Example

A document that enters a workflow is routed to a Requester, which forwards the document to a COM output channel. The message sent to the output channel contains message attributes that indicate the message is being sent from the Requester. The Correlation Manager uses these attributes to identify the suspended workflow and resume it from the Requester activity. Message Type attributes are configured to forward the message to another workflow, which processes and eventually routes the document to a Task. After a user takes action on the document, it is sent to a Poster, which sends the document back to the same COM output channel. The returning message contains different Message Type attributes, which prevent the document from being processed again by the second workflow. When the document is posted back, both workflows are complete.

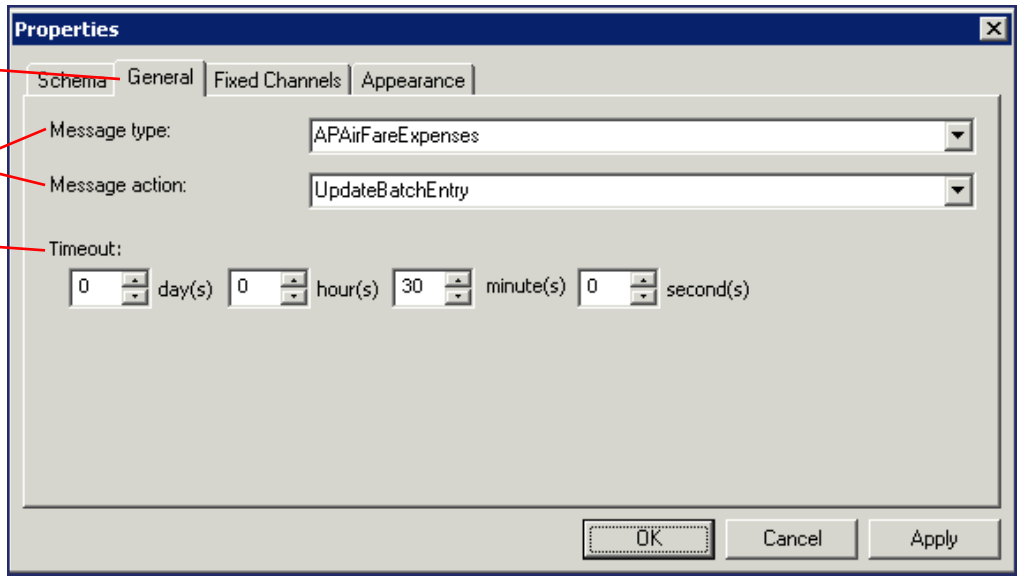
Set Up a Requester

To set up a Requester:

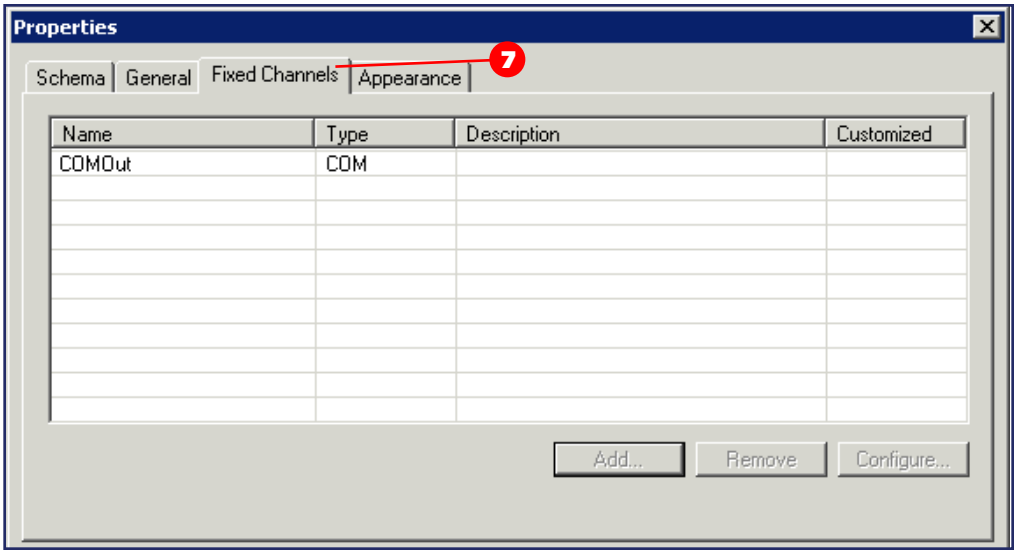
1. Double-click the **Requester** to open the **Properties** window.
2. Click the **Schema** tab.
3. Click the **Browse** button to select an **XSD** schema if you need to define the format of the document entering the Requester.



4. Click the **General** tab.
5. Select the **Message type** and **Message action**.
6. Use the **Timeout** fields to set the time period. The workflow will resume if no response is received after the timeout expires.



7. Click the **Fixed Channels** tab to assign one or more output channels.



If you modify the output channel configuration previously set up in the ESC Administration Console, note the customized Requester icon displays instead of the default one:

Conversion

Use this activity to convert a document from one format to another and to store values returned from other activities in containers or variables. The Conversion activity contains an Input Schema - the format of incoming documents - and an Output Schema - the format of documents that will be passed to the outbound Connection. After you set the Input and Output Schemas, you can create the conversion using the XML Mapper, a visual interface for mapping elements from the Input Schema to the Output Schema. The mapping is saved as an XSLT file (eXtensible Stylesheet Language Transformations). Use the XML Mapper to view the XSLT source and edit it if necessary. For more information about the XML Mapper, review the XML Mapper section later in this chapter.

Example

A document that contains a bill of materials (BOM) enters the workflow. The assembly part in the BOM may be a new part or an existing part. You can use a conversion to set up a Web Method call to the PartExists web service method, which will verify whether to add the part to the database or to create a revision for the existing part.

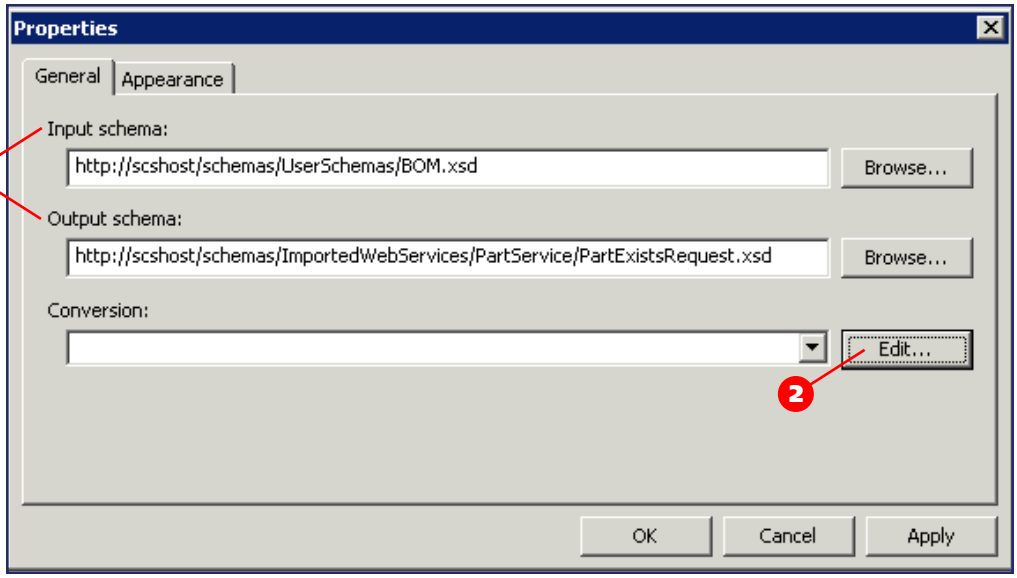
Set Up a Conversion

To set up a Conversion activity:

1. In the **Properties** window, select the **Input schema** and **Output schema** to indicate the structure of incoming and target documents.

   In this example, the Input schema is the schema for the bill of materials document entering the workflow. The Output schema is the request schema for the Web Method activity that will follow the conversion.

2. Click **Edit**.



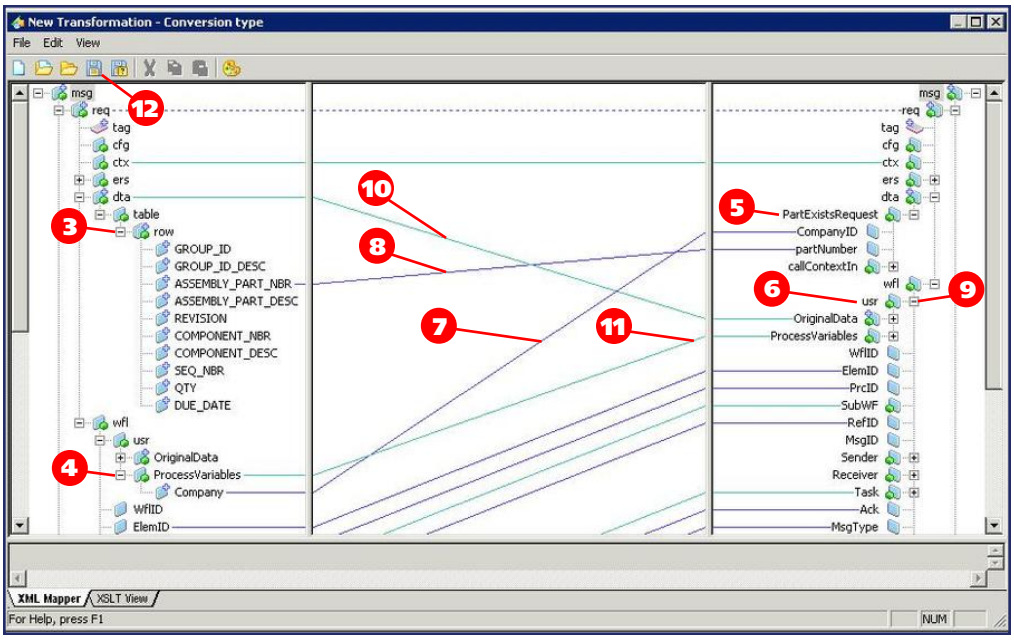
3. In the Tree View, expand the **req > dta > table > row** node in the incoming document.

4. Expand the **wfl > usr > ProcessVariables** node in the incoming document.

5. Expand the **req > dta > PartExistsRequests** node in the target document.

6. Expand the **wfl > usr** node in the target document.

7. Map the **Company** node in the incoming document to the **CompanyID** node in the target document.



   To create the mapping, click the Company node, hold the left mouse key, and drag the pointer to the CompanyID node in the target document.
   The mapping is represented by a line that connects the two nodes. Thus, the Company of the incoming document is supplied to the CompanyID node of the target document.

8. Map the **ASSEMBLY_PART_NBR** node in the incoming document to the **partNumber** node in the target document.

9. Verify if the **usr** node in the target document is mapped to a node in the source document. If it is, delete the mapping.

   To delete the mapping, click the link, right-click, and select **Delete selected link(s)**.

10. Map the **dta** node in the incoming document to the **OriginalData** node in the target document.

    This mapping creates a deep copy that stores all the information in the dta node of the incoming document in a message extension in the target document. Review the Message Extensions section later in this chapter for more information about how to store information in a container for later use.

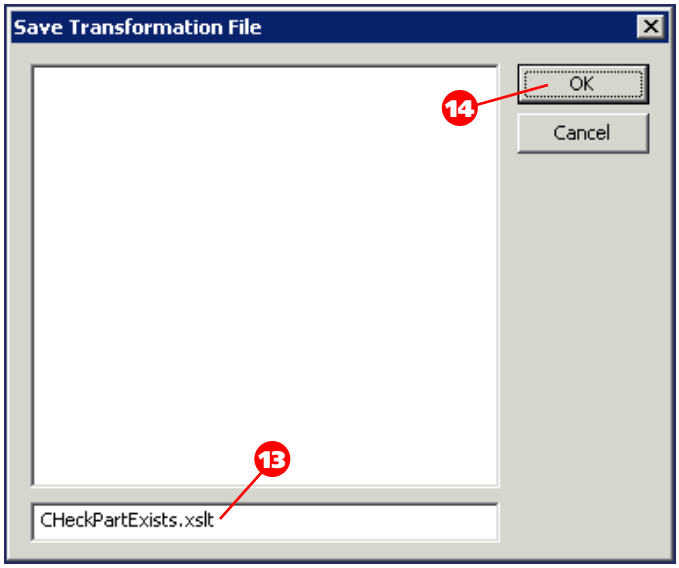
11. Map the **ProcessVariables** node in the incoming document to the **ProcessVariables** node in the target document.

12. Click **Save**.

13. The **Save Transformation file** window displays. Enter a name for the XSLT file.

    For descriptions of all the node types in the XML Mapper and the different types of mappings you can make, review the XML Mapper section later in this chapter.

14. Click **OK**.



Condition

Use this activity to test an incoming document against an automated processing rule and halt the document path if it does not satisfy the rule. If the document satisfies the processing rule, the Condition passes it to the outbound Connections. Although the Condition can be attached to more than one outbound Connection, the document is evaluated against only one rule. To evaluate a document against more than one rule, use a Choice activity.

Use Condition Rule

The Condition rule is an XPath expression evaluated against an incoming document as either true or false. If the document satisfies the rule, the system sends the document through the outbound Connections. If the document does not satisfy the rule, the path of the document is stopped, which can stop the entire workflow.

Example

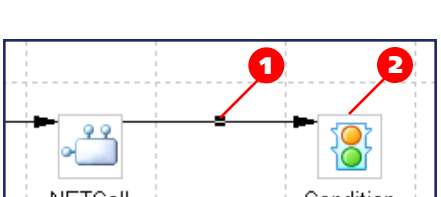
An incoming document contains information about a call record for a call center. The document is routed along two paths, each one serving as the inbound Connection to a Condition activity. The first Condition tests whether the CallSeqNum element equals zero and allows the document to pass to the outbound Connection if the test evaluates as true. Otherwise, the document is halted. The second Condition also tests whether the CallSeqNum equals zero, but the second Condition only allows the document to pass to the outbound Connection if the test evaluates as false. The result is that the incoming document is allowed to proceed along only one of the two paths.

Set Up a Condition

To set up a Condition:

1. Add the inbound **Connection**.

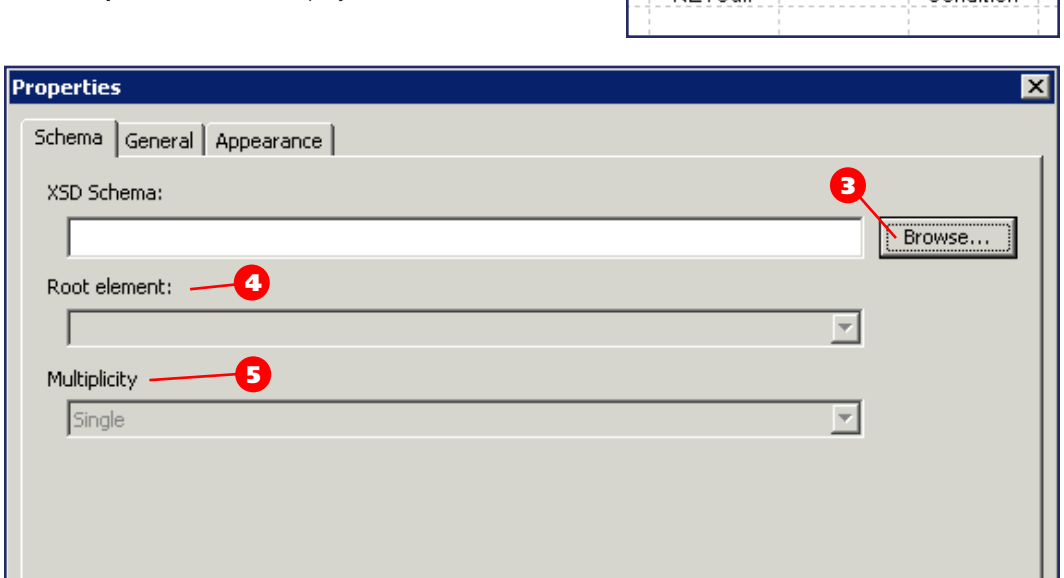
   The workflow activity that precedes the inbound Connection supplies the document to be evaluated against the Condition rule.



2. Double-click the **Condition**. The **Properties** window displays.

3. On the **Schema** sheet, click the **Browse** button to specify the schema of the incoming document.

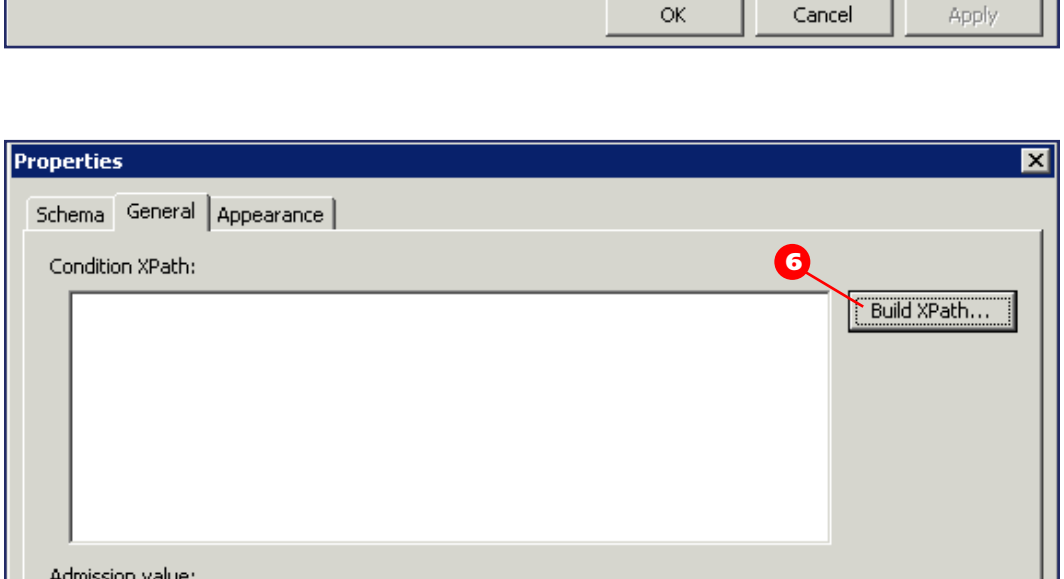
   In the **Open Schema** dialog left pane, you can switch between Web-Services schemas, .NET Reference schemas or user-defined schemas generated in Schema Export utility.



4. Select a **Root Element** for the selected schema.

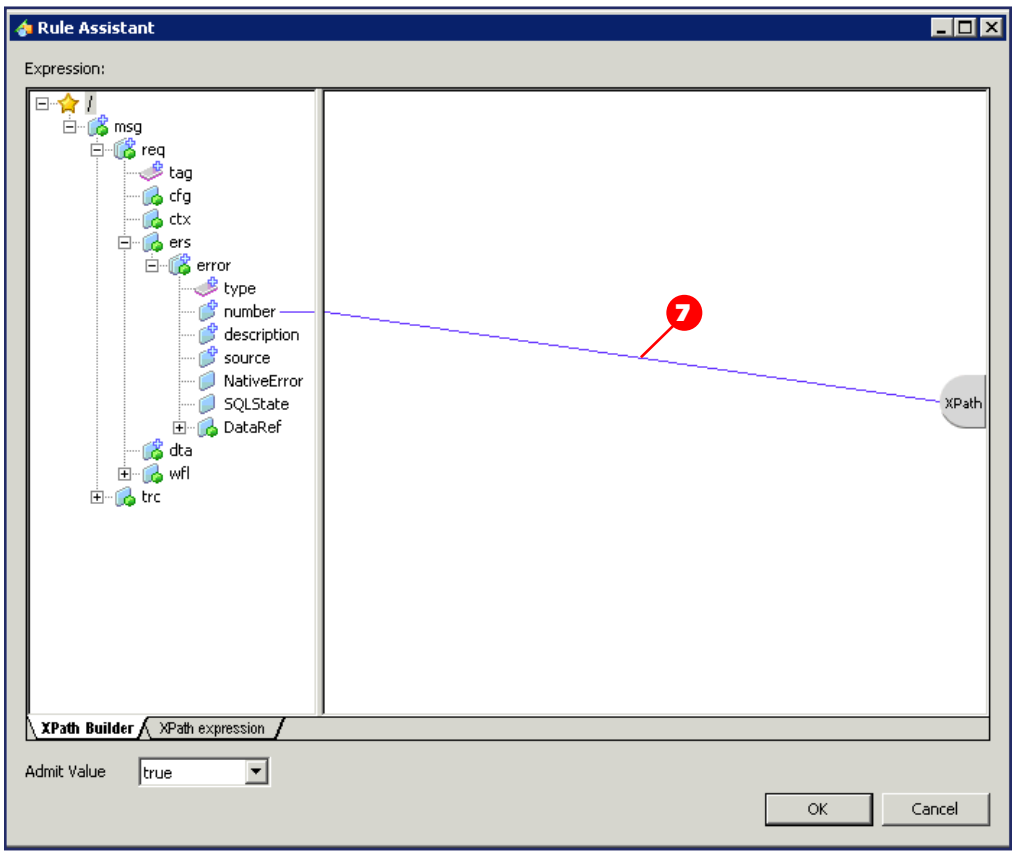
5. Select the **Multiplicity** for the selected schema. Select **Single** to use just one element under the dta element, or select **Multiple** to utilize a collection under the dta element.

6. Navigate to the **General** sheet and click **Build XPath** to create the Condition rule.



7. The **Rule Assistant** displays. To define a rule, expand the nodes in the document and map the node you want to test to the **XPath** marker on the right.

   In this example, the rule is designed to test whether an error number exists. In the following steps, you will customize the rule because each Service Connect transaction returns an error number where 0 (zero) is equivalent to no error.

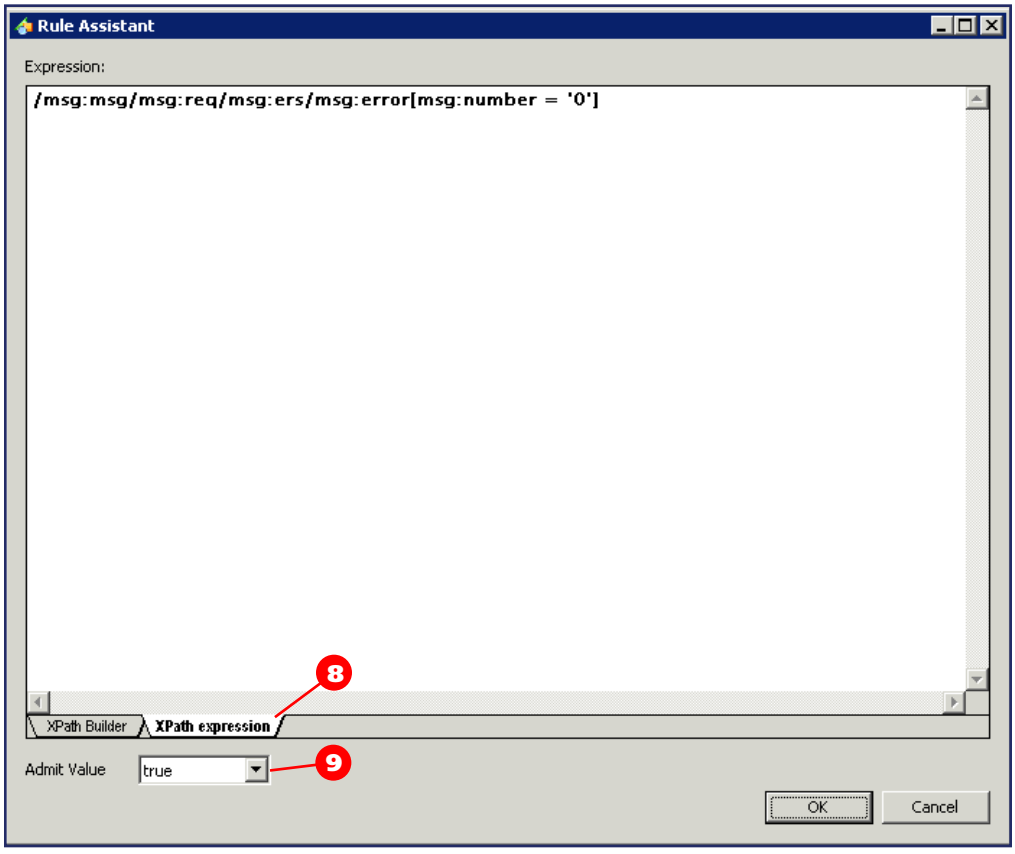


8. Click the **XPath expression** tab and add a filter expression to the XPath so the rule tests the value of the number node.

   This XPath expression will evaluate to true when the value of the msg:number node equals zero.

9. Click the **Admit Value** drop-down list to select either **true** or **false**.

The Admit Value field is set to true, which indicates the document will pass to the outbound Connection if the expression evaluates to true. Otherwise, this branch of the workflow will halt.

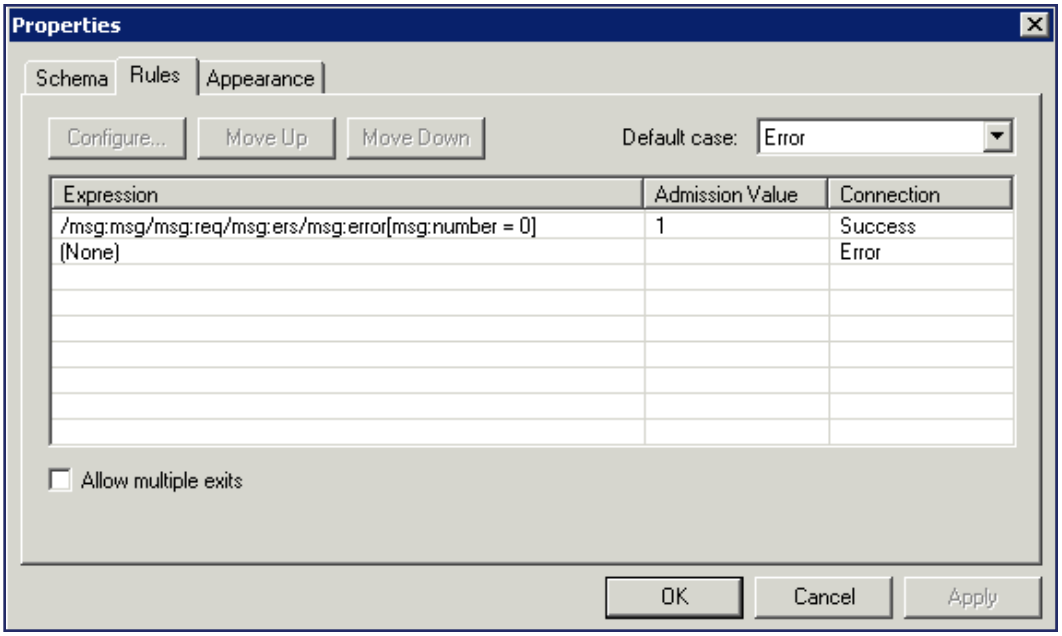


Choice

Use this activity to route a document based on automated processing rules. The Choice has one inbound Connection but can have several outbound Connections. Each outbound Connection is associated with a processing rule. A document that enters the Choice activity is evaluated against the rules to determine which of the outbound Connections the document will follow. You can configure Choice activities to send the incoming document along multiple outbound Connections or to just one.

Use Choice Rules

Each rule is an XPath expression that is evaluated against the incoming document as either true or false. Documents can progress through each outbound Connection for which they satisfy the rule. To restrict the passage of a document to only one outbound Connection, clear the Allow multiple exits check box.



Rules are applied to the incoming document in the order in which they display in the Rules dialog box. Thus, if the incoming document satisfies more than one rule but the document is allowed to pass through only one outbound Connection, the document will progress through the first Connection that satisfies the rule. To change the order of the rules, select a row in the grid and use the Move Up or Move Down button. All Choice elements have a Default case rule. If an incoming document does not satisfy any of the rules, the system passes the document to the outbound Connection associated with the Default case rule, which allows the workflow to continue.

Example

An incoming document contains part update information. The workflow uses a Web Method to update the Epicor application. The Web Method output is passed to a Choice activity. If the web service returns an error, the Choice activity routes the document to the Poster element that notifies a user about an error. If the web service does not report any errors, the update process ends successfully.

Set Up a Choice

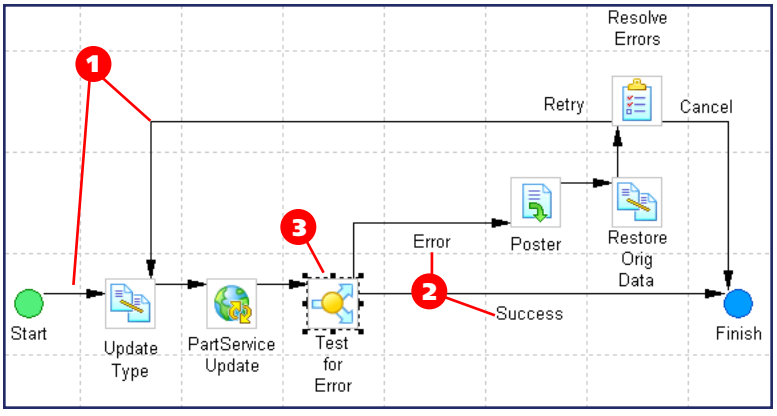
To set up a Choice:

1. Add the inbound **Connection** and the outbound **Connections**.

   The workflow element that precedes the inbound Connection supplies the document to be evaluated against the Choice rules. In this example, the preceding element is a Web Method call titled Check Part Exists.

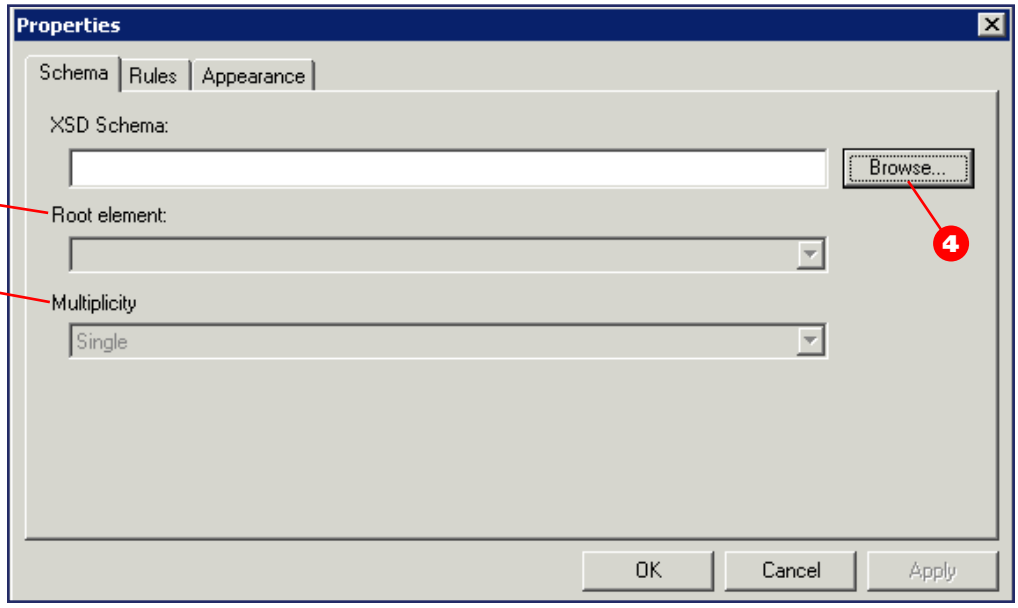
2. Set the **Caption** on each outbound Connection. Give each Connection a logical name to help you when defining the Choice rules.

3. Double-click the **Choice**.



4. The **Properties** window displays. On the **Schema** sheet, click the **Browse** button to specify the schema of the incoming document.

   In the **Open Schema** dialog left pane, you can switch between Web-Services schemas, .NET Reference schemas or user-defined schemas generated in Schema Export utility.



5. Select a **Root Element** for the selected schema.

6. Select the **Multiplicity** for the selected schema. Select **Single** to use just one element under the dta element, or select **Multiple** to utilize a collection under the dta element.

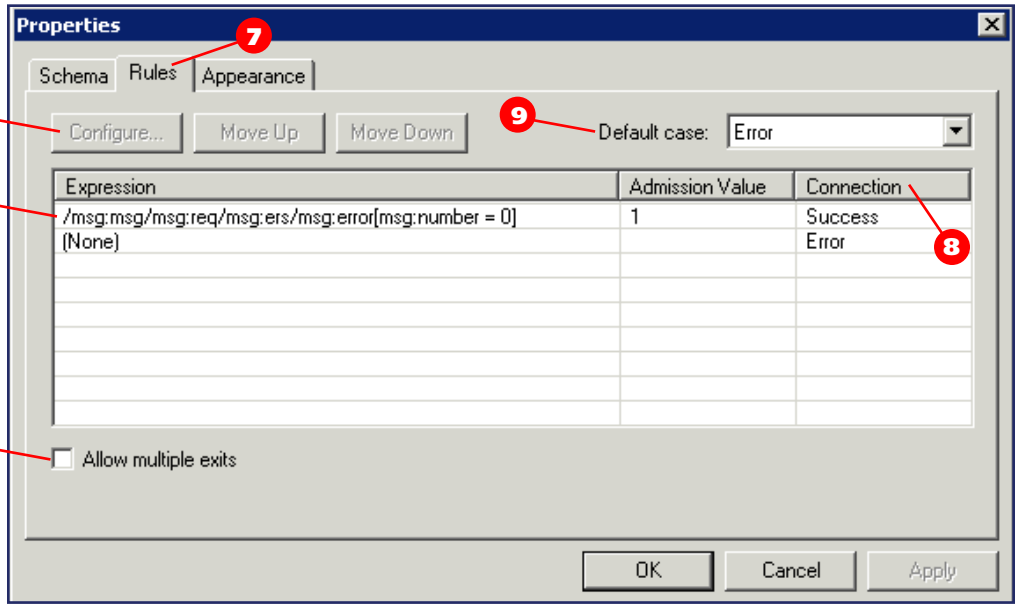
7. Navigate to the **Rules** sheet.

8. Notice the two outbound Connections are listed by name in the **Connection** column.

9. Select the appropriate **Default case**.

10. Select whether to **Allow multiple exits**.

    If the Allow multiple exits check box is selected, the document will be evaluated against each rule in the grid even after it has satisfied a rule, and this can be sent to more than one outbound Connection. If the check box is clear, then the document will be sent to the first outbound Connection for which it satisfies the rule or to the Default case Connection if it does not satisfy any rules.

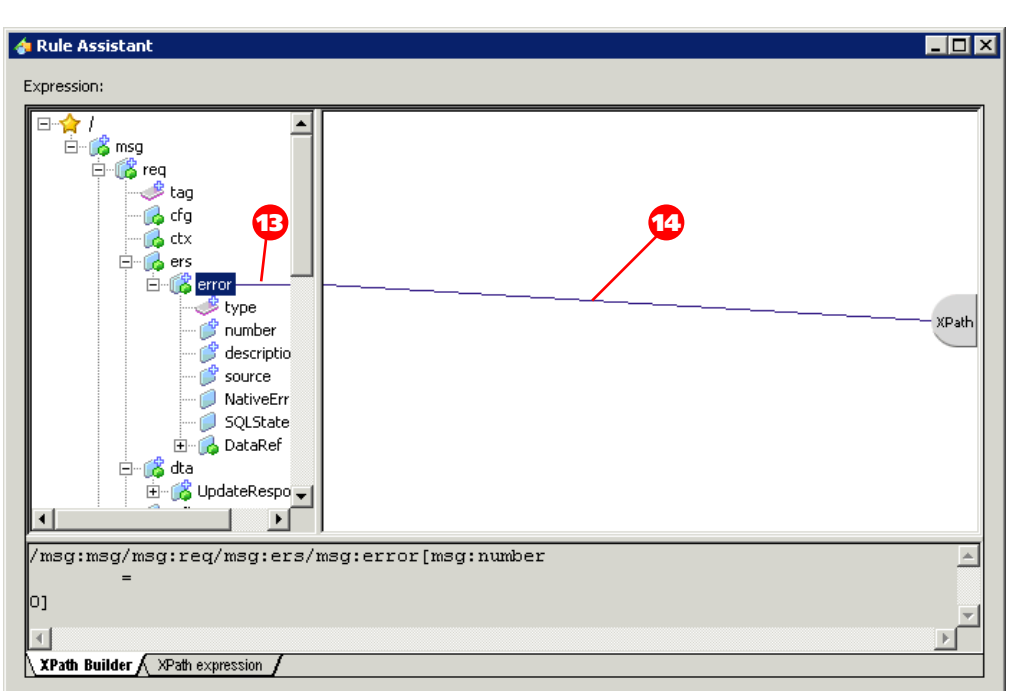


11. In the **Expression** column, select a Connection for which you want to define a rule.

12. Click **Configure**.

13. The **Rule Assistant** window displays. Expand the nodes in the incoming document to find the node you want to use for the rule.

14. Map the node you want to use for the rule to the **XPath** marker on the right.



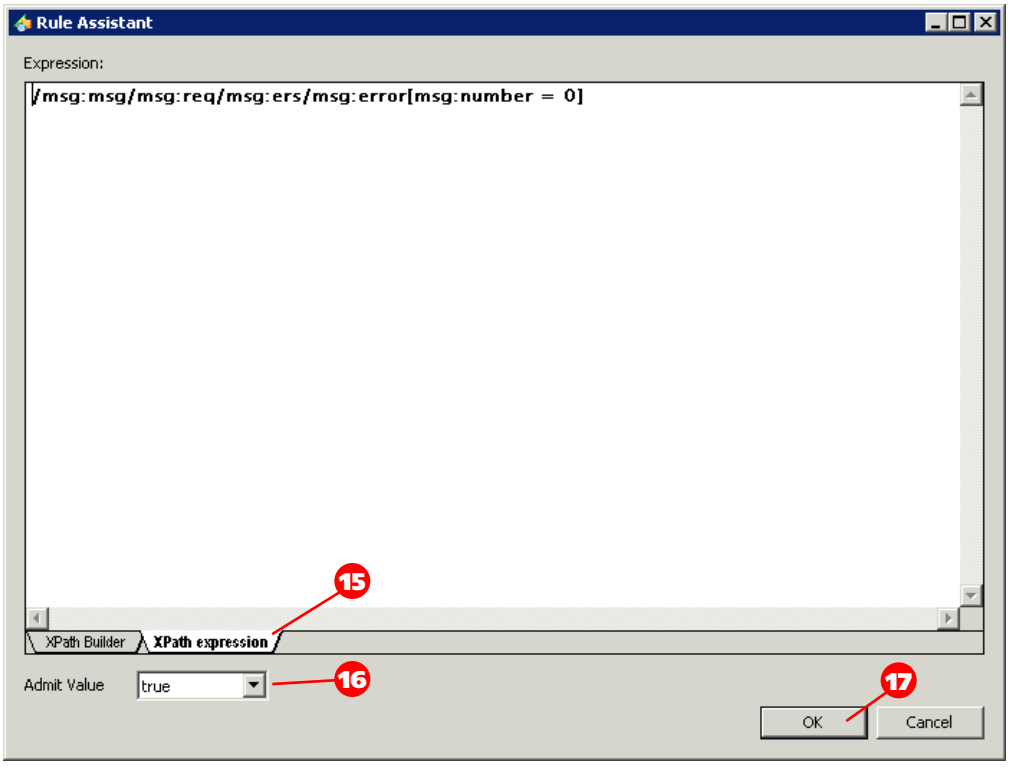
15. Click the **XPath expression** tab if you need to customize the rule.

    In this example, the rule has been customized to test the value of the dta:result node instead of testing for the presence of the node.

16. Click the **Admit Value** drop-down list to select either **true** or **false**.

    The Admit Value field is set to true, which indicates the document will pass to the outbound Connection if the expression evaluates to true. Otherwise, Service Connect evaluates the message against the next Choice rule.

17. Click **OK** to return to the **Properties** dialog box.



Web Method

Use this activity to call a service method, such as a web service method or a Windows Communication Foundation (WCF) method. You must import the method as a Service Reference in the ESC Administration Console before you can use it in a Web Method. Web Methods are the primary methods for communicating with target databases, except for Epicor iScala, which uses business logic components called Managers to interact with the database.

Example

An incoming document contains a customer record. You can configure a Web Method activity to call the GetByID method to verify whether the customer record already exists. If a customer record is not retrieved, another Web Method activity can call the Update method to add the new customer to the target database.

Set Up a Web Method

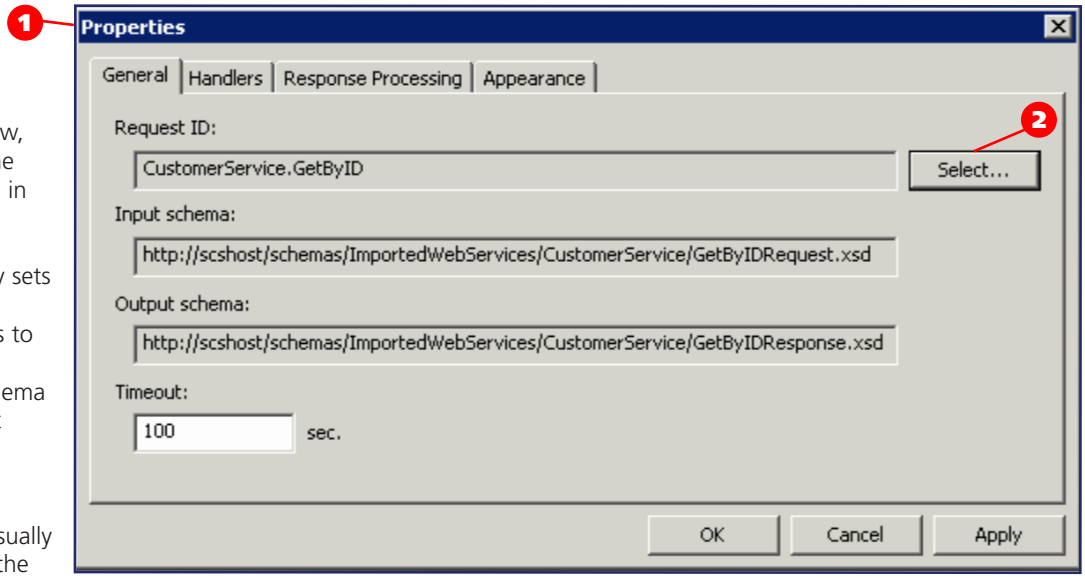
To set up a Web Method:

1. Double-click the **Web Method** to open the **Properties** window.

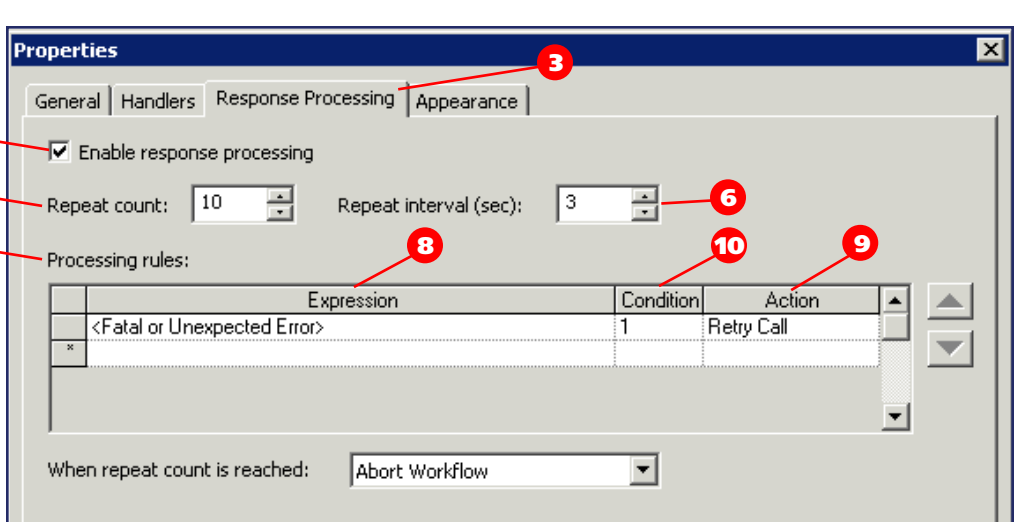
2. In the **Properties** window, click **Select** to choose the method you want to call in the Request ID field.

   The system automatically sets the Input schema to the schema that corresponds to the web service request. Likewise, the Output schema is set to the schema that corresponds to the web service response.

   The preceding activity, usually a Conversion, must use the Web Service request schema as its Output schema so the document coming into the Web Method is ready to pass to the Web Service. The following activity must use, or be ready to accept, a document formatted according to the Web Service response schema.



3. Optionally, click the **Response Processing** tab to configure repetitive loops for the web method.



The calls of a Web Method, .NET Call, or iScala manager from Service Connect workflows sometimes result in errors. If the same workflow element is called several seconds later, the call is successful. The reason for the errors depends on many factors such as installation peculiarities. Use this tab to configure the number and frequency of loop repetitions and the workflow behavior when the maximum number of loop repetitions is reached.

4. Select the **Enable response processing** check box to enable the response mechanism.

5. Specify the **Repeat count** - the number of attempts to repeat Web Method, iScala manager, or .NET calls. If the element call is successful and the specified number of repetitions is not reached, the workflow continues to execute. If the element call is not successful, but the specified number of repetitions is reached, the workflow behaves as specified in the **When repeat count is reached** field.

6. Specify the **Repeat interval**, in seconds, between the repeated Web Method, iScala manager, or .NET Calls.

   The maximum interval is 999 seconds. By default, the interval is 3 seconds.

7. In the **Processing rules** grid, you can add, edit, and delete the errors and expressions used to handle Web Method, iScala manager, or .NET responses.

8. The expressions used to handle responses display in the **Expression** column. By default, the <Fatal or Unexpected Error> expression is used. You can add, edit, or delete the expressions that display in this grid.

   At least one expression must display in the grid.

9. To perform an **Action** on the selected expression, click the button in the right corner of the cell. Available options:

   - **Build Rule** – This launches the XPath Builder where you can compose an expression which will be used to process errors of the specified type. This option is available for an empty grid row.
   - **Edit Rule** – This launches the XPath Builder where you can edit the selected expression. This option is available for the grid row, which contains an expression.
   - **Delete Rule** – This deletes an existing expression.
   - **Process Errors** – This adds the default <Fatal or Unexpected Error> expression. This option is available if the default expression is not used in the grid. You cannot include the default expression in the grid more than once.

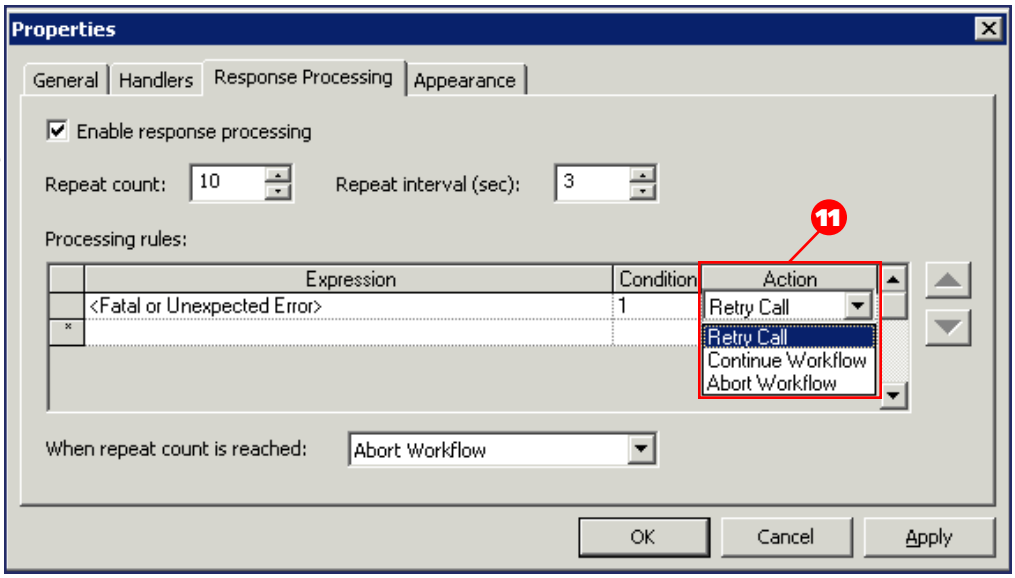
10. The **Condition** column displays the values, which determine if the action executes when the expression is true or false. Available options:

    - **0** – The action executes when the expression is **false**.
    - **1** – The action executes when the expression is **true**.

11. In the **Action** column, select how to handle the expression.

    The available values are:

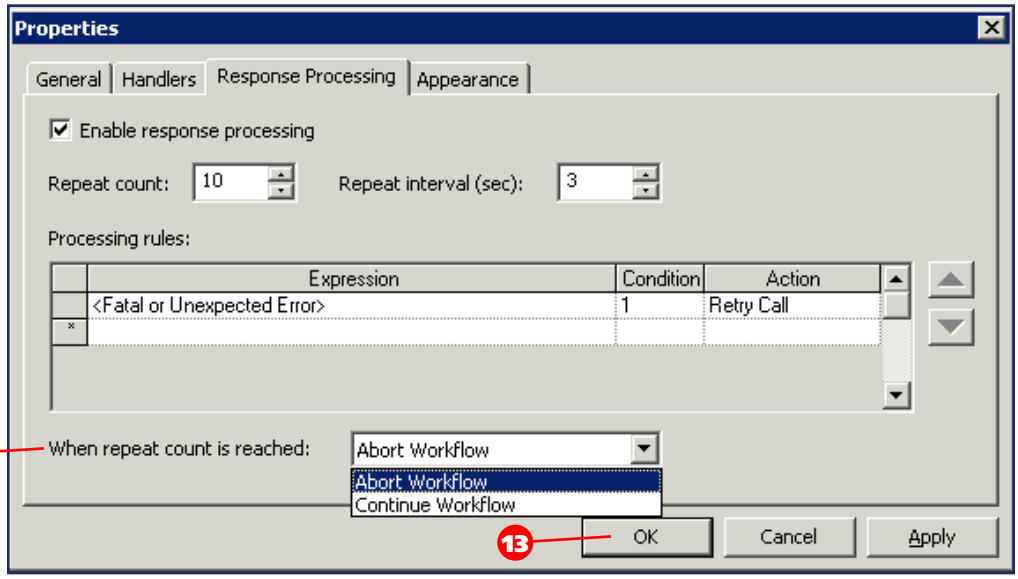
    - **Retry Call** – This repeats the Web Method, iScala manager, or .NET call with the parameters (number of times or intervals) specified on the Response Processing tab.
    - **Continue Workflow** – This continues the workflow execution, regardless of Web Method, iScala manager, or .NET call failure.
    - **Abort Workflow** – This aborts the workflow execution.



12. The **When repeat count is reached** options handle the workflow behavior when the number specified in the Repeat count field is reached.

    The available values are **Abort Workflow** or **Continue Workflow**.

13. Click **OK** to apply the selected parameters.



When response processing is enabled, the web method icon changes from  to  .

If you modify a Web Method configuration, for example, if you customize the handlers configuration in the Workflow Designer, Web Method icon changes to the following :

 .NET Call

Use this activity to call methods exposed by a .NET reference. You must import the method as a .NET Reference in the ESC Administration Console before you can use it in a .NET Call. When you call .NET assemblies in workflows, performance improves and you can replace web service calls with direct Epicor .NET object calls. This feature extends the scope of workflow functionality, providing additional leverage when working with any .NET language.
To use the .NET Call workflow element, it must be licensed.

Example

An incoming document contains customer information. The workflow uses a .NET Call to update the target database with the new customer information.

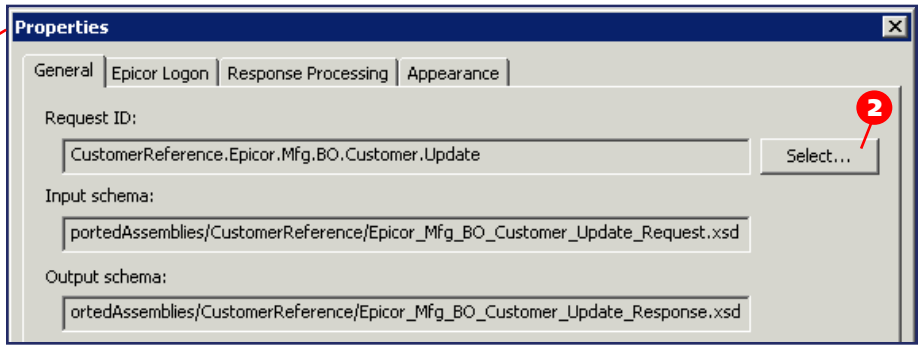
Set Up a .NET Call

To set up a .NET Call:

1. Double-click the **.NET Call** to open the **Properties** window

2. Click **Select** to choose the method you want to call in the Request ID field. You can also enter the value directly.

   The system automatically sets the Input Schema to the schema that corresponds to the .NET Call request. Likewise, the Output Schema is set to the schema that corresponds to the .NET Call response.



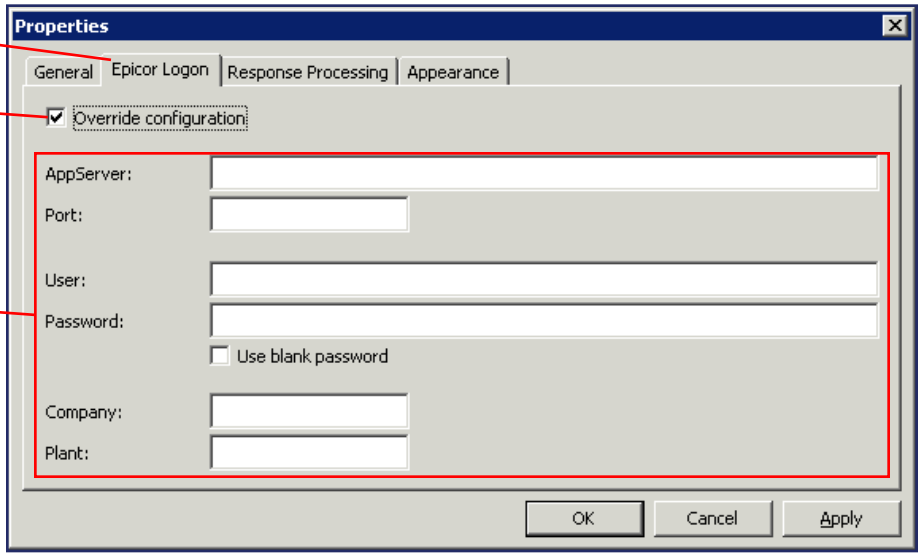
3. If the method used is an Epicor Assembly and you need to override a logon configuration entered during the .NET Reference import in the Epicor Service Connect Administration Console, click the **Epicor Logon** tab.

   Review the .NET References section in Chapter 3: Connectivity Components for more information.

4. Select the **Override configuration** check box.

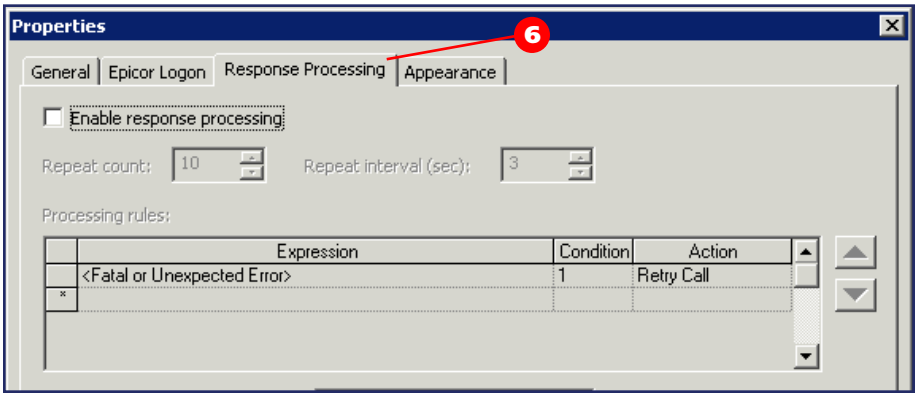
5. Enter your company-specific settings.

   If the Epicor logon configuration was customized in the Workflow Designer, .NET Call icon changes to the following:



6. On the **Response Processing** tab, you can set .NET Call element to re-call a .NET element repeatedly in case of an error.

Review the Web Method section earlier in this chapter for more information on response mechanism functionality.



DB Operation

Use this activity to run one of several SQL statements (SELECT, INSERT, DELETE, UPDATE, and EXEC) against one or several databases. SQL statements can use part of an XML document to update or select database data. You can put the result of the SELECT statement into an XML document for further workflow processing.

You can use the DB Operation to perform the following actions:

- Read, write, and update third-party databases during workflow execution.
- Read values from application databases on-the-fly; it may be useful if one or several fields are not returned by standard BO methods.
- Read additional BO values from custom user tables.

Service Connect interacts with SQL through ActiveX Data Objects (ADO). If SQL database contains bit type fields, after DBOperation execution TRUE (1) value will be -1 (string).
Use case: if you need to pass the result of DBOperation to a Web Service request and one of the request fields is bool, use boolean functoid to convert "-1". Otherwise an error occurs.

To use the DB Operation workflow element, it must be licensed.

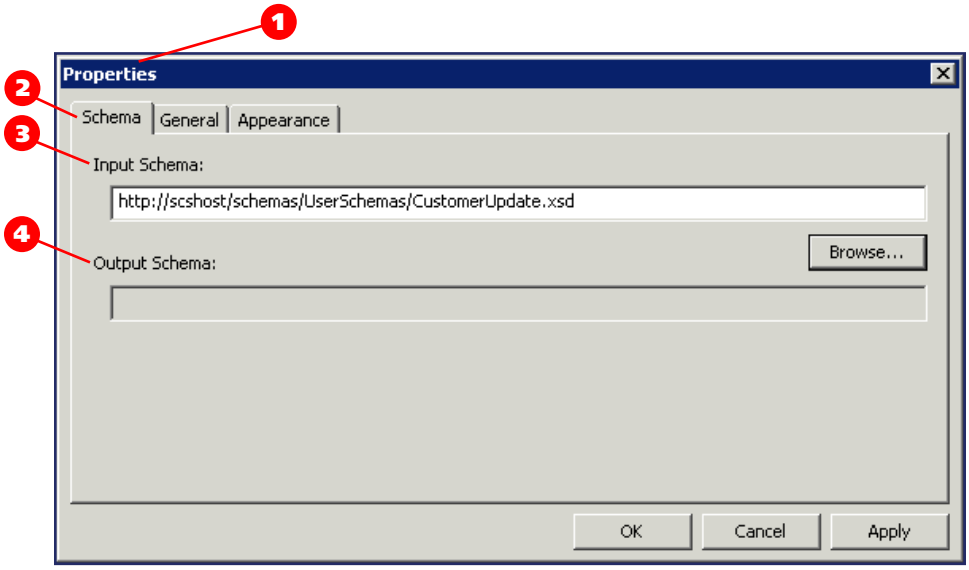
Example

Create a new customer record and use the DB Operation activity to run a query against the database to retrieve data. As a result, an .xml file that contains all the customer records will be dropped in the output folder you specify. You can use the .xml file for further processing.

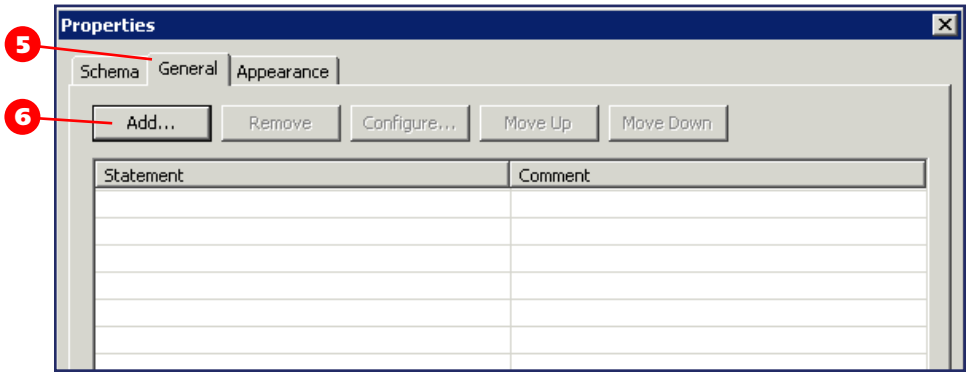
Set Up a DB Operation

To set up a DB Operation:

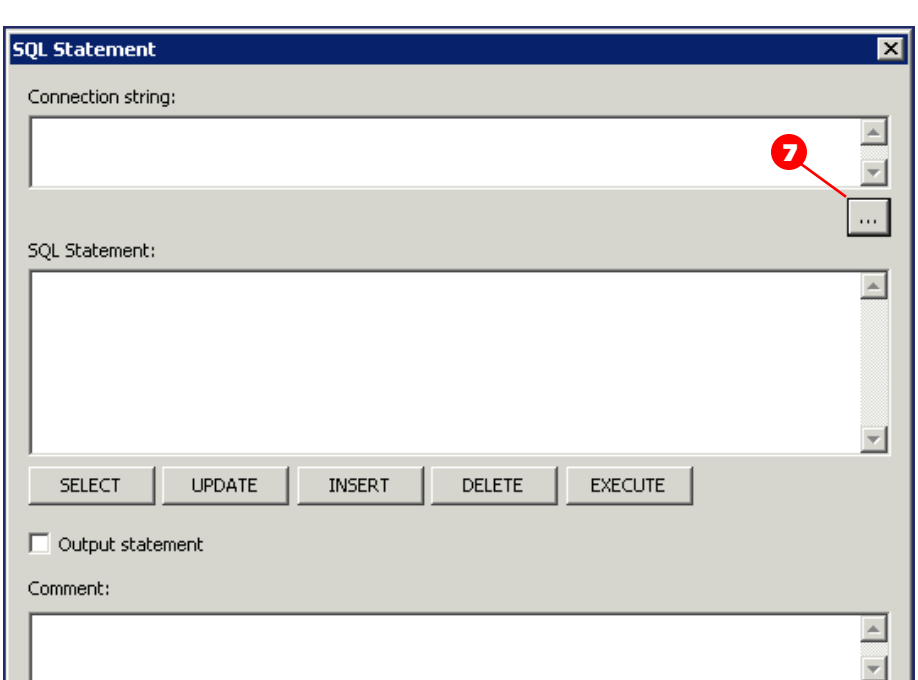
1. Double-click the **DBOperation** to open the **Properties** window.
2. Click the **Schema** tab.
3. Click the **Browse** button to find and select the **Input Schema** path to use, update, or select database data.
4. The **Output Schema** field is read-only and is defined by the output statement.



5. Click the **General** tab.
6. Click the **Add** button to add a new statement.



7. In the **SQL Statement** window, click the **... (Ellipse)** button to build the connection string.

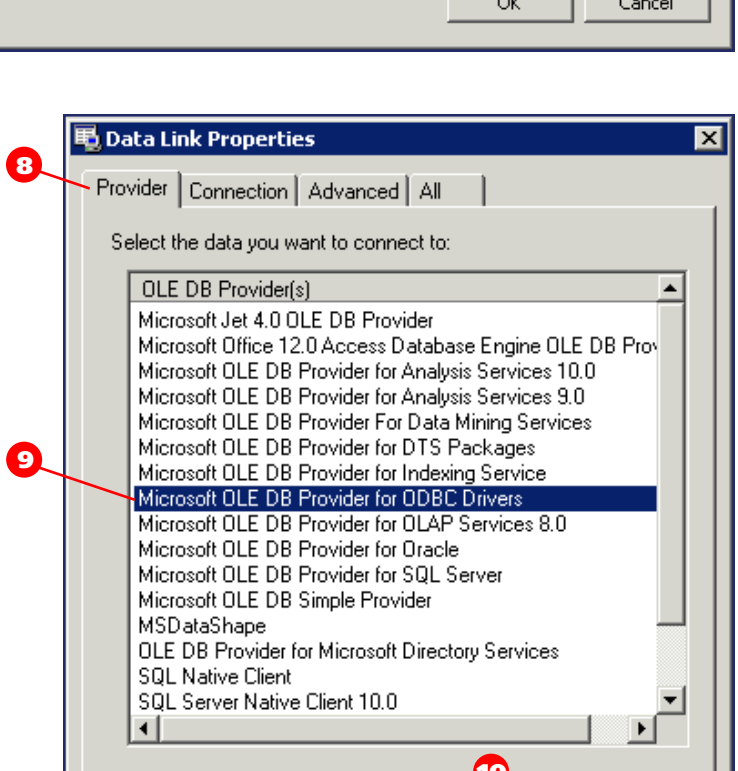


8. Click the **Provider** tab.

9. Select an OLE DB Provider.

   In this example, select Microsoft OLE DB Provider for ODBC Drivers.

10. Click **Next**.



11. Click the **Connection** tab.

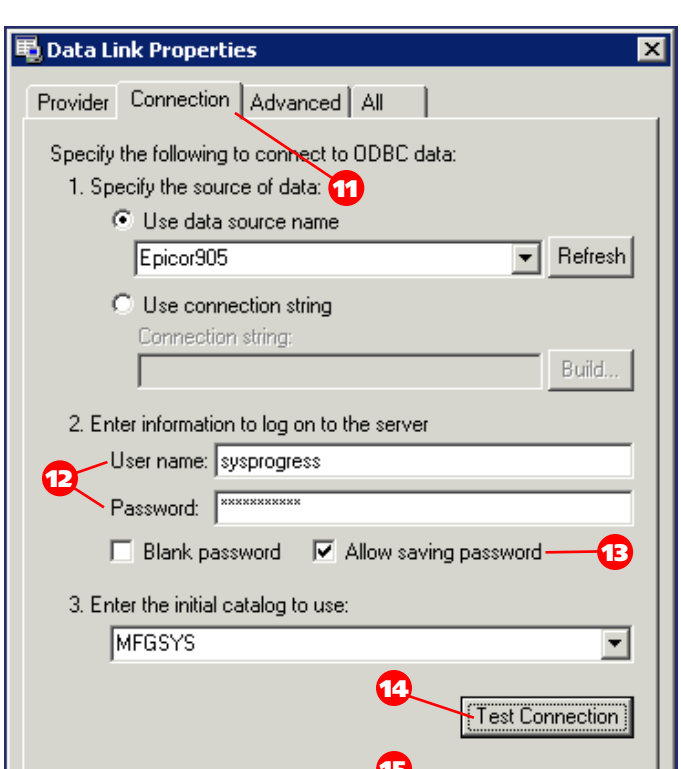
12. In the **Username** and **Password** fields, enter your ODBC credentials.

13. When you connect to a database that requires a user name and password, select the **Allow saving password** check box.

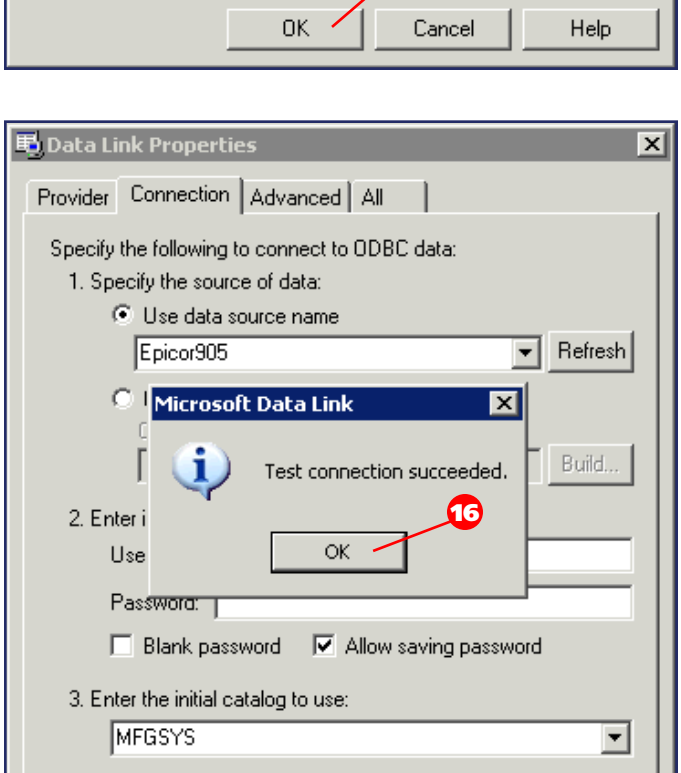
14. To test the established connection, click **Test Connection**.

15. Click **OK**.

    You can edit a generated connection string manually. You can insert XML links into the connection string, for example, to address different databases depending on XML document content.



16. To the confirmation message, click **OK**.



17. Enter or build the **SQL Statement** using the SELECT, UPDATE, INSERT, DELETE, and EXECUTE buttons. If XML links are found in the connection string, you will be prompted to resolve them.

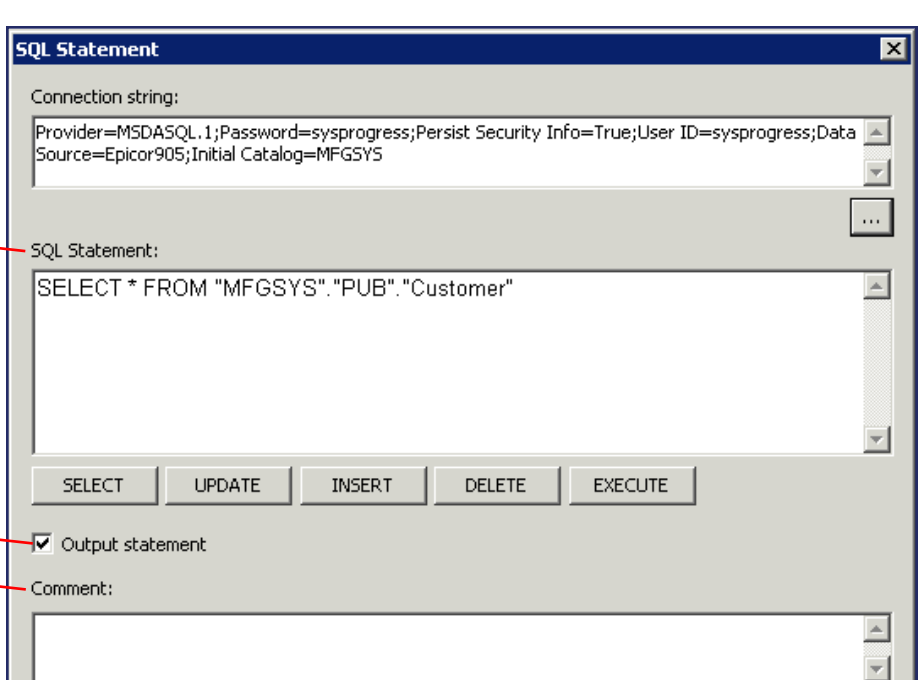
    In this example, the query returns all records from the Customer table.

    When you generate an SQL statement via INSERT, UPDATE, DELETE, or EXECUTE functionality, Service Connect checks if this results in direct writes to an Epicor transactional database. The corresponding warning displays.

18. Optionally, if you want one of the SELECT/EXEC statements to put the data to the output XML document, select the **Output statement** check box. You can use this data for further workflow processing.

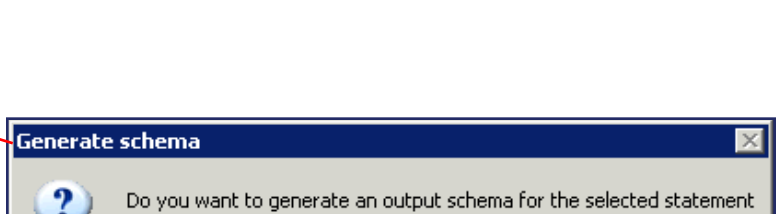
19. You can enter a comment in the **Comment** field.

20. Click **OK**.



21. If the **Output statement** check box was selected, the **Generate schema** window displays.

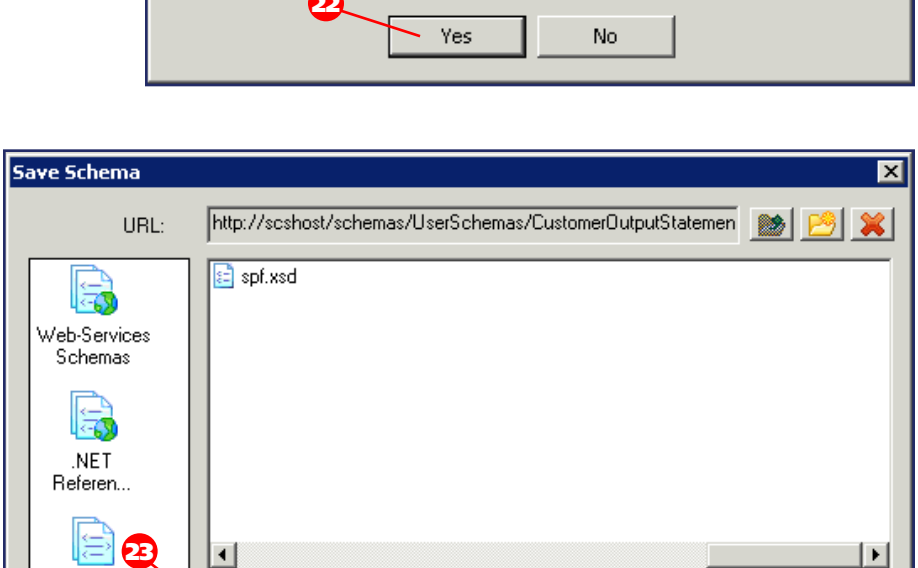
22. If you want to generate a schema for an output statement, click **Yes**.



23. Enter a **File Name** for the new schema.

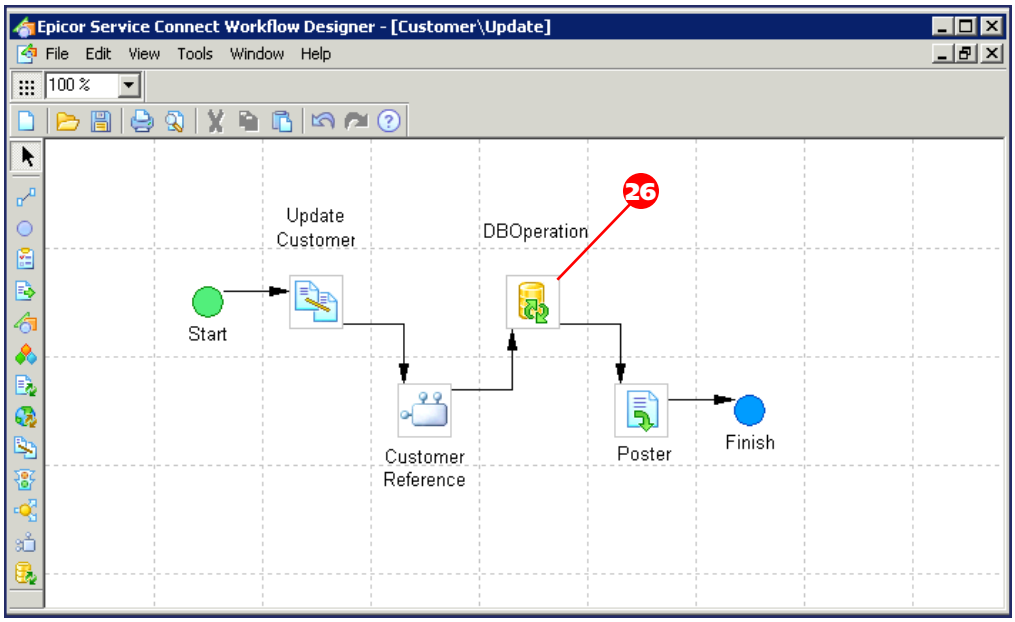
24. Click **OK**.

25. In the **Properties** window, click **OK**.



26. Build the workflow with the **DBOperation** element.

In this example, the DB Operation is followed by the Poster element which drops an xml output from the SQL query in the folder specified in the output channel.



Break

Use this activity to break a cycled sub-workflow. Default behavior of synchronously executed sub-workflow assumes that the main workflow execution continues after sub-workflow completion, no matter what the result of the sub-workflow execution is. Moreover, in case when cycling is set-up, all sub-workflow iterations that are defined by the main workflow document are executed, no matter what the result of any particular iteration is.
In some cases, it is necessary to break a sub-workflow execution. You can use the Break element to define an ordered set of rules that are executed against incoming document in order to decide how exactly the main workflow execution should be altered.

The main workflow can change how to call a sub-workflow in the following ways:

- Looping is interrupted and the trace status is set to **Completed**.
- Looping is interrupted and the trace status is set to In **Progress**.
- Looping is interrupted and the trace status is set to **Abortive**.
- Looping is not interrupted and the sub-workflow continues. The **Ignore** status means that the Break element should not do anything at all. It passes the unmodified document without setting a break code in process context. In this case you avoid additional Choice or Condition elements that control whether the execution of the main workflow should be altered or not.

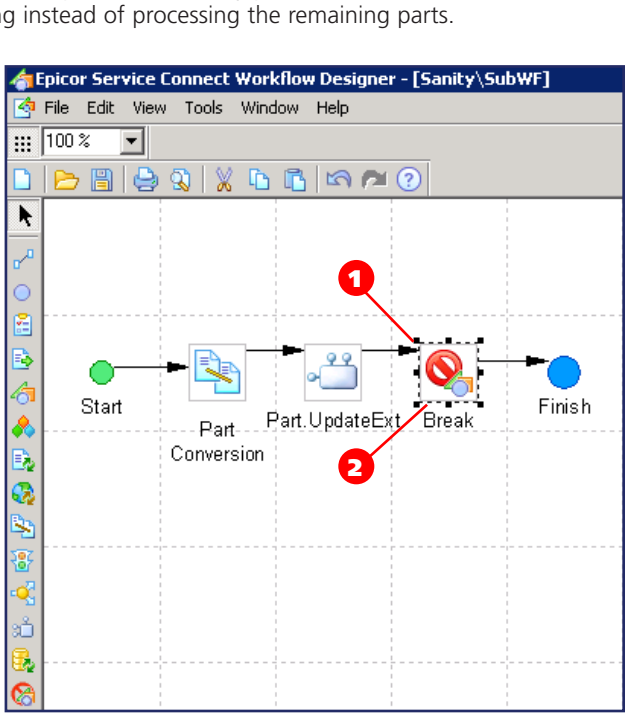
Example

You use a complex file with many part numbers to update the application. Each part is individually processed in a sub-workflow. If an error occurs in any of the parts, you want the sub-workflow to stop looping instead of processing the remaining parts.

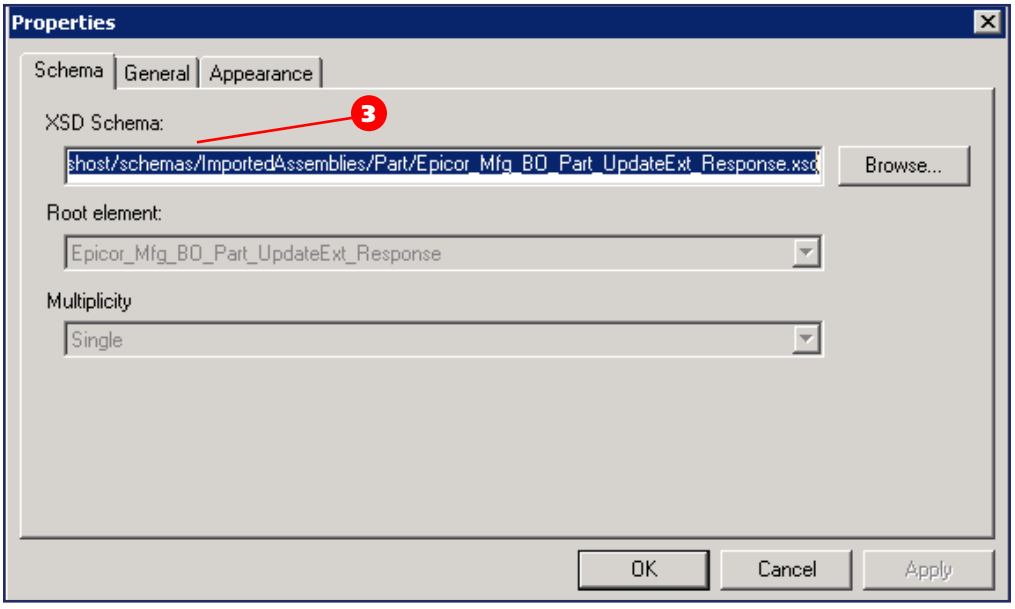
1. Depending on your business logic, place the **Break** element in a sub-workflow.

   The Break element should not necessary be followed by the Finish element, but can be placed anywhere the workflow logics requires.

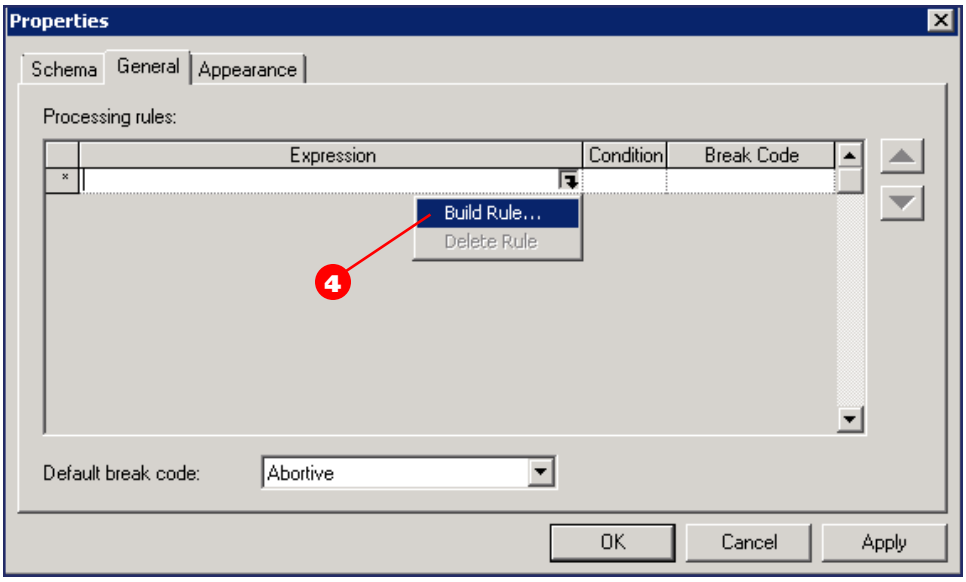
2. Double-click the **Break**.



3. The **Properties** window displays. On the **Schema** sheet, click the **Browse** button to find and select an input schema for the element.



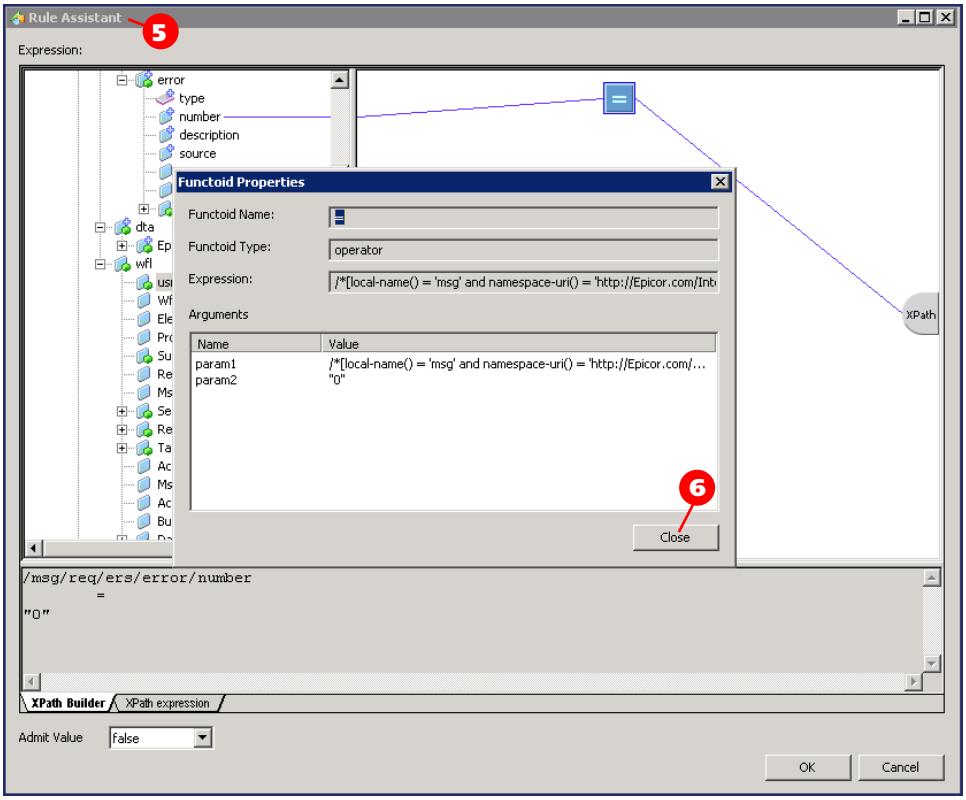
4. Click the **General** sheet. Click the **down arrow** button next to the **Expression** field, and select **Build** rule.



5. Use the **Rule Assistant** to compose an expression, which will be verified against incoming document.

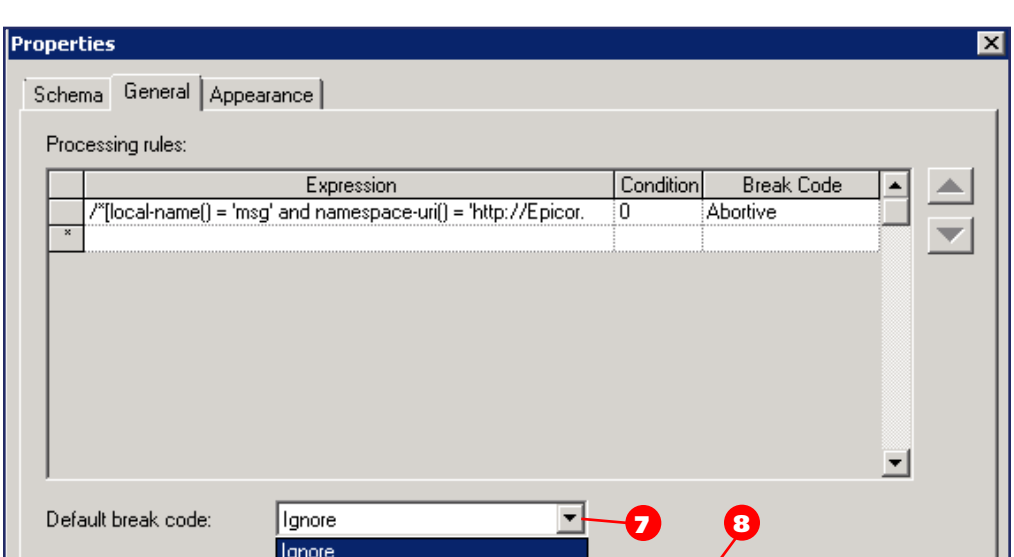
   In this example, the break element looks for any error in the update response from .NET call.

6. Once the rule is created, in the Functoid Properties window, click **Close**, and in the Rule Assistant window, click **OK**."

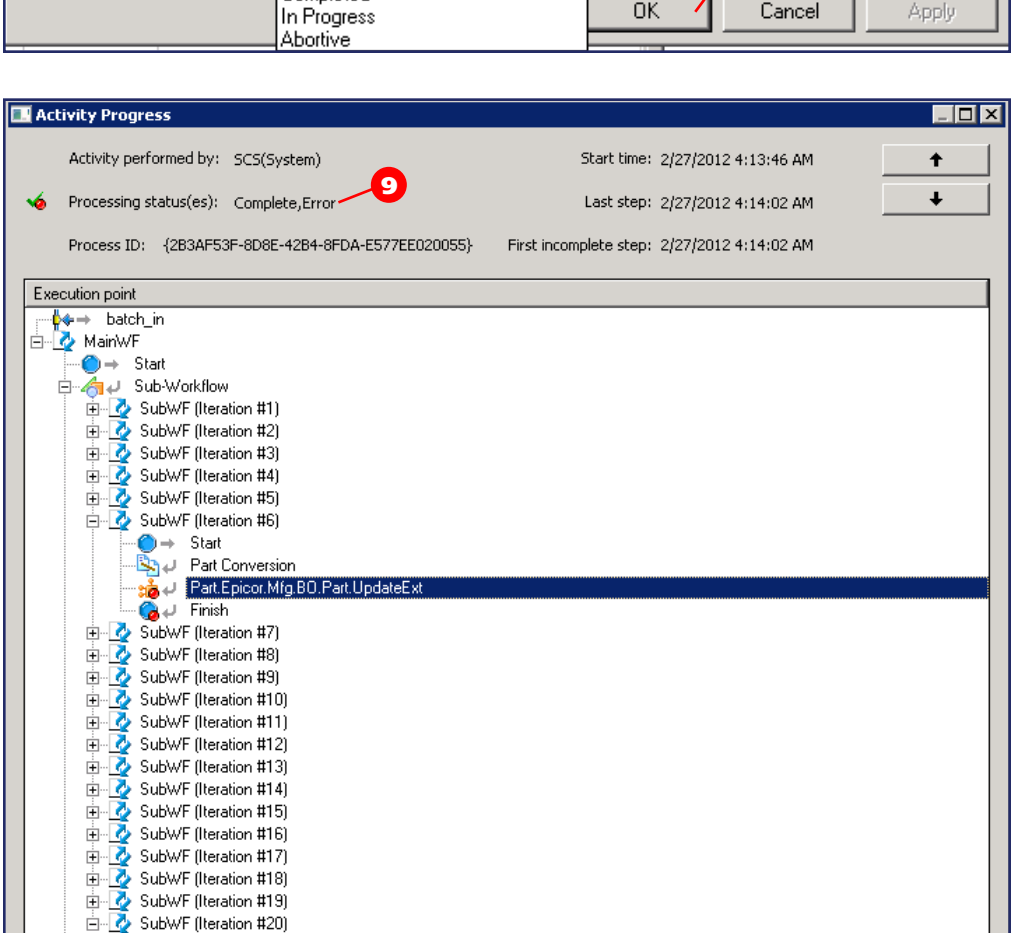


7. Click the **Default break code** drop-down list to select the status that will be assigned to the workflow when the specified condition is met.

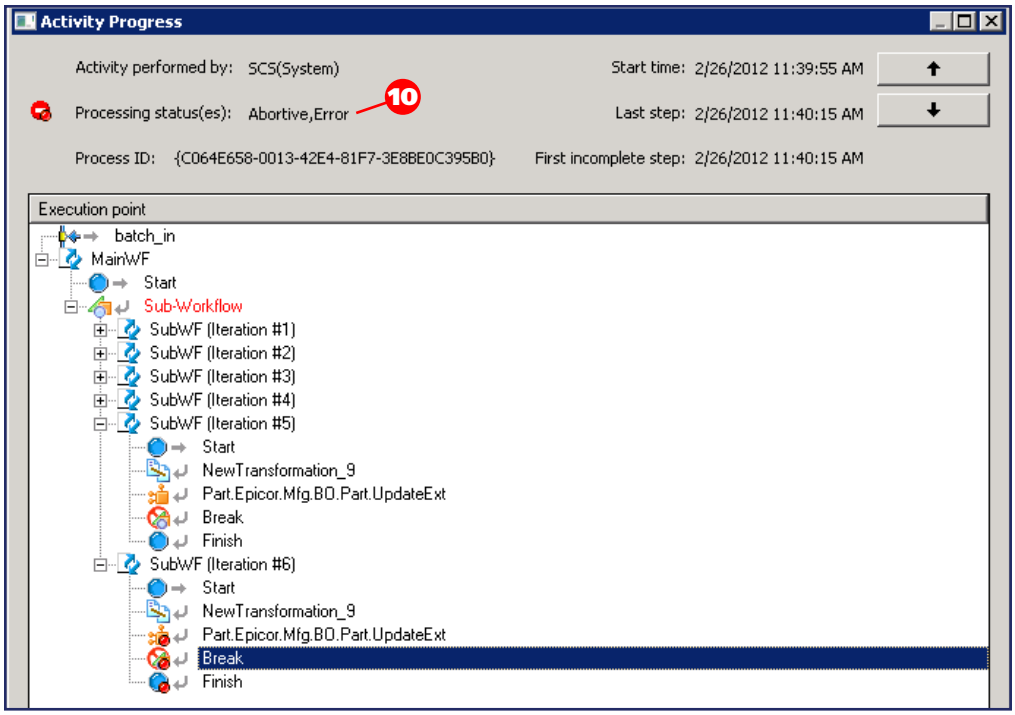
8. Click **OK**.



9. Notice the difference when the corrupted source file is processed without using the break element. The activity results in **Complete, Error status**, all the part iterations are processed.



10. When the Break element is used in the sub-workflow, the error introduced in the 6th iteration halts the process and the activity results in the **Abortive** state.



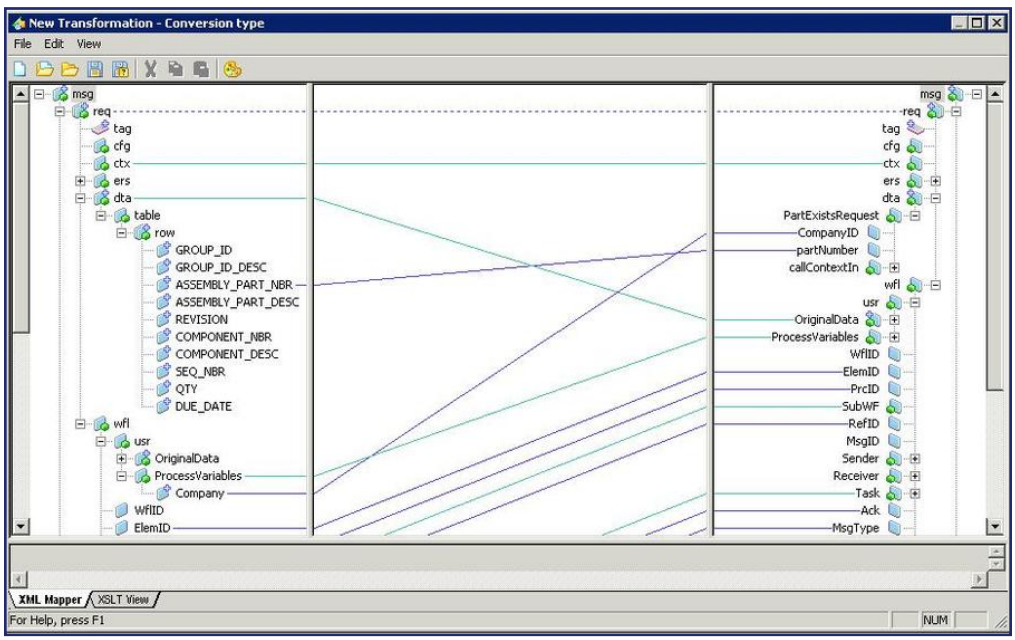
XML Mapper

The XML Mapper is a tool you can use from within a Conversion activity to transform an XML document from one format to another. The initial format of the XML document, referred to as the incoming document, is based on the Input Schema of the Conversion activity. The eventual format of the XML document is based on the Output Schema. The resulting document is referred to as the target document. To transform the file, nodes in the incoming document are mapped to nodes in the target document. The XML Mapper provides a graphical interface that shows both the incoming and target documents as expandable trees. Mappings between nodes are represented as lines that connect the nodes in the incoming document to the nodes in the target document. In addition to the graphical interface, the XML Mapper also has an XSLT view, where you can view and edit the source code directly.

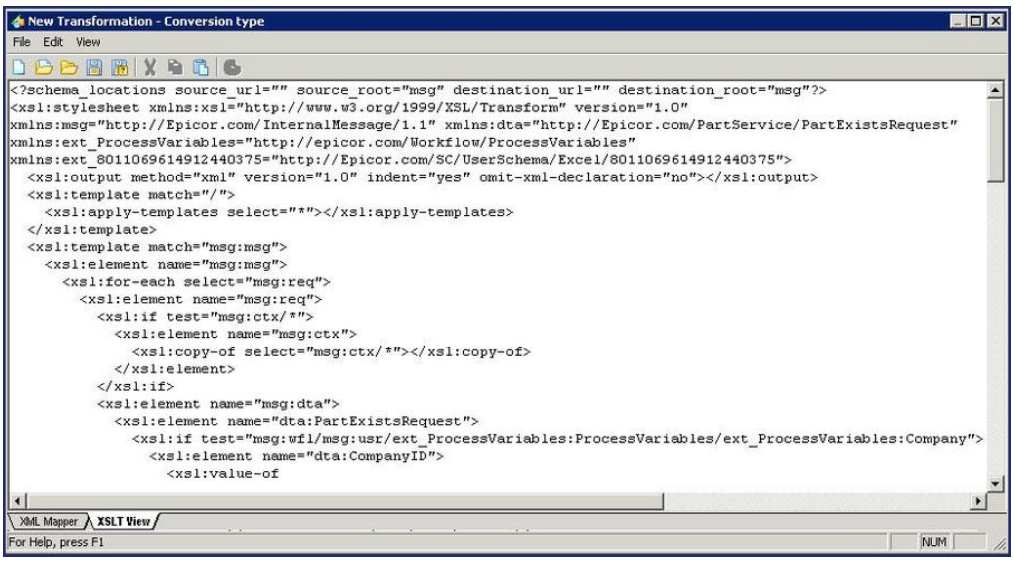
You can open the XML Mapper from the Configuration field of a Conversion Properties sheet by clicking Edit.

This section of the document explains the XML Mapper interface and describes the various tools and techniques you can apply to a transformation.

The following graphic displays a transformation represented in the XML Mapper graphical interface. In this example, the ASSEMBLY_PART_NBR and Company nodes in the incoming document on the left side are mapped to the corresponding nodes in the target document on the right side. The target document uses the request schema for a Web Method call.



This graphic displays the XSLT View of the same transformation file.



XML Mapper Graphical Interface

The XML Mapper is divided into three main areas:

- The incoming document pane on the left
- The center pane, which is used for Functoids
- The target document pane on the right

The documents in the incoming and target document panes are represented as expandable trees. Each branch of the tree represents an XML node in the document. For a description of the major nodes in a Service Connect XML document, review the Internal Envelope Structure section in Chapter 1: Epicor Service Connect Overview. Typically, transforms will use nodes located within the business data node (dta) and user data node (usr, a child of wfl).

Document Nodes

There are four basic types of nodes:

- Simple
- Complex
- Attribute
- Fake

Simple Nodes

Simple nodes represent XML elements that contain parsed character data (PCDATA). Simple nodes are typically mapped to each other using a one-to one relationship, or many-to-many if the nodes represent collections. The following table shows how simple nodes appear in the XML Mapper.

PCDATA refers to the text values in an XML document that are not markup. Such values can include strings, datetimes, or integers.

Node Icon	Description
	A single, simple node. This element can display only once within its current context.
	A simple collection node. A collection means that more than one of this element can occur within the current context. The icon represents all occurrences of the element.

Complex Nodes

Complex nodes represent XML elements that can contain PCDATA and other elements. Complex nodes are typically mapped to each other using a deep copy, which means all the child node values of the complex node in the incoming document are automatically copied to the child nodes of the complex node in the target document. For a deep copy to function, the structure of the complex nodes must be the same in both documents. The following table shows how complex nodes appear in the XML Mapper.

Node Icon	Description
	A single, complex node. The element can appear only once within its current context. If the element has child nodes, you can expand and collapse it. **Example:** The business data node (dta) is an example. Only one dta node is allowed in the Service Connect internal message envelope, and the node can contain a complex hierarchy of child nodes.
	A complex collection node. This element can appear multiple times within its current context. This icon represents all occurrences of the element. If the element has child nodes, you can expand and collapse it. **Example:** A sales order document can contain header information plus multiple rows, one for each line item. The row element is a complex collection because there can be more than one row, and each row can contain several child nodes to describe the item sold on that row. Tip: As a rule, you create for-each loop only between collection. If either left or right node is not a collection then the for-each is not directly available because of the following reasons: • If the left node is not a collection then only one node can be evaluated and then this looping makes no sense. • If the right node is not a collection then looping creates several output nodes in place where only one is available. Such output xml does not correspond to scheme and can be rejected. To create a for-each loop between nodes that are not collections, drag the link with SHIFT key pressed. The Mapper displays a warning notifying that the selected source schema node is not a collection.

Attribute Nodes

Attribute nodes, which provide additional information about an element node, appear as child nodes to their respective elements. Attribute nodes appear as follows.

Fake Nodes

Fake nodes are nodes in the target document mapped to a node in the incoming document, but the target node no longer exists. Fake nodes can occur, for example, when Web References are removed from the ESC Administration Console. Fake nodes appear as follows.

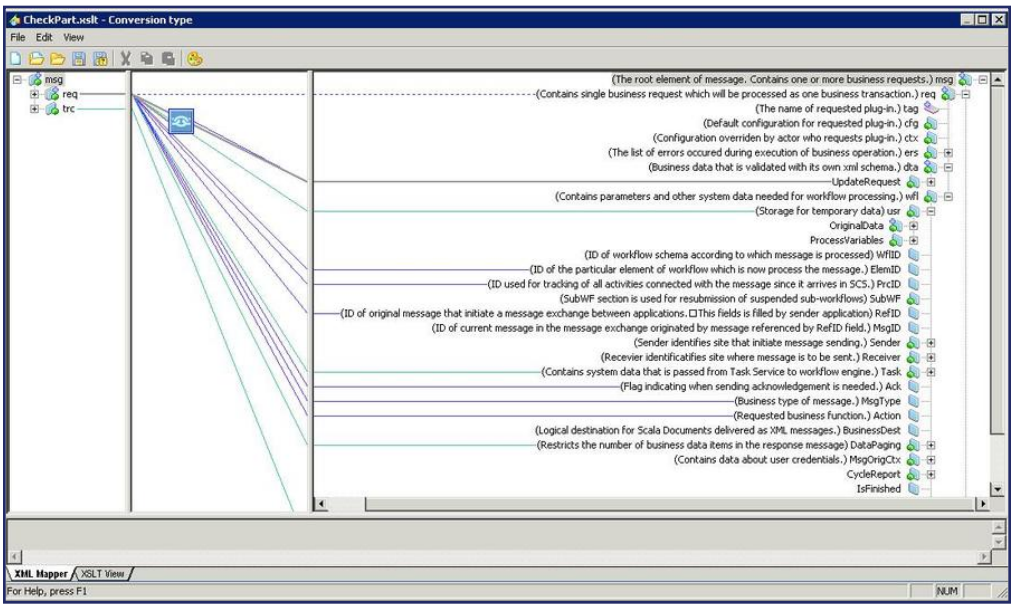
Node Characteristics

The node icons in the XML Mapper can be augmented to indicate other node characteristics. The following table shows the icons used in the XML Mapper.

Node Icon	Description
	A required node. The schema requires that this node is present in the output document. Required nodes do not always need to have a value. Sometimes, they can just be present. If the node's data type is string, you can force an empty node using the Set Literal Value dialog box. Review the Literal Values section in this chapter for more information. Other data types, such as integer, require an appropriate value, and an empty node will generate an error. Review the Node Annotations and Data Types section in this chapter for more information about how to view a node's data type.
	A node where a literal value has been set. Review the Literal Values section in this chapter for more information.
	A target node set to receive a value from an incoming document node. When one node is mapped to another, a condition is placed on the target node. You can use the XPath builder tool to edit the condition. Review the Node Conditions section in this chapter for more information.
	A node where custom XSLT code has been added to the transformation file. You can add custom code using the XSLT View or by opening the XSLT file in a text or XSLT editor. Unless you are very familiar with XSLT, Epicor does not recommend you add custom code.

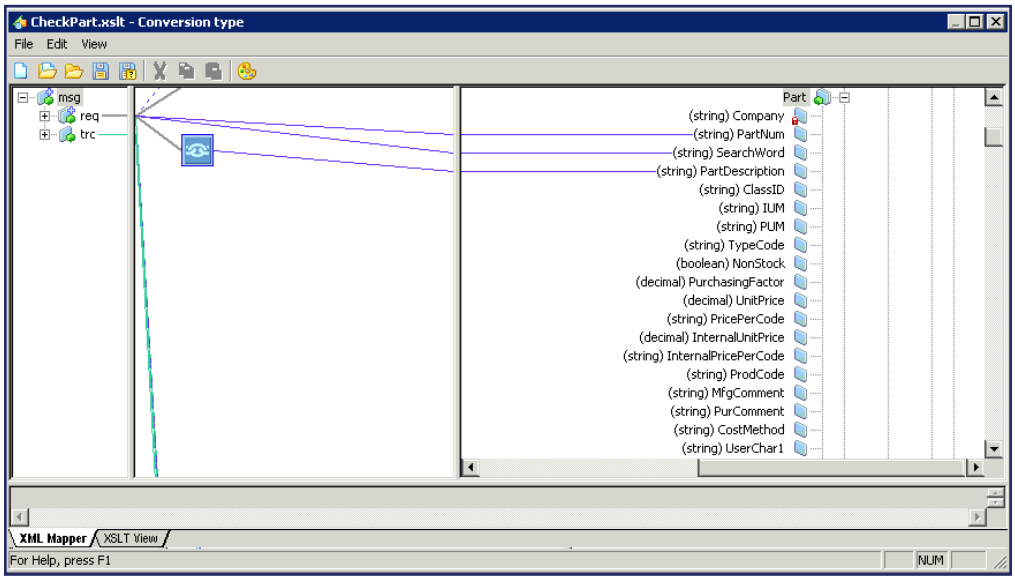
Node Annotations and Data Types

The XML Mapper can show or hide additional information about the incoming and target documents that can help define the structure of the documents and data types each node requires.



Annotations are descriptions of each node in the Service Connect internal envelope structure. To show the node annotations, right-click anywhere in the incoming or target document pane, highlight Annotations, and select Show annotations. The following graphic shows the annotations for a target document. Notice the OriginalData and ProcessVariables nodes do not have annotations. It is possible to add annotations to user-defined schemas. For an example of how to add annotations to a schema, see the ScaInternalMsg.xsd file, located in the SCS\Schemas\epicor folder where Service Connect is installed.

Data types indicate the type of data required by the XSD schema for the node. To show the data types, right-click anywhere in the incoming or target document pane, select Annotations, and then select Show types. The following graphic shows the data types for the target document.

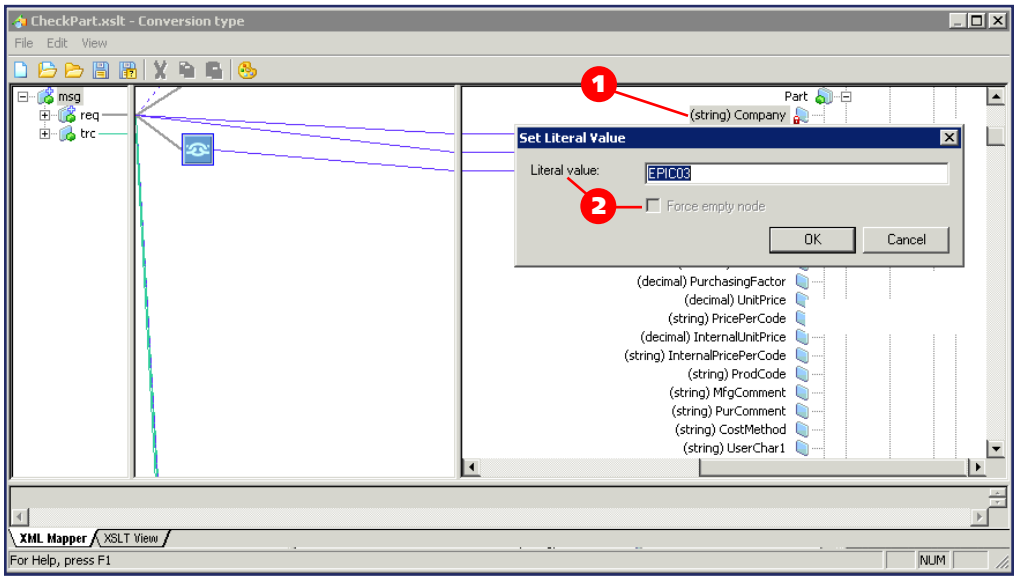


Apply a Literal Value

On some occasions, you must supply a value to a node in the target document, but there is no node to map to it from the incoming document. For these instances, you can apply a literal value to a node.

To apply a literal value to a node:

1. Right-click a node in the target document and select **Set Literal Value**.
2. The **Set Literal Value** window displays. Enter a **Literal value** or select the **Force empty node** check box to indicate the node should display in the target document but will not have a value.



The value should correspond to the data type the target document node expects. Although the Set Literal Value dialog box does not specify the data type of the value, Service Connect will attempt to convert literal values to the appropriate type.

In this example, the value of the Company node is set to the string value EPIC03.

Set Up a Node Condition

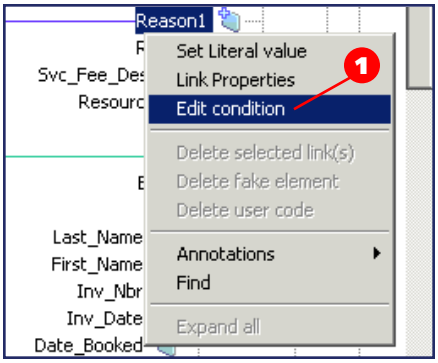
A node condition acts as criteria that specifies whether the node in the target document is created. If the condition is satisfied, the node in the target document is created. If the condition is not satisfied, the node is omitted.

Use node conditions with caution. If a node is omitted from the target document, it could violate the document's schema and cause an error condition.

To set up a node condition:

1. Right-click a node in the target document and select **Edit condition**. The node must be linked to a node in the incoming document or have a literal value set.

Use the **XPath builder** to construct the criteria that must be satisfied to create the node in the target document. Review the Link Conditions section later in this chapter for a brief example of how to use the XPath builder to create criteria.



Node Mappings

A mapping is a link between a node in the incoming document to a node in the target document. The XML Mapper uses a line that connects two nodes to represent a mapping. Just as different types of nodes are in a document, you can make different types of mappings between nodes.

You can map incoming document nodes to multiple target document nodes. Each target document node accepts only one mapping or literal value.

The following table shows visual representations of each type of mapping.

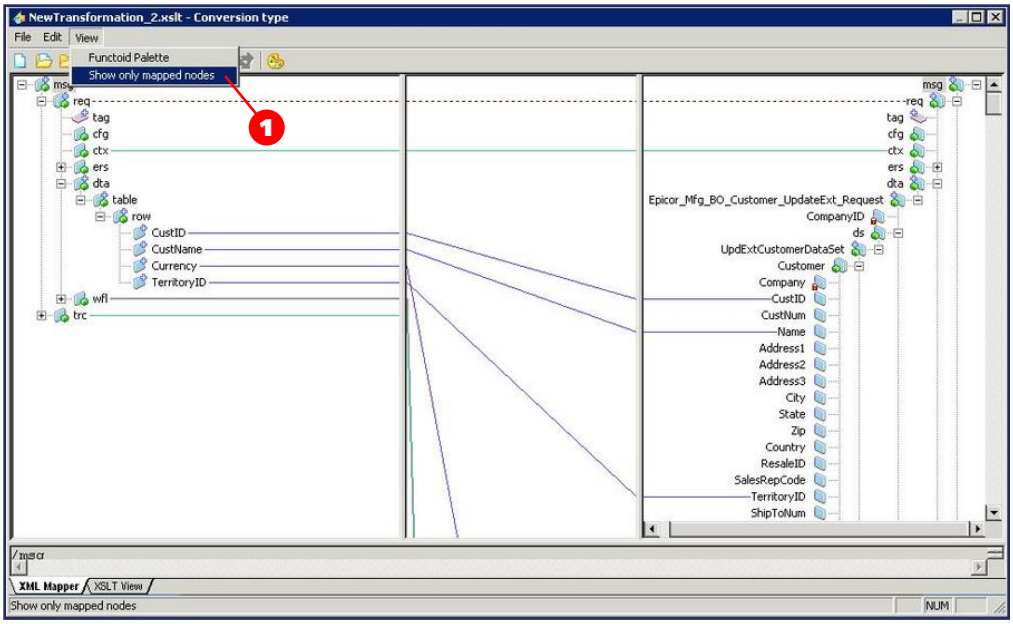
Mapping	Description
	A mapping between two single, simple nodes. This mapping is represented as a solid, blue line.
	A mapping between two single, complex nodes. This mapping is referred to as a deep copy because all the child nodes from the incoming document will be mapped to the target document. The structure of the complex nodes must be identical for this type of mapping to function. This mapping is represented as a solid, green line.
	A mapping between two simple collection nodes. For every XML element represented by the incoming collection node, an identical XML element will be created in the target collection node. This mapping is represented as a dashed, blue line.
	A secondary collection mapping. This mapping is also referred to as a merge because you can use it to merge values to a target document collection node that has already been mapped to a collection node from the incoming document. Thus, the child nodes from both the primary and secondary collection nodes of the incoming document can be mapped to the child nodes of the collection node in the target document. This mapping is represented as a dashed, gray line.
	A mapping where a condition has been applied. This mapping indicates the node value is modified by a link condition or functoid. This mapping is represented as a solid, purple line.
	A group of mappings that are collapsed together. To view individual mappings, expand the nodes. This mapping is represented as a solid, gray line thicker than the other mapping lines.
	A broken mapping. This mapping indicates a node in the target document was once mapped to a node in the incoming document, but the node in the incoming document cannot be found. This mapping is represented as a solid, red line. When the mapping line reaches the center pane, it ends in a squiggle.

View Mapped Nodes Only

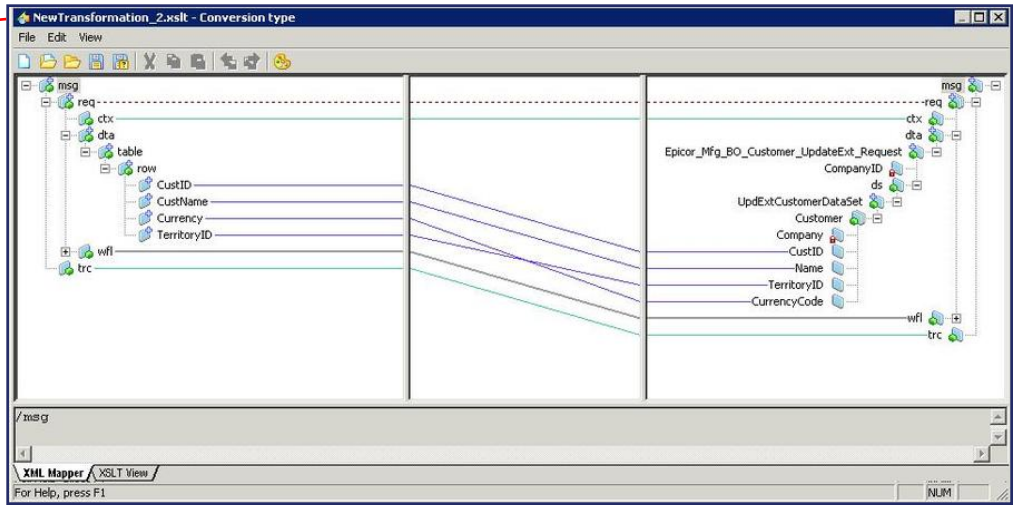
It is possible to only display nodes mapped between the incoming and target document.

To show only mapped nodes in XML Mapper:

1. From the **View** menu, select **Show only mapped nodes**.



2. View the simplified graphical interface that only includes mapped nodes and literal values.

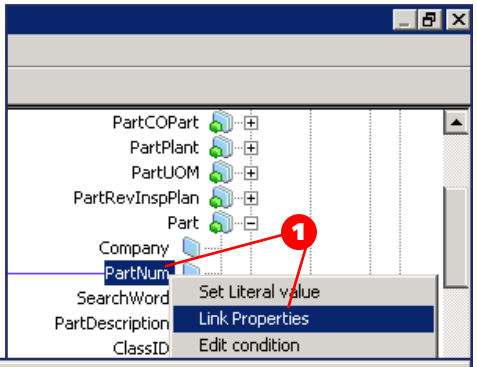


Set Up a Link Condition

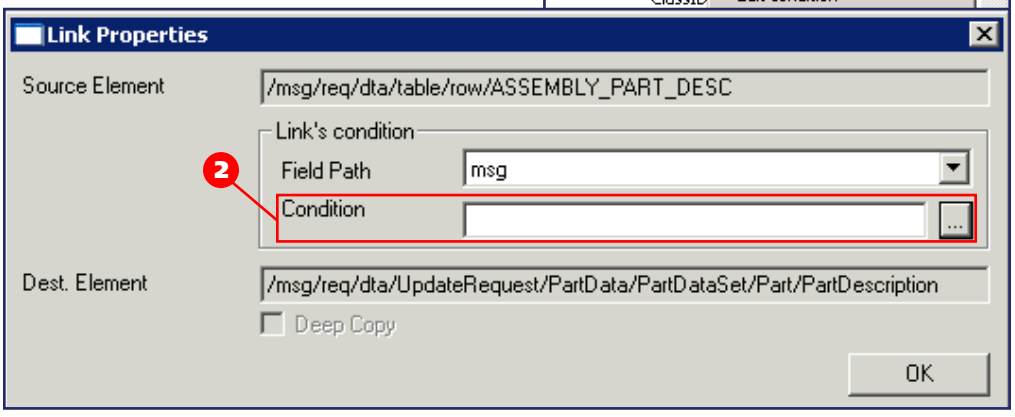
A link condition acts as selection criteria that can be evaluated against information passed from an incoming document node to a target document node.

To set up a link condition:

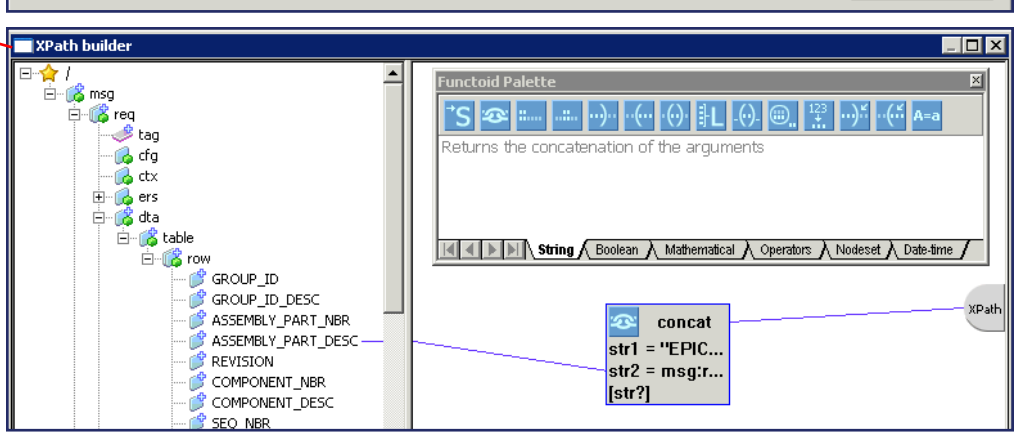
1. Right-click a node in the target document and select **Link Properties**. The node must be linked to a node in the incoming document.



2. The **Link Properties** window displays. In the **Condition** field, enter an XPath expression to use as the selection criteria or click the **... (Ellipse)** button next to the Condition field to use the XPath builder.



3. The **XPath builder** displays. Create a rule that will act as the selection criteria. The Functoid Palette is available to help create the rule.

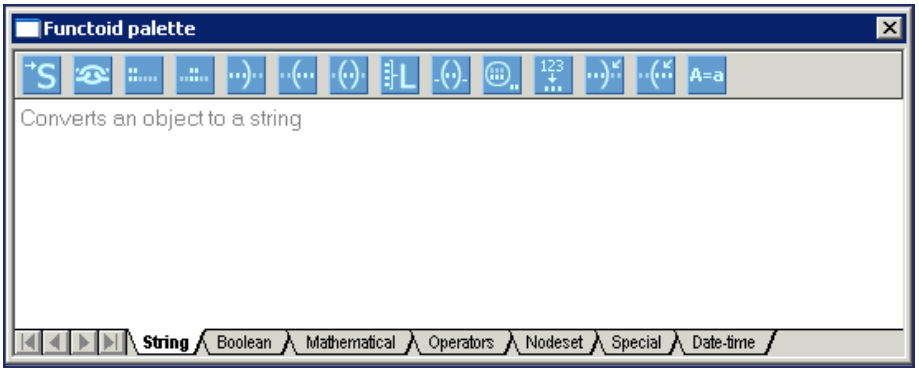


In this example, the PartDescription node in the target document is a concatenation of a string and an incoming document node.

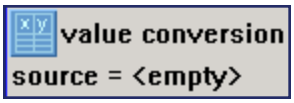
Functoids

Functoids are a collection of XPath functions and extension functions you can add in the center pane of the XML Mapper window to perform a variety of tasks, such as comparisons, mathematical operations, data type conversions, and so on. You can use the Functoid Palette to add functoids to a Conversion. Some functoids require that one or more nodes be mapped to it from the incoming document. You can map the result of the functoid to another functoid or to a node in the target document.

This graphic displays the Functoid Palette.

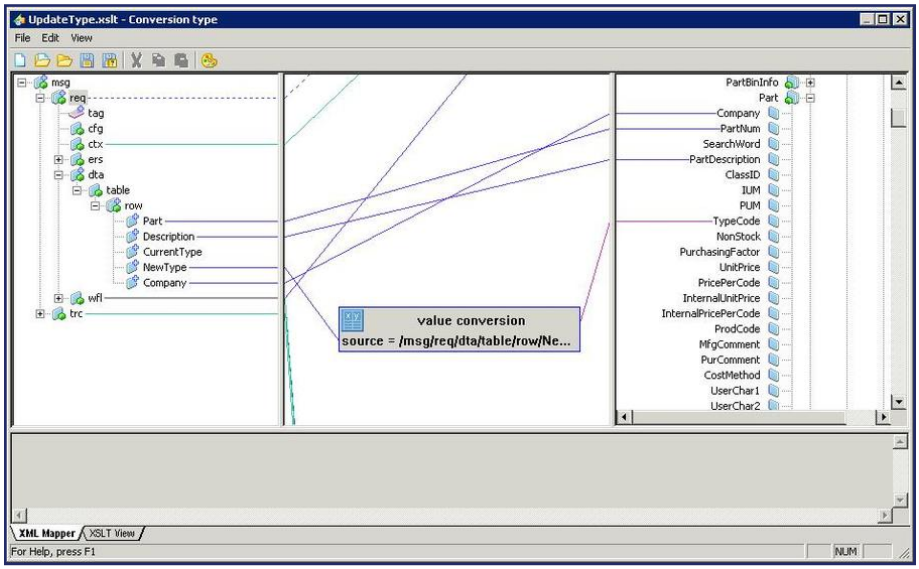


This graphic shows how a functoid appears when it is added to the center pane of the XML Mapper. You can map a node in the incoming document to the equation source = <empty>. In this case, the functoid will return the conversion of the values from the supplied node. You can map the phrase value conversion to a node in the target document. You can double-click a functoid to view or modify its properties in a dialog box.



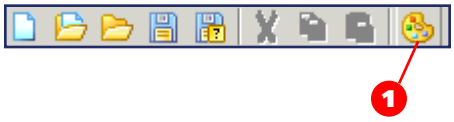
This graphic shows the value conversion functoid where an incoming document node is mapped as the input argument and the result of the functoid is mapped to a target document node.

Functoids can be shown in a collapsed state to simplify the appearance of the center pane. Right-click a functoid to expand or collapse it. Right-click a blank area in the center pane to expand or collapse all functoids. A functoid must be expanded before you can create mappings to or from it, but you can still double-click the collapsed functoid to access the functoid's dialog box.



Add a Functoid to a Conversion

To add a functoid to a Conversion:



1. Click the **Functoid Palette** button on the **Standard** toolbar.
2. Select the appropriate tab at the bottom of the Functoid Palette.

   Point to a functoid button in the palette to see its description.
3. Click a functoid button and drag the functoid to the center pane of the XML Mapper.
4. If the functoid requires input arguments, do one or more of the following:
   - Map a node from the source document.
   - Open the functoid's Properties dialog box to add literal values.
   - Map the output of another functoid as the input argument.
5. Map the functoid to another functoid or map the functoid to a node in the target document.

The following tables list each functoid, sorted by type.

String Functoids

Functoid	Name	Description
	string	Converts an object to a string.
	concat	Returns a concatenation of the arguments. You can map the arguments in from an incoming document node or another functoid, or they can be literal values.
	starts-with	Returns true if the first argument starts with the second argument; otherwise, it returns false.
	contains	Returns true if the first argument contains the second argument; otherwise, it returns false.
	substring-before	Returns the portion of the first argument that precedes the first occurrence of the second argument. It returns an empty string if the first argument does not contain the second argument.
	substring-after	Returns the portion of the first argument that follows the first occurrence of the second argument. It returns an empty string if the first argument does not contain the second argument.
	substring	Returns the substring of the first argument, starting at the position specified in the second argument and the length specified in the third argument.
	length	Returns the number of characters in a string.
	normalize-space	Strips insignificant white space from a string. Any duplicate spaces and trailing white space characters are eliminated.
	translate	Returns the first argument with occurrences of characters in the second argument that have been replaced by the character located at the corresponding position in the third argument.
	number	Converts numbers to strings. The conversion results from formatting the number specified in the first argument (number) using the format specified in the second argument (string) and applying the rules defined in the decimal format identified by the third optional argument (string) and the corresponding <xsl:decimal-format> element.
	rightSubstringAfter	Returns the substring of the first argument string that precedes the last occurrence of the second argument string in the first argument string or an empty string if the first argument string does not contain the second argument string.
	rightSubstringBefore	Returns the substring of the first argument string that follows the last occurrence of the second argument string in the first argument string or an empty string if the first argument string does not contain the second argument string.
	CompareNoCase	Compares two strings in case-insensitive mode.

Boolean Functoids

Functoid	Name	Description
	boolean	Converts an argument to a Boolean.
	not	Returns true if the argument is false; otherwise, it returns false.
	true	Returns true.
	false	Returns false.
	lang	Returns true or false depending on whether the language of the context node, as specified by xml:lang attributes, is the same as, or is a sublanguage of, the language specified by the argument string. The language of the context node is determined by the value of the xml:lang attribute on the context node, or, if the context node has no xml:lang attribute, by the value of the xml:lang attribute on the nearest ancestor of the context node that has an xml:lang attribute. If there is no such attribute, then lang returns false. If there is such an attribute, then lang returns true if the attribute value equals the argument ignoring case or if there is some suffix starting with – such that the attribute value equals the argument ignoring that suffix of the attribute value and ignoring case.

Mathematical Functoids

Functoid	Name	Description
	number	Converts the argument to a number.
	add	Adds two or more numbers and returns the sum.
	subtract	Subtracts numbers and returns the difference.
	multiply	Multiplies two or more numbers and returns the product.
	div	Divides numbers and returns the quotient. This is a floating-point division, which means the value returned may include a fractional part -10 div 4 equals 2.5, not 2 or 3.
	mod	Returns the modulus, that is, divides number 1 by number 2 and returns the remainder.
	negative	Returns the negative value of the argument.
	ceiling	Returns the smallest integer that is not less than the argument.
	floor	Returns the largest integer that is not greater than the argument.
	round	Returns the integer closest in value to the argument.
	sum	Returns the sum of all the nodes in the node-set. Each node is first converted to a number value.

Functoid	Name	Description
	min	Returns the minimal value of all the nodes in the node-set. Each node is first converted to a number value.
	max	Returns the maximum value of all the nodes in the node-set. Each node is first converted to a number value.
	avg	Returns the average value of all the nodes in the node-set. Each node is first converted to a number value.

Operators

Functoid	Name	Description
	equal	Compares whether the first argument equals the second.
	not equal	Compares whether the first argument does not equal the second.
	less than	Compares whether the first argument is less than the second argument.
	greater than	Compares whether the first argument is greater than the second argument.
	less than or equal to	Compares whether the first argument is less than or equal to the second argument.
	greater than or equal to	Compares whether the first argument is greater than or equal to the second argument.
	and	Logical and. You can use this functoid to string other functoids together into a logical expression.
	or	Logical or. You can use this functoid to string other functoids together into a logical expression.
	union	Returns the union of several operands, which must be node-sets. This preserves the document order and does not return duplicates. Use this functoid to move different nodes from different places in the Input Schema to the sequence sub-nodes in the Output Schema.

Nodeset Functoids

Functoid	Name	Description
	last	Returns a number equal to the context size from the expression evaluation context.
	position	Returns a number equal to the context position from the expression evaluation context.
	count	Returns the number of nodes in the node-set argument.
	id	Selects elements by their unique ID. When the argument to id is of the node-set type, then the result is the union of the result of applying the id to the string-value of each of the nodes in the argument node-set. When the argument to id is of any other type, the argument is converted to a string as if by a call to the string function; the string is split into a whitespace-separated list of tokens; the result is a node-set that contains the elements in the same document as the context node that has a unique ID equal to any of the tokens in the list.

Functoid	Name	Description
	local-name	Returns the local part of the expanded name of the node in the node-set argument that is first in the document order.
	name	Returns a string that contains a QName that represents the expanded name of the node in the node-set argument that is first in the document order.
	namespace-uri	Returns the namespace Uniform Resource Identifier (URI) of the expanded name of the node in the node-set argument that is first in the document order.

Special Functoids

Functoid	Name	Description
	conditional	Chooses from a default value or a list of alternatives based on a user-defined condition. Review the Conditional Functoid Example section later in this chapter for more information.
	value conversion	Converts input values with user-defined rules. You can define the rules in the Value Conversion Functoid dialog box or save them as a set of predefined rules. You can use predefined rules with more than one value conversion functoid. Review the Value Conversion Functoid section later in this chapter for information about how to define conversion rules.
	DB lookup	Performs a database lookup using specified selection criteria.
	createGUID	Creates guid in registry format; for example, {xxxxxxx-xxx ... xxxx}.
	toCDATA	Wraps the node to the CDATA section.

Date-time Functoids

Functoid	Name	Description
	toDateTime	Converts an object to the dateTime data type.
	currentDateTime	Returns the current date.
	local current date	Returns the local current date of the ESC server.
	format-date	Converts a date to a string using the format string specified.
	format-time	Converts time to a string using the format string specified.
	yearFromDateTime	Returns a numeric value that represents the year component in the dateTime value.
	monthFromDateTime	Returns a numeric value between 1 and 12, both inclusive, that represents the month component in the dateTime value.
	dayFromDateTime	Returns a numeric value between 1 and 31, both inclusive, that represents the day component in the dateTime value.
	hoursFromDateTime	Returns a numeric value between 0 and 23, both inclusive, that represents the hours component in the dateTime value.
	minutesFromDateTime	Returns a numeric value between 0 and 59, both inclusive, that represents the minute component in the dateTime value.

Functoid	Name	Description
	secondsFromDateTime	Returns a numeric value greater than or equal to zero and less than 60 that represents the seconds and fractional seconds in the dateTime value.
	subtractDateTimes	Returns the duration that corresponds to the difference between the normalized value of the first argument and the normalized value of the second argument.
	addDurationToDateTime	Returns the dateTime computed by adding the duration specified in the second argument to the dateTime specified in the first argument.
	subtractDurationFromDateTime	Returns the dateTime computed by subtracting the duration specified in the second argument from the dateTime specified in the first argument.
	after-midnight	Returns a number that represents the number of seconds after midnight in the dateTime value.
	stringToDateTime	Converts a string to a dateTime value using a specified format.

Date-time Functoids Format String

The following format string elements are accepted in the Value field of Date-time functoids.

1. dateTimeFromString, format-date functoids

d	Day of month as digits with no leading zero for single-digit days.
dd	Day of month as digits with leading zero for single-digit days.
ddd	Day of week as a three-letter abbreviation. The function uses the LOCALE_SABBREVDAYNAME value associated with the specified locale.
dddd	Day of week as its full name. The function uses the LOCALE_SDAYNAME value associated with the specified locale.
M	Month as digits with no leading zero for single-digit months.
MM	Month as digits with leading zero for single-digit months.
MMM	Month as a three-letter abbreviation. The function uses the LOCALE_SABBREVMONTHNAME value associated with the specified locale.
MMMM	Month as its full name. The function uses the LOCALE_SMONTHNAME value associated with the specified locale.
y	Year as last two digits, but with no leading zero for years less than 10.
yy	Year as last two digits, but with leading zero for years less than 10.

yyyy	Year represented by full four or five digits, depending on the calendar used. Thai Buddhist and Korean calendars both have five digit years. The "yyyy" pattern will show five digits for these two calendars, and four digits for all other supported calendars.
gg	Period/era string. The function uses the CAL_SERASTRING value associated with the specified locale. This element is ignored if the date to be formatted does not have an associated era or period string.

2. dateTimeFromString, format-time functoids

h	Hours as 0-12.
hh	Hours as 00-12.
H	Hours as 0-23.
HH	Hours as 00-23.
m	Minutes as 0-59.
mm	Minutes as 00-59.
s	Seconds as 0-59.
ss	Seconds as 00-59.
tt	Insert AM or PM, display hours as 12-hour clock.
t	Insert A or P, display hours as 12-hour clock.

The format string elements are case sensitive.
Date-time and Duration formats are compatible with the w3org specification (http://www.w3.org/TR/xpath-functions/).
For date-time the format is **YYYY-MM-DDThh:mm:ss.(s)**, for duration: **PnYnMnDTnHnMnS**.

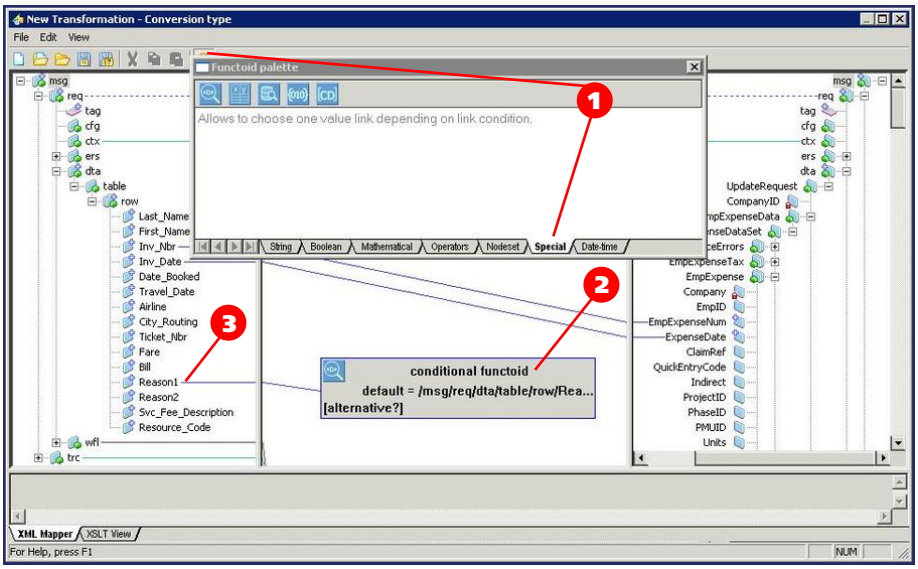
Conditional Functoid Example

An incoming document contains several expenses. If an expense is project-related, the Reason1 node in the incoming document contains the project code, which you want to map to the ProjectID node in the target document. If the expense is internal, the Reason1 node contains text that explains the reason for the expense, which you do not want to map to the ProjectID node. The following steps show how to use a conditional functoid to evaluate the value of another node, Bill, to determine whether the expense is project-related. If Bill = "B", then the Reason1 node contains a project expense. If Bill = "N", then the Reason1 node contains a text description.

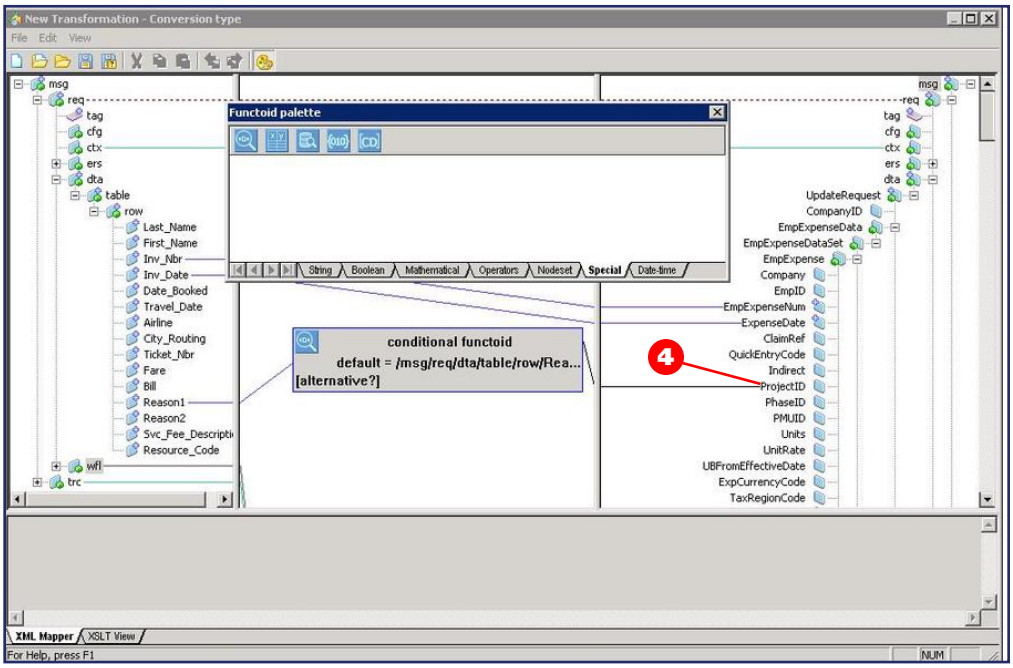
1. Click the **Functoid Palette** button on the **Standard** toolbar. Click the **Special** tab of the Functoid palette.

2. Add a **conditional functoid** to the XML Mapper.

3. Map the **Reason1** node in the incoming document to **default**.



4. Map the conditional functoid to the **ProjectID** node in the target document."

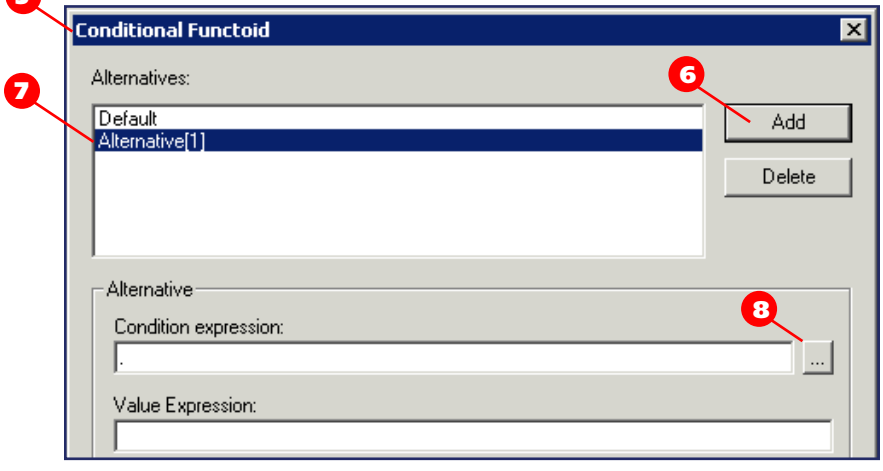


5. Double-click the functoid to open the **Conditional Functoid** window.

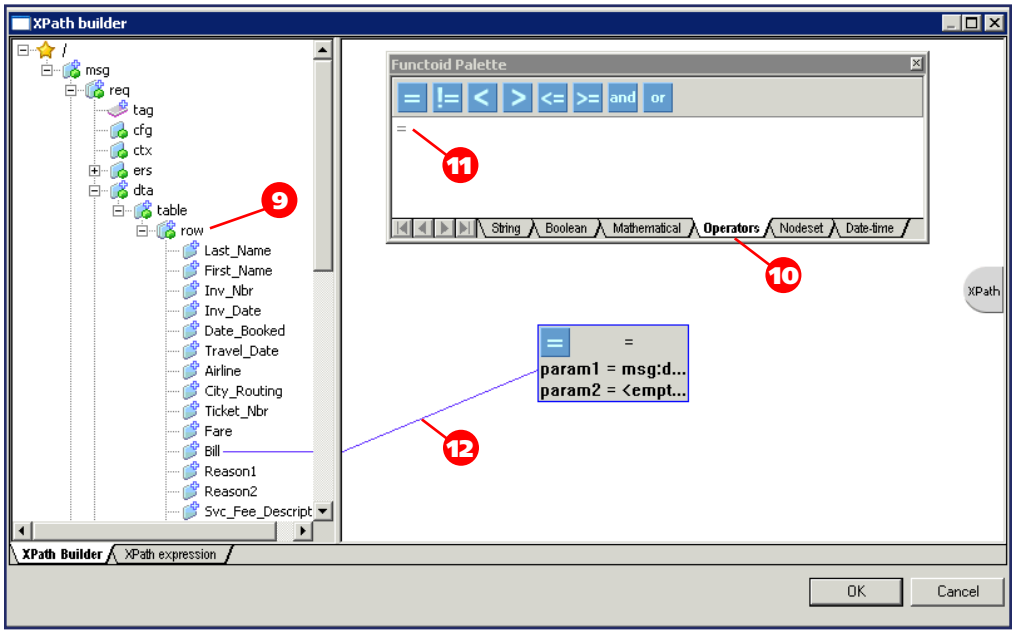
6. Click **Add**.

7. From the **Alternatives** list, select **Alternative[1].**

8. Click the **... (Ellipse)** button next to the **Condition expression** field.



9. In the **Tree View**, expand the **req > dta > table > row** node. Right-click in the right pane and select **Show Functoid Palette**.



> Not all the functoids that display in the XML Mapper also display in the XPath Builder. The list of functoids available in the XPath Builder depends on the context from which the XPath Builder is called. Conditional, Value Conversion, Union, DBOperation, ToCDATA, CreateGUID functoids cannot be used in XPath Builder because they are predefined XSL constructions (<xsl:choose>, <xsl:otherwise>, <xsl:variable>) and are not suitable for XPath.

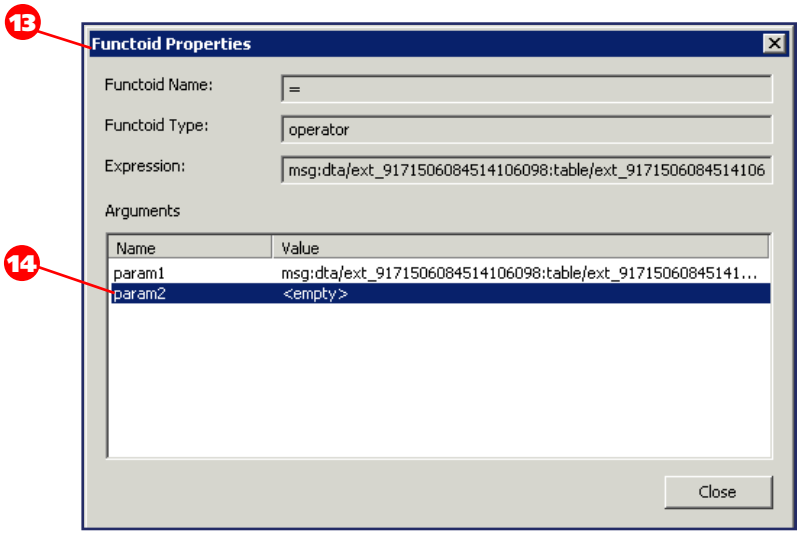
10. In the **Functoid Palette,** click the **Operators** tab.

11. Add an **equals** (=) functoid to the right pane of the XPath Builder.

12. Map the **Bill** node to **param1**.

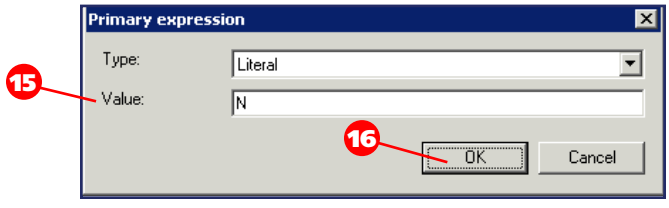
13. Double-click the functoid to open the **Functoid Properties**.

14. Double-click **param2** where the Value is **<empty>**.



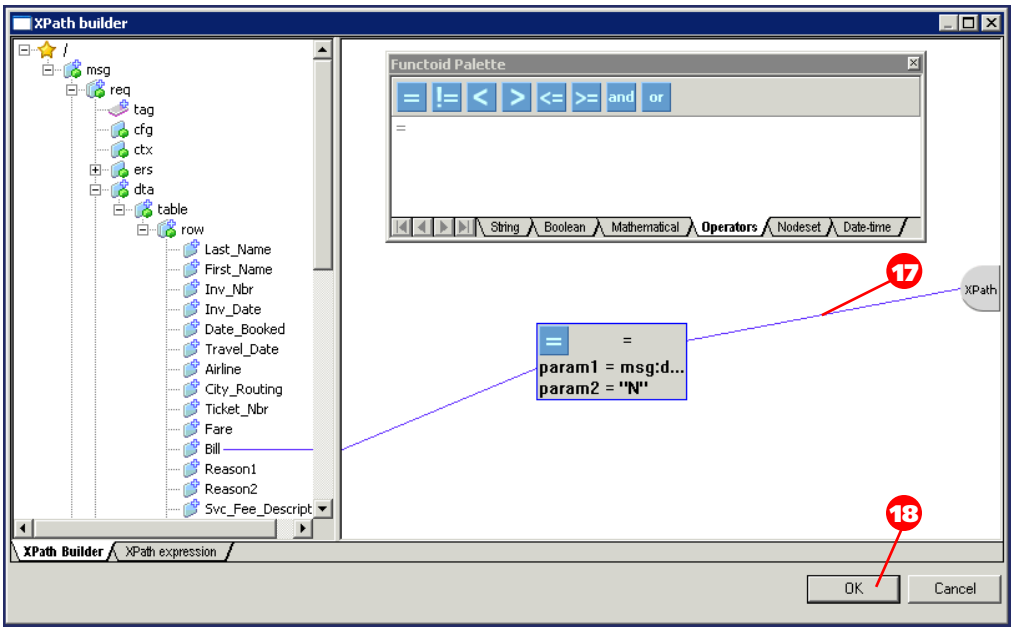
15. The **Primary expression** window displays. Enter **N** as the **Value**.

16. Click **OK** and click **Exit** the Functoid Properties window**.**

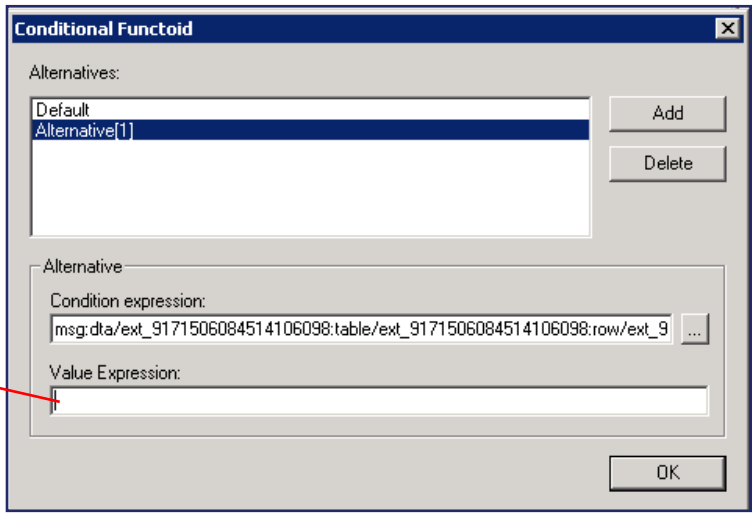


17. Map the functoid to the **XPath** marker.

18. Click **OK** again to return to the Conditional Functoid dialog box.



19. In the **Value Expression** field, press the **Space bar** and then press **Backspace**. This sets the value of the field to an empty string for the Alternative[1] condition.



The default condition for the conditional functoid will map the value of the Reason1 node in the incoming document to the ProjectID node in the target document. However, the alternative condition specifies that if the value of the Bill node for the same expense equals N, which indicates an internal expense, the functoid will map an empty string to the ProjectCode node in the target document. Thus, the value of the Reason1 node in the incoming document will only be mapped to the ProjectID node of the expense update request for project-related expenses.

Conversion Value Functoids

A conversion value functoid uses a collection of value pairs to define conversion rules. Each pair consists of a source value and a target value. If a value from an incoming document node or other functoid matches the source value, it is converted to the target value. You can set a default value to handle an instance where the incoming value does not match any of the source values. The default value can be the target value from one of the value pairs, or it can be a unique value.

You can define conversion rules in two ways:

- **In place** - Rules are defined in the Value Conversion Functoid dialog box.
- **Predefined** - Rules are defined in an external XSLT transform file. You can import the XSLT file by reference into the current transform which allows the rules to be applied to the functoid.

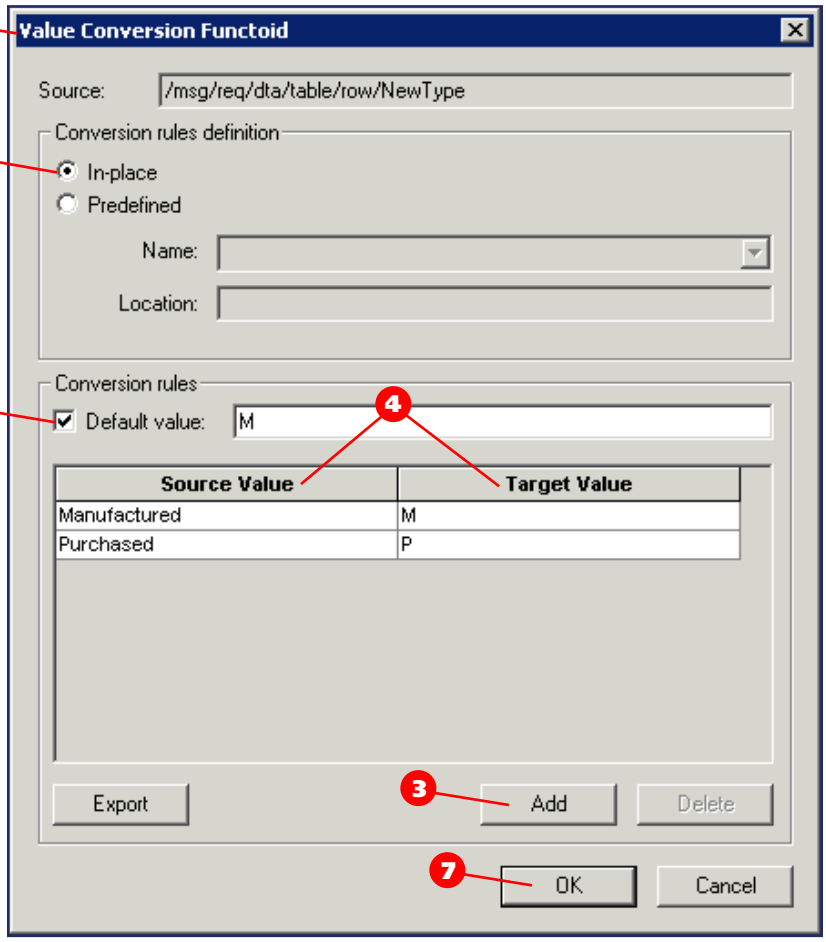
Define Conversion Rules In Place

You can quickly add conversion rules in the Value Conversion Functoid dialog box. This procedure begins after the functoid is added to the center pane and a value is supplied as an input argument.

To define rules in place:

1. Double-click the functoid to open the **Value Conversion Functoid** dialog box.
2. Verify **In-place** is selected as the conversion rule option.
3. Click **Add** to add a row to the Conversion rules grid.
4. Enter the **Source Value** and **Target Value** for the conversion rule.
5. Repeat these steps to add additional conversion rules.
6. You can select the **Default value** check box to supply a value to use if the incoming value does not match any of the Source Values in the grid.

   The default value can be one of the defined Target Values, or it can be a unique value.
7. Click **OK**.



Pre-Defined Conversion Rules

Use pre-defined conversion rules to save conversion rules to a separate file. Thus, the rules can be used by more than one functoid. To define and then use the rules, you must perform the following steps:

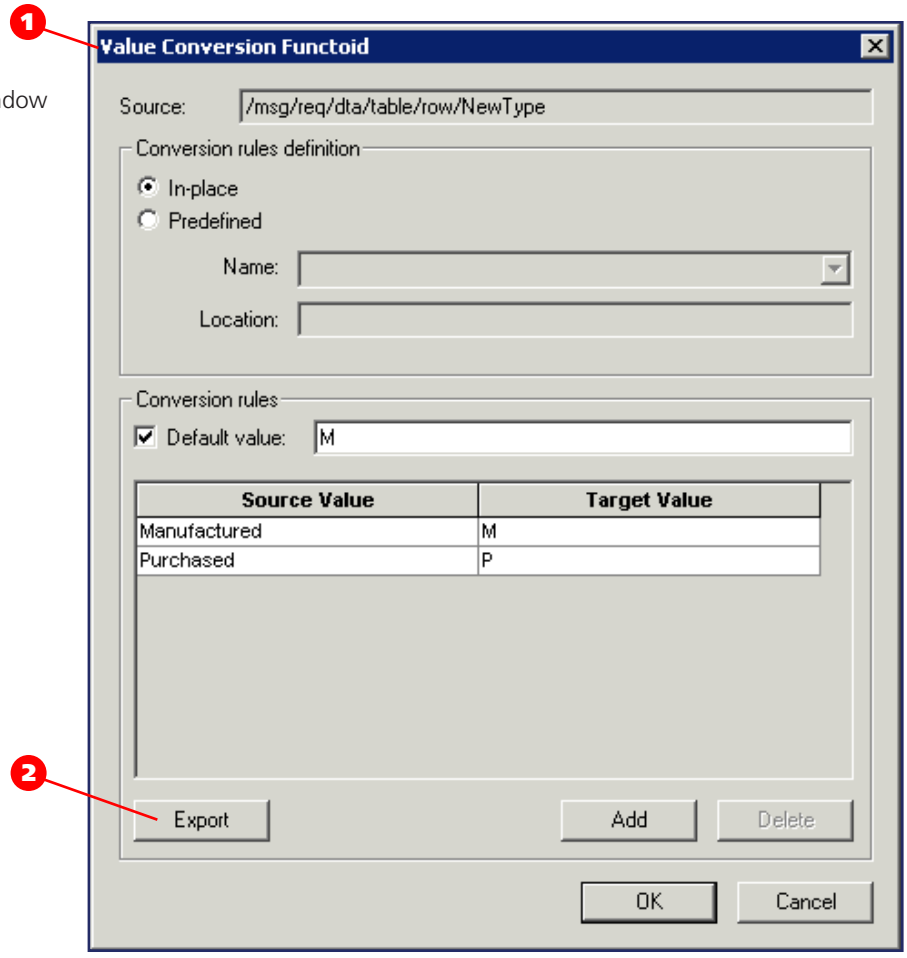
- Create the rules. You can write the rules to an XSLT file; or, you can create the rules in the Value Conversion Functoid dialog box and export them.
- Add a reference to the rules file in the XML Mapper.
- Use the rules file for the conversion functoid.

Create Rules

You can create your own rules file and save it to <Service Connect installation folder>\System\Services\DES\Processes\Custom\Common\Transformations\ValueConversion. However, you may want to create your own rules file as a template first.

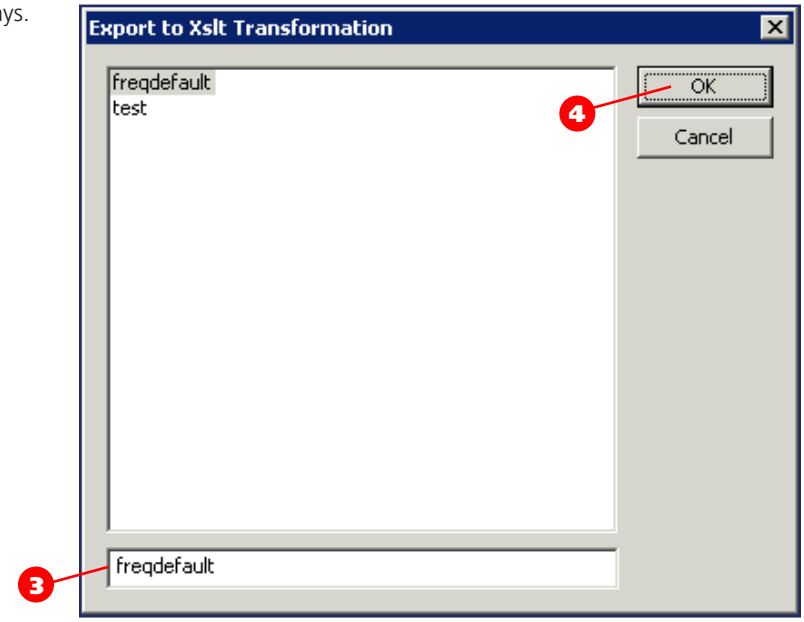
To create pre-defined conversion rules:

1. Follow the steps in the Define Conversion Rules in Place section above to create conversion rules in the **Value Conversion Functoid** window but **do not** click OK.
2. Click **Export**.



3. The **Export to Xslt Transformation** window displays. Enter a name for the transform file.

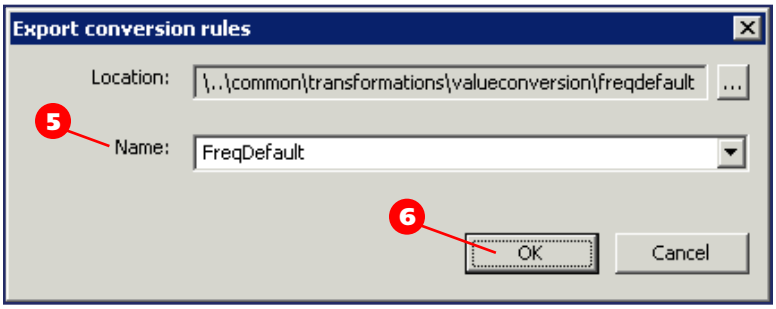
4. Click **OK**.



5. The **Export conversion rules** window displays. Optionally, enter or select a **Name** for the rules.

   The name identifies which set of rules you want to use in the functoid. It is useful to have unique names for each set of rules, especially when more than one set of rules is stored in the same file. Also, you can select the rules name to overwrite an existing set of rules.

6. Click **OK** until you exit all dialog boxes.



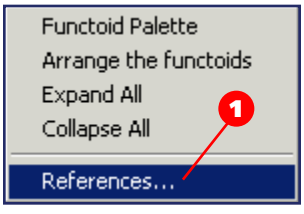
You can save more than one set of rules to the same file. You can then access all the rules by referencing just one file in the XML Mapper. Select an already existing rules file during this step to add a set of rules to it.

Add a Reference to the Rules

Adding a reference makes the rule sets in the rules file available for use in any conversion functoid.

To add a reference:

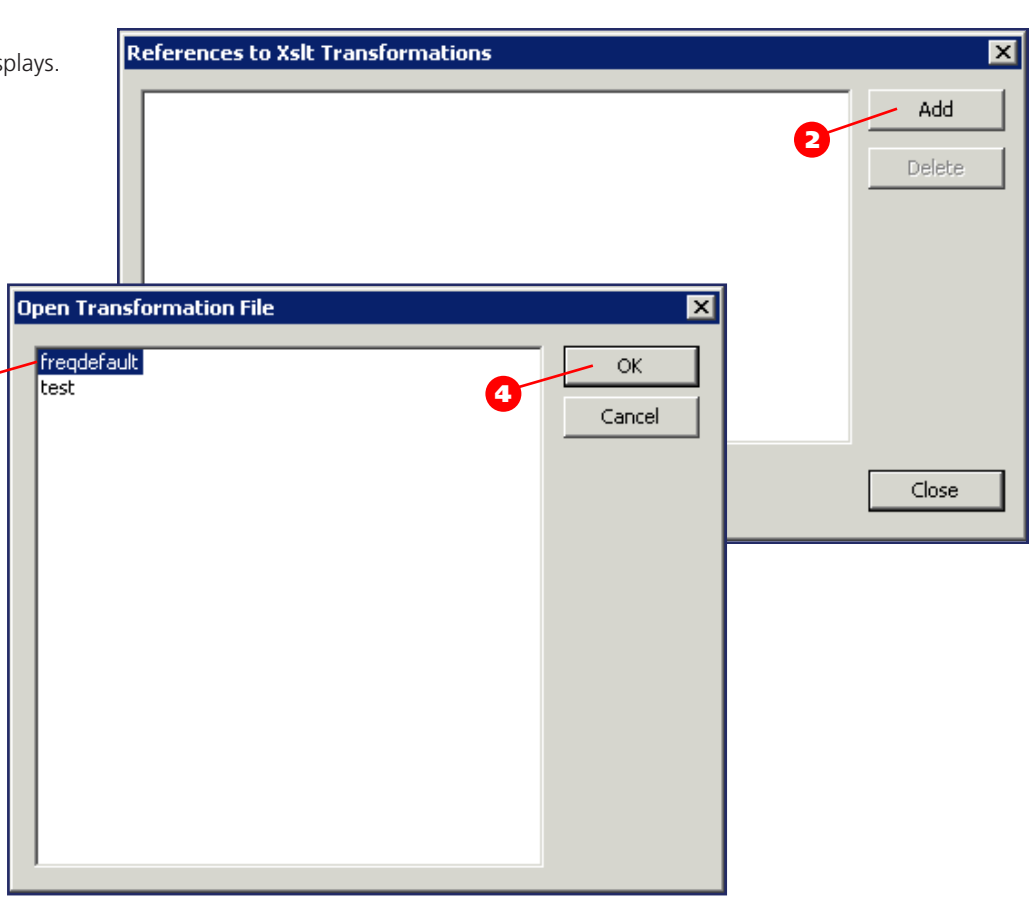
1. Right-click the center pane of the XML Mapper and select **References**.



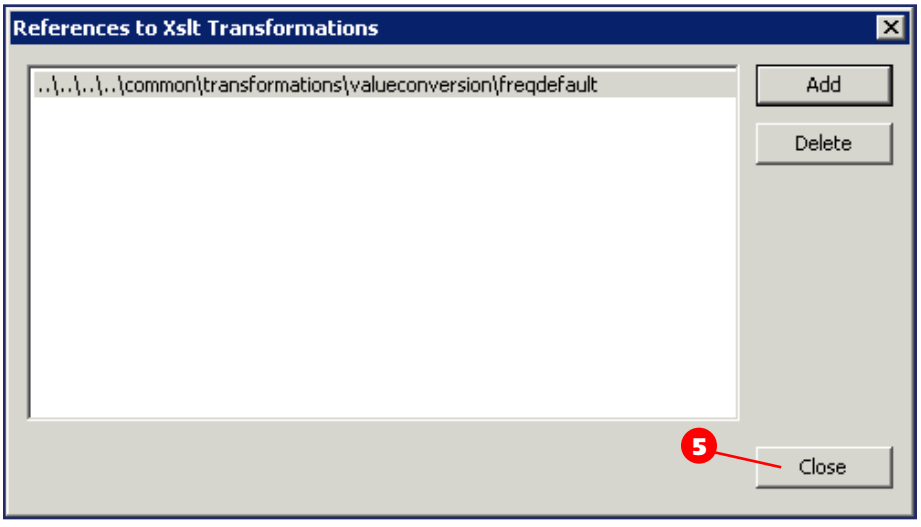
2. The **References to Xslt Transformations** window displays. Click **Add**.

3. The **Open Transformation File** window displays. Select the conversion rules file.

4. Click **OK**.



5. The **References to Xslt Transformations** window displays. Click **Close**.



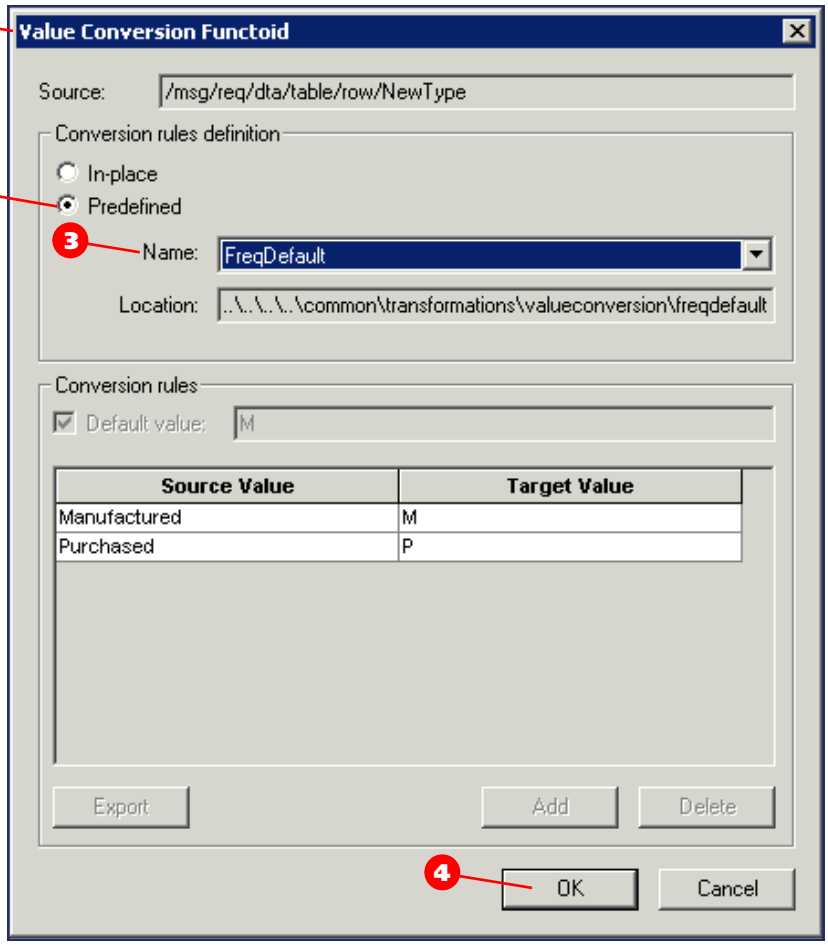
Use the Rules File for the Conversion Functoid

This procedure begins after the functoid is added to the center pane and a value is supplied as an input argument. Remember, you must add the rules file as a reference before you can perform these steps.

To add a reference to the conversion rules in the XML Mapper:

1. Double-click the functoid to open the **Value Conversion Functoid** window.
2. In the **Conversion rules definition** section, select **Predefined**.
3. Select the **Name** of the rule set you want to use.
4. Click **OK**.

You cannot edit the conversion rules. To modify the conversion rules, use the In-place option and recreate the rules. Then, export the rules using the same file and name as the rules you want to replace.

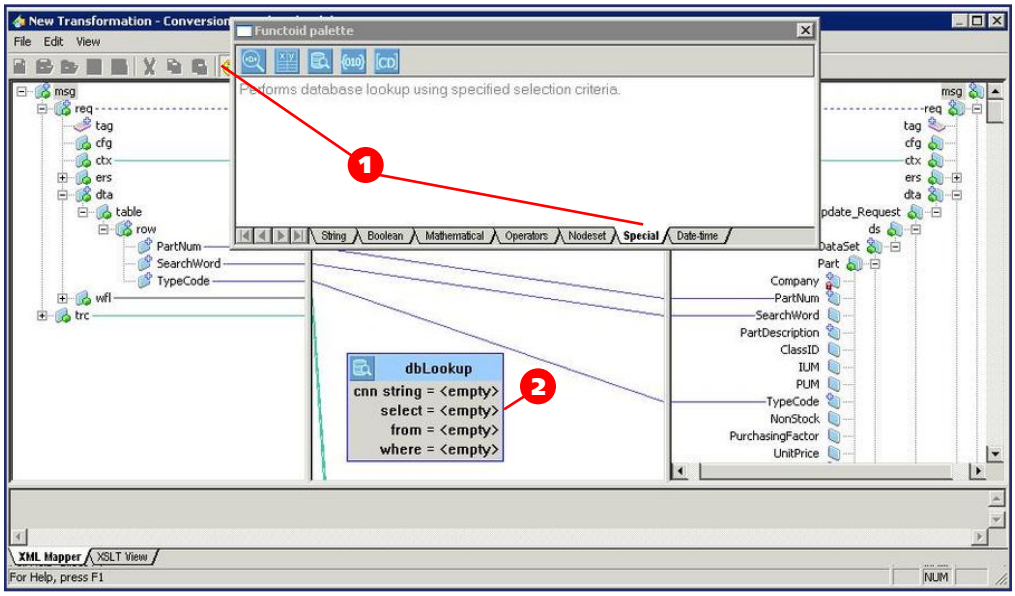


DB Lookup Functoid Example

Update the PartDescription node of the target element with the part number value that comes from the database. This value will be looked up in the target database based on the PartNum value from the incoming document. As a result, the PartDescription node will update for all parts that exist in the database. For non-existing parts, the PartDescription node will not contain a description.

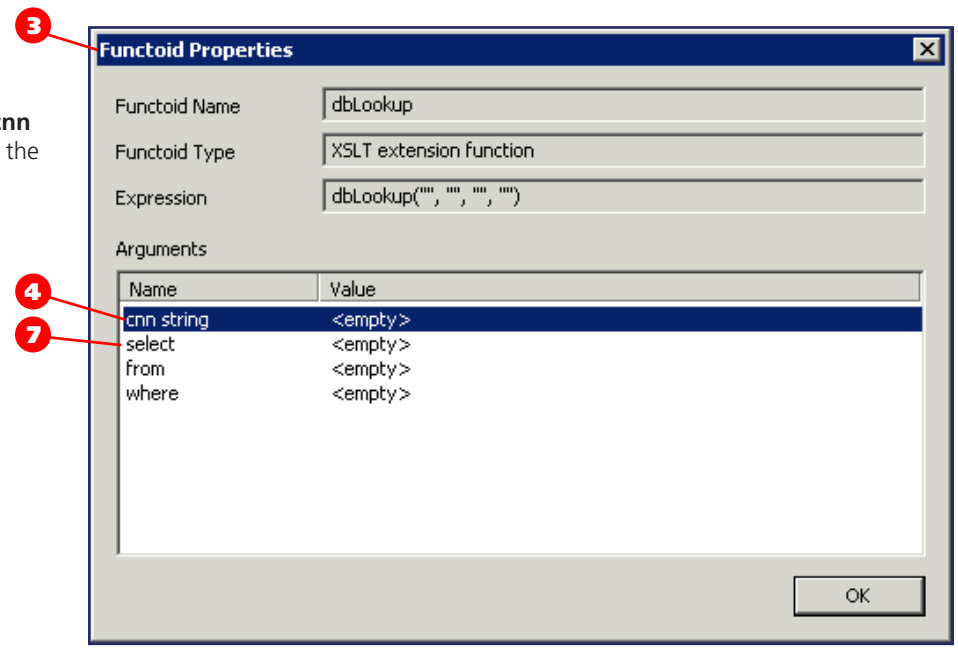
To add the DB Lookup functoid:

1. Click the **Functoid Palette** button on the **Standard** toolbar. Click the **Special** tab.
2. Add a **dbLookup** functoid to the XML Mapper.



3. Double-click the functoid to open the **Functoid Properties** window.

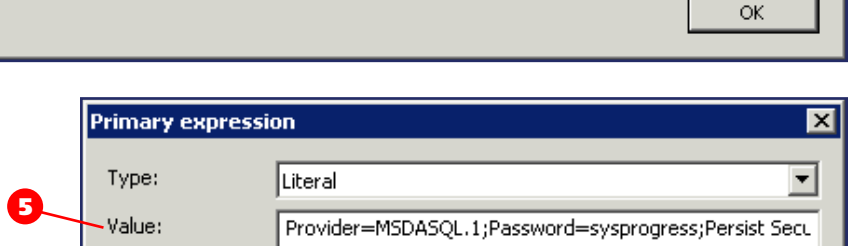
4. In the **Arguments** list, double-click **cnn string** to establish the connection to the database.



5. In the **Value** field, enter the connection string.

6. Click **OK**.

For more information on how to build a connection string, review the DB Operation section earlier in this chapter.

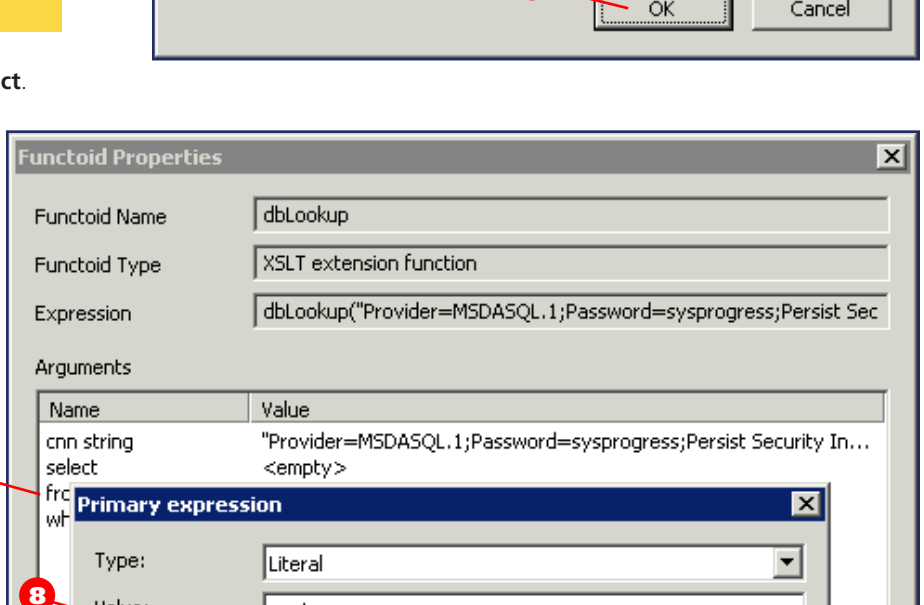


7. In the **Arguments** list, double-click **select**.

8. Specify the **Value** you want to select from the database.

   In this example, the partnum column is selected.

9. In the Arguments list, double-click **from**.

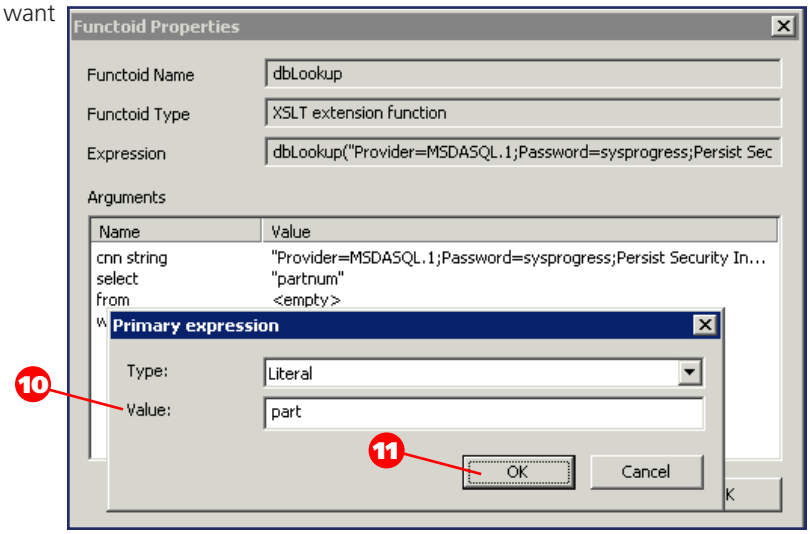


10. In the **Value** field, enter the table from which you want to select data.

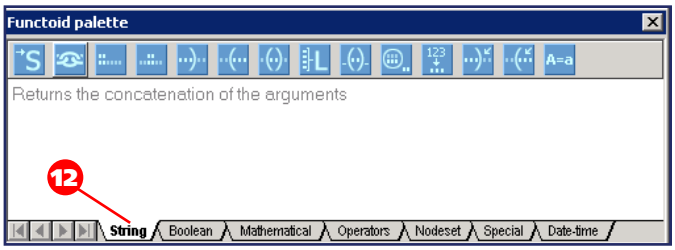
In this example, the part table is selected.

11. Click **OK** to close the Functoid Properties window.

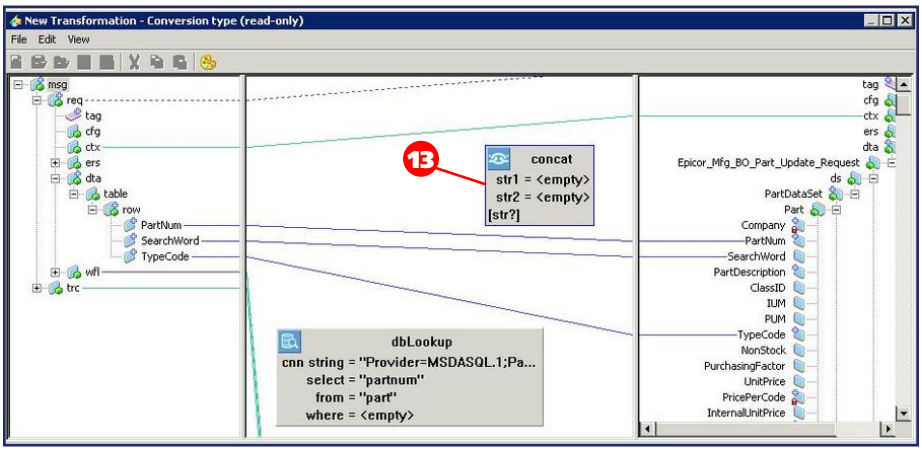
Next, use another functoid to generate the where expression.



12. Click the **String** tab of the Functoid Palette.

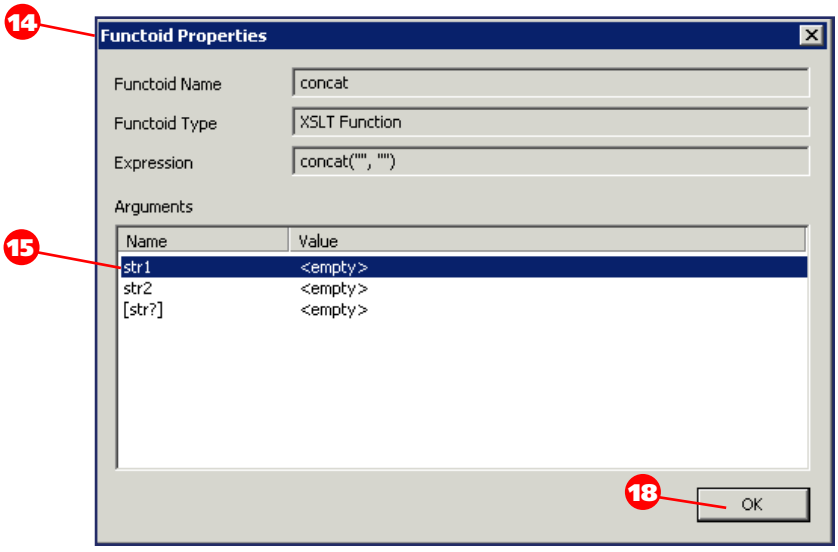


13. Add a **concat functoid** to the XML Mapper.



14. Double-click the functoid to open the **Functoid Properties** window.

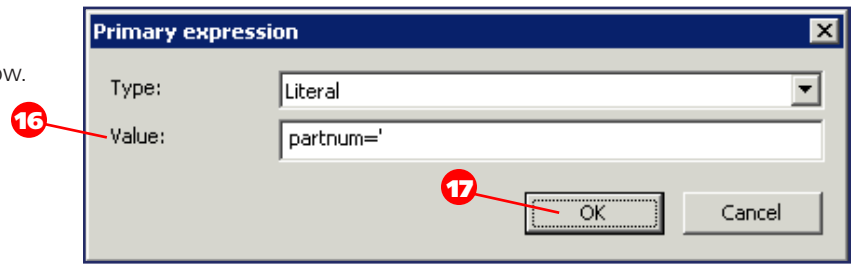
15. In the **Arguments** list, ouble-click **str1.**



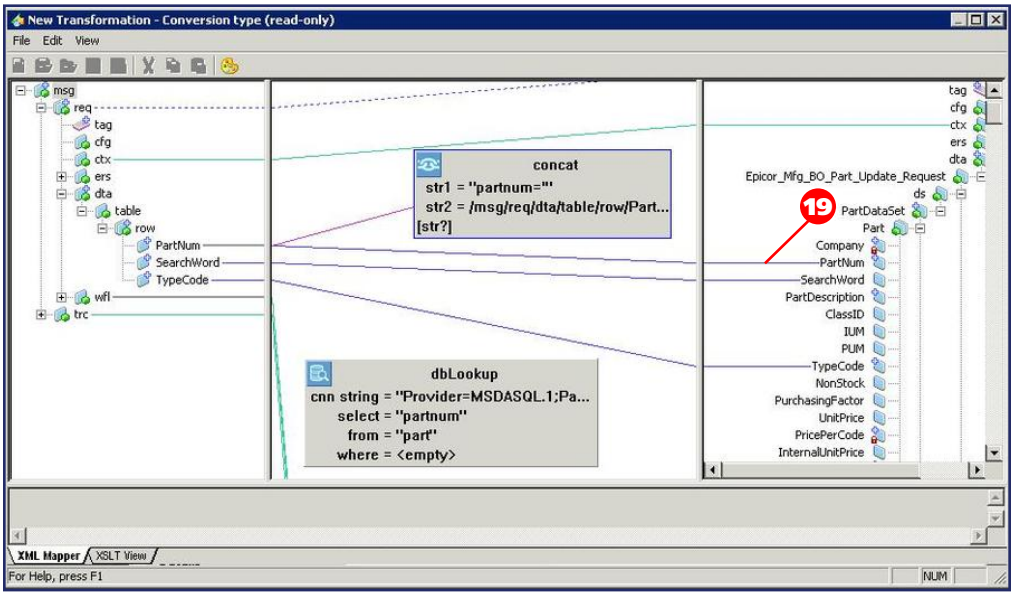
16. In the **Value** field, enter the first argument.

17. Click **OK** to close the Primary expression window.

18. Click **OK** to close the Functoid Properties window.



19. Map the **PartNum** node in the incoming document to the **str2** value in the functoid.

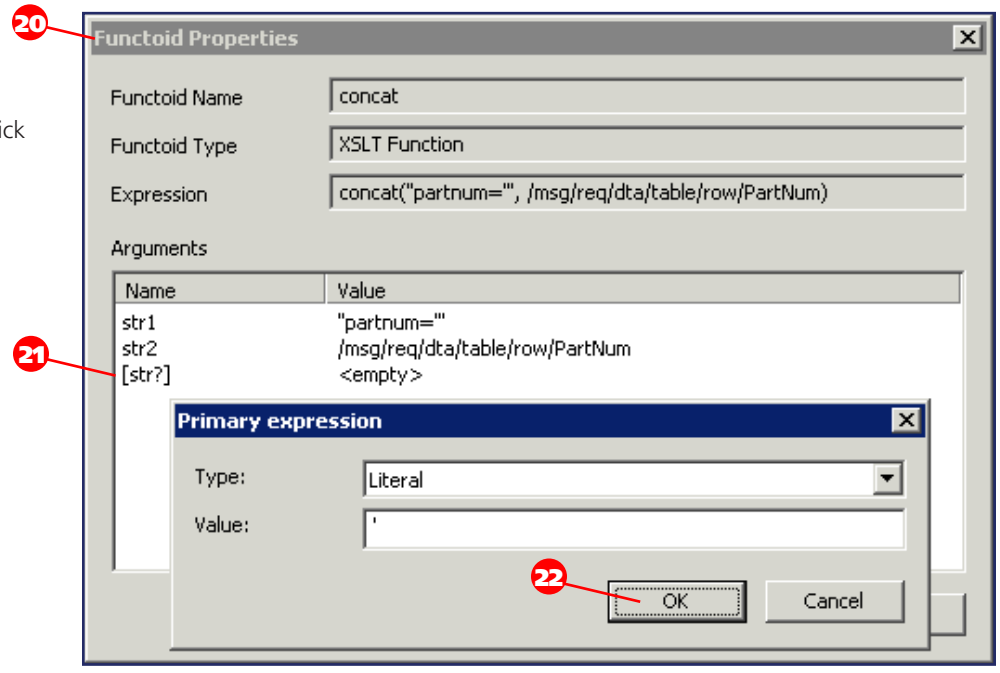


20. Double-click the functoid to open the **Functoid Properties** window.

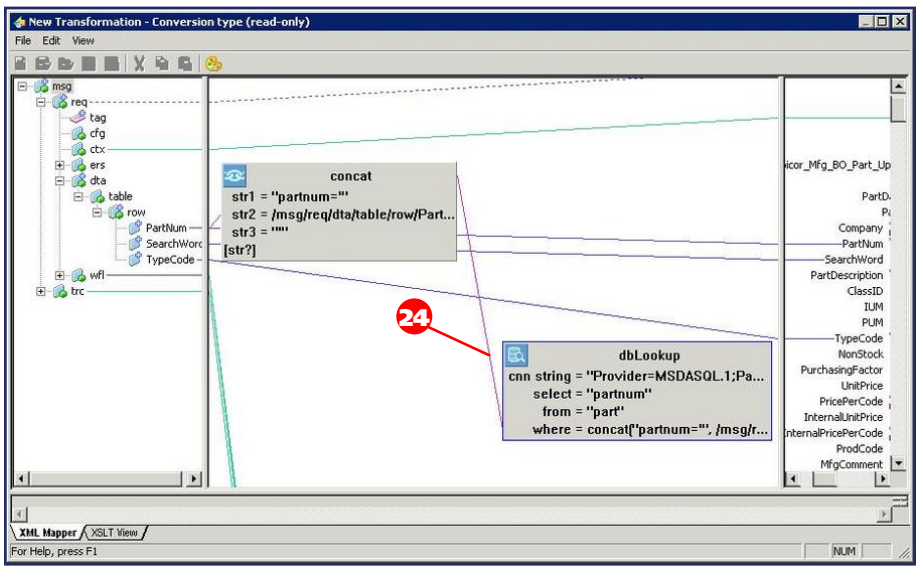
21. In the **Arguments** list, double-click the argument where the value is **<empty>** and finish the statement.

22. Click **OK** to close the Primary expression window.

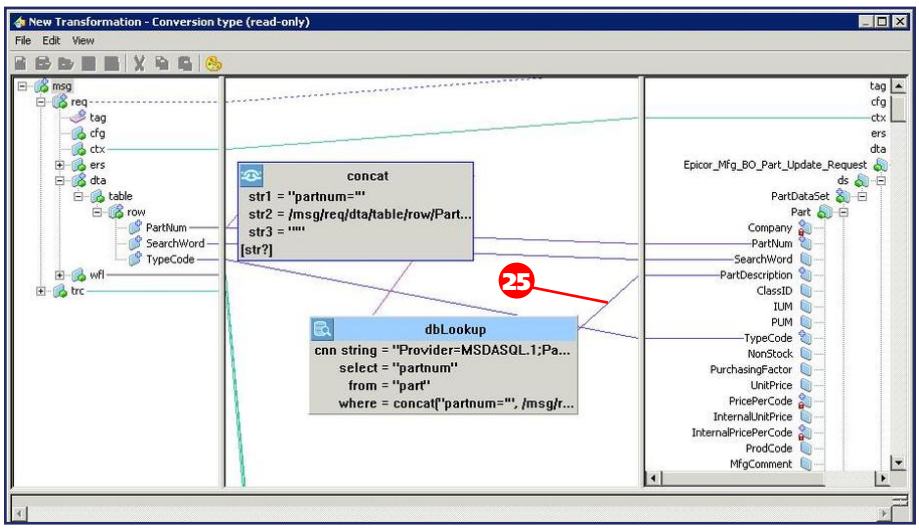
23. Click **OK** to close the Functoid Properties window.



24. Map the **concat** functoid to the **where** argument of the **dbLookup** functoid.



25. Map the **dbLookup** functoid to the **PartDescription** node in the target document.



Epicor Log Converter

The Epicor Log Converter parses the Epicor application log file to generate Service Connect workflows.

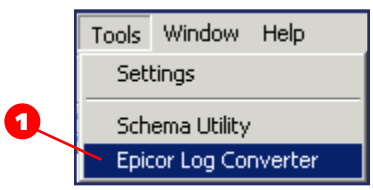
The parser searches for the following sections in the Epicor log file:

- <businessObject>BOName</businessObject>
- <methodName>MethodName</methodName>

The parser then creates a sample workflow where the found methods are transformed sequentially into the Web Service or .NET call components with or without error handling and Conversions in between. The Web Service Request ID is formed as BOName.MethodName.

Convert an Epicor Log File

1. From the **Tools** menu, select **Epicor Log Converter**.



2. Click the **Browse** button to find and select the Epicor log file you want to convert.

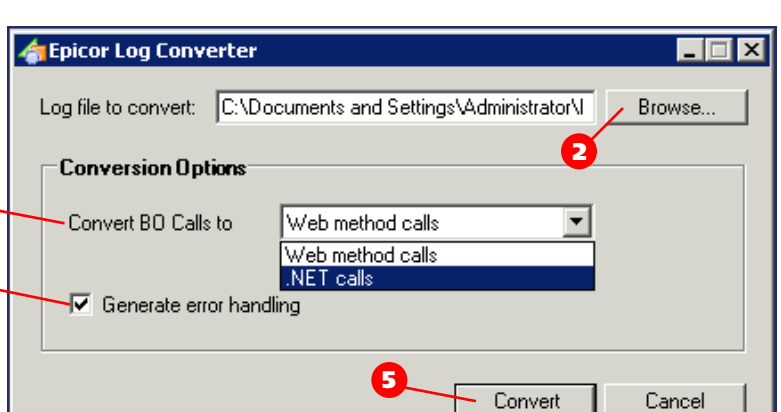
3. Click the **Convert BO Calls** drop-down list to select if you want to convert BO calls to **Web method calls** or **.NET** calls.

4. If you want errors to be handled, select the **Generate error handling** check box.

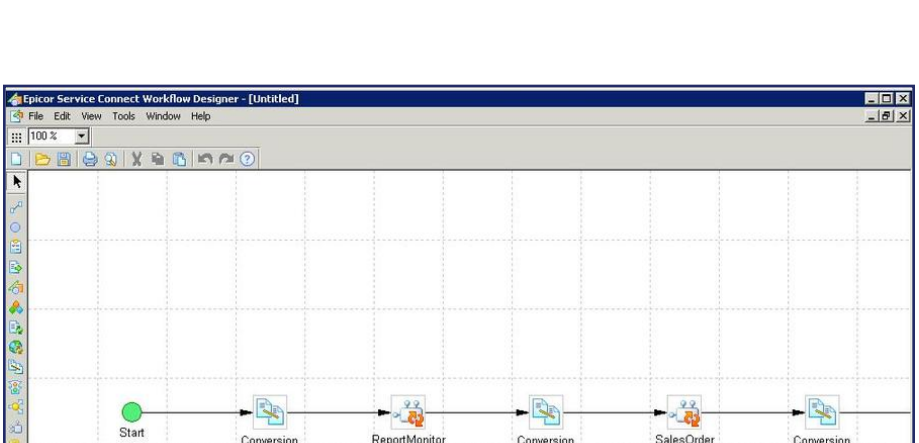
   If you enable error handling, response processing will be enabled for all the Web Services or .NET calls on the generated workflow, and the Repeat count number will be set to 10.

   If you do not enable error handling, response processing will be disabled.

5. Click **Convert**.



The log file is parsed, and the workflow is opened in Workflow Designer.



Generating Schema from Sample Data

Use the Generate Schema from Sample Data tool in the Workflow Designer to generate and import schemas into Service Connect. Frequently, this utility is used to create schemas for information that is sent into Service Connect. Several types of documents can be passed into Service Connect - Excel®, Comma Separated Value (CSV), fixed with text files, and external XML. However, to be accepted into a workflow, the data must be converted to an XML format Service Connect recognizes. Creating a schema tells Service Connect what type of data to expect and ensures the data can be transformed or used as needed by workflow activities.

You can also use this tool to generate schemas based on other schemas, such as the schemas created for the request and response values of Web Method calls. You can use these schema types to create message extensions, which are explained later in this chapter.

To generate a schema for a Microsoft Office 2007 or 2010 file, you must first install the Office System Driver package, which you can download from the Microsoft Website. Go to www.microsoft.com/downloads and search for office system driver.

Example

In the following example, the incoming data is an Excel spreadsheet that contains part information that will be updated in the Epicor application. To get the information into Service Connect, the Excel data must be transformed into XML. Service Connect will use a schema to help with the transformation.

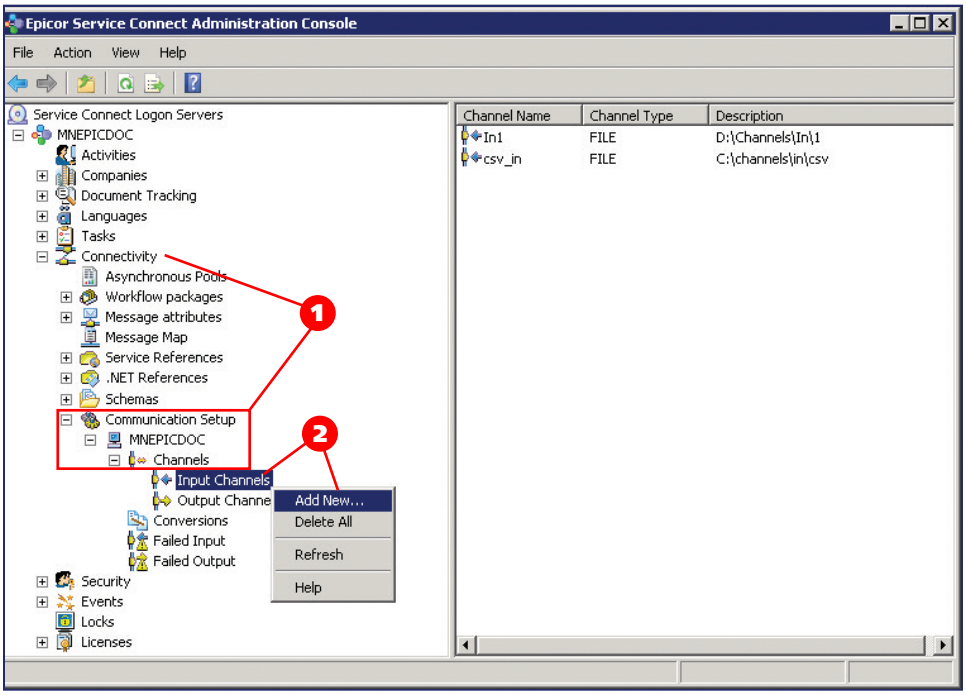
Configure the Input Channel

Before you generate a schema for your spreadsheet, configure the Input Channel that will accept the spreadsheet to recognize it.

To configure the Input Channel:

1. In the **Tree View**, expand the following nodes: **Connectivity > Communication Setup >** <Server Name> **> Channels.**

2. Right-click **Input Channels**, and select **Add New**.

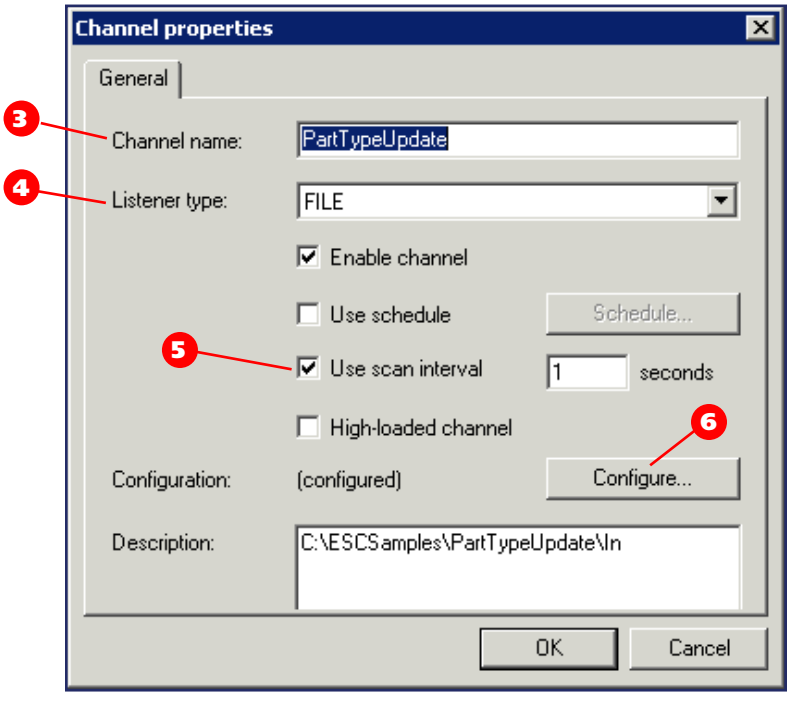


3. The **Channel properties** window displays. In the **Channel name** field, enter **PartTypeUpdate**.

4. In the **Listener** type, select **FILE**.

5. Select the **Use scan interval** check box.

6. Click **Configure**.



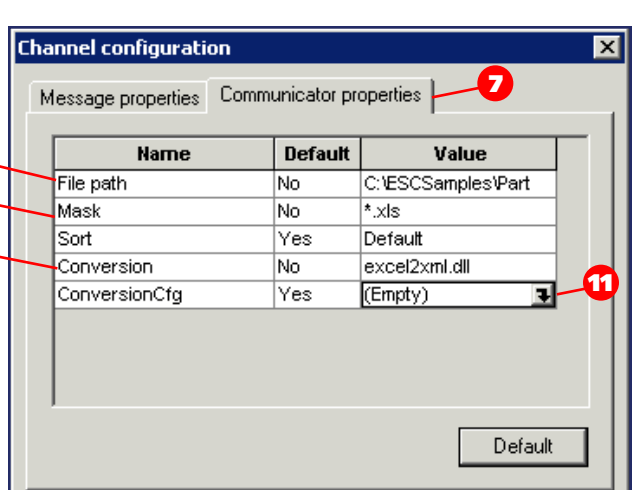
7. The **Channel configuration** window displays. Click the **Communicator properties** tab.

8. Click the **down arrow** button next to the **File path** field, navigate to the folder where the speadsheet is stored and click **OK**.

9. Verify the **Mask** field displays ***.xls.**

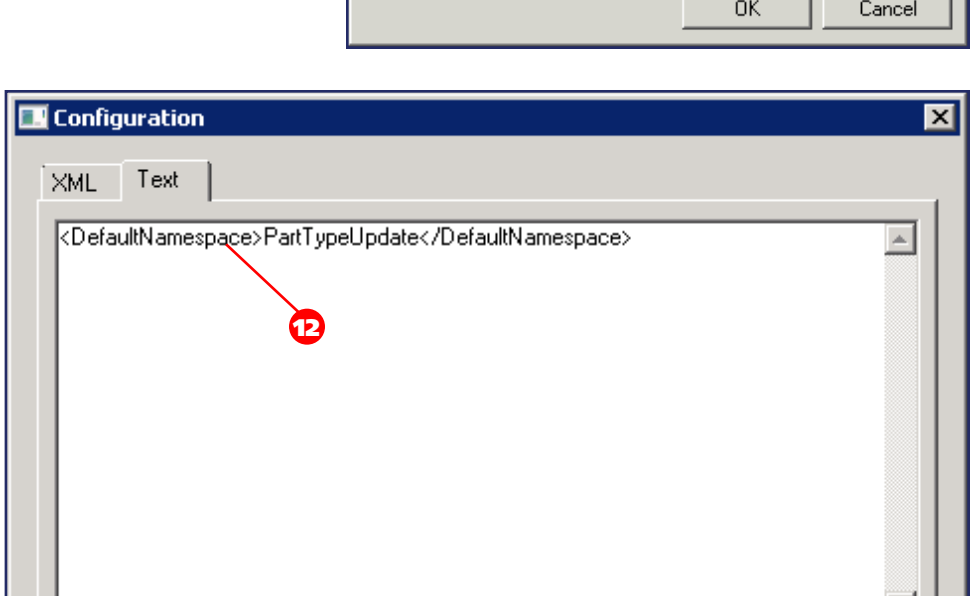
10. Verify the **Conversion** field displays **excel2xml.dll.**

11. Click the **down arrow** button next to the **ConversionCfg** field.



12. The **Configuration** window displays. Enter **<DefaultNamespace> PartTypeUpdate </DefaultNamespace>**.

    When you type this value in the Configuration dialog box you configure Service Connect to use a namespace you declare instead of generating a namespace based on the number of columns in the spreadsheet. Since the Service Connect generated namespace is based on the number of columns in the incoming data, if you add or remove any columns from the spread sheet, when a new schema is generated, Service Connect will generate a different namespace for the schema. The different namespace will affect any Conversion activities you use to process the data.



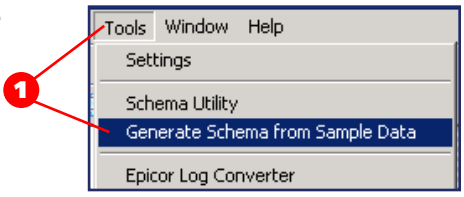
    Each column header in the spreadsheet becomes an element beneath the row element. The maxOccurs attribute is set to unbounded for the row element. Thus, Service Connect expects the incoming spreadsheet to have one or more rows of data. Also, the TargetNamespace attribute contains the namespace entered during the configuration step.

13. Click **OK** until you exit all dialog boxes.

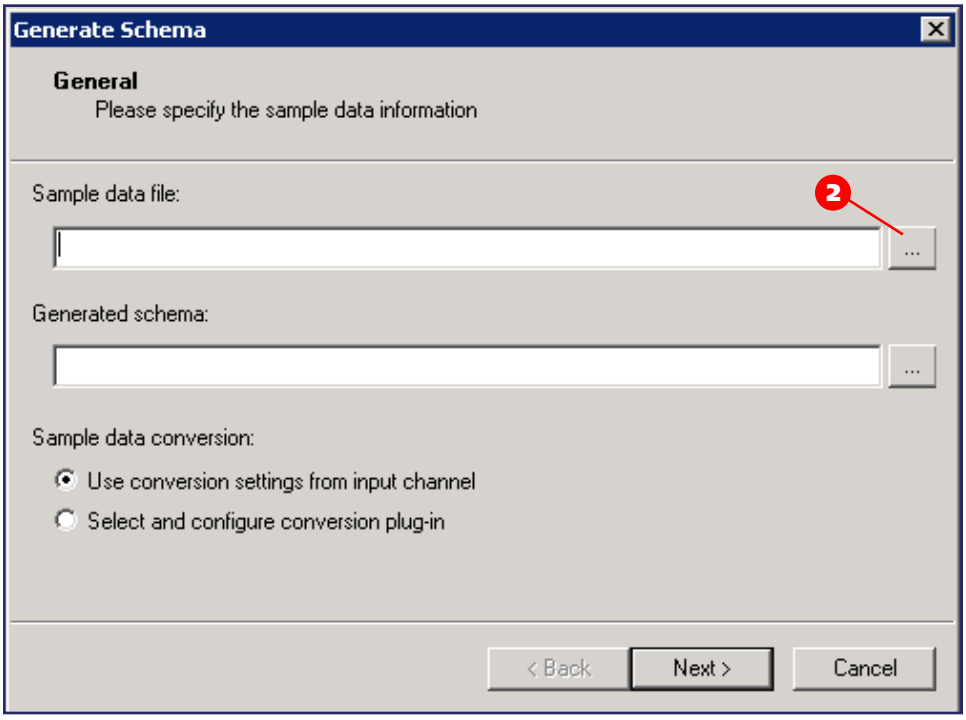
Generate a Schema

To generate a schema for the Excel data:

1. From the **Tools** menu, select **Generate Schema from Sample Data**. The **Generate Schema** window displays.

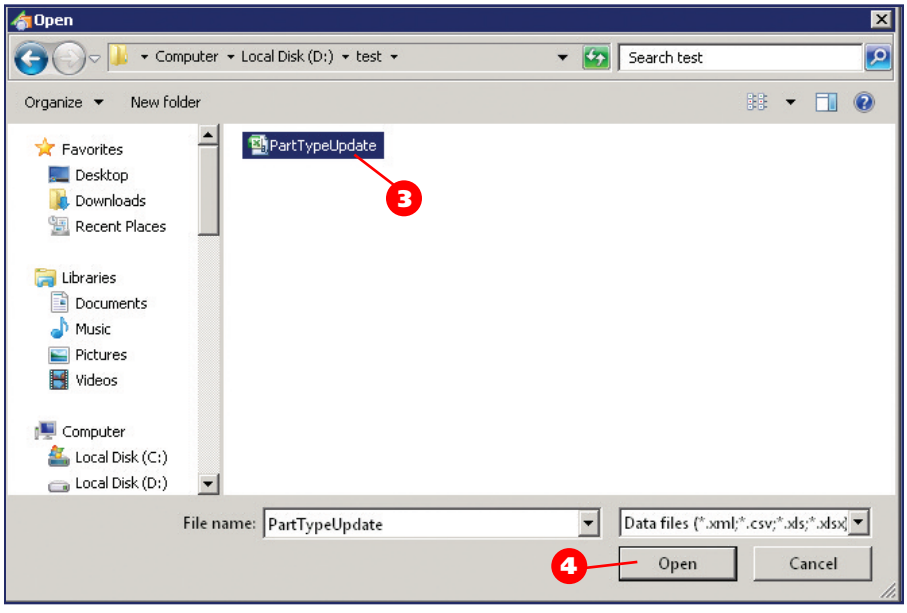


2. Next to the **Sample** data file field, click the **.... (Ellipse)** button.

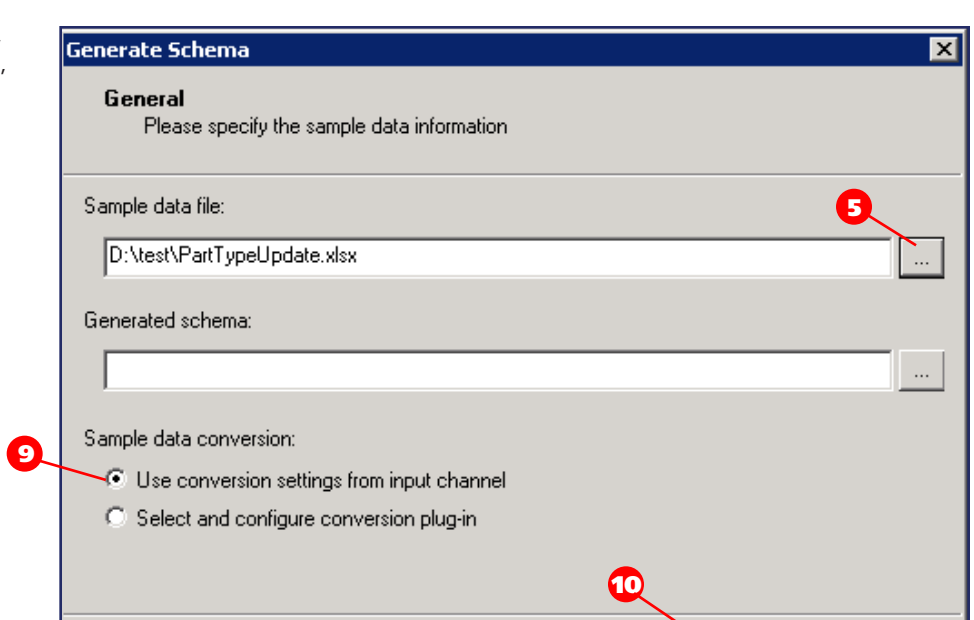


3. The **Load Sample File** window displays. Find and select the Excel file.

4. Click **Open**.



5. On the **Generate Schema** window, next to the Generated schema field, click the **Browse** button.



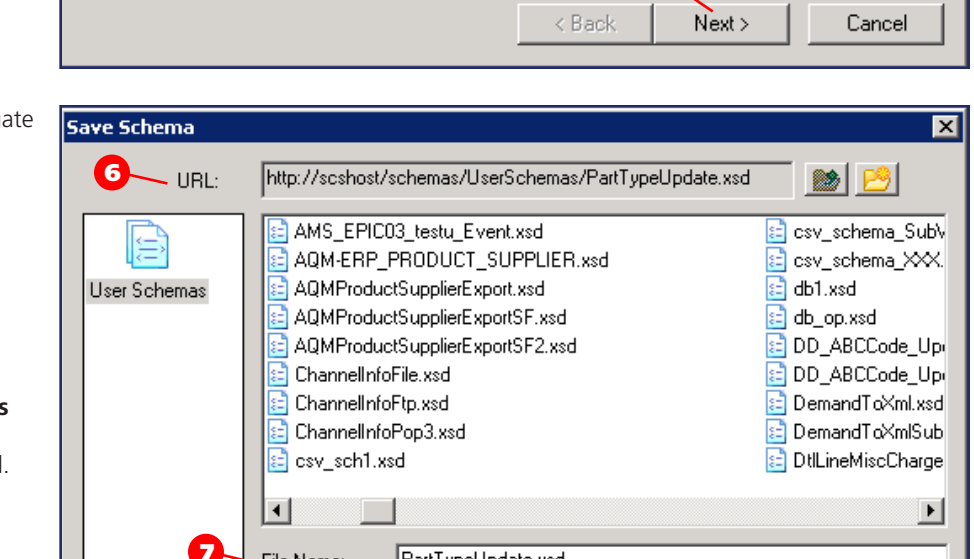
6. In the **Save Schema** window, navigate to the folder where to save the schema.

7. In the **File Name** field, enter the schema file name.

8. Click **OK.**

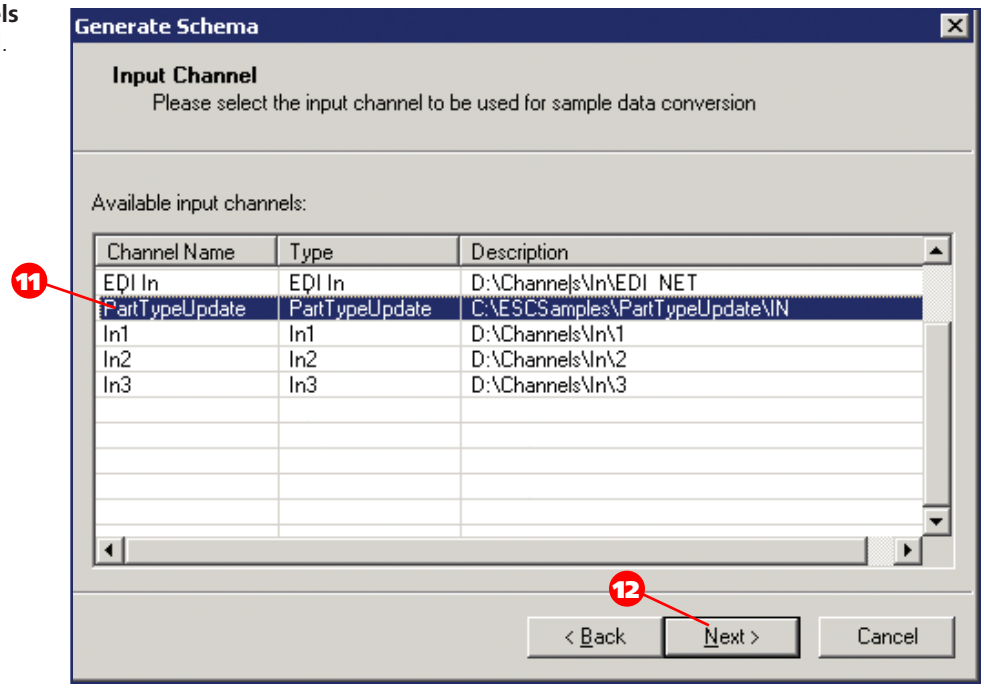
9. On the **Generate Schema** screen, select the **Use conversion settings from input channel** option to use standard conversion on the channel.

10. Click **Next**.



11. From the **Available input channels** list, select the channel you created.

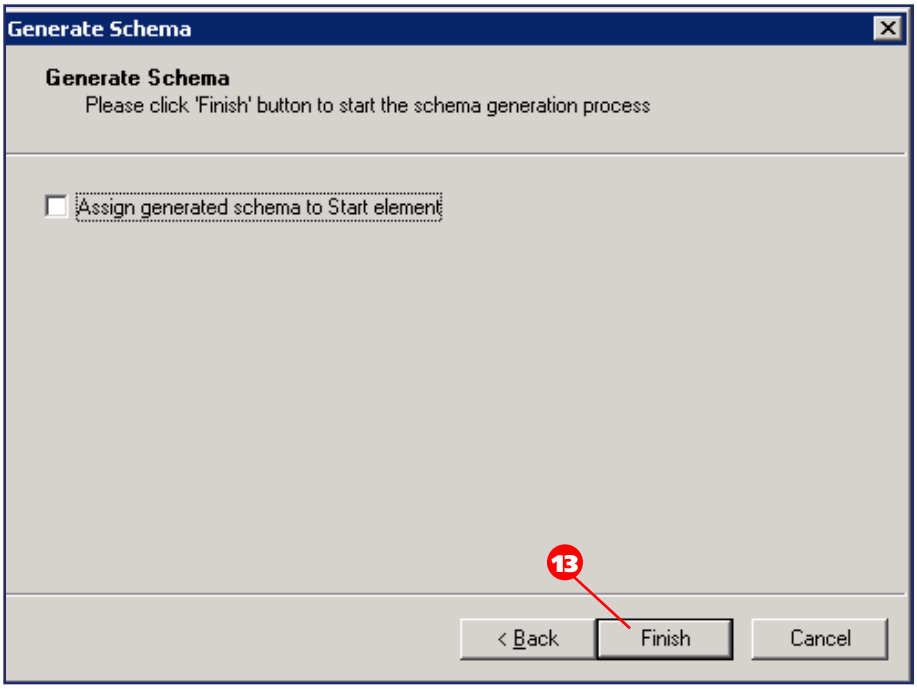
12. Click **Next**.



13. Click **Finish**.

Service Connect generates a schema for your spreadsheet and imports it. The schema is now available for use within a workflow as a Service Connect User Schema.

You can navigate to the folder you specified for the schema storage, locate the new schema and open it for review in a text editor.



Process Properties

Use the Process Properties window to set up process parameters such as document version, default asynchronous pool of the process and document tracking options. Use this tool to configure message extensions or process variables specific for the particular workflow and define company and plant information.

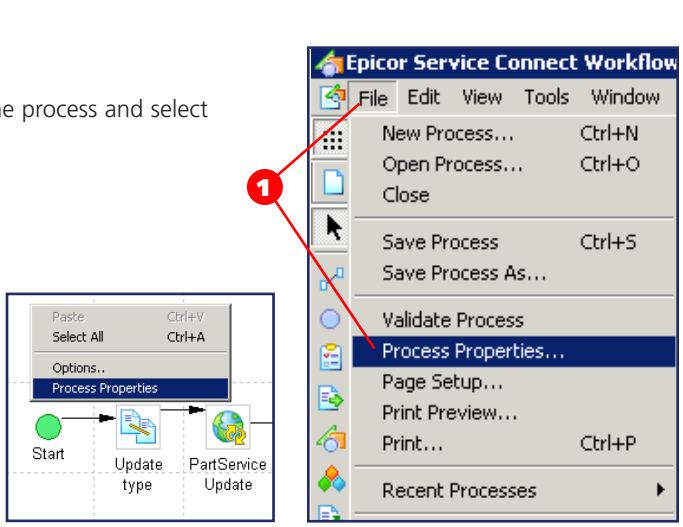
General Settings

Use the General tab to define version, asynchronous pool or workflow description of the current process.

To define general Process Properties of the workflow:

1. From the **File** menu, select **Process Properties**.

   You can alternatively right-click an empty space anywhere on the process and select **Process Properties** from the menu.



2. The **Process properties** window displays. Click the **Message Extensions** tab.

3. To change the document version, enter a new version number in the **Document version** field.

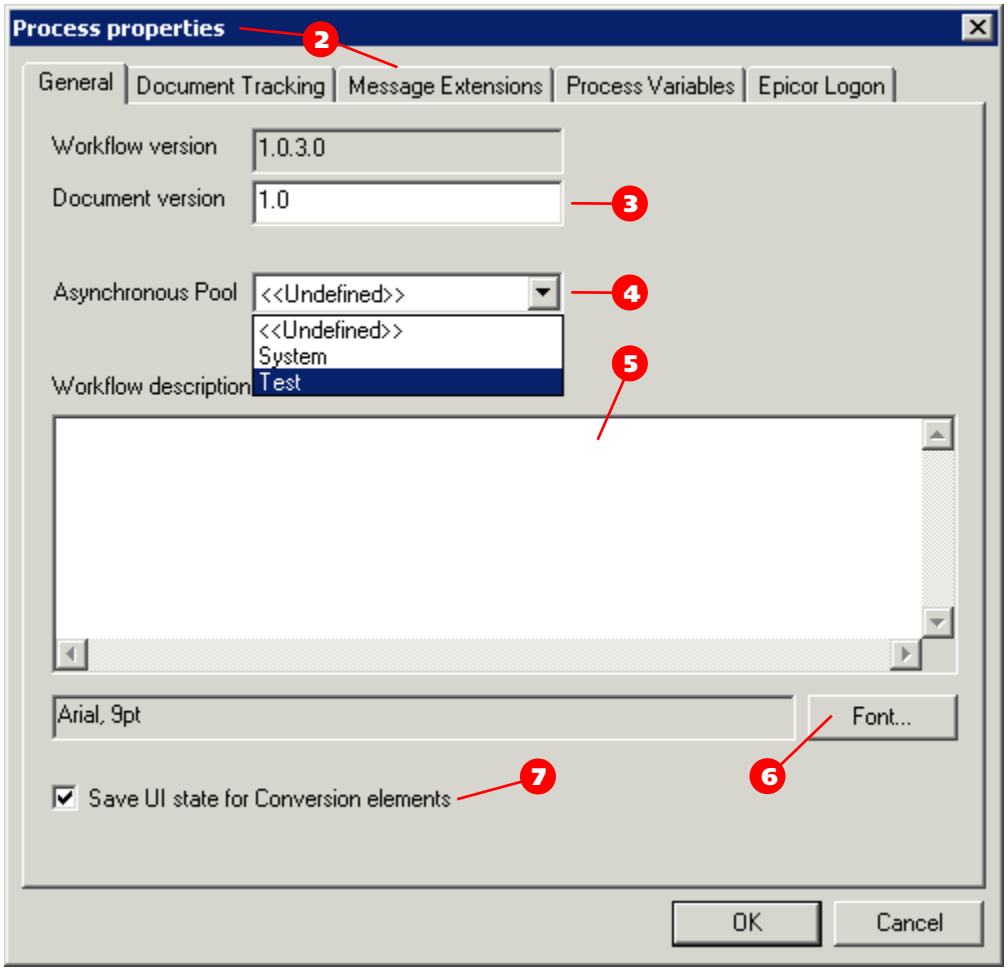
   If you change the document version, and then try to save the Process, Workflow Designer warns you that this Process already exists. You can either replace the old Process with the new one, or save the new Process with a different name.

4. You can use the drop-down list to assign an **Asynchronous pool** to the workflow.

   Asynchronous pool is used to handle incoming asynchronous call (request) for message processing.

5. You can use the **Workflow description** box to enter process related comments.

   The comment displays in the top left corner of the process by default, but you can change its size and position by dragging it with the mouse. After you add a comment, you can edit its text right in place by double-clicking it.



6. To select the font and color for the comment, click the **Font** button.

7. The **Save UI State for the Conversion elements** check box is selected by default. It controls if all node selections, expanded nodes, scrollbar positions and functoid positions are preserved, when the workflow is saved.

   You can disable the setting if it causes performance issues or if you do not want to see the last UI state of the XML Mapper.

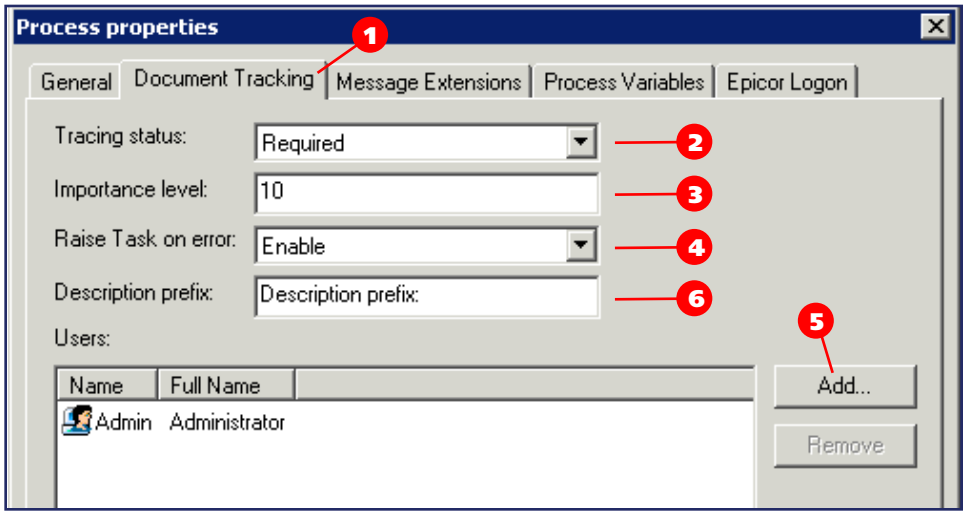
Document Tracking

Use the Document Tracking tab to control the way messages are processed by the workflow.

To use the Document Tracking properties of the workflow:

1. In the **Process Properties** window, select the **Document Tracking** tab.
2. Click the **Tracing status** drop-down list to select if Document Tracking is **Enabled** or **Required** for messages processed with the process.

   To activate Document Tracking for the workflow which is exposed as web service, set the Tracing status field to Required.



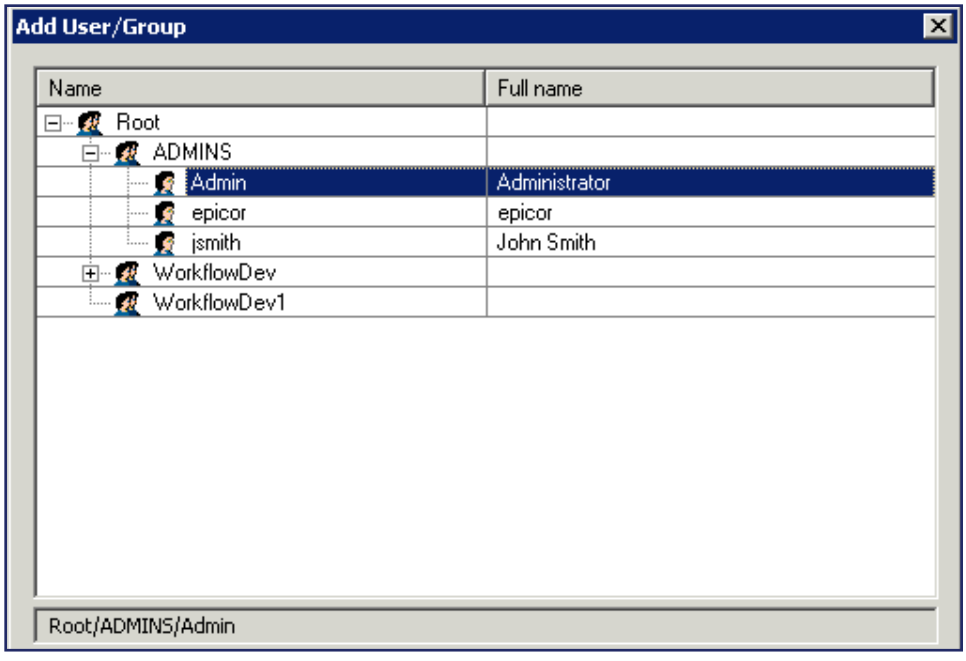
3. You can use the **Importance level** field to specify the importance level for the current process.

   By default, 10 is used as the value of this parameter. This value is compared to the threshold value, specified in the ESC Administration Console, and the traces with importance level less than the threshold value are deleted from the database. If the value of the Importance Level parameter for a process equals to the threshold value, the trace is not deleted.

4. You can have the Workflow Engine to create a Task element and stop processing with the Pending status in case a Web method or .Net method is finished with an error. To do this, click the **Raise task on error** drop-down list and select **Enable**.

5. Click the **Add** button to control whom the Task is assigned to using the **Add User/Group window**.

   The message in the Task element has error section from the previous failed response and the original request of the last executed element. The Task has two links: one to re-process the same element again, second – to step over the failed element and go to the next element.



6. In the **Description prefix** field, define the Task description that will display in the Task Monitor.

Message Extensions

Message extensions, also known as containers, are used to store data so it will be available to workflow activities after a Web Method or a .NET call. The data can be information passed into the workflow or derived from other workflow activities, such as a Conversion or a prior Web Method or .NET call.

Message extensions are useful because the Input Schema for a call must always be the request schema for the used method, and the Output Schema for a call must always be the response schema for the used method. These schemas only have nodes for the information sent to and from a Web Service or a .NET method, so a container must be used to propagate any other data required later in the workflow.

Example

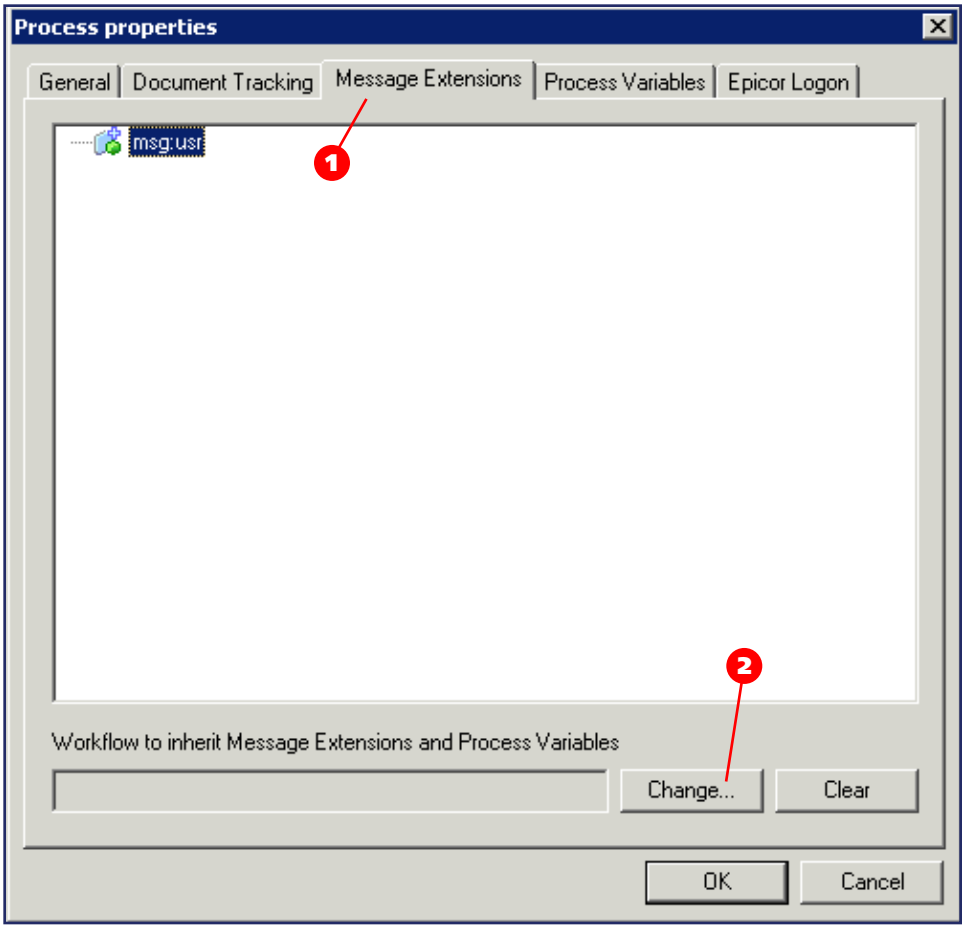
In the following example workflow, the incoming data is an Excel spreadsheet that contains part update information.

The workflow in this example checks the updated information for errors. In order to restore the original data sent into the workflow, a message extension must be added. Use message extension to fix data and resubmit it for update in the following workflow activities.

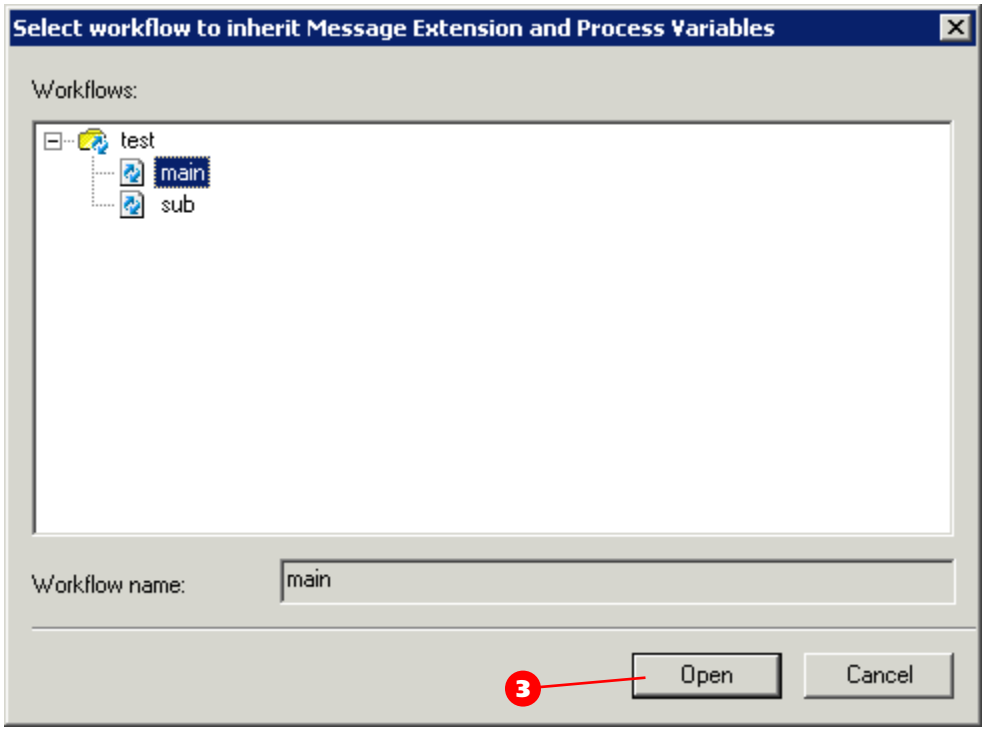
Create a Message Extension

To create an extension:

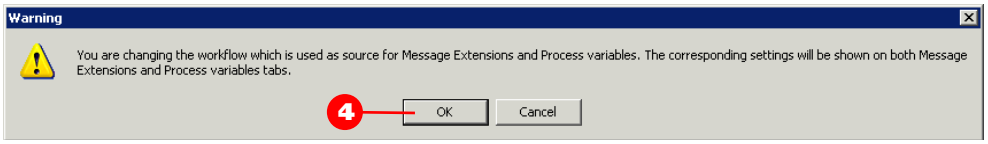
1. Click the **Message Extensions** tab.
2. If you want to select a workflow as a source for Message Extensions and Process Variables, click **Change**.



3. In the **Select workflow to inherit Message Extensions and Process Variables** window, select the workflow and click **Open**.



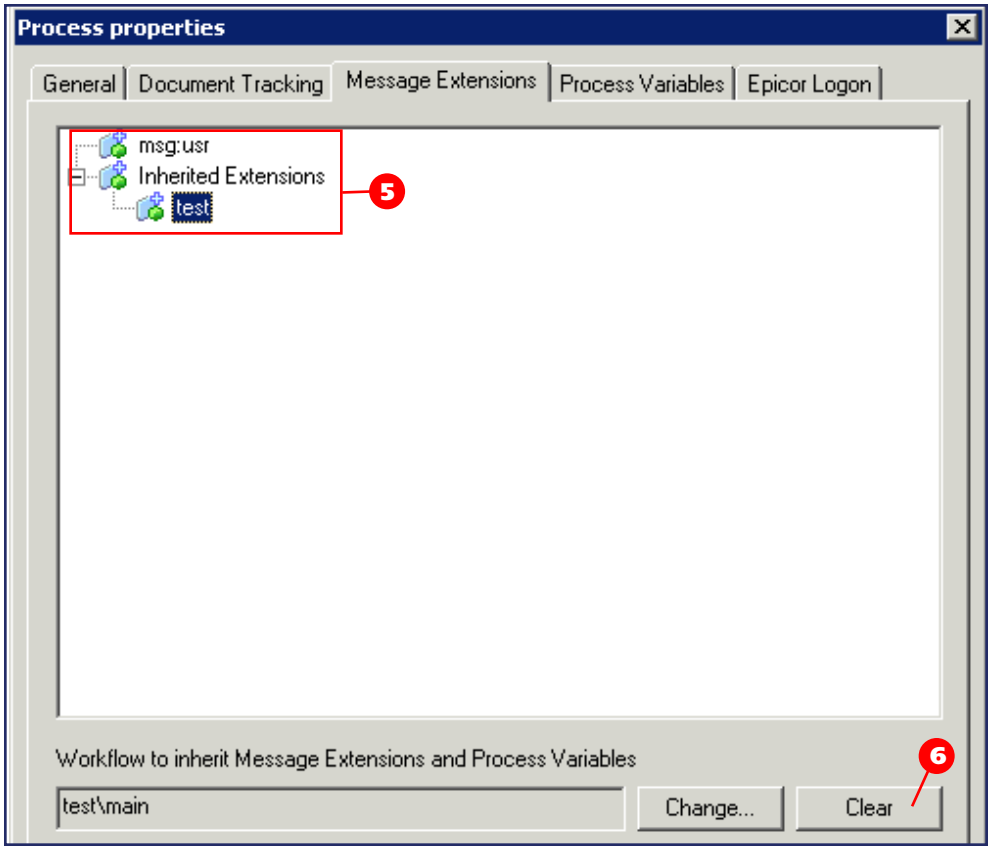
4. To the **Warning** message, click **OK**.



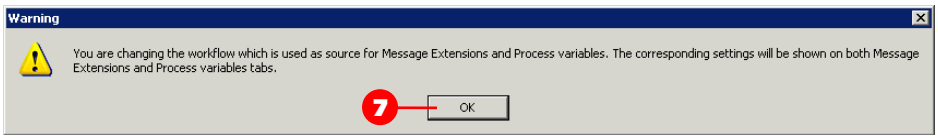
5. Notice the tree displays the **Inherited Extensions** node.

> The Validate workflow functionality checks for name conflicts between inherited Extensions/Variables and local ones specified in this workflow. In case such conflicts are found, a warning displays. If an inherited variable/extension has the same name as a local one, the local variable/extension will be seen in XML Mapper, XPath builder and runtime.

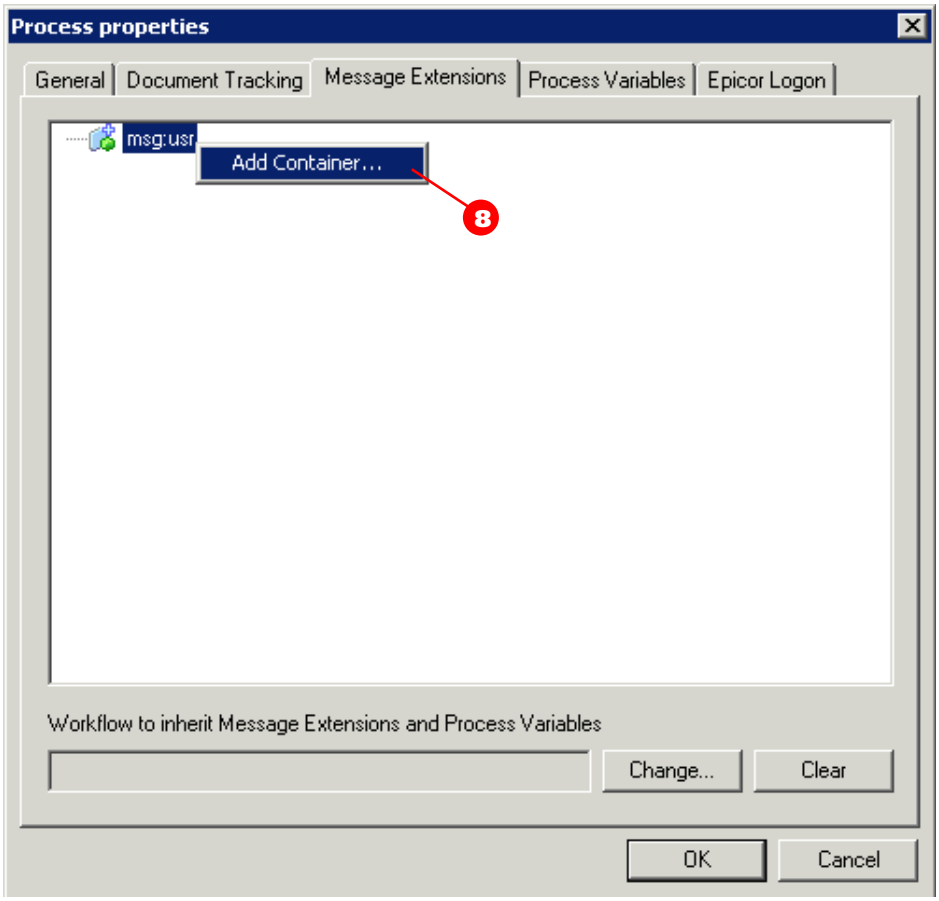
6. To cancel the Message Extensions inheritance, click the **Clear** button.



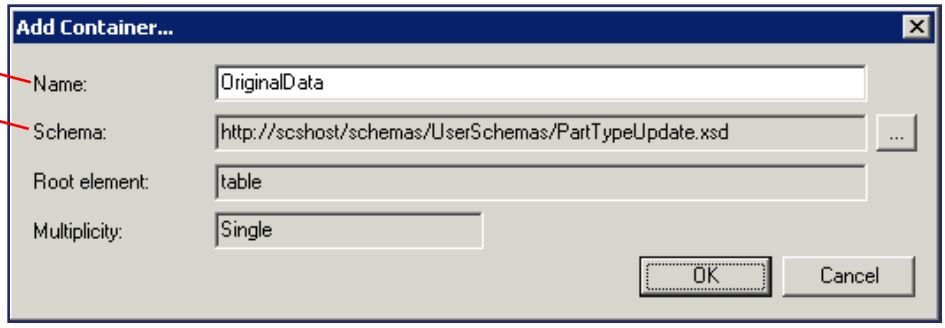
7. To the **Warning** message, click **OK**.



8. Right-click **msg:usr** and select **Add Container**.



9. The **Add Container** window displays. Enter a **Name** for the container.

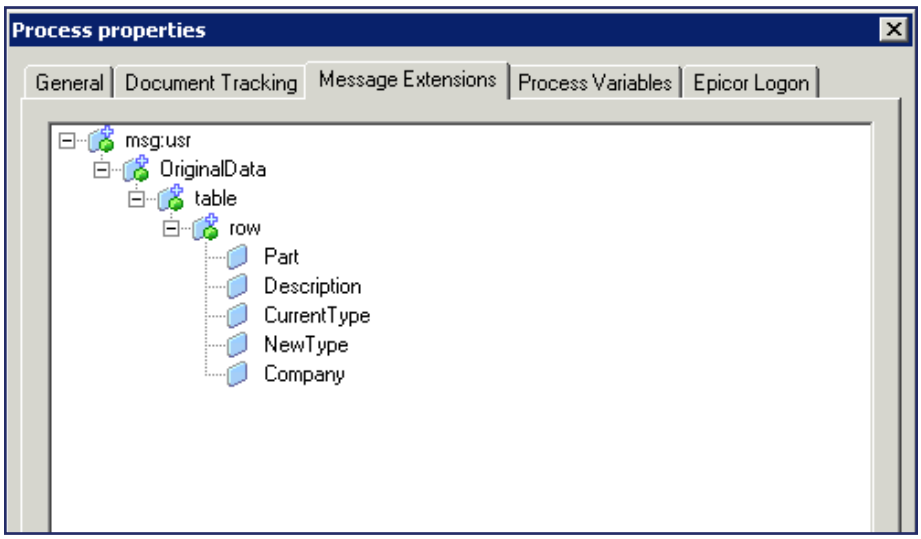


You can name the container almost anything you like using a combination of letters, numbers, underscores, and dashes. You could use the name OriginalData if you are storing a message as it enters a workflow and before it is processed or converted to any other format. Storing the message in its original XML format preserves the information for later use after Web Method or .NET calls.

10. Click the **... (Ellipse)** button to find and select the **Schema** that will define the data to store in the container. In this case, the schema is used to import the Excel spreadsheet into Service Connect.

After the container is created, the data structure displays in the Process Properties dialog box.

You can now use the container to restore the original data.



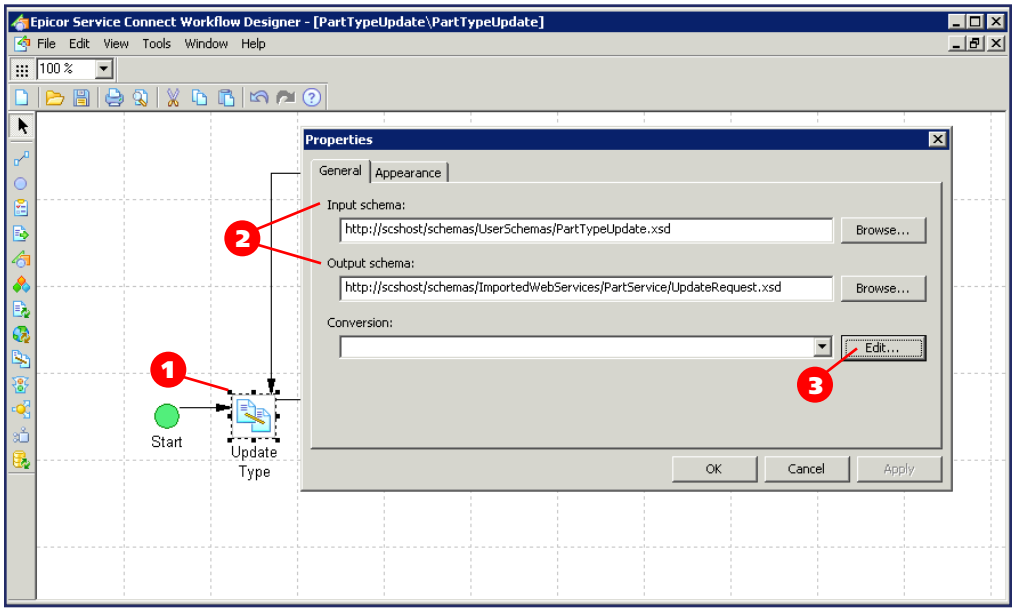
To use the container:

1. Open the **Properties** of the **Conversion** activity where you want to use the container.

2. Select the **Input schema** and **Output schema**.

   In this example, the Input Schema for the Conversion is the schema used to import part information into Service Connect. The Input Schema name is PartTypeUpdate.xsd. The Output Schema is the Web Service request schema used for the Web Method call, which is the next activity in the workflow. The Output Schema name is UpdateRequest.xsd.

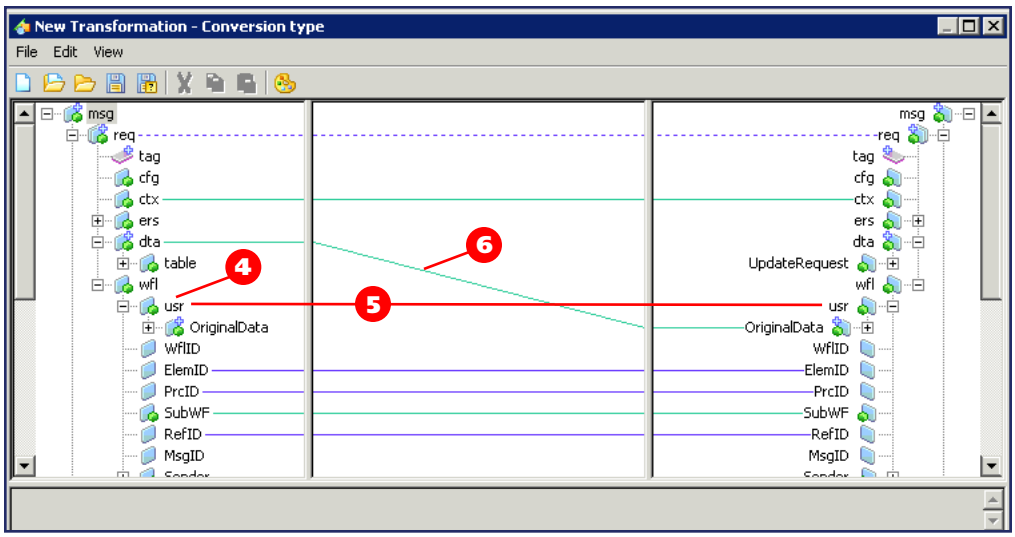
3. Click **Edit** to open the XML Mapper.



4. Expand the following nodes on both sides of the XML Mapper: **req > wfl > usr node**.

   The OriginalData container appears beneath the usr node on both sides. The container is empty because the document that enters the workflow is stored in the dta node.

5. Delete the mapping between the **usr** nodes.



Service Connect automatically maps the usr nodes in every conversion to propagate message extensions and process variables. However, in this example, you must delete the mapping because the OriginalData container in the target document can accept only one mapping, and you want to add the data from the **dta** node in the incoming document.

6. Map the **dta** node in the incoming document to the **OriginalData** node in the target document.

   Notice the connection line is green. The green line indicates the mapping between the nodes is a deep copy, and all the child nodes are automatically mapped. A deep copy is possible only when the structure of both complex nodes is identical. In this case, they are identical because the schema of the incoming document was the same schema used to create the container.

   After the Web Method call, you can find the information in the container under the wfl/usr node where it is available for further processing. Data returned from the Web Method call will appear in the dta node.

To combine the data from the container and the Web Method:

1. Open the **Properties** of the **Conversion** that follows the Web Method.

   In this example, the Restore Original Data conversion is used. To make the data available to edit in the Task activity, restore the original data to the dta node of the internal envelope.

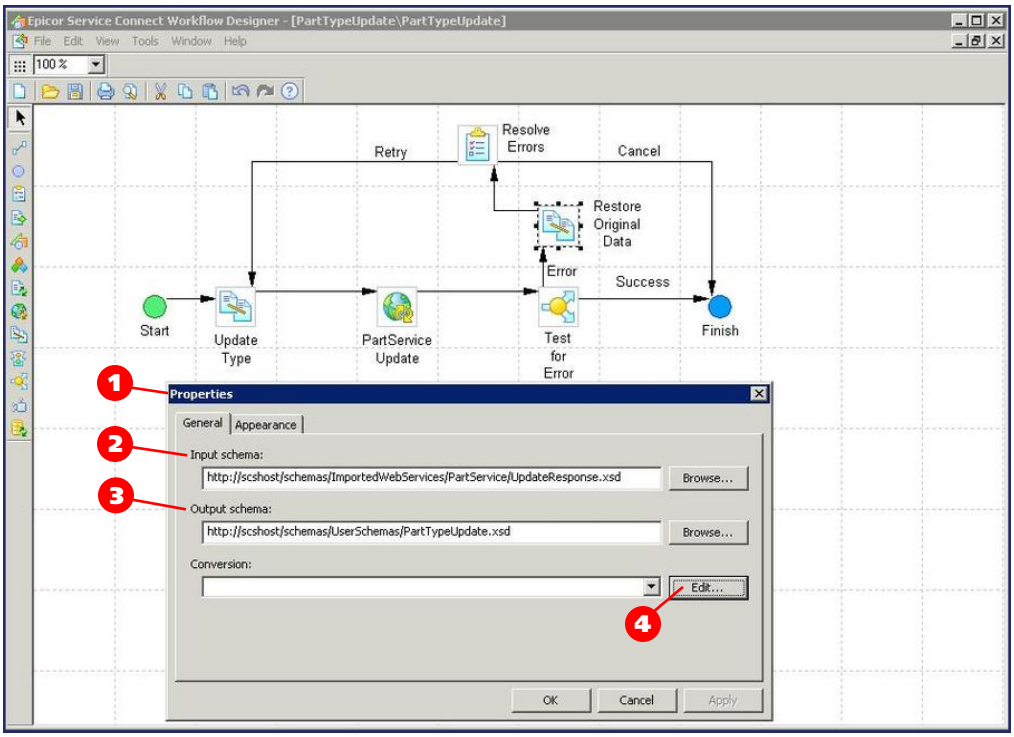
2. Select the **Input schema**.

   In this example, the Input Schema for the Conversion is the part service response schema.

3. Select the **Output schema**.

   In this example, the Output Schema for the Conversion is the schema used to import part information into Service Connect.

4. Click **Edit** to open the XML Mapper.

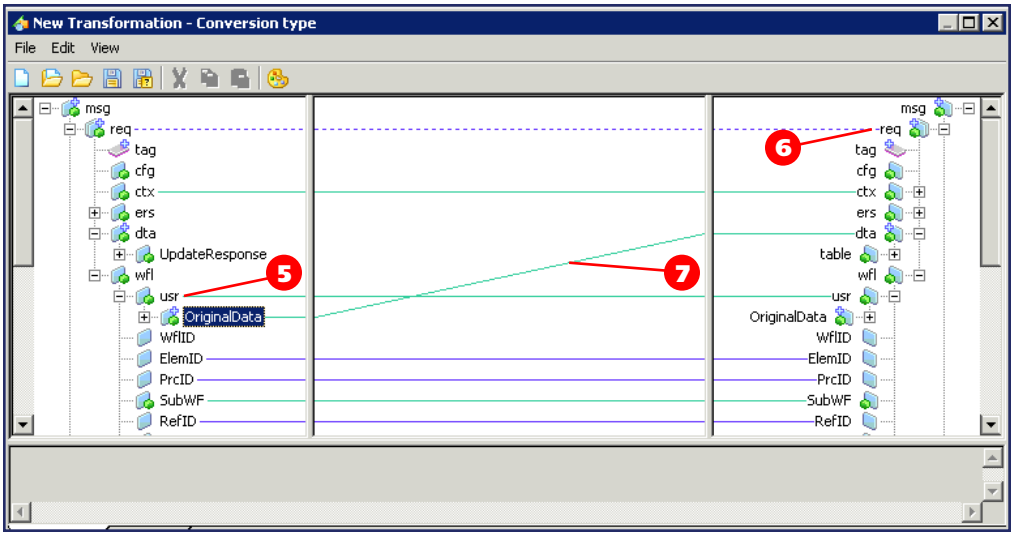


5. Expand the following nodes in the incoming document: **msg > req > wfl > usr**.

   The OriginaData node contains the restored original data.

6. Expand the following nodes in the target document: **msg > req**.

7. Map the **OriginalData** node on the left to the **dta** node on the right.



Process Variables

Process variables are similar to message extensions. Use process variables to store data in the usr node of documents so you can access the information when needed. While message extensions are designed to hold complex data structures based on a schema, variables are designed to hold a single value. In addition, you can assign process variables a data type and a default value.

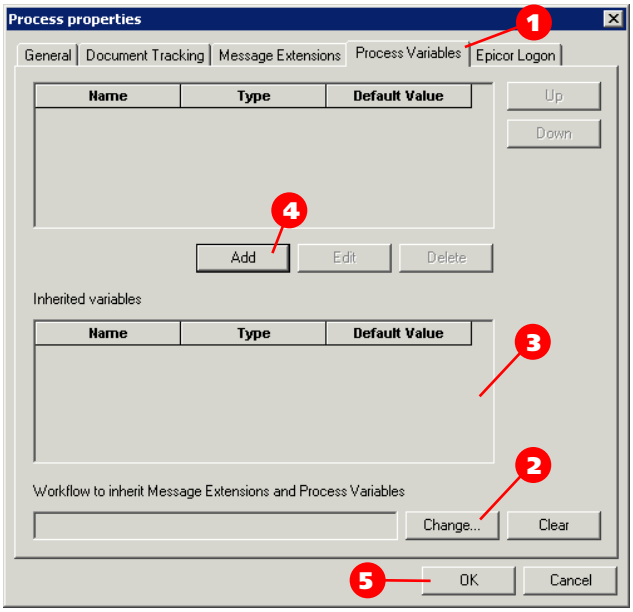
Example

Suppose you have a workflow where you need to use the Company ID. You can set up a process variable to hold the company code.

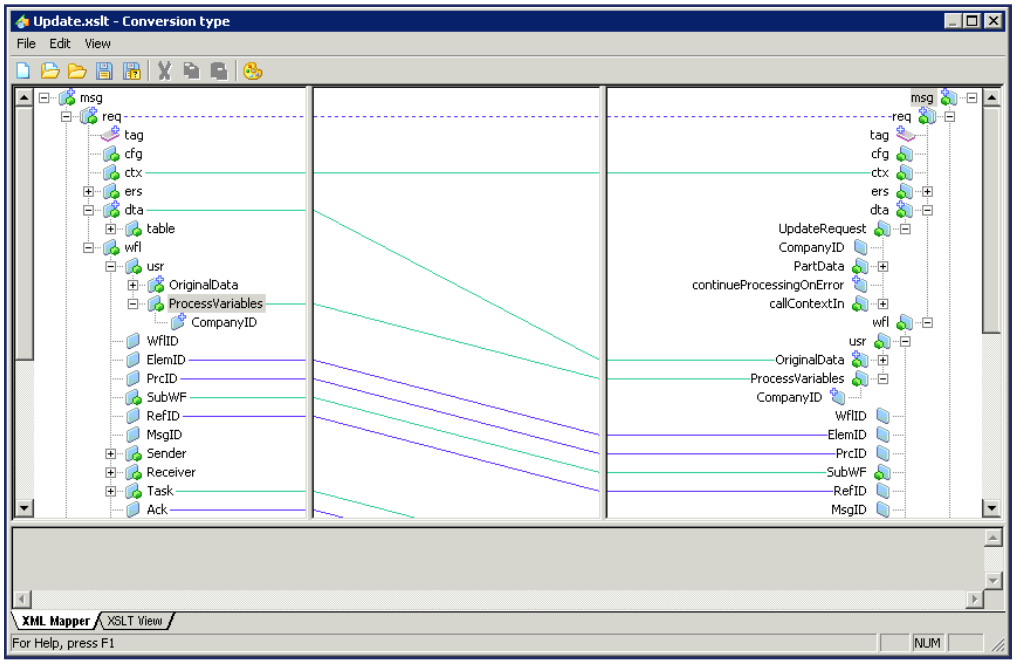
Set Up a Process Variable

To set up a process variable:

1. Click the **Process Variables** tab.
2. To inherit Process Variables and Message extensions from another workflow, click the **Change** button to find and select a workflow.
3. Once a workflow is selected, variables display in the **Inherited variables** grid. For more information, view the **Message Extensions** topic above.
4. Click **Add**.
5. Define the variable and click **OK** until you exit all dialog boxes.



The variable will be available for use from within the usr node under ProcessVariables.

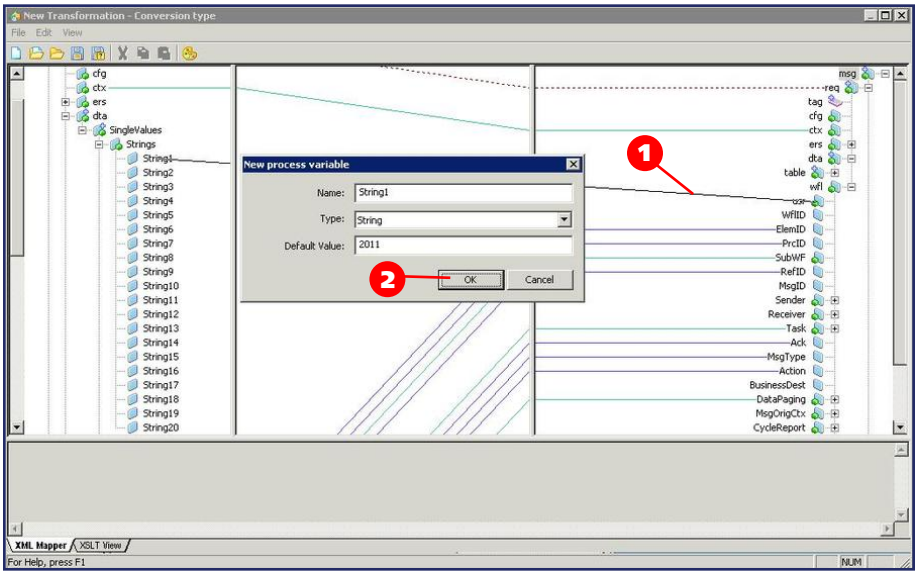


Drop nodes to the usr section

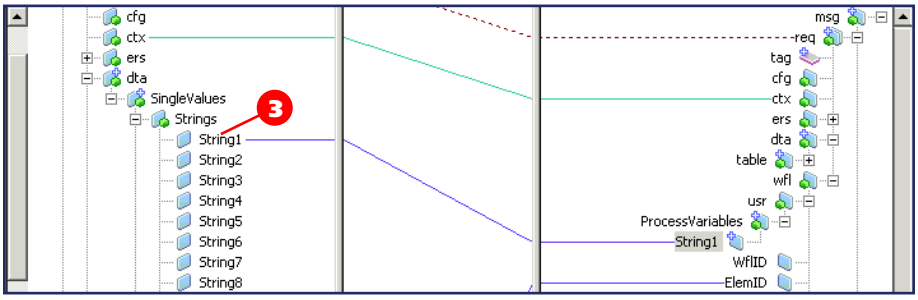
When you configure a Conversion element and drop nodes to the <usr> section, XML Mapper creates Process variables and Message extensions.

Example:

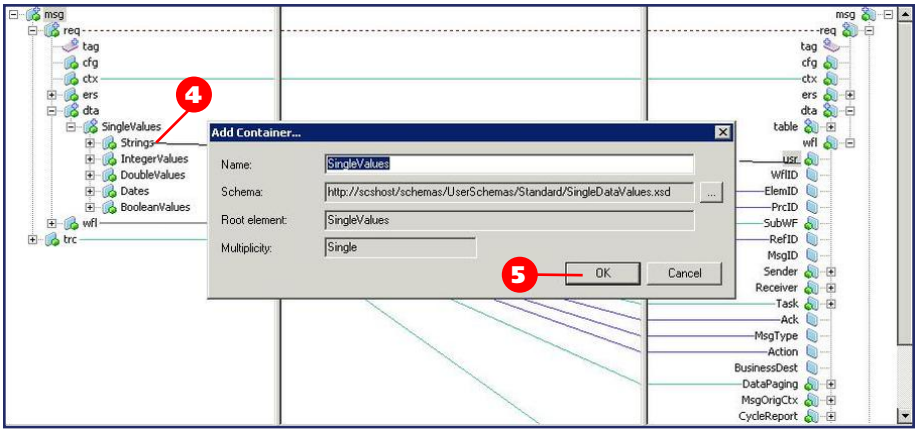
1. In XML Mapper, drop a node to the <usr> section of the target document.
2. In the **New process variable** window, edit the properties of the variable and click **OK**.



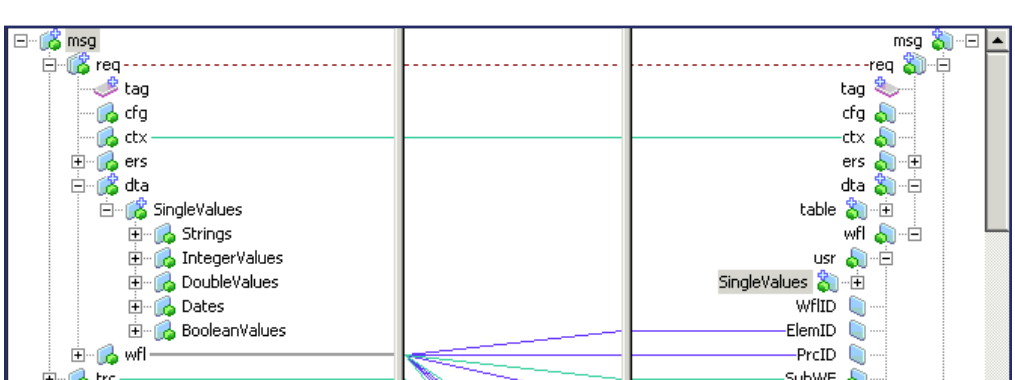
3. When you map a simple type node to the <usr> section, a new Process variable is created and mapped.



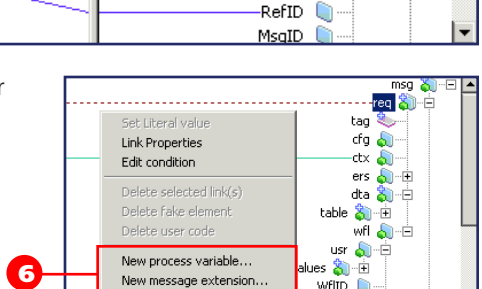
4. When you map a complex type node under the source <dta> section to the target <usr> section, a new message extension in source schema is created.
5. In the **Add Container** window, edit the properties of the message extension and click **OK**.



For complex nodes, no automatic mapping is created.



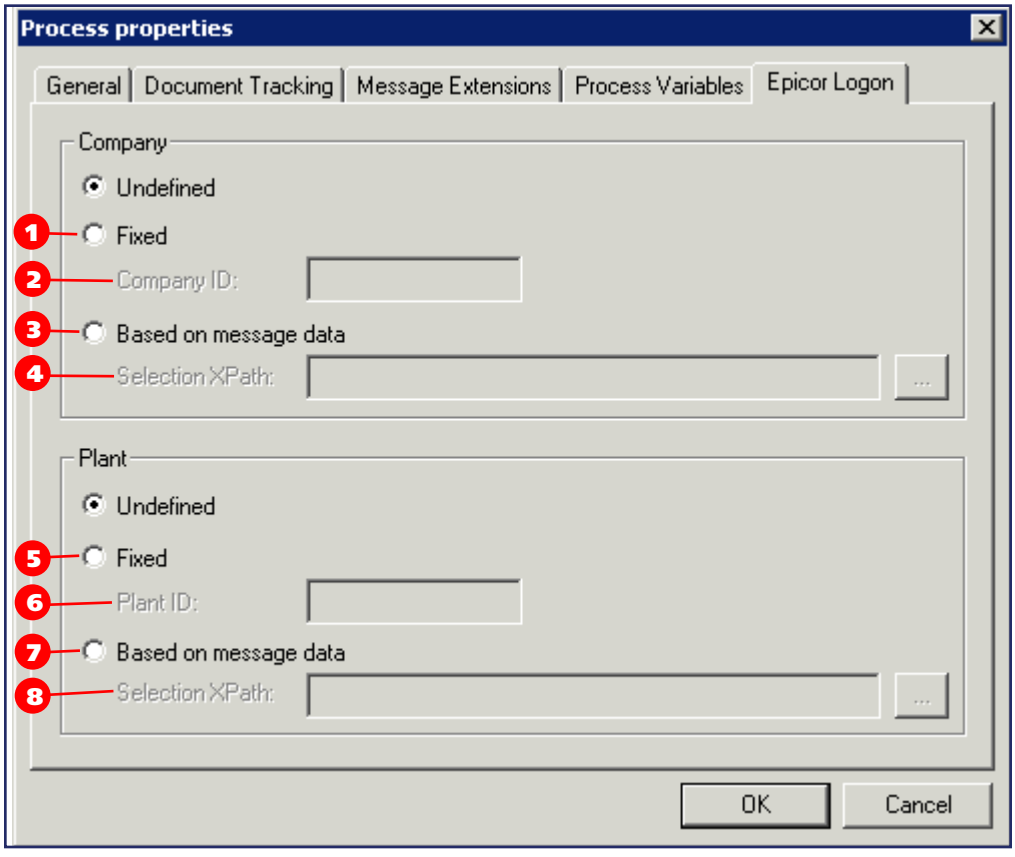
6. You can alternatively right-click the right pane and select **New process variable** or **New message extension** to create variables or containers.



Epicor Logon

Use the Epicor Logon tab to set up a Company and Plant information.

1. To set a particular Epicor Company, on the **Epicor Logon** tab, in the Company pane, select the **Fixed** option.
2. In the **Company ID** field, enter a company code.
3. If you want the company to be selected during runtime according to the input message data, in the Company pane, select **Based on message data**.
4. Enter the XPath in the **Selection XPath** field, or click the **... (Ellipse)** button to build the XPath in XPath Builder.
5. To set a particular Epicor Plant, in the Plant pane, select the **Fixed** radio-button.
6. In the **Plant ID** field, enter a plant code.
7. If you want the plant to be selected during runtime according to the input message data, in the Plant pane, select **Based on message data**.



8. Enter the XPath in the **Selection XPath** field, or click the button next to the field to build the XPath in XPath Builder.

DES Poster

The processes provided with the Epicor Service Connect are referred to as **Standard**. The processes created manually or derived from a standard process by the user are referred to as **Custom**. When testing a custom process, Epicor.DESPoster tool is used to process a particular document.

Use DESPoster to customise standard processes, create your own custom processes and to validate them. You can also load internal and external envelope XMLs, so you don't have to type them manually.

To access the Epicor.DESPoster tool, navigate to the SCSTools folder of your Epicor Service Connect installation, for example: C:\Program Files\Epicor Service Connect\Tools\SCSTools.

Summary

This chapter introduced Workflow Designer functionality you can use to develop workflows for Service Connect. The next chapter describes how to integrate Service Connect with your Epicor solution.

Chapter 5
Epicor Service Connect Integration

This chapter explains the steps required to integrate Epicor Service Connect with the following Epicor solutions:

- Epicor ERP
- Epicor Enterprise Financials and SCM (Supply Chain Management) and Epicor for Service Enterprises
- Epicor iScala

After you follow the integration steps, you should be able to create workflows. For more details about how to create workflows, refer to Chapter 4: Workflow Designer or to the appropriate sample workflow documentation for your Epicor solution, posted on EPICweb.

Where applicable, the procedures in this chapter offer tips and best practices information. Also, for Epicor ERP and Epicor for Service Enterprises, there are procedures that explain how to use the tracing tools within those applications. If you use Epicor Vantage, use the Epicor ERP section to learn how to integrate with Service Connect.

Service Connect and Epicor ERP

This section explains how to integrate Service Connect with Epicor ERP, how to use the Epicor ERP tracing tools, and how to call a Service Connect workflow from the Epicor ERP Business Process Management (BPM) module.

The process for Epicor Vantage is almost identical. Notes throughout this section highlight information where the processes are different.

Integration Overview

You must perform the following tasks to create workflows that interact with Epicor ERP.

- Make Epicor ERP web services available to Service Connect.
- Import the web services as Service Connect Service References.
- Import .NET References.
- Create document schemas.
- Create message attributes.
- Create input and output channels.

Epicor ERP Web Services Setup

You must expose the Epicor ERP Web Services and import them as Service References in the ESC Administration Console. You can then use these service references from within a workflow.

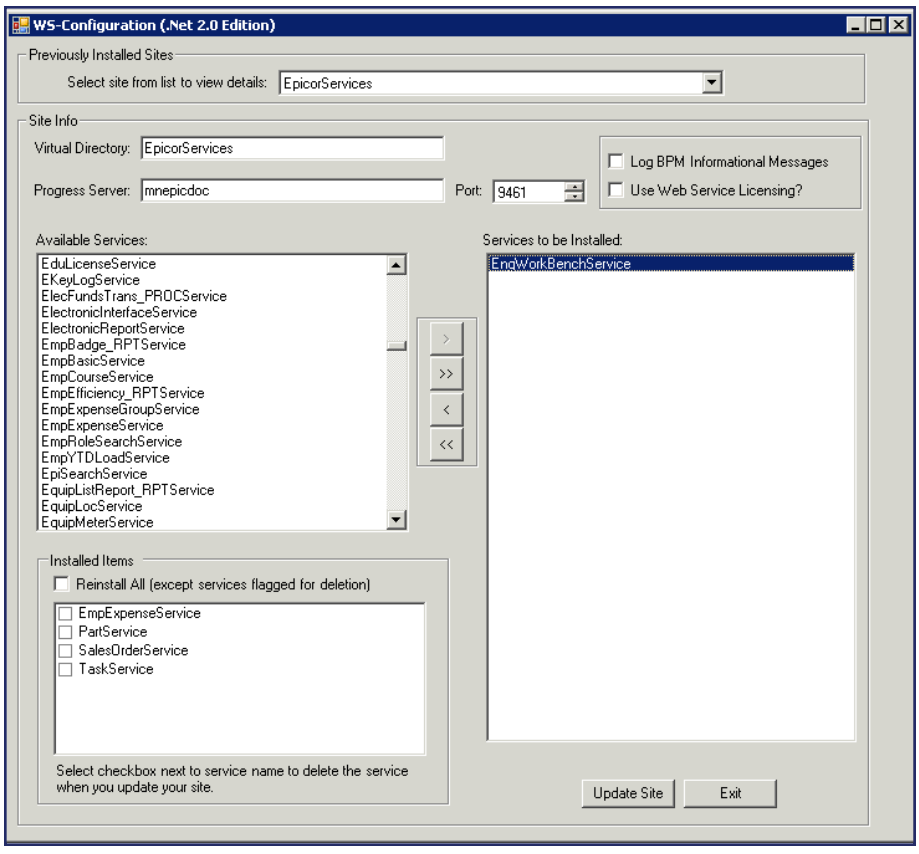
Expose Epicor ERP Web Services

On your Epicor server, go to the **Epicor905** folder and open the **Web Services** folder.

Open the **WS-Configuration_ QuickStart.doc** file and follow the instructions to install the Web Services you need.

Note the **Virtual Directory** name if you changed it from the default - EpicorServices.

For Epicor Vantage, the web services are located in the mfgsys803 folder. The default virtual directory name for Vantage is VantageServices.



Expose Epicor WCF Web Services

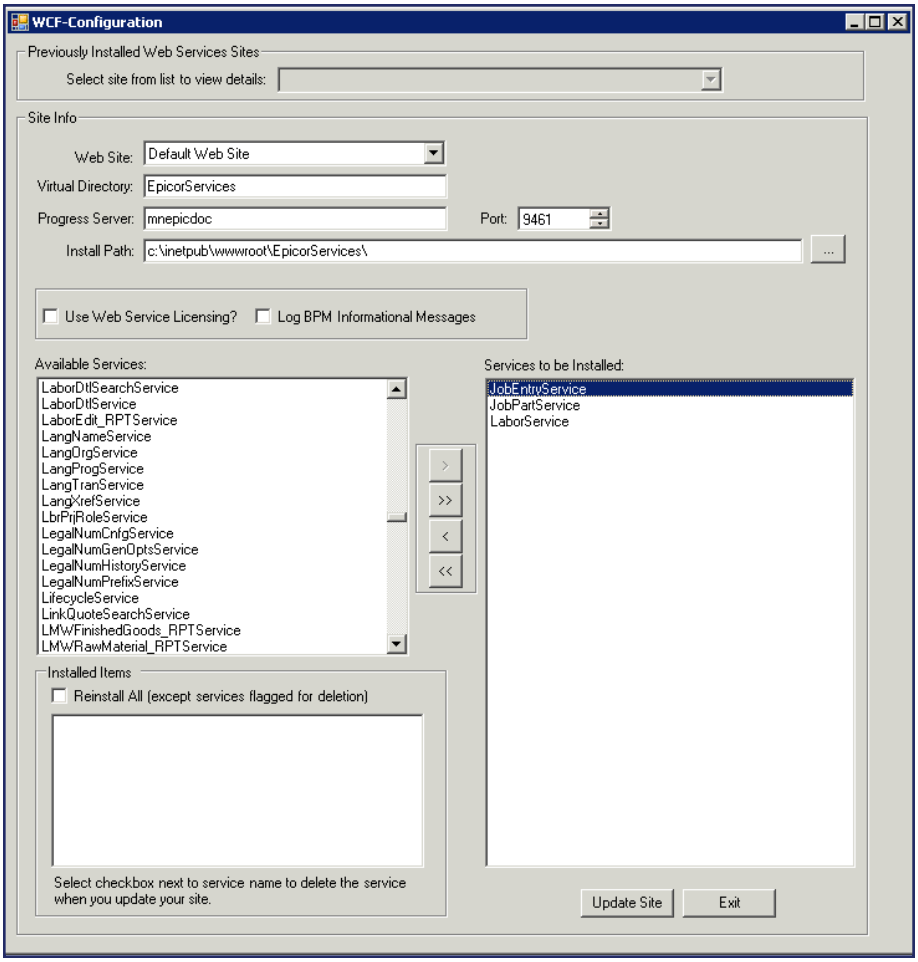
Windows Communication Foundation (WCF) is Microsoft®'s unified programming model for building service-oriented applications.

On your Epicor server, go to the **Epicor905** folder and open the **WCF Services** folder.

Open the **WCF-Service Installation Guide.pdf** file and follow the instructions to install the WCF Services you need.

The following image displays the WCF-Configuration tool that runs against the IIS6 web server.

If you are new to Service Connect, Epicor suggests you start to use WCF services. If you already have numerous Service Connect workflows or extensive custom coding around web services, you may want to install both the older WSE services and the WCF services and start to transition to the WCF version.

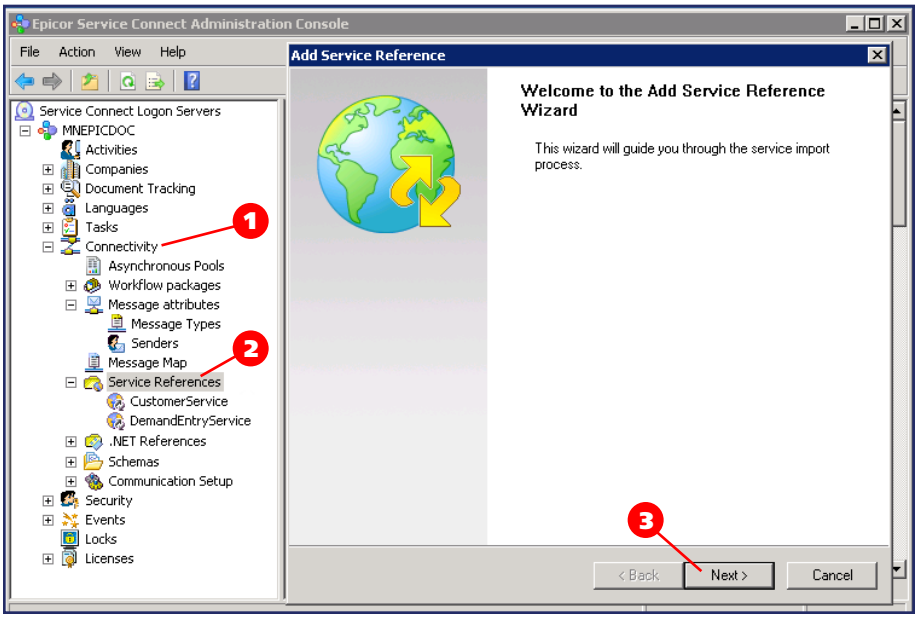


Import Service References into Service Connect

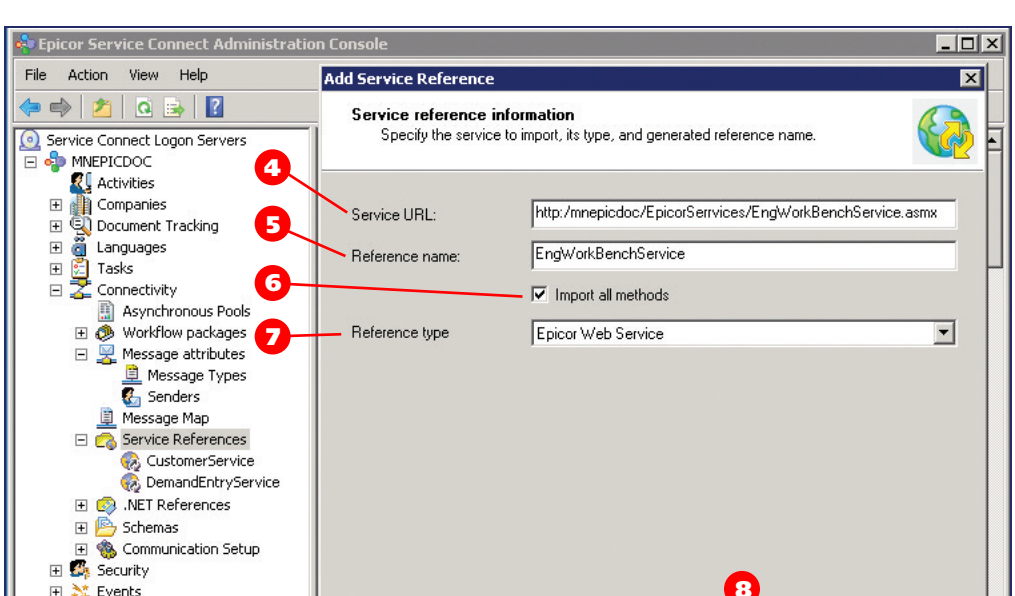
Service References are configured links to the published application services, such as Web Services or Windows Communication Foundation® services, which are outside of Service Connect. You use these references within the Workflow Designer.

To import Service References:

1. In the Epicor Service Connect Administration Console, expand the **Connectivity** node. [ESC9505_001]
2. Right-click the **Service References** node and select **Add Reference**.
3. In the **Add Service Reference** wizard welcome screen, click **Next**.



4. In the **Service reference information** screen, enter the **Service URL** of an Epicor web service you made available in the previous procedure. The URL syntax should appear as follows: http://<servername>/EpicorServices/<servicename>.asmx



The URL for Vantage services is as follows: http://<servername>/VantageServices/<servicename>.asmx. The URL for WCF services is as follows: http://<servername>/EpicorServices/<servicename>.svc

5. Notice the **Reference Name** field automatically displays the service name. The name will be used to identify the Web Reference in the ESC Administration Console and will prefix the web service methods in Web Method calls.

6. Leave the **Import all methods** check box selected. If you do not select Import allmethods, you can add the methods one-by-one later and configure them individually.

7. Select **Epicor Web Service** as the **Reference** type.

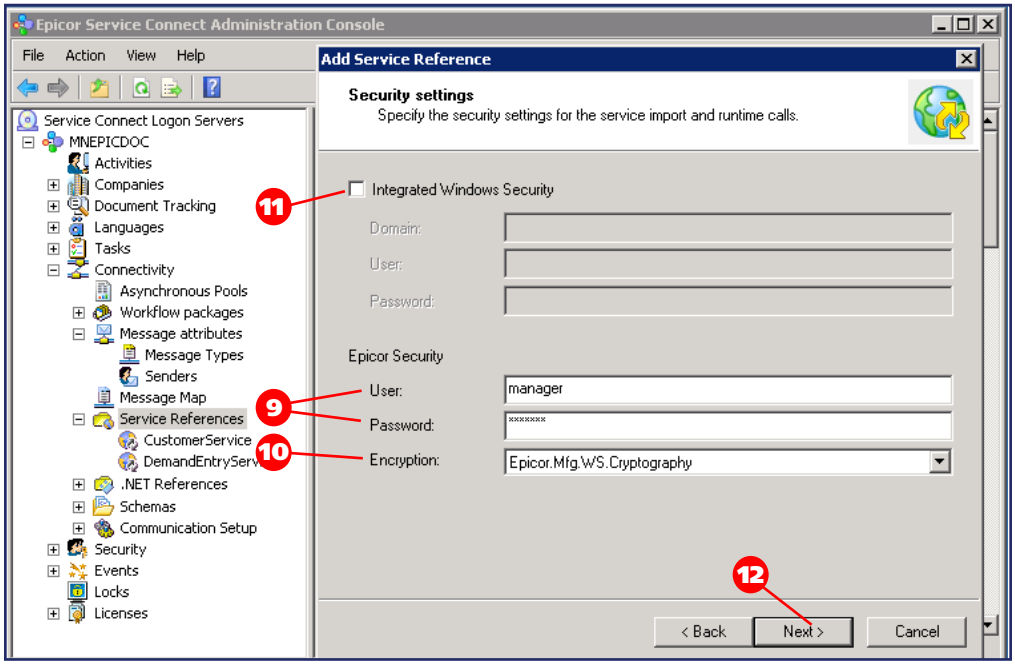
8. Click **Next**.

Select this Reference Type for Epicor Vantage as well. If you select Generic WCF Reference, update the Binding Settings in the following screen.

9. Under Epicor Security, enter an Epicor ERP **User** name and the corresponding **Password**.

10. For **Encryption**, select **icor.Mfg.WS.Cryptography**.

11. Use the **Integrated Windows Security** option only when you have set up Windows Authentication for the web services.



You can select the Integrated Windows Security check box and leave the Domain, User, and Password fields blank if you want to grant access to the web services using the same account Service Connect uses to run ScaDESRouter. For more information about how to use Windows Authentication with the web services, refer to the Enabling Windows Authentication section in the Web Services Developer Guide. This document is located in the Web Services folder on your Epicor ERP server.

12. Click **Next**.

13. Review the summary of the import parameters and click Next.
14. When the import is complete, a detailed log is provided on the Review import log screen.
15. Review the import log and click Finish.

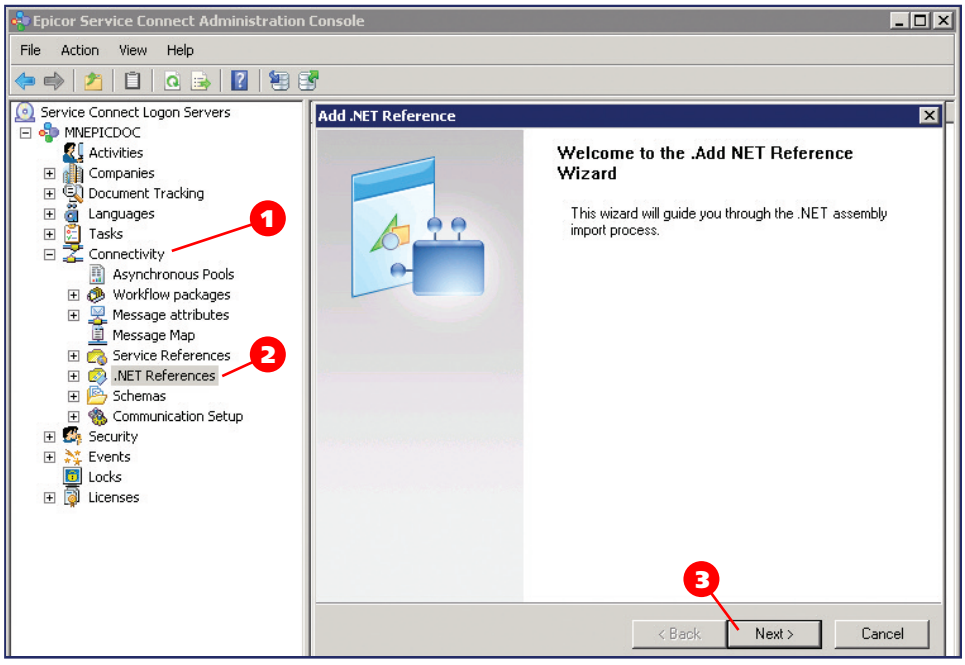
The schemas for the Web Services are imported into Service Connect and are available to use by a Web Method workflow activity.

Import .NET Reference into Service Connect

.NET References are links to specific assembly references. You use these references within the Workflow Designer.

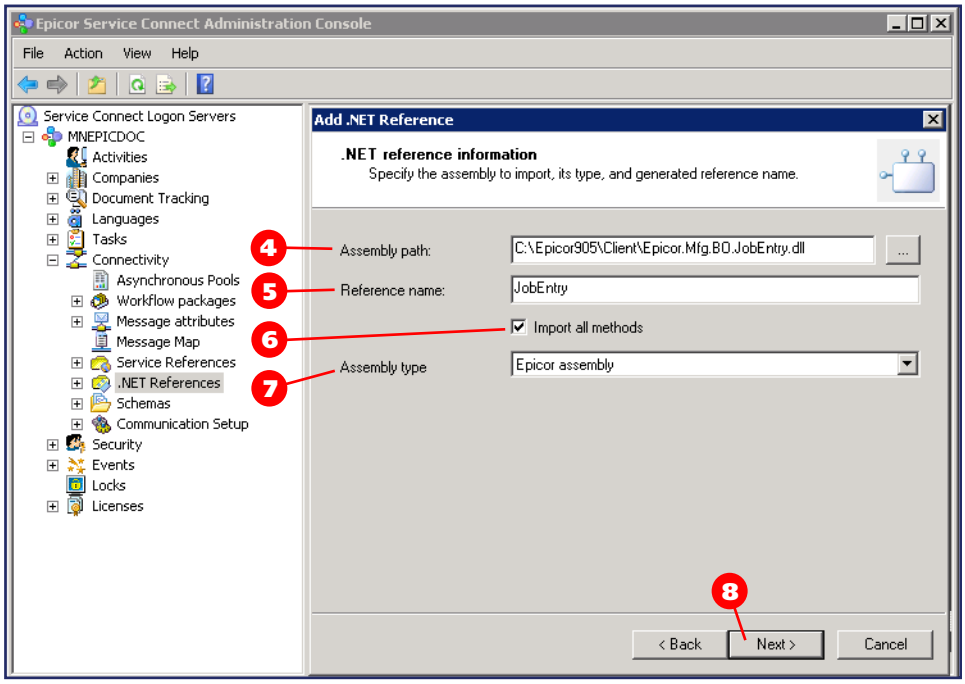
To import .NET References:

1. In the Epicor Service ConnectAdministration Console, expand the **Connectivity** node.
2. Right-click the **.NET References** node and select **Add Reference**.
3. In the **Add .NET Reference** wizard welcome screen, click **Next**.

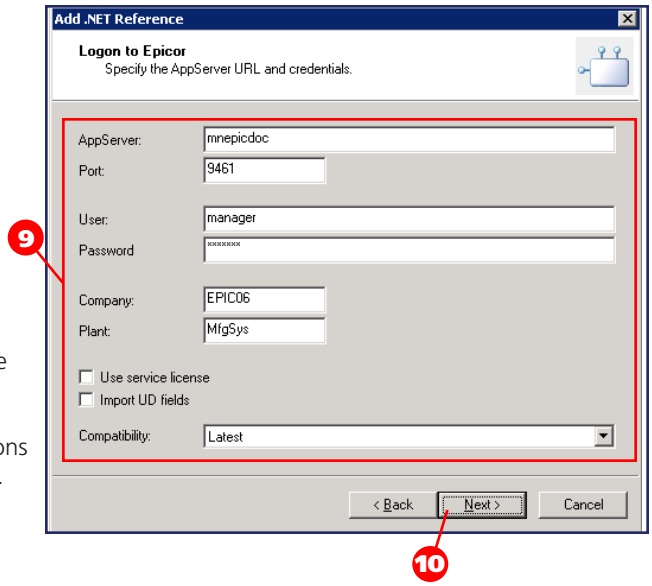


4. Next to the **Assembly path** field, click the browse button, navigate to the .NET assembly, and click **OK**." This is the path on the Service Connect server.
5. Notice the **Reference name** field automatically displays the .NET reference name. This name will be displayed in the imported .NET references list.
6. Select the **Import all methods** check box to import all .NET reference methods. If you do not select the Import all methods check box, you can add the methods one-by-one later and configure them individually.
7. Select **Epicor assembly** as the **Assembly type**.

   Select this type when you add a reference to Epicor Business Objects.
8. Click **Next.**



9. Enter the logon credentials applicable to your environment.
10. Click **Next**.
11. Review the summary of the import parameters and click **Next**.
12. When the import is complete, a detailed log is provided on the **Review import log** screen.
13. Review the import log and click **Finish**.



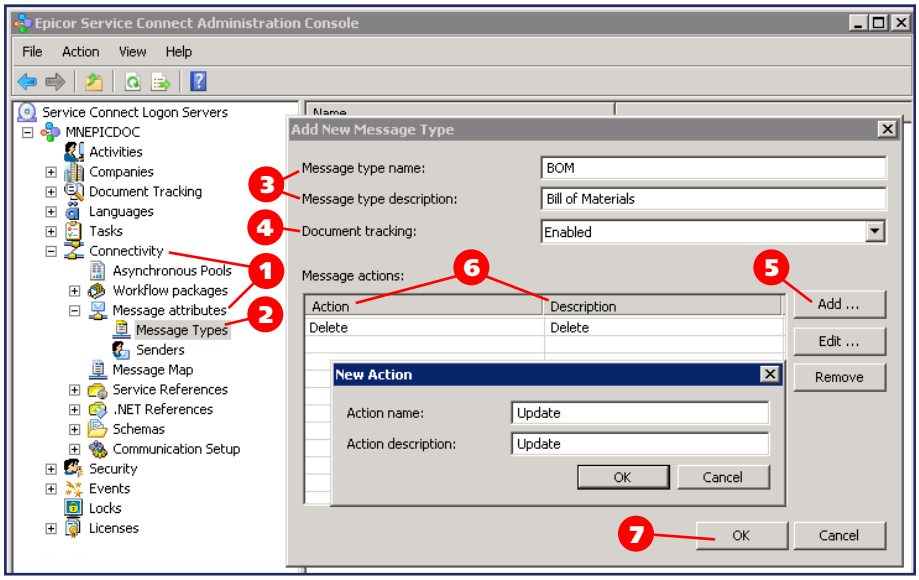
You can use the Test context menu item on Service and.NET reference methods to call particular methods from the ESC Administration Console to ensure you specified reference settings correctly. For example, you can test the GetList method to verify response accuracy of an imported reference. For detailed instructions on how to test a method, refer to Chapter 3: Connectivity Components.

Add a Message Type

Use Message Types to classify the different documents you plan to process in Service Connect.

To add a message type:

1. In the Epicor Service Connect Administration Console, expand the **Connectivity** node and then the **Message attributes** node.
2. Right-click **Message Types** and select **Add new Message Type**.
3. In the **Message type** window, enter a **Message type name** and **Message type description** that reflect the types of documents the workflow will handle.
4. Select **Enabled** in the **Document tracking** field to be able to view the processing of documents associated with the message type.
5. Click **Add**.
6. In the New Action window, enter an **Action** and **Description** that indicate the action the workflow will take when a document assigned to this message type enters the system.
7. Click **OK** until you exit all dialog boxes.

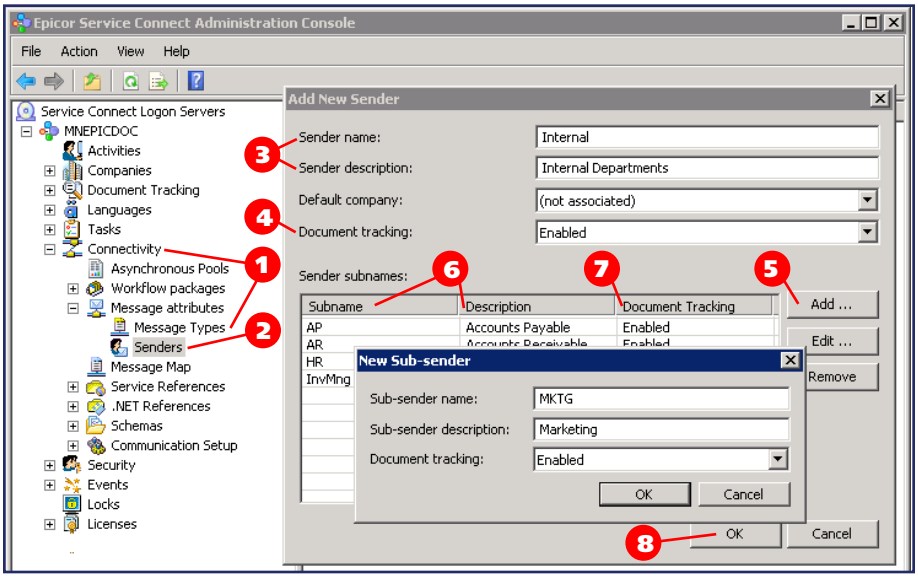


Add a Sender

Use Senders to define the origin of the documents sent to Service Connect. The origin can be an application name, your company name, or a computer name. Within a Sender, you can also define sub-names to more narrowly define a document's origin.

To add a Sender:

1. n the Epicor Service Connect Administration Console, expand the **Connectivity** node and then the **Message attributes** node.
2. Right-click **Senders** and select **Add New Sender**.
3. In the **Add New Sender** window, enter a **Sender name** and **Sender description** to identify the document's origin.
4. Select **Enabled** in the **Document tracking** field to be able to view the processing of documents associated with this Sender."
5. Click **Add.**
6. In the **New Sub-sender** window, enter a **Sub-sender name** and **description**.
7. Select **Enabled** in the **Document Tracking** field.
8. Click **OK** until you exit all dialog boxes.

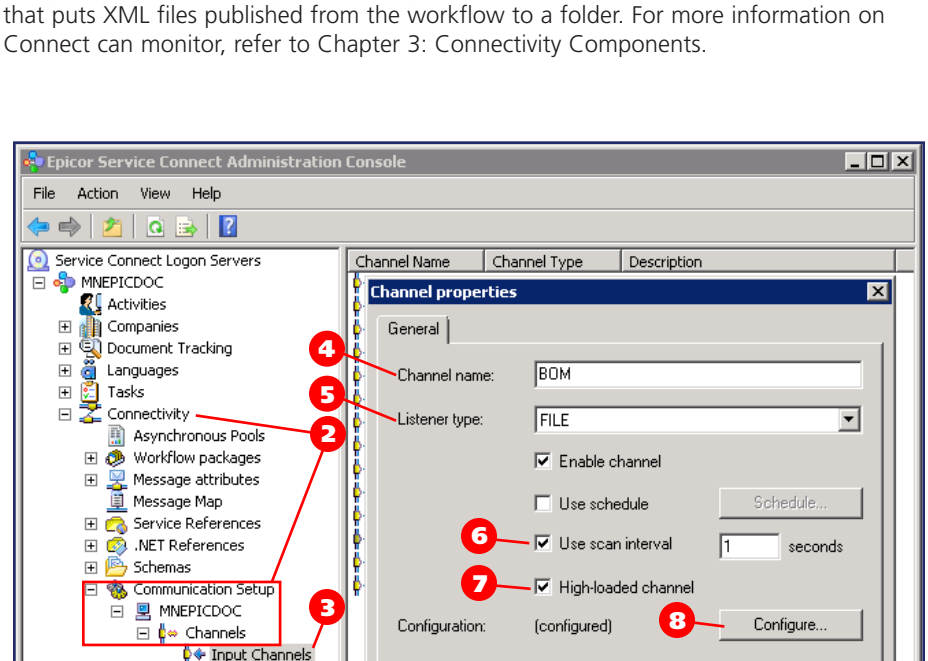


Create Input and Output Channels

An input channel serves as a workflow entry point. Output channels serve as a way to publish documents during workflow processing. The following steps show you how to create an input channel that accepts an Excel spreadsheet saved to a folder on a computer's file system and how to create an output channel that puts XML files published from the workflow to a folder. For more information on the types of communication channels Service Connect can monitor, refer to Chapter 3: Connectivity Components.

To add an input channel:

1. IOpen Windows Explorer and create a folder on a file system the Service Connect server can access. For example, create **C:\ESCSamples\BOM\In** and **C:\ESCSamples\BOM\Out**.
2. In the Epicor Service Connect Administration Console, expand the following nodes: **Connectivity**, **Communication Setup**, **machine name**, and **Channels**.
3. Right-click **Input Channels** and select **Add New**.
4. In the **Channel properties** window, enter a **Channel name**.
5. Select a **Listener type**.
6. Optionally, select the **Use scan interval** check box and select an appropriate interval in seconds.
7. Optionally, select **High-loaded channel**.



When this option is selected, the system will try to allocate a separate thread for the channel.

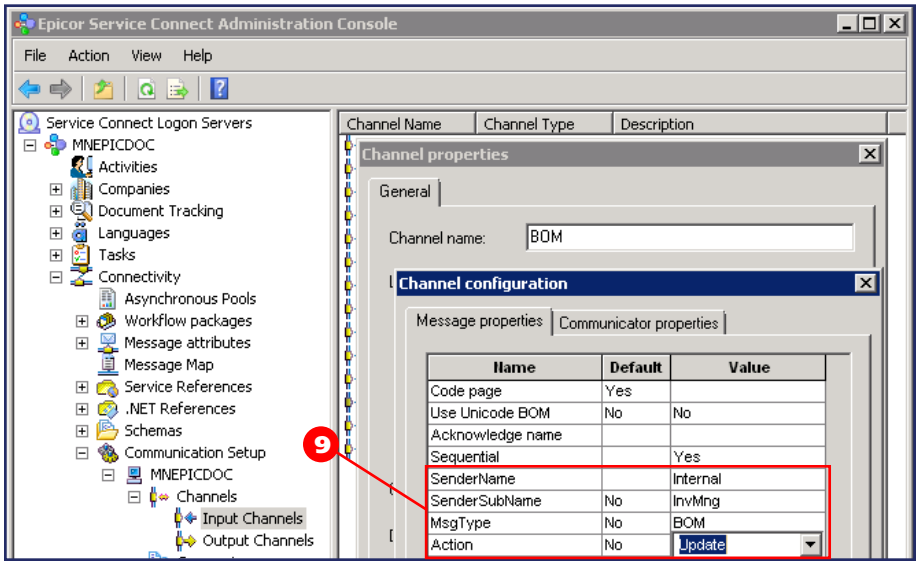
Review the Communication Setup and Channels section in Chapter 3: Connectivity Components for more information on how to allocate a separate thread for the channel.

8. Click **Configure**.

9. In the **Channel configuration** window, for the following fields, select the values you created in the previous procedures.

   - SenderName
   - SenderSubName
   - MsgType
   - Action

   The values in these fields are added to the document when it enters the input channel.



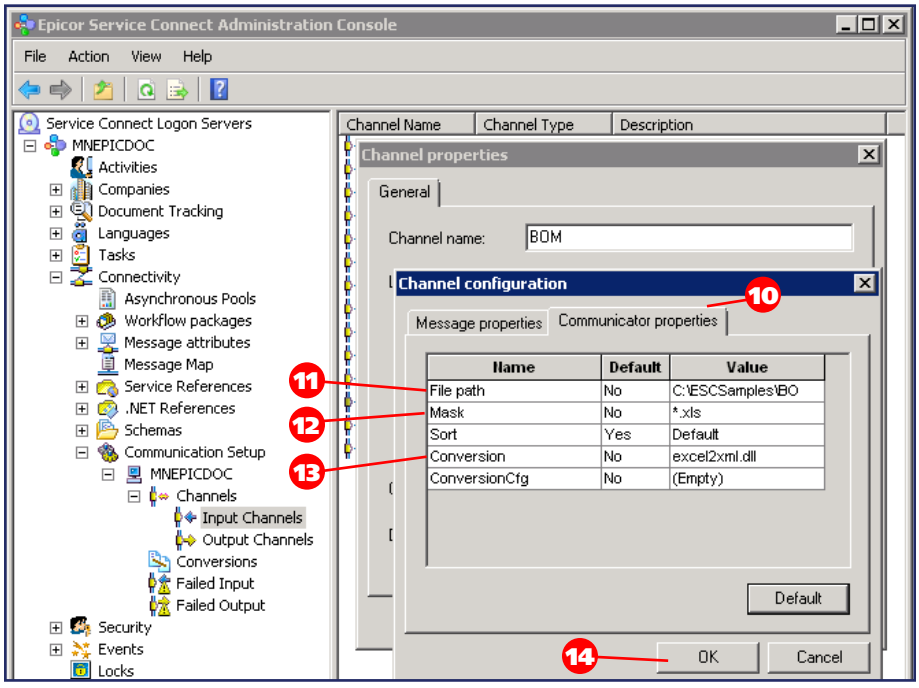
10. Click the **Communicator properties** tab.

11. Enter the **File path** to the new folder created at the beginning of this procedure for the file input.

12. Enter ***.xls** as the **Mask**.

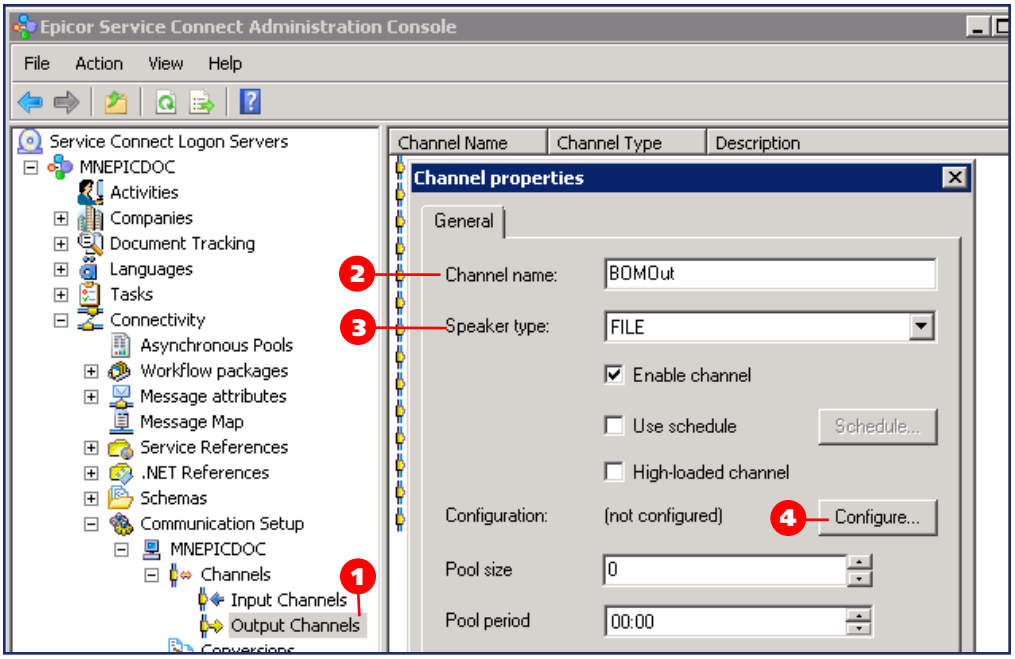
13. Select **excel2xml.dll** as the **Conversion**.

14. Click **OK** until you exit all dialog boxes.

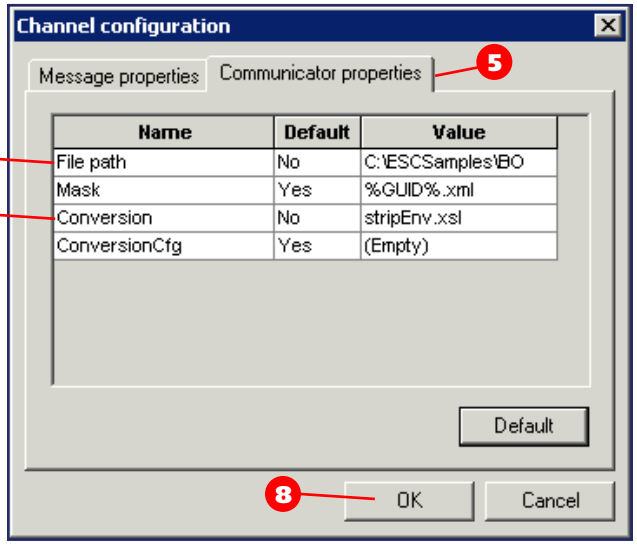


To add an output channel:

1. Right-click **Output Channels** and select **Add New**.
2. In the **Channel properties** window, enter a **Channel name**.
3. Select **FILE** as the **Speaker type**.
4. Click **Configure**.



5. In the **Channel configuration** window, click the **Communicator properties** tab.
6. Enter the **File path** to the folder where you want to publish the output.
7. Select **stripEnv.xsl** as the **Conversion**.
8. Click **OK** to exit all dialog boxes.

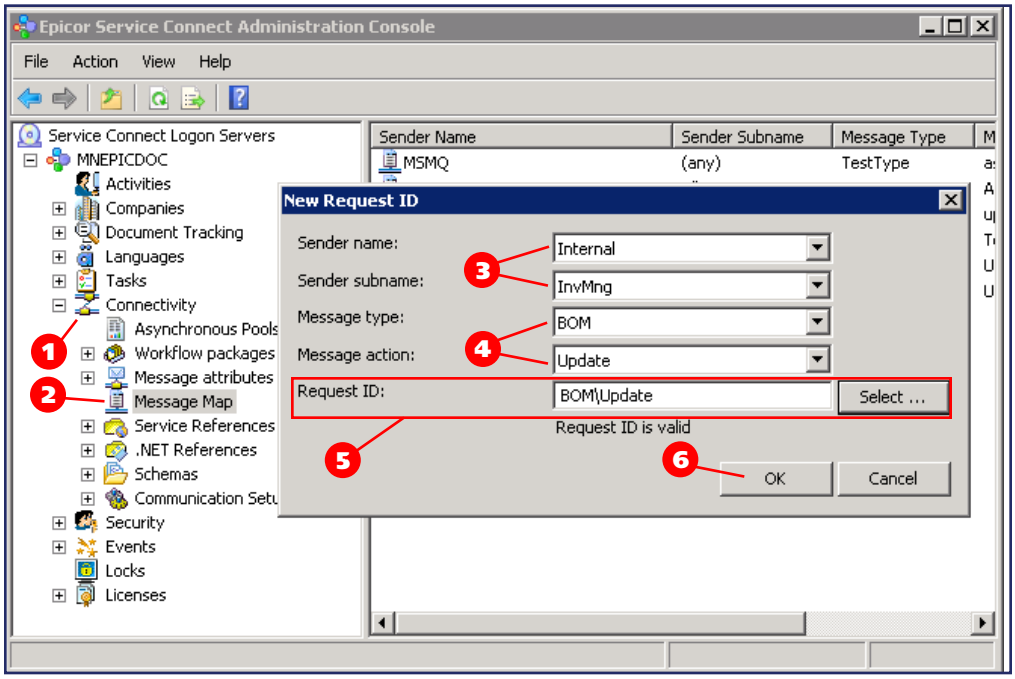


Create the Message Map

Message maps are used to route a document once the document is processed in an input channel. Service Connect uses the message map to send the document to the appropriate workflow based on the Sender and Message Type.

To create a message map:

1. In Epicor Service Connect Administration Console, expand the **Connectivity** node.
2. Right-click **Message Map** and select **Add new Request**.
3. In the **New Request ID** window, select the **Sender name** and **Sender subname** set up previously.
4. Select the **Message type** and **Message action** set up previously.
5. If you have created a workflow, click **Select** to add it to the **Request ID** field; otherwise, leave the Request ID blank and complete it later.
6. Click **OK**.



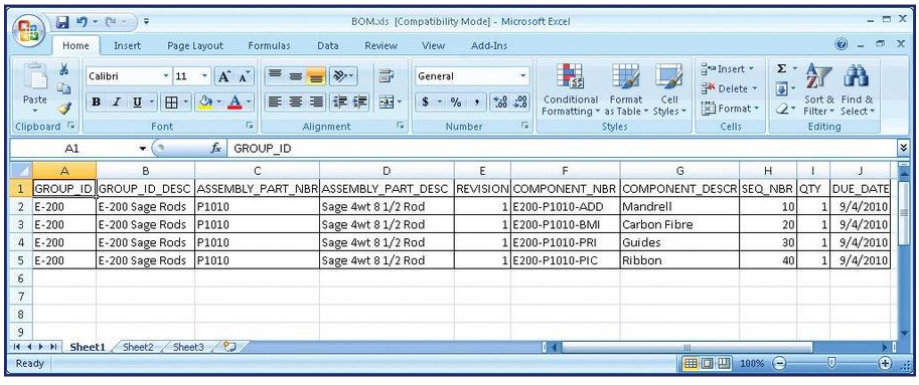
Create Document Schemas

The data in the inbound document must conform to a schema Service Connect recognizes. To meet this need, you can generate a schema based on the incoming document. Any schema you generate is known as a user schema, as opposed to a Web Service or a .NET reference schema, which is generated when you add a Service Reference or .NET Reference in the ESC Administration Console.

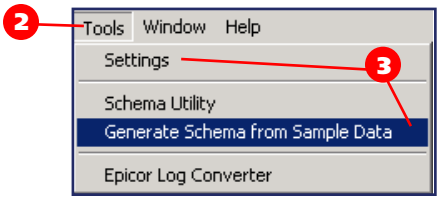
This example shows how to generate a schema based on a Microsoft Excel® spreadsheet.

To generate the schema and import it into Service Connect:

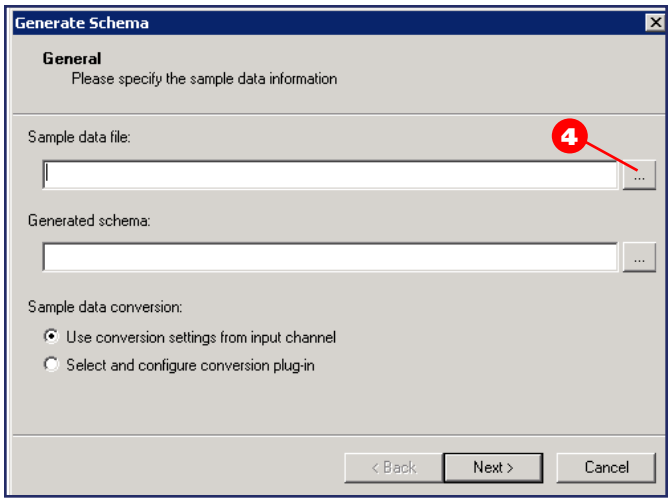
1. Create or locate a spreadsheet similar to the one shown here.



2. Log into the **Epicor Service Connect Workflow Designer**.
3. From the **Tools** menu, select **Generate Schema from Sample Data**.

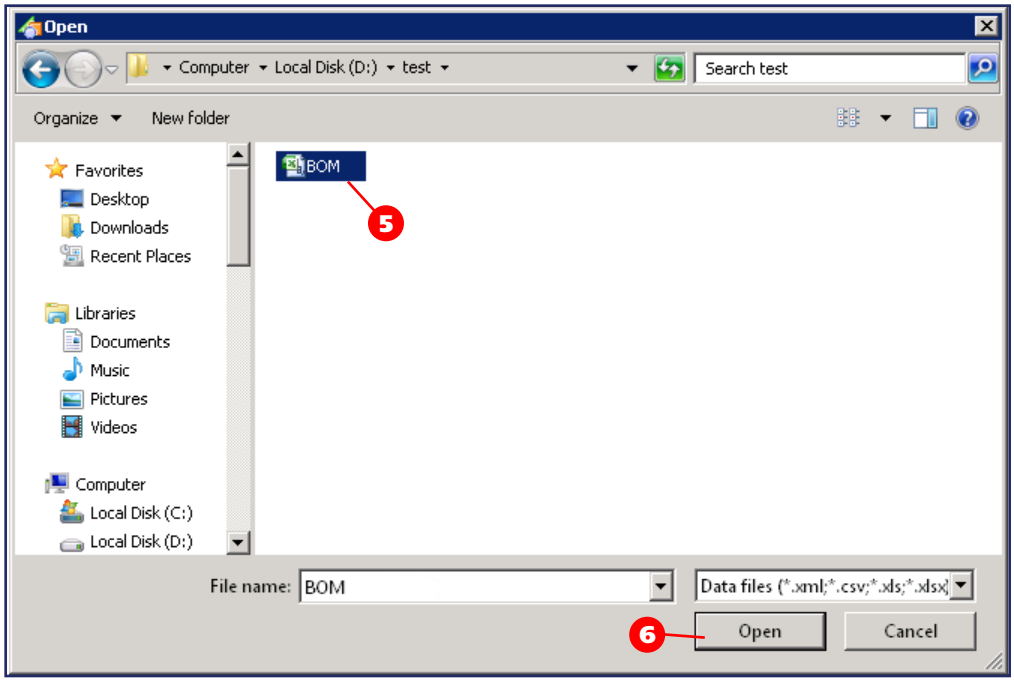


4. The **Generate Schema** window displays. Next to the **Sample data file** field, click the browse button.

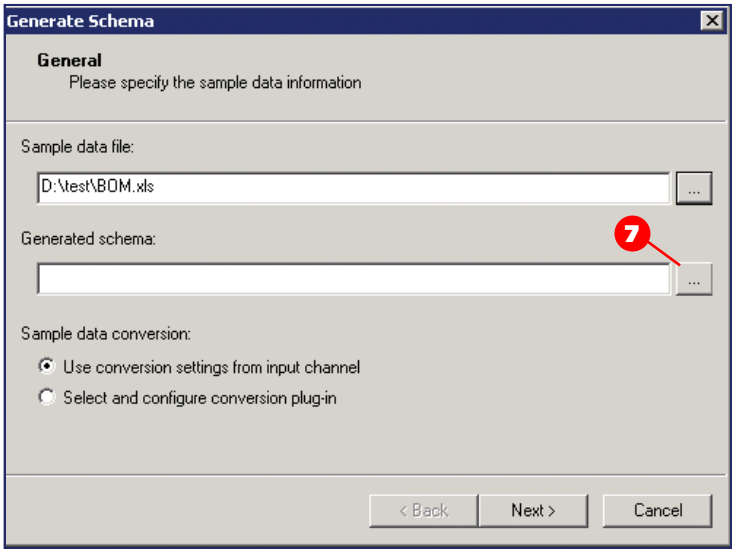


5. In the standard **Open** window, browse to and select the Excel file.

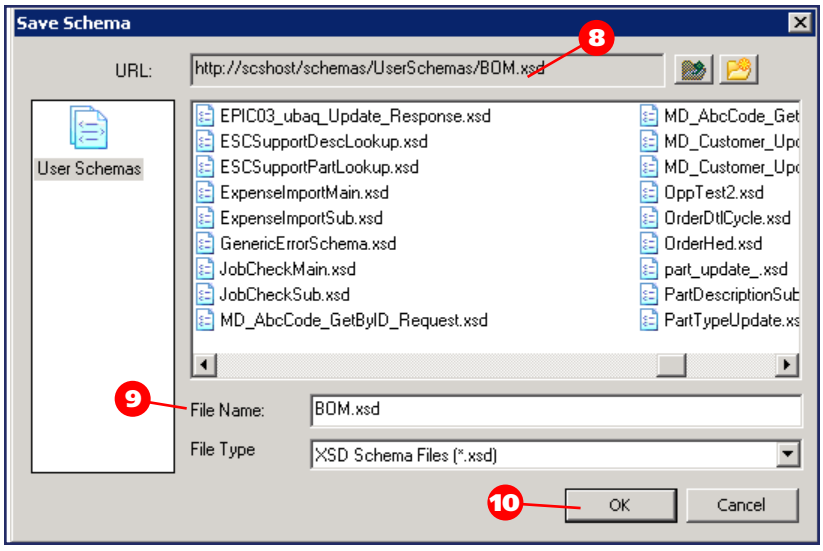
6. Click **Open**.



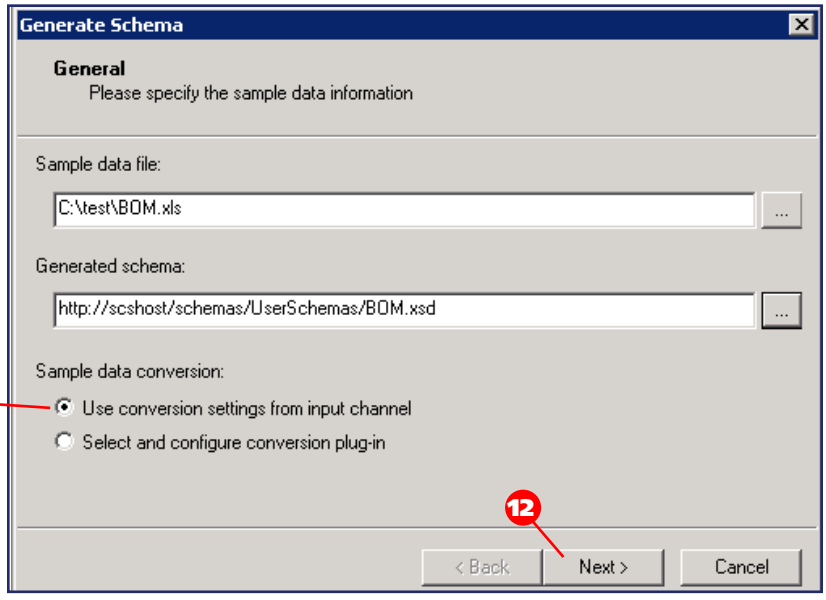
7. On the **Generate Schema** screen, next to the **Generated schema** field, click the **... (Ellipse)** button.



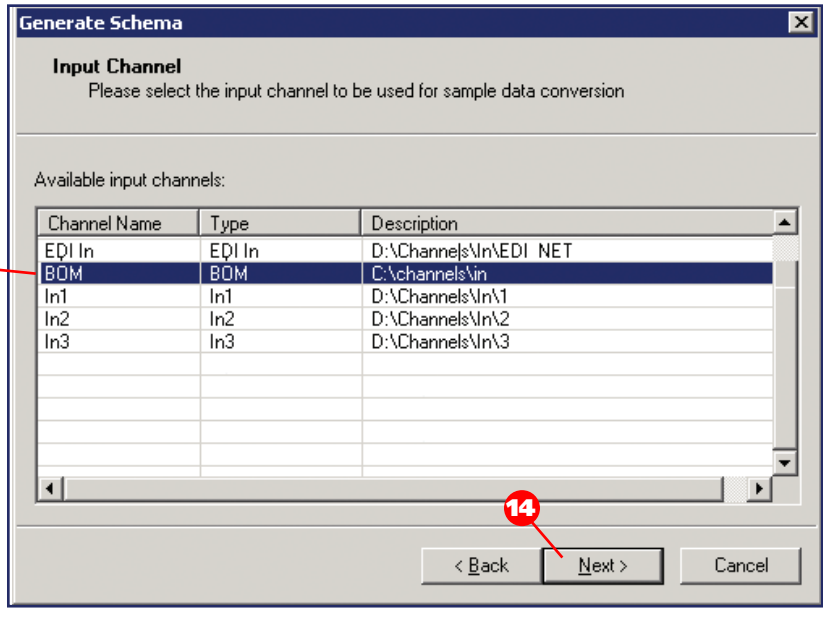
8. In the **Save Schema** dialog, navigate to the folder where to save the schema.
9. In the **File Name** field, enter the schema file name.
10. Click **OK**.



11. On the Generate Schema screen, select the **Use conversion settings from input channel** option to use standard conversion on the channel.
12. Click **Next**.



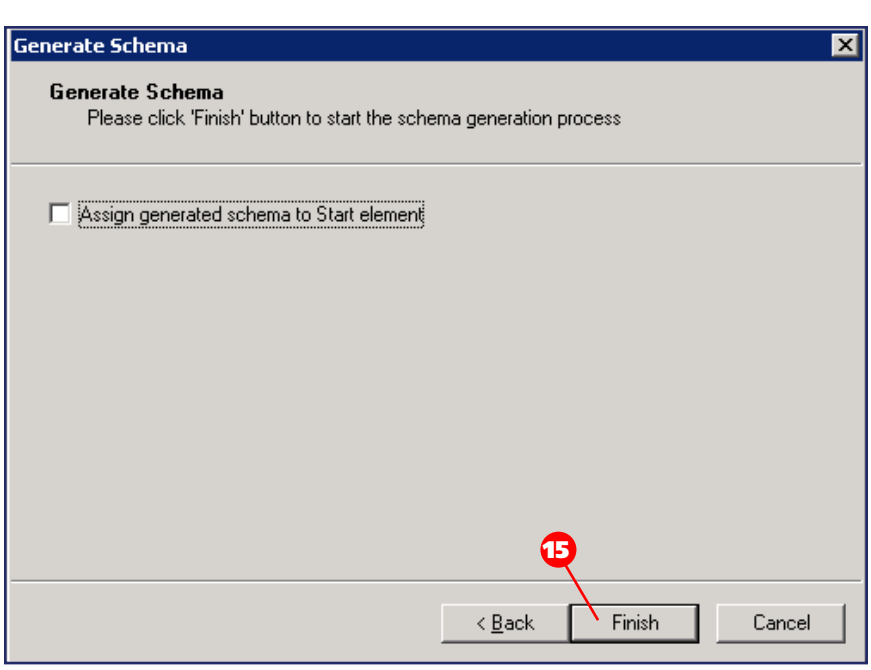
13. From the **Available input channels** list, select a FILE channel which uses the **excel2xml.dll** conversion. In this example, you select a channel named **BOM**.
14. Click **Next**.



15. Click **Finish**.

Service Connect generates a schema for your spreadsheet and imports it. You can now use the schema within a workflow to perform business processes with the document data.

You can now use the schema in workflow activities to perform business processes with the document data.



Epicor ERP Tracing

Use the Tracing Options program to set up a trace log that captures all public business object calls, which includes calls the client initiates as well as calls from one object to another. When you activate this log, any public business calls sent to the server are recorded in the log.

Start a Trace Log

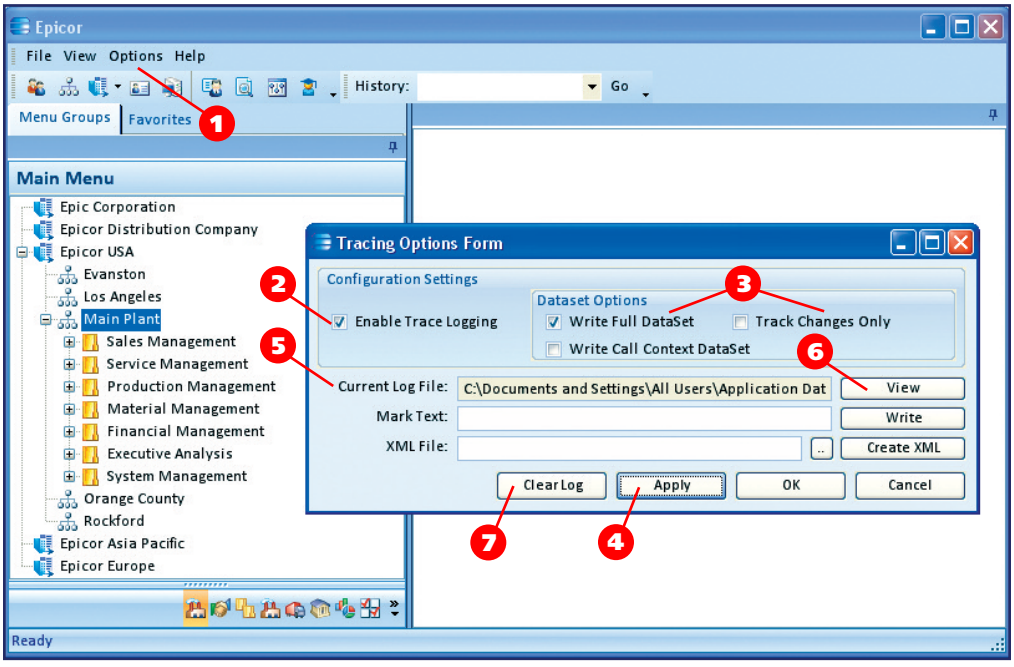
You use the tracing log to track Epicor ERP business object calls. You can then use this information to see how successful operations are performed within the application, so you can map the business logic as needed to a Service Connect workflow.

To start a trace log:

1. In Epicor ERP, from the **Options** menu, select **Tracing Options**.
2. The **Tracing Options Form** displays. Select the **Enable Trace Logging** check box.
3. Select **Write Full DataSet** if you want to view complete XML datasets passed to and returned from the server, or, select **Track Changes Only** to view only changed in the dataset.

   You can only select one of the Dataset Options.
4. Click **Apply**.

   The application begins a trace log.

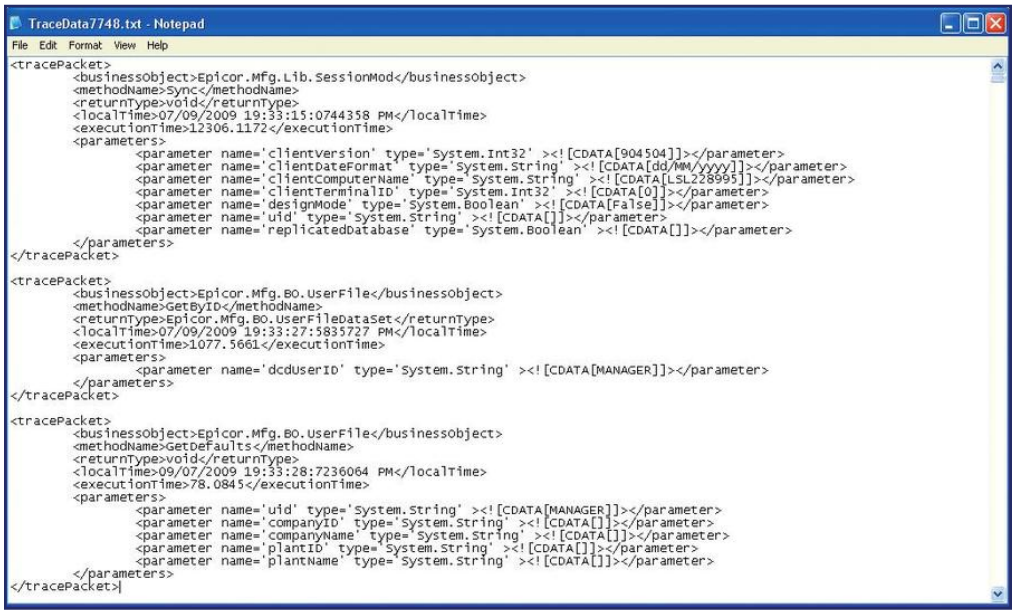


5. All business object calls from this point forward are entered into the log file shown in the **Current Log File** field.

6. Click **View** to review the log file.

7. Click **Clear Log** to clear the log.

The trace log may look similar to the one shown here.



Exit the System Monitor to reduce the number of trace packets in the log. To exit the System Monitor, right-click the System Monitor icon in the Notification Area (right side of the taskbar) and select Exit. You can restart the System Monitor from the following menu path: System Management / Utilities / System Monitor.

You can use the Epicor Log Converter to convert an Epicor log to a Service Connect workflow. To open the Converter, in the Workflow Designer, select Epicor Log Converter from the Tools menu on the Main Menu.

Create an XML Trace Report

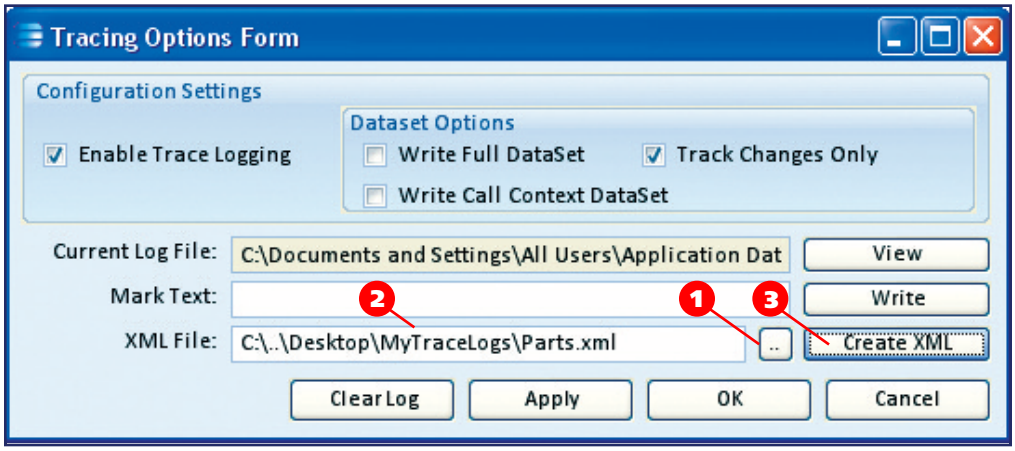
You can save the trace file to an XML file. The system generates an XSL stylesheet so the XML file is formatted in a report layout.

To create an XML trace report:

1. In the **Tracing Options Form**, click **Browse**.

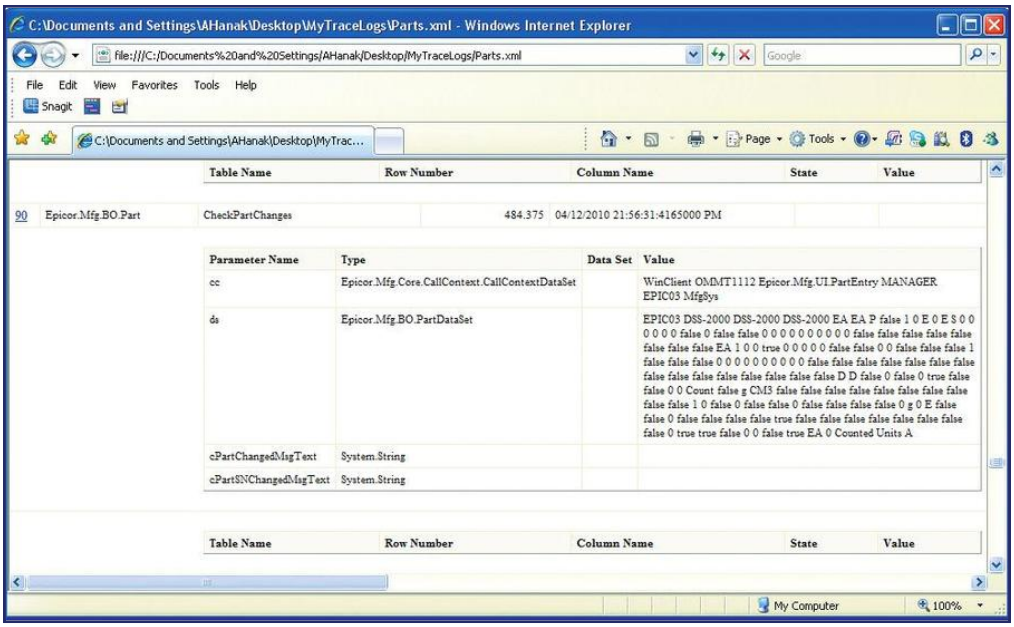
2. Browse to a location on your computer where you want to create the file.

3. Click **Create XML**.



To view the XML trace, go to the folder where the XML file was created and double-click the file. The following shows a sample XML trace.

If you send the XML file to someone else, be sure to include the file named TracePacket_Default.xsl.



Enable IIS-Level Tracing

It is also possible to enable web service tracing directly from the Epicor ERP web services. To learn more about this type of trace and how to enable it, review the Data Processing Tracing section in the Epicor ERP Web Services Developer Guide. This document is located in the Web Services folder on your Epicor ERP server.

Service Connect and Epicor ERP Business Process Management

Use Business Process Management (BPM) to create workflows that automate, execute, and monitor business processes. The workflows, known as directives, extend the base functionality of Epicor ERP without requiring customizations or affecting your ability to take future releases of the software. A directive is a set of conditions and actions associated with either a business object method or a database table. If the specified conditions are met, the actions are executed. To further extend the versatility of BPM, you can call Service Connect workflows from a BPM directive.

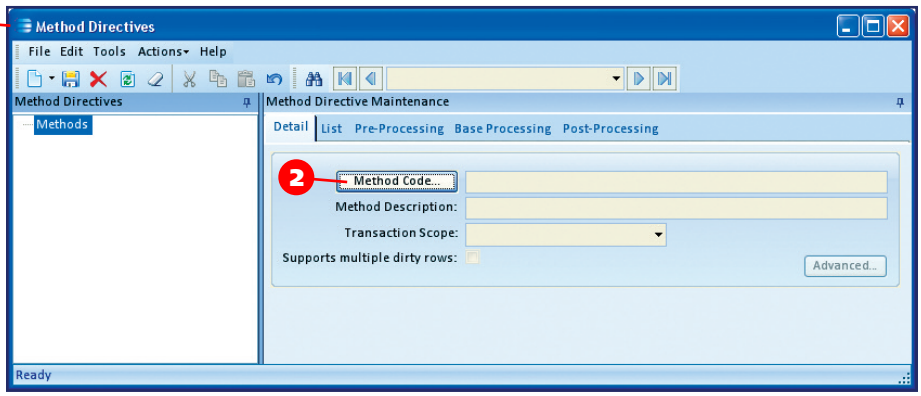
The following procedures show how to call a Service Connect workflow from a directive associated with a business object. Directives for database tables, known as data directives, can also call workflows. For more information about BPM, refer to the Epicor ERP application help or the Epicor ICE 2.5 Tools User Guide.

Locate a Business Object Method

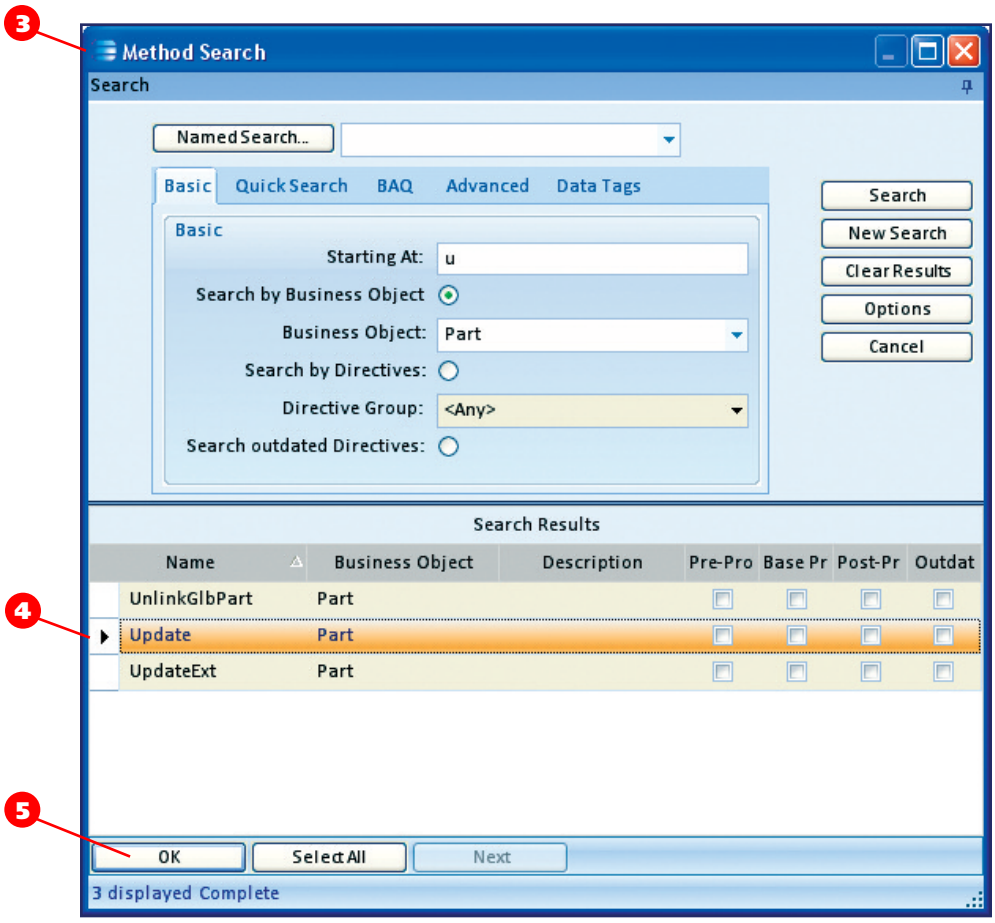
The first step to create a method directive is to identify the business object method that will trigger the BPM workflow.

To locate a business object method:

1. In Epicor ERP, open **Method Directives** using the following menu path: **System Management > Business Process Management > Setup > Method Directives**.
2. Click **Method Code**.



3. Use the **Method Search** to locate a business object method.
4. Select one or more methods.
5. Click **OK**.

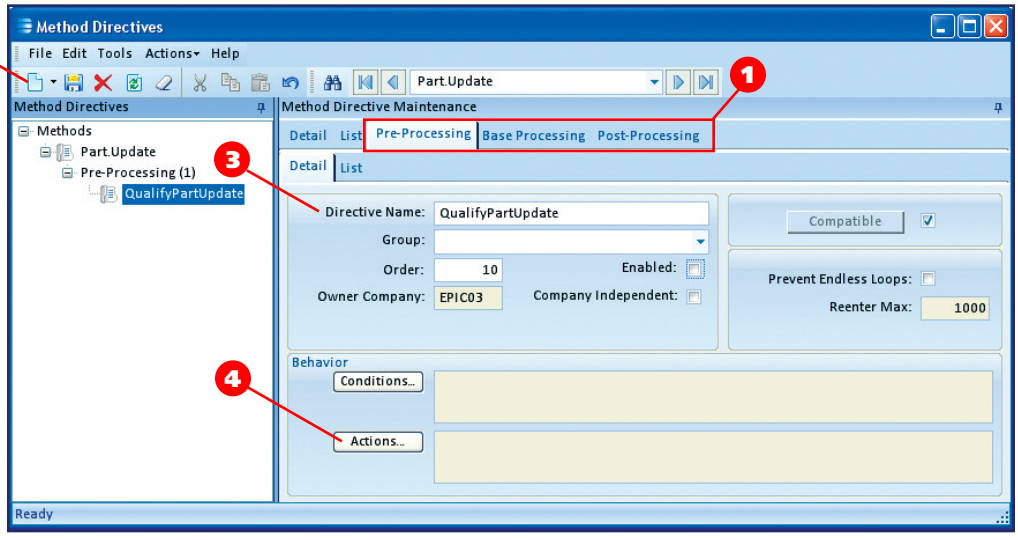


Create a Directive that Calls a Service Connect Workflow

A complete method directive usually (but not always) contains one or more condition statements that are evaluated when the business object method is executed. If the conditions are satisfied, the application performs the actions associated with the directive. However, this example only shows how to set up the action that calls a Service Connect workflow.

To create an action that calls a Service Connect workflow:

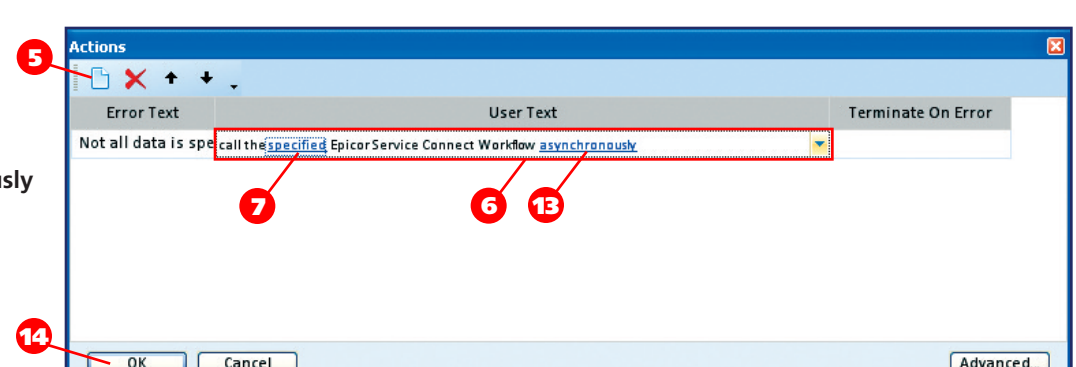
1. Click the **Pre-Processing**, **Base Processing**, or **Post-Processing** tab. For details about the differences between pre-, base, and post-processing directives, refer to the application help or the Epicor ICE 2.5 Tools User Guide.
2. Click **New**.
3. Enter a **Directive Name**.
4. Click **Actions.**



5. In the **Actions** window, click **New**.

6. Select **call the specified Epicor Service Connect Workflow asynchronously** in the **User Text** field.

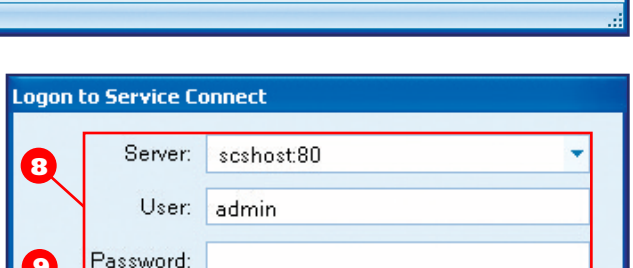
7. Click the word **specified**.



8. In the **Logon to Service Connect** window, enter the **Server**, **User**, and **Password** required to access Service Connect. These are the credentials used to log into Service Connect, and not the credentials for a Windows user account.

9. Click **OK**.

   The system loads all the Service Connect workflow packages available on the server.



10. In the **Select Workflow** window, browse to the workflow you want to call from the method directive in the **Chosen Workflow** field.

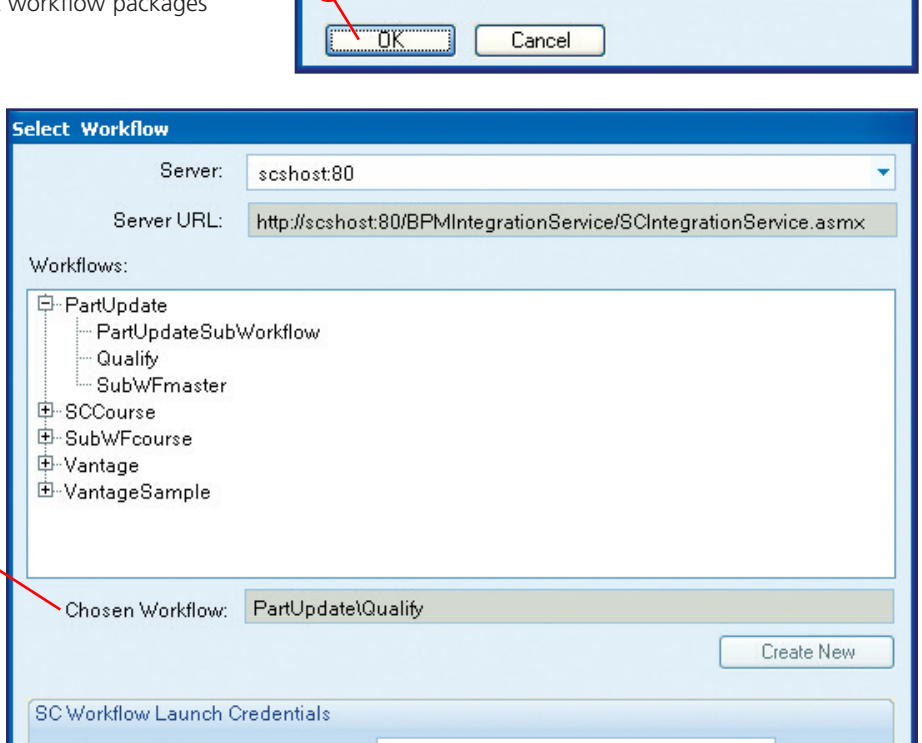
11. Enter the **SC User** and **SC Password**.

12. Click **OK**.

13. Click the word **asynchronously** to toggle between synchronous and asynchronous execution.

    Asynchronous workflow calls are executed according to the schedule defined in the BPM Action Process program. Synchronous workflow calls are made when the action executes.

14. Click **OK** until you exit all dialog boxes.



BPM Integration Completion

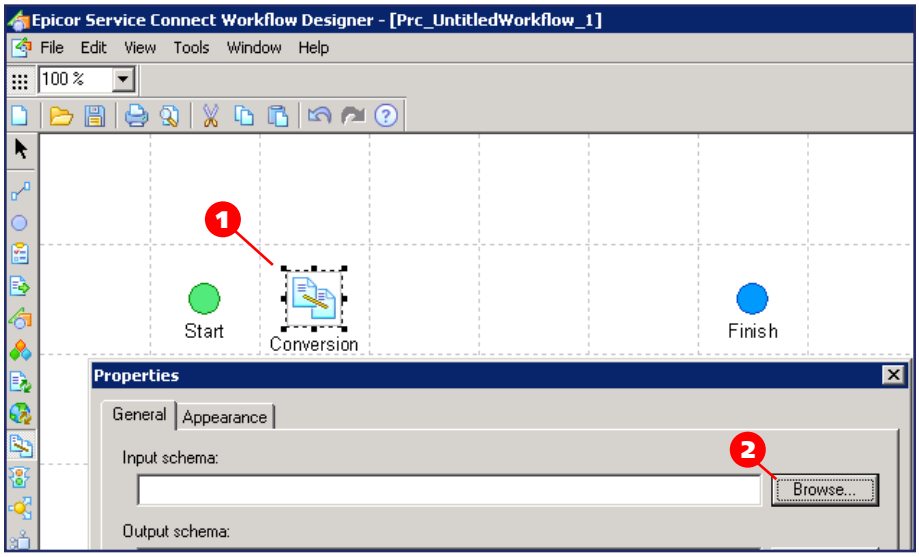
When the BPM directive executes, Epicor ERP sends the pre-update dataset for the business object method to Service Connect. Since Epicor ERP calls the Service Connect workflow directly, it is not necessary to set up message attributes, a message map, or an input channel to route the information.

Epicor ERP creates schemas for the dataset the application sends to Service Connect. The schemas are located in the User Schemas directory and are named with the business object and method name. To use the data from Epicor ERP in the Service Connect workflow, you must add the schema as the input schema for the first Conversion activity, or if the first activity in the workflow is not a Conversion, you can assign the schema to the Start activity.

Example

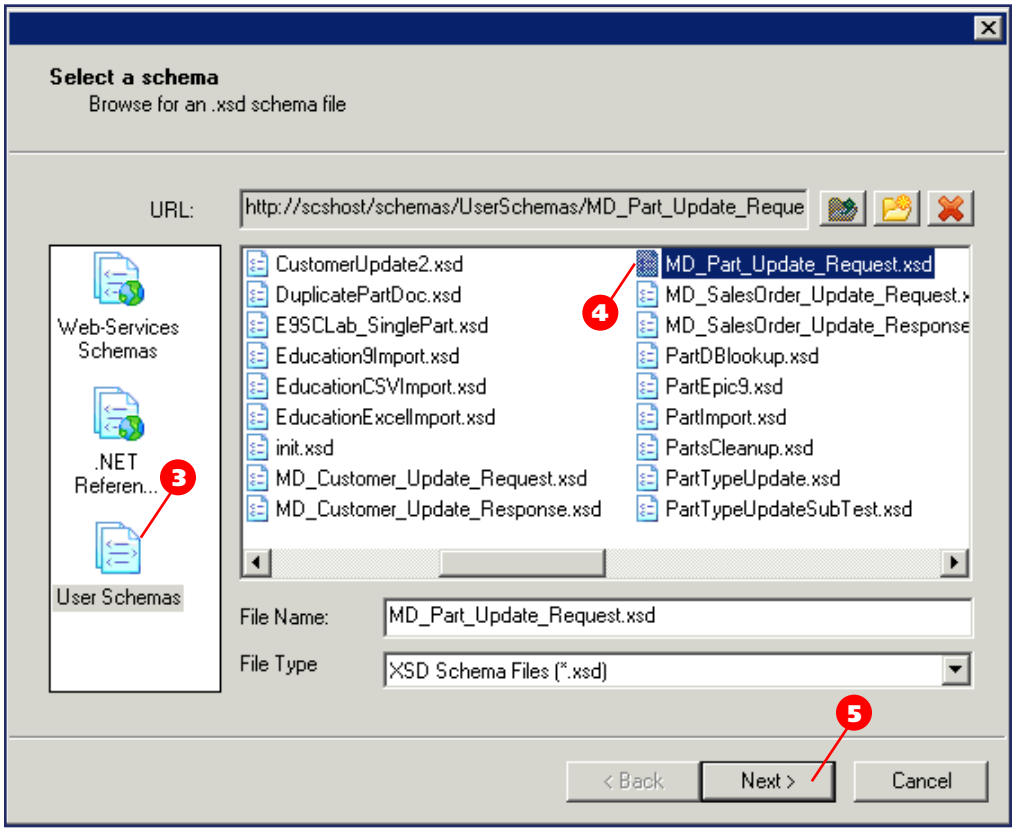
Using the Part.Update method, complete the integration:

1. Right-click the **Conversion** activity that immediately follows Start and select **Properties**.
2. In the **Properties** window, click **Browse** to select the Input schema.



3. Click **User Schemas**.
4. Browse to and select the request schema for the business object method.

   In this example, MD stands for method directive.
5. Click **Next**.
6. Click **Finish**.



After this procedure, you can configure the Output Schema and the transformation for the Conversion activity. Refer to Chapter 4: Workflow Designer for more information on how to define Conversions.

Service Connect and Epicor Enterprise

This section explains how to integrate Service Connect with Epicor Enterprise Financials and SCM and Epicor for Service Enterprises. Integration with Epicor Enterprise Financials requires the installation of Epicor Integration Hub, a collection of web services designed to integrate Epicor Enterprise with Epicor for Service Enterprises and other applications. In addition to integration information, this section also explains how to use the tracing tools available in Epicor for Service Enterprises.

Integration Overview

You must perform the following tasks to create workflows that interact with Epicor Enterprise Financials and SCM and Epicor for Service Enterprises.

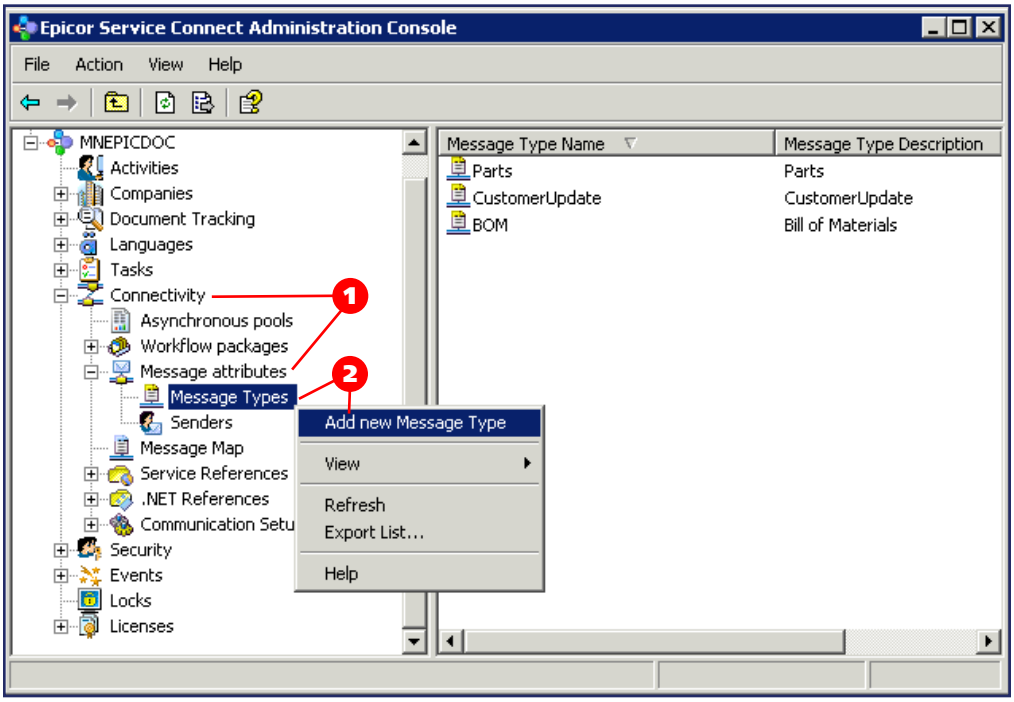
- Define a Message Type.
- Add a Sender.
- Create a Message Map.
- Add a Service Reference to the application Web Services.
- Add an Input and an Output Channel.
- Generate a schema to handle incoming data.

Add a Message Type

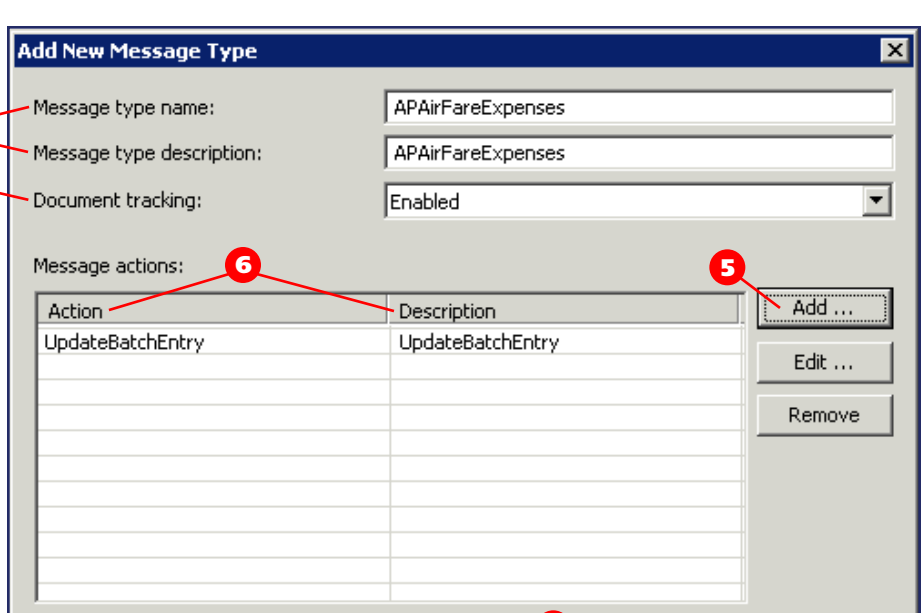
Use Message Types to classify the different documents you plan to process in Service Connect.

To add a message type:

1. In the ESC Administration Console, expand the **Connectivity node** and expand the **Message attributes** node.
2. Right-click **Message Types** and select **Add new Message Type**.



3. In the **Add New Message Type** window, enter a **Message type name** and **Message type description** that reflect the types of documents the workflow will handle.
4. Select **Enabled** in the **Document tracking** field, to be able to view the processing of documents associated with the message type.
5. Click **Add**.
6. Enter an **Action** and **Description** that indicate the action the workflow will take when a document assigned to this message type enters the system.
7. Click **OK** until you exit all dialog boxes.

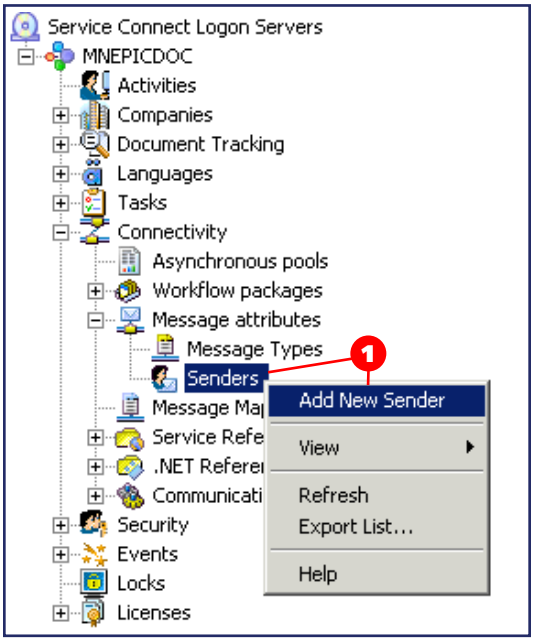


Add a Sender

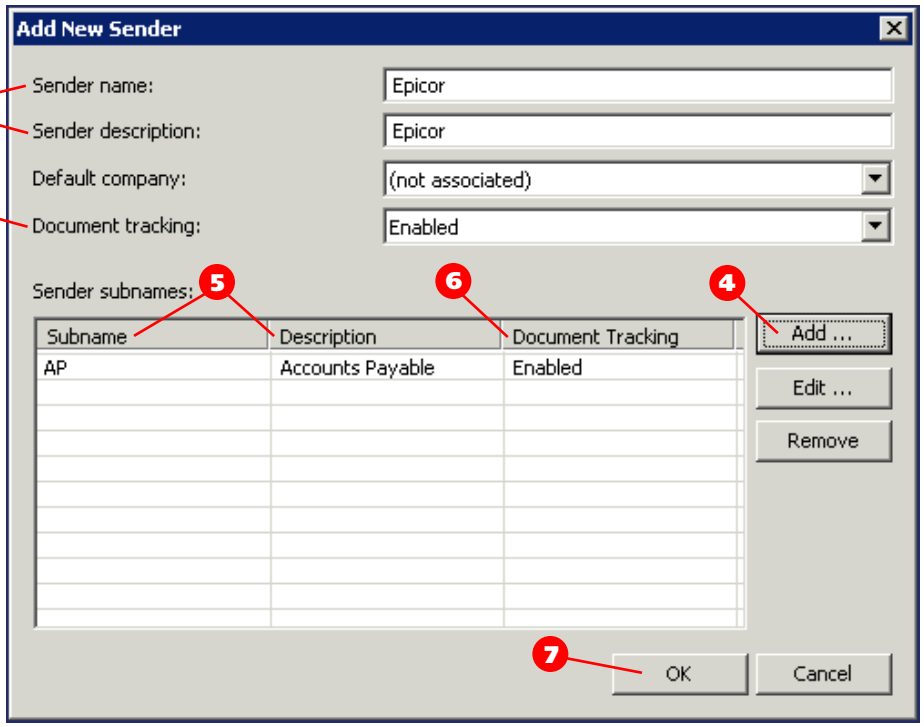
Use Senders to define the origin of the documents sent to Service Connect. This can be an application name, your company name, or a computer name. Within a Sender, you can also define subnames to more narrowly define a document's origins.

To add a Sender:

1. Right-click **Senders** and select **Add New Sender**.



2. In the **Add New Sender** window, enter a **Sender name** and **Sender description** to identify the document's origin.
3. Select **Enabled** in the **Document tracking** field, to be able to view the processing of documents associated with this Sender.
4. Click **Add.**
5. Enter a **Subname** and **Description**.
6. Set the **Document Tracking** field to **Enabled**.
7. Click **OK** until you exit all dialog boxes.

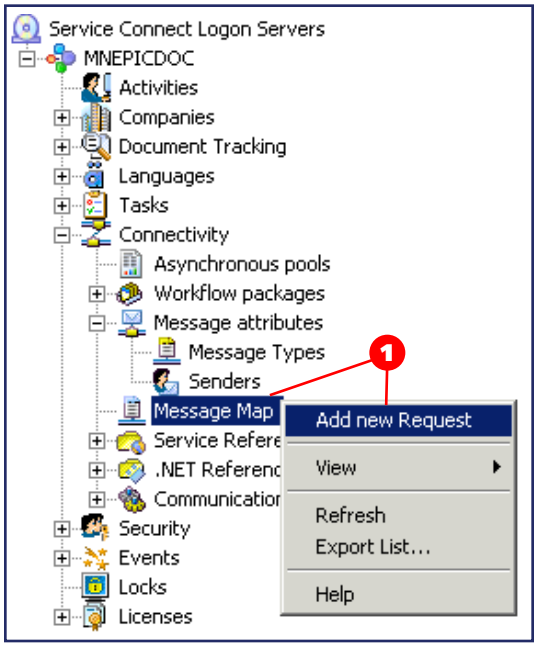


Create a Message Map

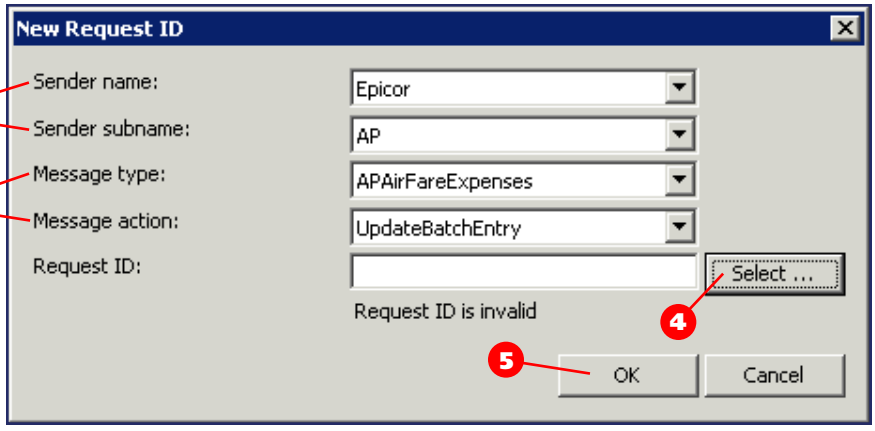
Message maps are used to route a document once the document is processed in an input channel. Service Connect uses the message map to send the document to the appropriate workflow based on the Sender and Message Type.

To create a message map:

1. Right-click **Message Map** and select **Add new Request**.



2. In the **New Request ID** window, select the **Sender name** and **Sender subname** set up previously.
3. Select the **Message type** and **Message action** set up previously.
4. If you have created a workflow, click **Select** to add it to the Request ID field; otherwise, leave the Request ID field blank and complete it later.
5. Click **OK**.



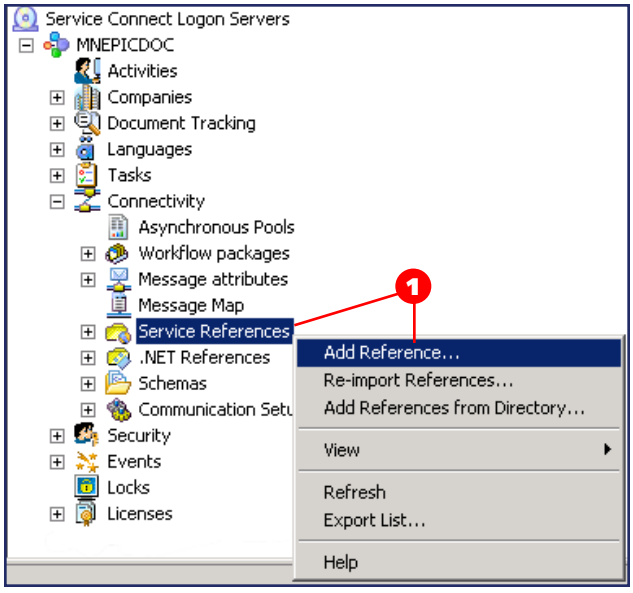
The result of your entries is that when a document is received in an input channel (yet to be defined), Service Connect will route the document to the workflow selected as the request ID if the sender name, subname, message type, and message action defined on the input channel match the entries in the message map.

Import Service References

Service references are links to the web services or Windows Communication Framework® (WCF) services of applications outside Service Connect. After you import service references into the ESC Administration Console, you can call web service methods from inside a workflow.

To add a service reference:

1. Right-click **Service References,** and select **Add Service Reference**.
2. Click **Next** in the first wizard screen.



Epicor for Service Enterprises and Integration Hub Web Services come in two ways: standard and Schema Values (SV). The SV Web Services all begin with the letters "sv" in their filename; for example, svExpense.asmx. When you add a Web Reference to Service Connect, always use the SV Web Services because they contain the data type information Service Connect needs to process the Web Service schema.

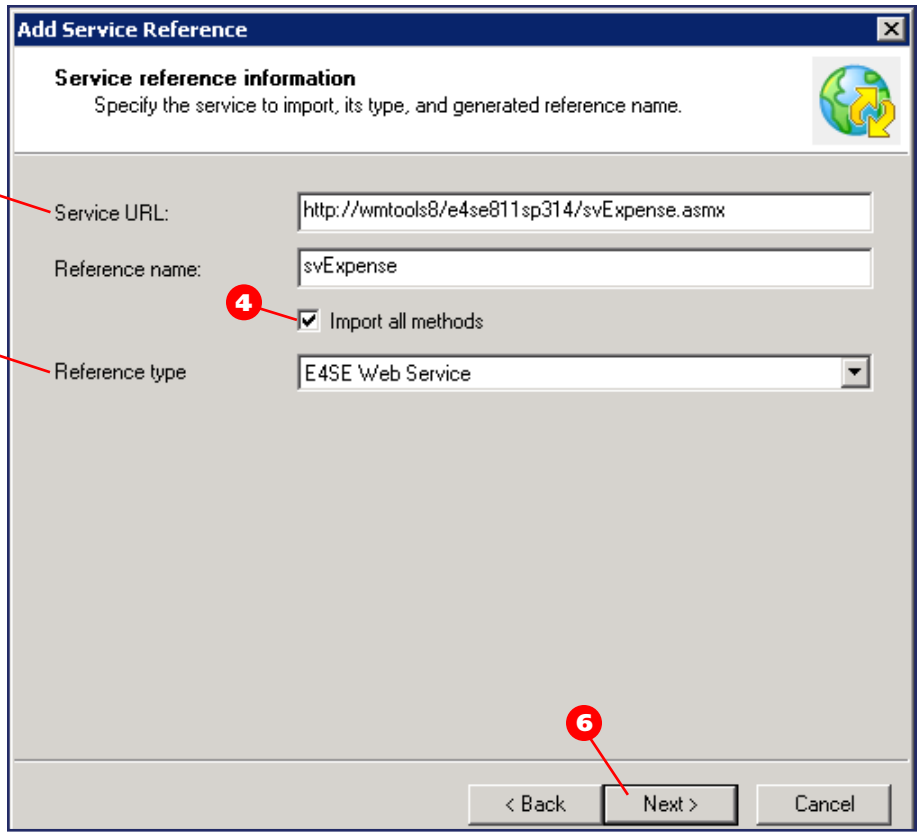
3. Enter the **Service URL**.

   The name will be used to identify the service reference in the ESC Administration Console and will prefix the web service methods in Web Method calls.

4. Leave the **Import all methods** check box selected.

5. Select **E4SE Web Service** as the **Service Reference Type**.

6. Click **Next**.



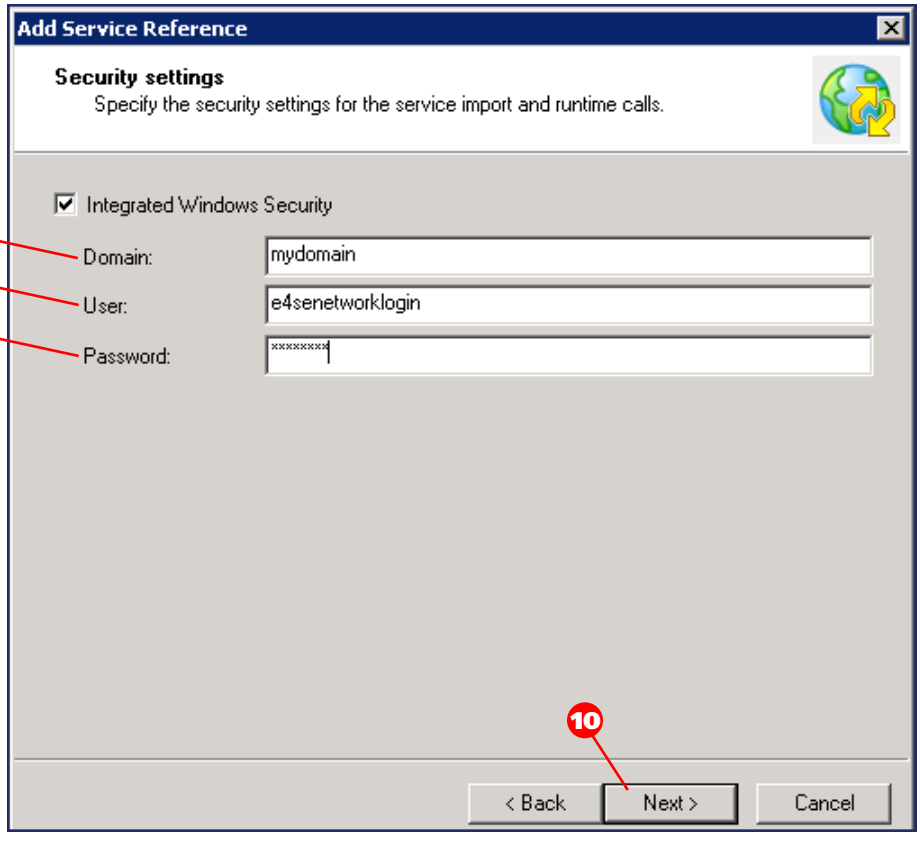
7. Enter the **Domain** used by a user account with trusted access to the location where the web services are installed.

   This user account will be used for secure access to the web services. For this example, you may want to use the Epicor for Service Enterprises domain user account, usually named e4senetworklogin.

8. Enter the **User**.

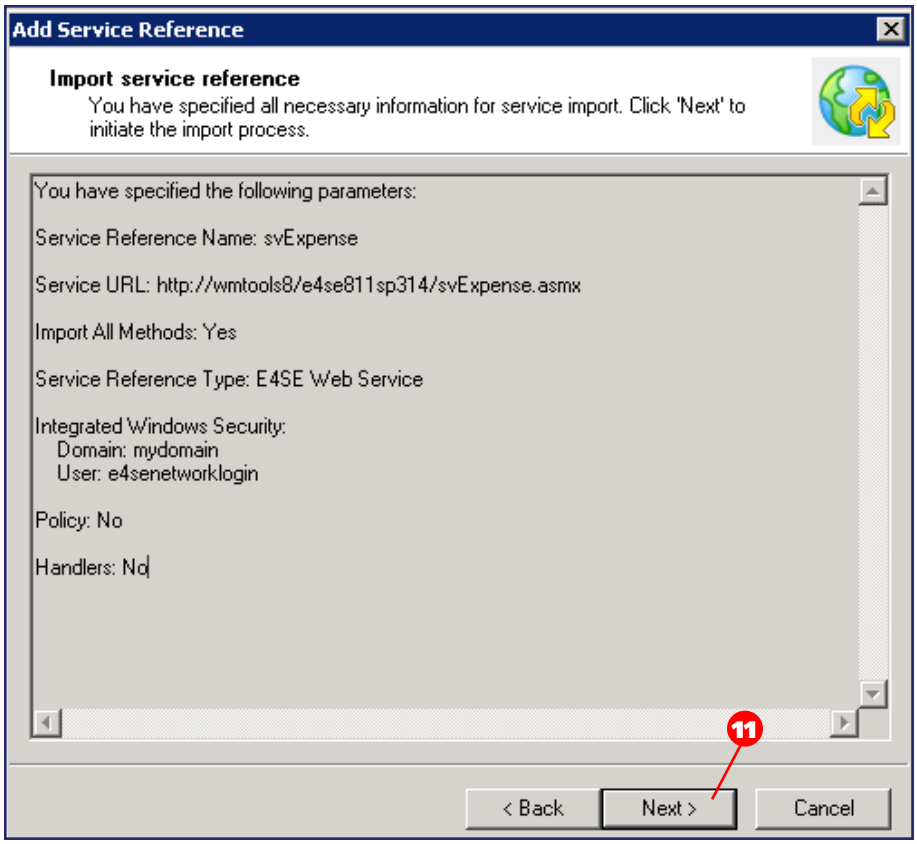
9. Enter the **Password**.

10. Click **Next**.



11. Review the Web Service Import information and click **Next.**

12. Click **Finish** after the import is complete.



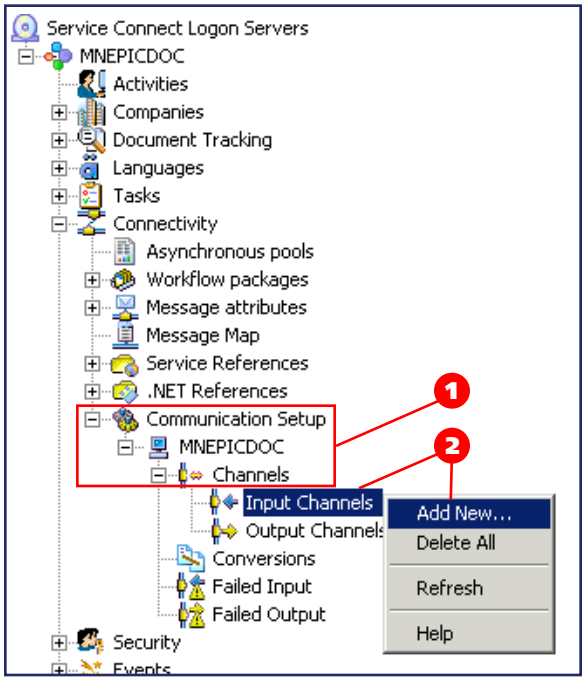
Add an Input Channel

An input channel serves as an entry point into the workflow. The following steps show you how to create an input channel that accepts a Microsoft Excel document as an email attachment. The input channel monitors the inbox of a POP3 mail account. For more information on the other types of input channels Service Connect can monitor, refer to Chapter 3: Connectivity Components.

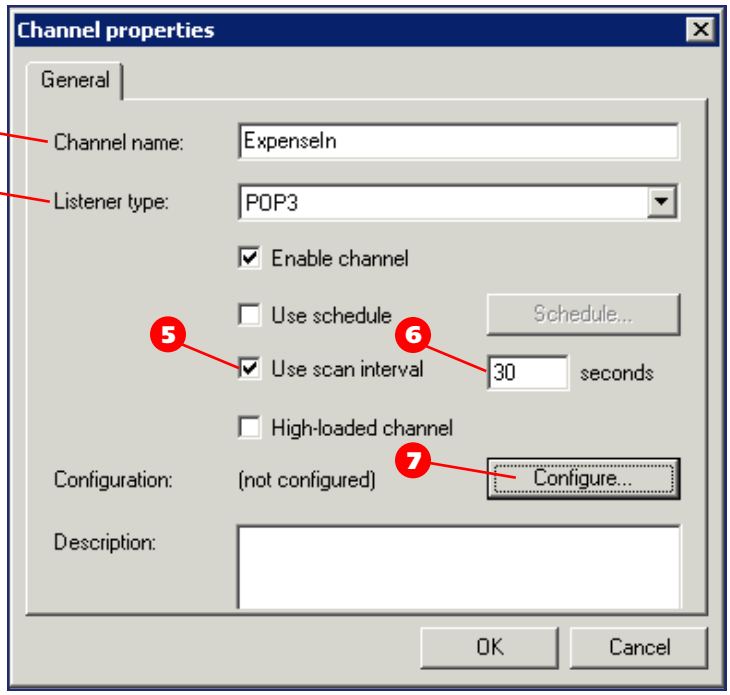
To create an input channel:

1. Expand the following nodes: **Communication Setup**, the server name, and **Channels**.

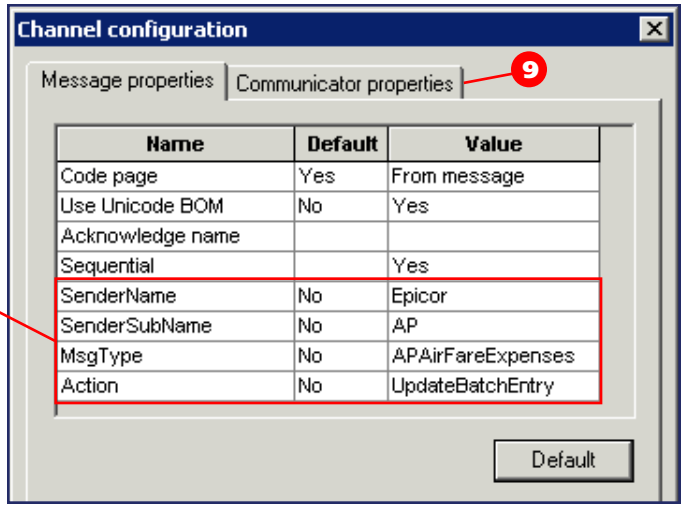
2. Right-click **Input Channels** and select **Add New**.



3. In the **Channel properties** window, enter a **Channel name**.
4. Select **POP3** as the **Listener type**.
5. Select the **Use scan interval** check box.
6. Increase the interval value set to **30** seconds. Some POP3 services block requests from a specific address if it connects to the mailbox too often.
7. Click **Configure**.

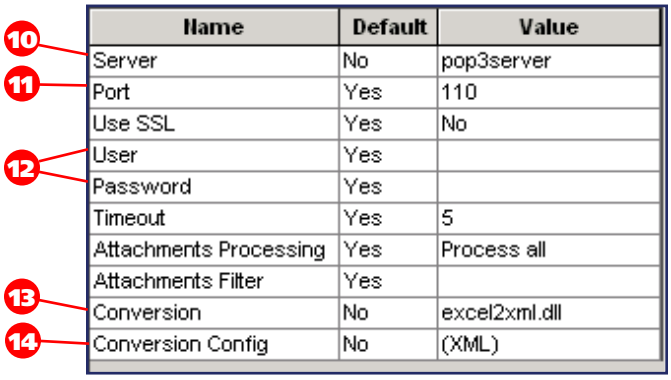


8. In the **Channel configuration** window, for the following fields, select the values used for the message map created earlier:
   - SenderName
   - SenderSubName
   - MsgType
   - Action
9. Click to the **Communicator properties** tab.



10. Enter the **Server** name of a POP mail server.

    If the server is within a trusted domain, you can enter just the server name. If the server is outside the trusted domain, enter the Fully Qualified Domain Name (FQDN). For example, if pop3server.mycompany.net is within a trusted domain, you can enter pop3server.
11. Enter the **Port** number.
12. If the POP mail server is outside the trusted domain, enter the **User** and **Password** for the user account that can access the server; otherwise, leave these fields blank.



13. Select the appropriate Conversion to handle the incoming document. Since the incoming document in this example is an Excel spreadsheet, use **excel2xml.dll** as the **Conversion**.

    The next two steps explain how to configure the input channel conversion to use a default namespace you define when the incoming document is transformed to the XML format Service Connect uses. These steps are optional but are recommended as a preferred practice.

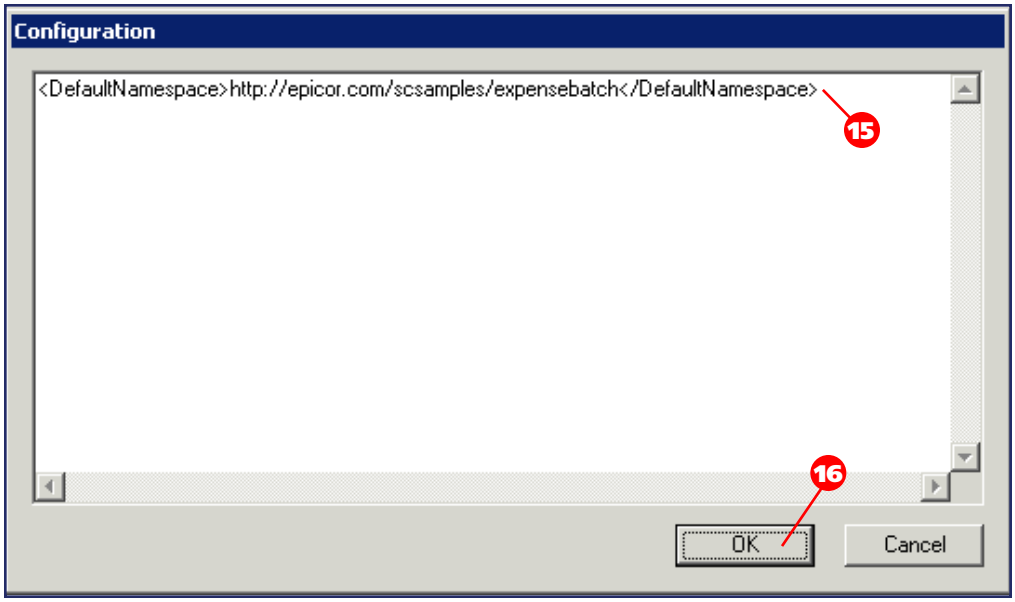
14. Click the down arrow button next to the **Conversion Config** field.

    You must click the field first to display the down arrow button.

15. Enter **<DefaultNamespace> http://epicor.com/ scsamples/expensebatch </DefaultNamespace>**.

    Typing this value into the XML dialog box configures Service Connect to use a namespace you declare for all documents that pass into this input channel. You will also add this namespace information to the schema that handles the incoming spreadsheet. Review the Generate a User Schema for the Inbound Document section for more information on how to use a namespace.

16. Click **OK** until you exit all dialog boxes.



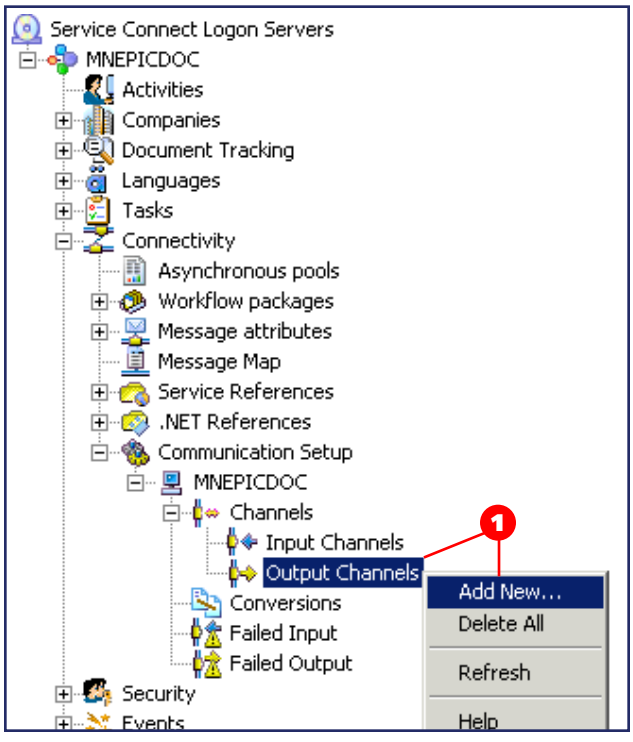
Add an Output Channel

An output channel serves as an exit for documents leaving the workflow. If the workflow performs its functions without sending documents to other applications from Service Connect, an output channel may not be required.

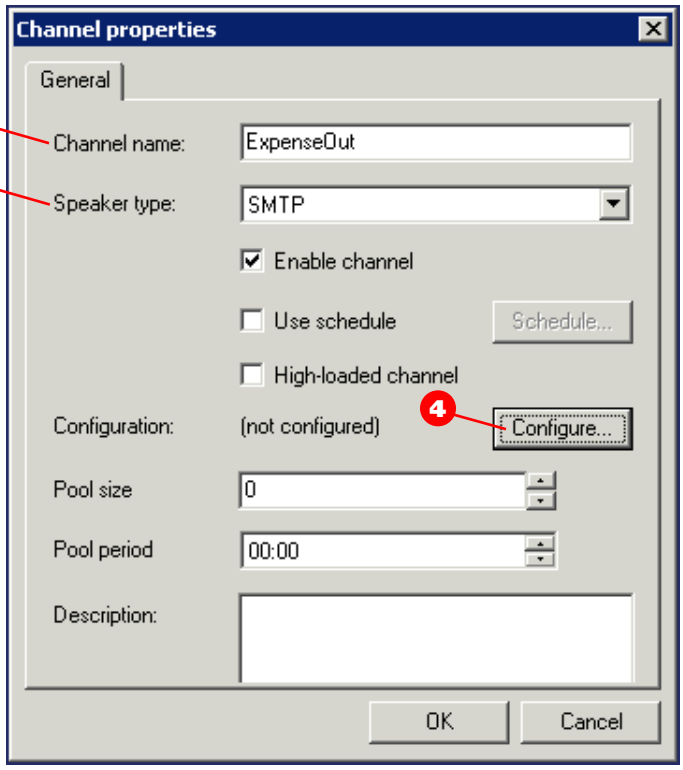
The following procedure shows you how to add an output channel you can use to send an email to someone to notify them about the workflow status. For information about the other types of output channels Service Connect can use to send information, refer to Chapter 3: Connectivity Components.

To add an output channel:

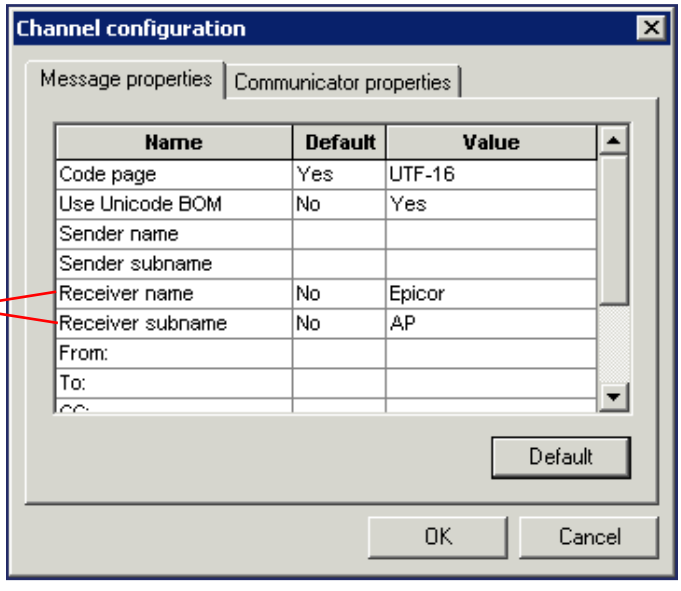
1. Right-click **Output Channels** and select **Add New**.



2. In the **Channel properties** window, enter a **Channel name**.
3. Select **SMTP** as the **Speaker type**.
4. Click **Configure**.



5. In the **Channel configuration** window, select the appropriate sender name and subname as the **Receiver name** and **Receiver subname**.



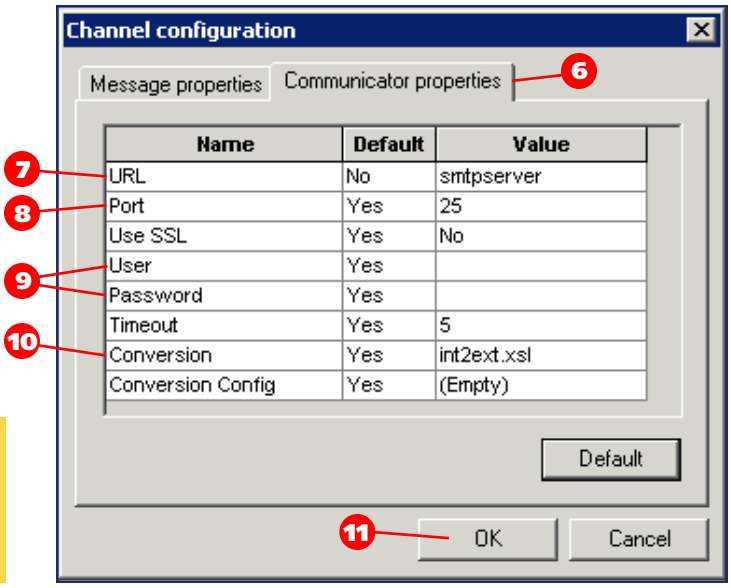
6. Click the **Communicator properties** tab.

7. Enter the server name you want to use to send the email response in the **URL** field.

   If the server is within a trusted domain, you can enter just the server name. If the server is outside the trusted domain, enter the Fully Qualified Domain Name (FQDN). For example, if smtpserver.mycompany.net is within a trusted domain, you can enter smtpserver.

8. Select the outgoing **Port** on the local machine, usually port 25.

   The default settings for some virus protection programs block port 25 to prevent mass mailing worms from sending mail. If the completed process does not send an email as designed, check your virus protection program's port blocking rules.



9. If the SMTP server is outside the trusted domain, enter the **User** and **Password** for an account on the server.

10. Select **int2ext.xsl** as the **Conversion**.

11. Click **OK** until you exit all dialog boxes.

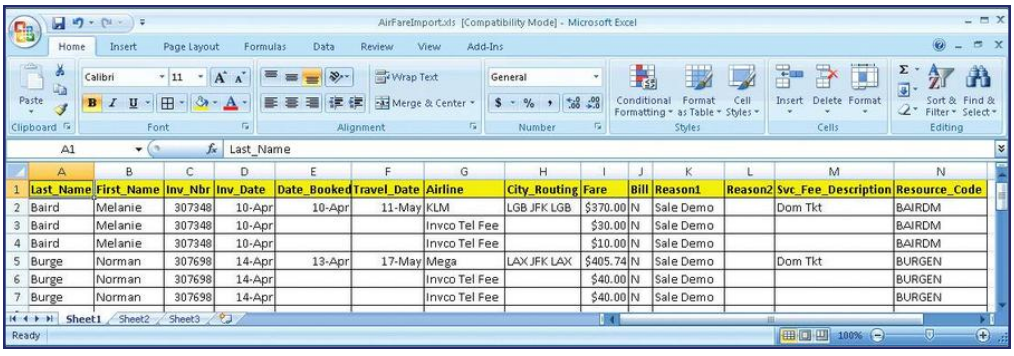
Generate a Schema for an Inbound Document

The data in the inbound document must conform to a schema Service Connect recognizes. To meet this need, you can generate a schema based on the incoming document. Any schema you generate is known as a user schema, as opposed to a Web-Service schema, which is generated when you add a Service Reference to the ESC Administration Console.

This example shows how to generate a schema based on an Excel spreadsheet.

When you generate a schema based on an Excel spreadsheet, Service Connect assumes the first row in the spreadsheet contains the column headers and that all subsequent rows contain the data. Also, if the spreadsheet contains multiple rows of data, ensure the spreadsheet you use to generate the schema also contains multiple rows of data; otherwise, Service Connect will assume the incoming document only contains one row and build the schema accordingly.

Here is an example of a spreadsheet that can pass into a Service Connect workflow.



Last_Name	First_Name	Inv_Nbr	Inv_Date	Date_Booked	Travel_Date	Airline	City_Routing	Fare	Bill	Reason1	Reason2	Svc_Fee_Description	Resource_Code
Baird	Melanie	307348	10-Apr	10-Apr	11-May	KLM	LGB JFK LGB	$370.00	N	Sale Demo		Dom Tkt	BAIRDM
Baird	Melanie	307348	10-Apr			Invco Tel Fee		$30.00	N	Sale Demo			BAIRDM
Baird	Melanie	307348	10-Apr			Invco Tel Fee		$10.00	N	Sale Demo			BAIRDM
Burge	Norman	307698	14-Apr	13-Apr	17-May	Mega	LAX JFK LAX	$405.74	N	Sale Demo		Dom Tkt	BURGEN
Burge	Norman	307698	14-Apr			Invco Tel Fee		$40.00	N	Sale Demo			BURGEN
Burge	Norman	307698	14-Apr			Invco Tel Fee		$40.00	N	Sale Demo			BURGEN

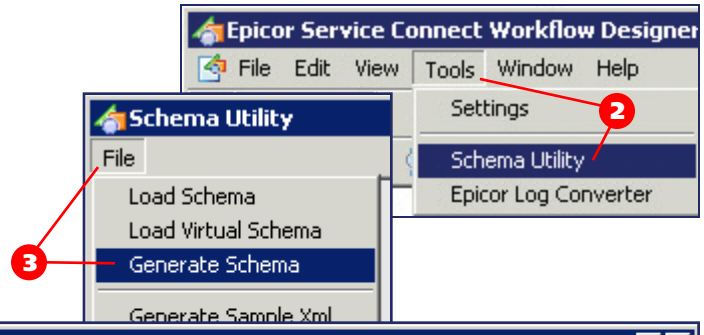
The spreadsheet columns include:

- **Last_Name** - The last name of a resource.
- **First_Name** - The first name of a resource
- **Inv_Nbr** - The invoice number associated with the expense
- **Inv_Date** - The invoice date

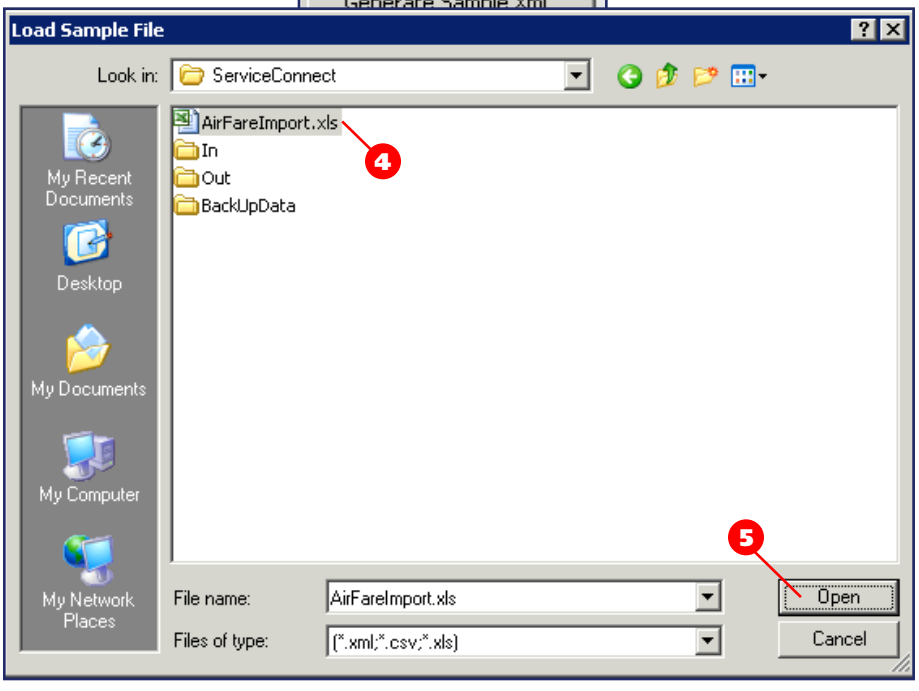
- **Date_Booked** - The date travel was booked
- **Travel_Date** - The date the resource traveled
- **Airline** - The name of the airline used
- **City_Routing** - The To, From, and layover information
- **Fare** - The fare paid
- **Bill** - Whether the expense is billable where N = No and B = Yes
- **Reason1** - For billable expenses, the project code; for non-billable expenses, the reason for travel
- **Reason2** - For billable expenses, the task name; for non-billable expenses, blank
- **Svc_Fee_Description** - The fare amount description
- **Resource_Code** - The resource's ID in Epicor for Service Enterprises

To generate a user schema:

1. Log into the Workflow Designer.
2. From the **Tools** menu, select **Schema Utility**.
3. The **Schema Utility** displays. From the **File** menu, select **Generate Schema**.

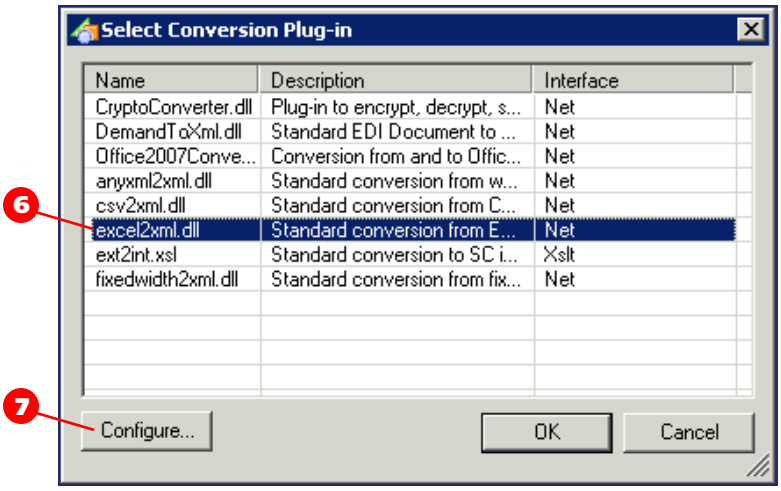


4. In the **Load Sample File** window, browse to and select the Excel file.
5. Click **Open**.



6. In the **Select Conversion Plug-in** window, select **excel2xml.dll**.

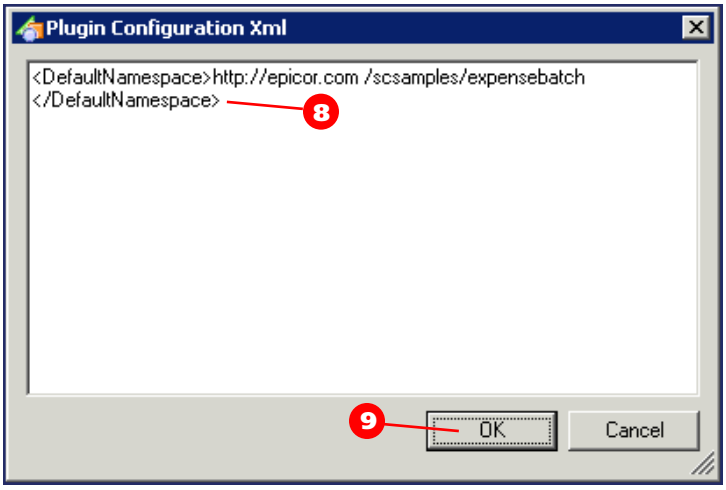
7. Click **Configure**.



8. In the **Plugin Configuration Xml** window, enter **<DefaultNamespace>http://epicor.com /scsamples/expensebatch </DefaultNamespace>**.

   Entering this value into the Plugin Configuration dialog box configures Service Connect to use a namespace you declare. A Service Connect generated namespace is based on the number of columns in the incoming data. Thus, if you add or remove any columns from the spreadsheet, when a new schema is generated, Service Connect will generate a different namespace for the schema. The namespace affects any Conversion elements used to process the data.

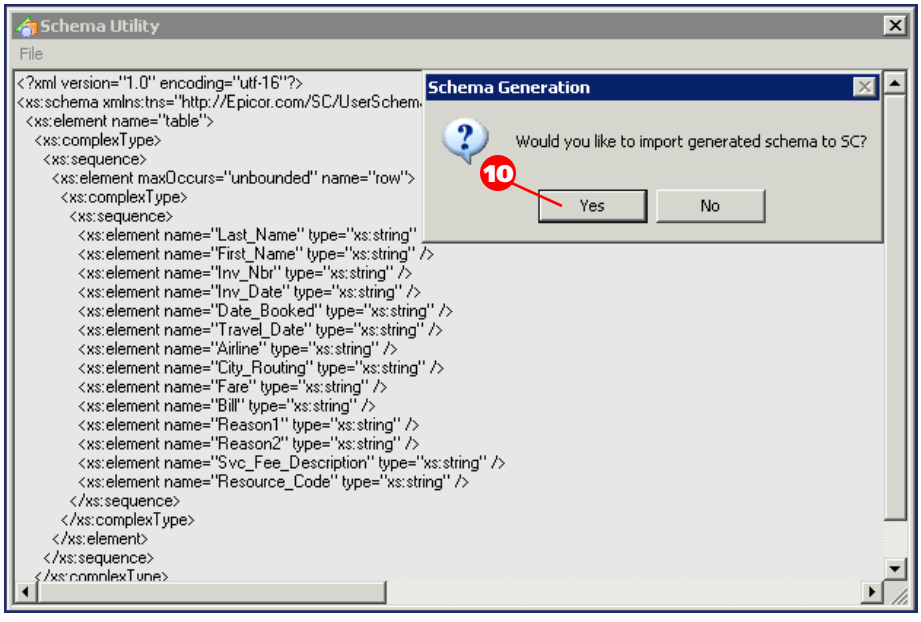
9. Click **OK**.



   Each column header in the spreadsheet becomes an element beneath the row element. Notice the maxOccurs attribute is set to unbounded for the row element. Thus, Service Connect expects the incoming spreadsheet to have one or more rows of data.

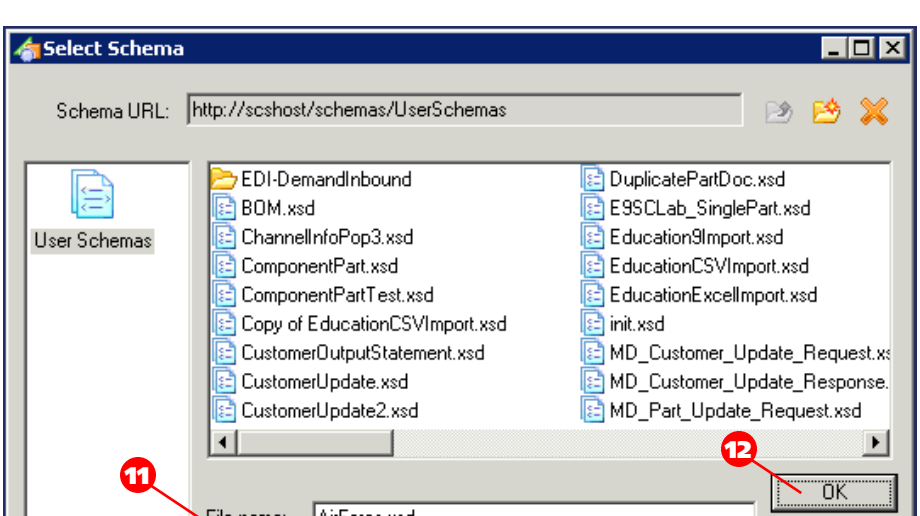
   You must save the schema where Service Connect will recognize it as a user-generated schema.

10. Click **Yes** to the **Schema generation** message to import the schema into Service Connect.



11. Enter a name for the schema file in the **File name** field.
12. Click **OK**.
13. Click **Save**.
14. From the **File** menu, select **Exit**.

The schema is now available for use within a workflow process as a Service Connect User Schema.



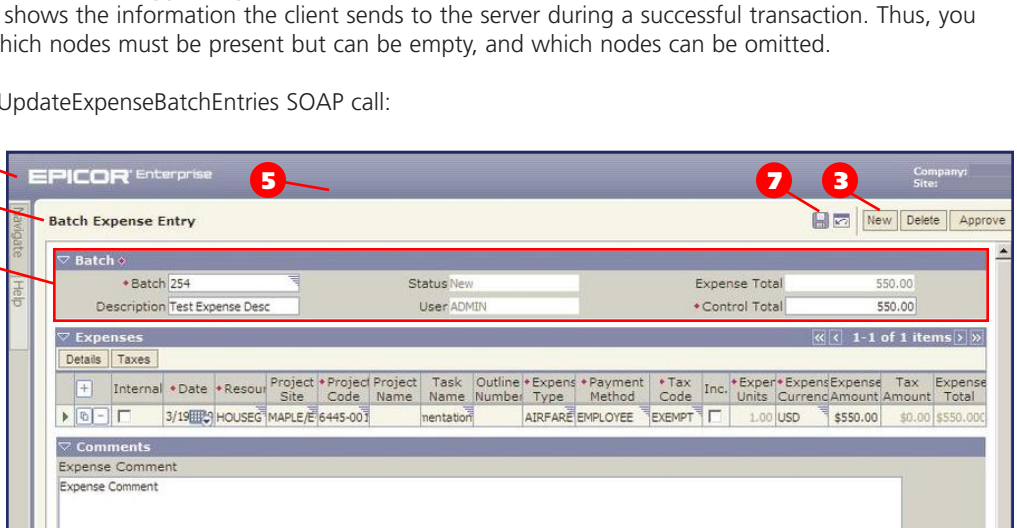
Special Technique: How to Use the Epicor for Service Enterprises Debugger

The Epicor for Service Enterprises Debugger captures and records all client-to-server activity that occurs between the DHTML Client and the Epicor for Service Enterprises web services. These are the SOAP messages that flow back and forth between the client and server. You can open the Debugger from the Epicor for Service Enterprises user interface.

The following example shows how to use the debugger to generate a sample XML file for a call to the UpdateExpenseBatchEntries web service method. The sample file shows the information the client sends to the server during a successful transaction. Thus, you can see which nodes are required, which nodes must be present but can be empty, and which nodes can be omitted.

To use the Debugger to capture the UpdateExpenseBatchEntries SOAP call:

1. Log into **Epicor for Service Enterprises**.
2. Open the **Batch Expense Entry** form using the menu path: **Project Suite > Delivery Management > Batch Expense Entry**.
3. Click **New** to create a new expense batch.
4. Enter a sample batch expense with one expense row but **do not** click Save. Verify all required information is entered.



5. Click in the blue area of the application header to the right of the Epicor Enterprise logo.
6. Press **Alt + D** to activate the Debugger.

   Debug mode = on displays in the browser status bar.
7. Click **Save**.

   The Debugger window opens and shows all the XML SOAP calls to and from the server.

8. Use the **Select Action** field to copy the information in the Debug window to the clipboard.

9. In an XML Editor, paste the results into a new, empty XML document. If you use an XML editor that does not indent XML, the XML pasted into the document displays as one line of text.

10. Look for the RESPONSE element where the soap:Body element contains the **GetAllExpenseBatch EntriesResponse** information.

11. Delete all the XML except the information between the opening and closing **GetAllExpenseBatch EntriesResponse** elements.

12. **Save** or **print** the document.

The result is that you have only the XML the server returns after an expense batch is successfully entered. You can use this XML document as a template to help set up the batch expense entry Conversion in Service Connect.

Service Connect and Epicor iScala

Epicor Service Connect was initially based on iScala Connectivity Solutions. Service Connect has significantly evolved during the time it has been developed independently.

In iScala 2.3, Epicor Service Connect replaced iScala Connectivity Solutions functionalities and provides a number of additional connectivity features.

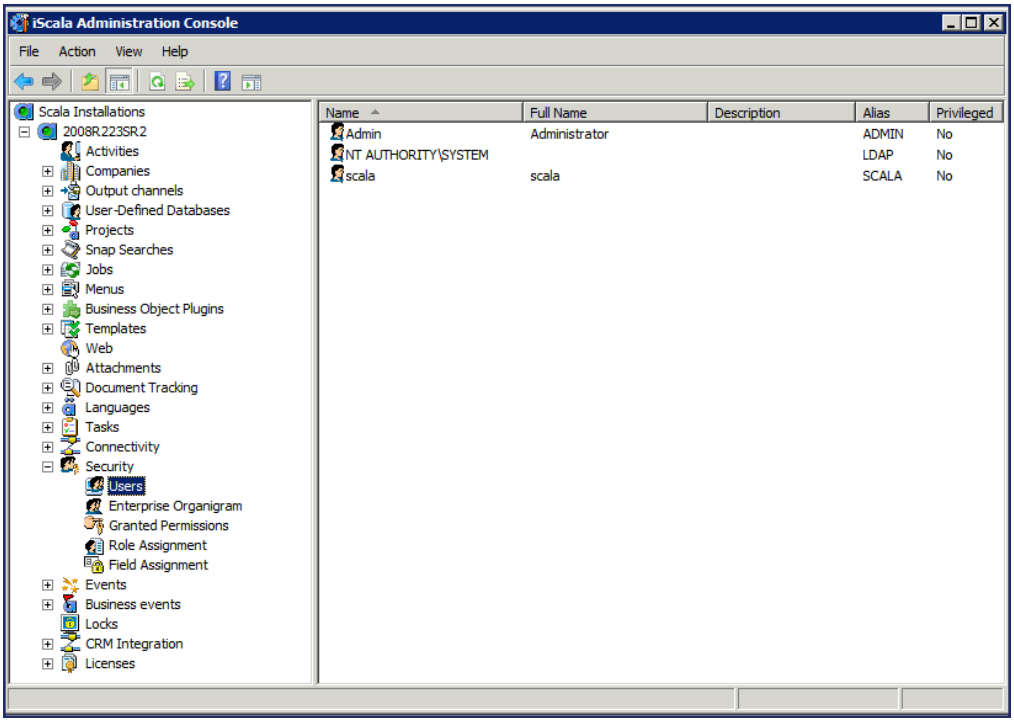
When Service Connect is installed as part of iScala, some data and integration parameters will require migration. Most of this process is done automatically during installation, but some issues require user attention.

Security

To set up and maintain security in your iScala system, the ESC Administration Console provides the following features:

- User Accounts
- Enterprise Organigram
- Permission Settings
- Role Assignment

In the ESC Administration Console, you can create and delete iScala users as well as manage user activities and user permissions for various
iScala features.



The above mentioned items are configured in the Activities and Security nodes, which display in the Administration Console tree in several places:

- **As root nodes –** To manage security items, users, and user activities in all the companies of the current installation.
- **Under each company node in the tree –** To manage company-specific security items, users, and activities.

The ability to view these settings depends on your administration rights.

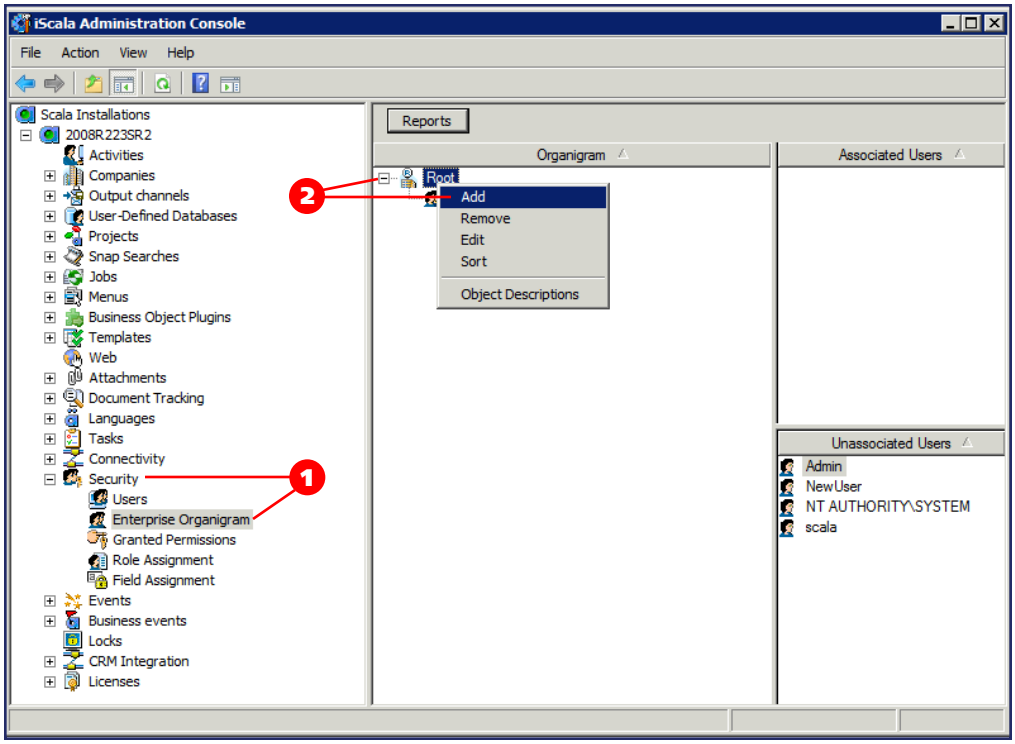
Grant Access Rights

To grant access rights in iScala:

1. In the iScala Administration Console, expand the **Security** node and select **Enterprise Organigram**. All users in the current installation of iScala are grouped hierarchically.

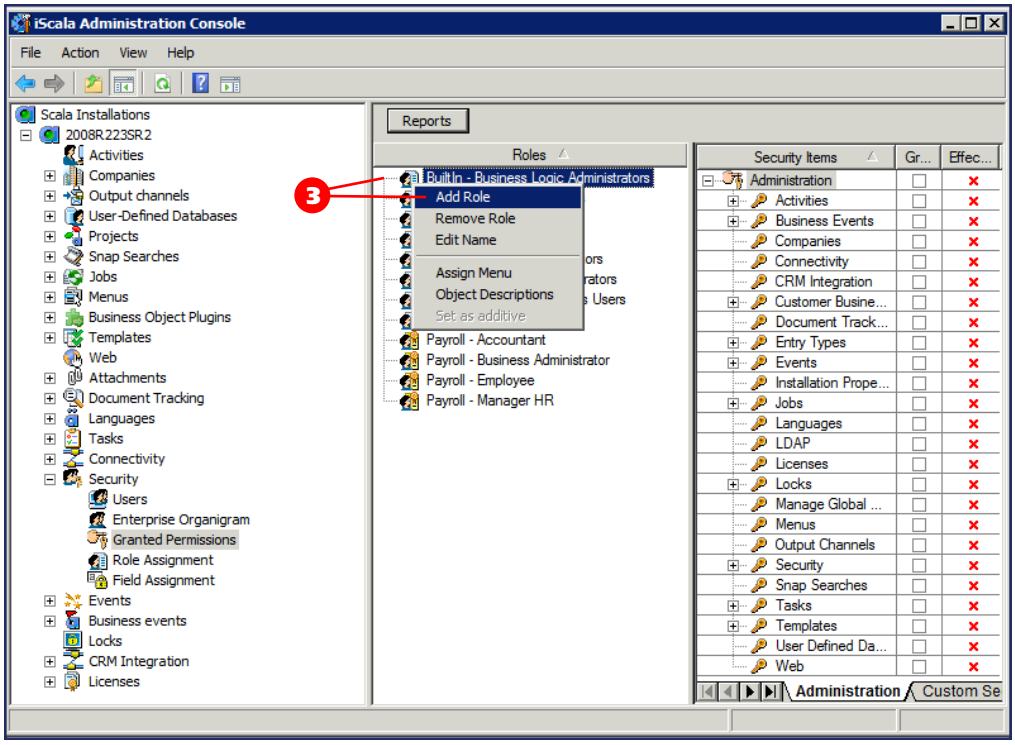
   An organigram is a diagram of the company's structure.

2. Right-click **Root** and select **Add** to create a group. Type the group name to replace the default NewGroup1 text.



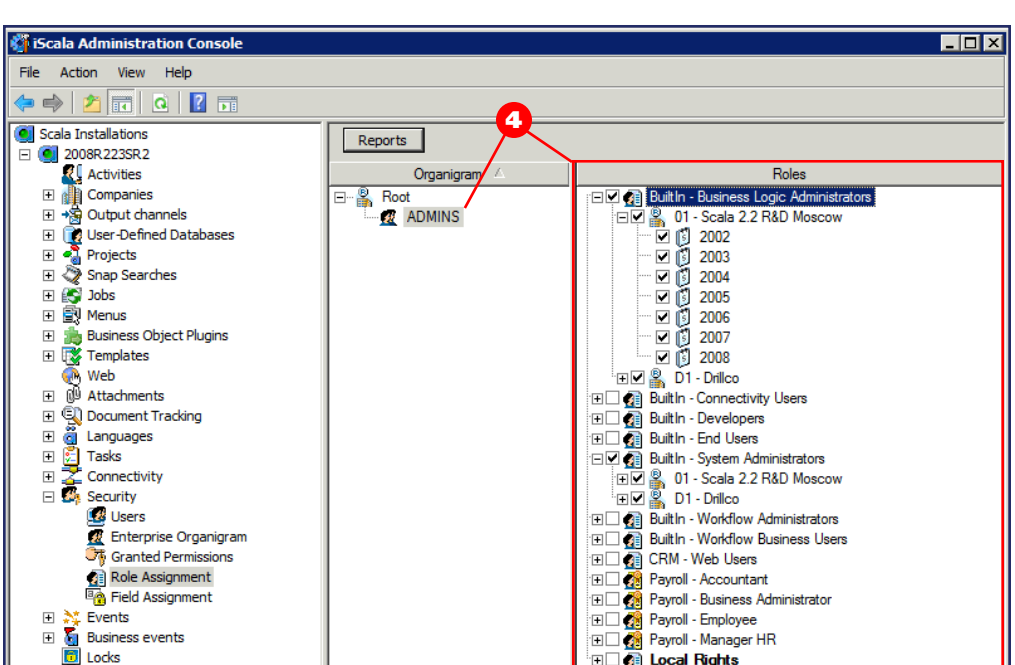
3. In **Granted Permissions**, right-click the **Roles** list and select **Add Role** to create a new role. To set permissions for the new role, click the various tabs at the bottom of the right pane and grant rights to specific security items.

   Alternately, you can find an already existing role that has the required permissions. Examples of built-in roles include System Administrators, Workflow Administrators, Workflow Business Users, and End Users.



4. In **Role Assignment**, click the user group in the **Organigram** tree and select the role(s) check boxes in the **Roles** pane to map the organigram user group to the role(s).

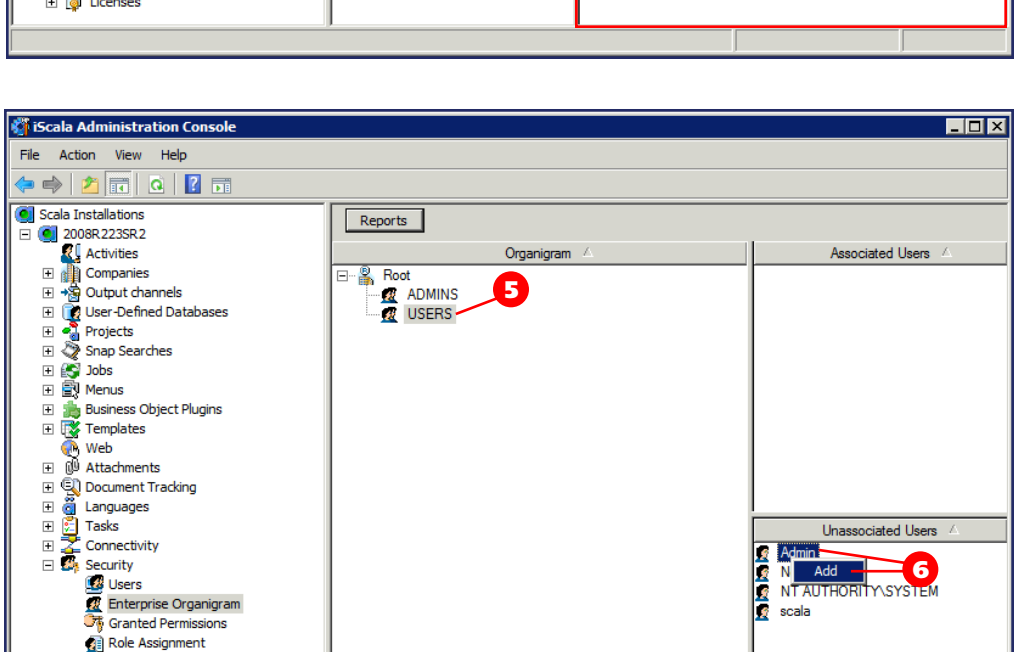
   To assign a role in all companies, select the check box near the role name, or in specific companies only, expand a role node and select certain companies. You can also assign roles for all financial years available for a company or for specific years only.



5. In the Enterprise Organigram center pane under **Organigram**, click the group to which you want to add the user.

6. In the **Unassociated Users** list, right-click the user you want to add to the group and select **Add**.

You can include one user in several organigram branches simultaneously.



User Parameters

In the Administration Console, you can perform the following actions with Service Connect user accounts:

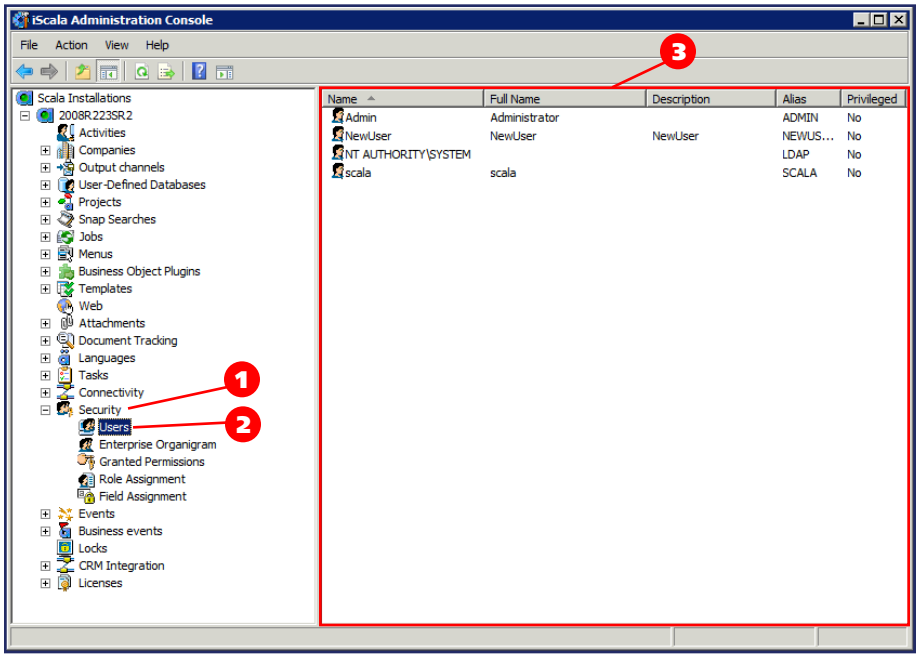
- View a list of all users.
- Add new users by:
  - Creating a new user
  - Importing a Windows user or a group of Windows users
  - Importing a list of users from a file; for example, to add users from another iScala installation
- Export a user list to a file.
- Check Windows user names.
- View and edit user properties.
- Delete a user.

View a List of All Users

Use the following steps to view all users:

1. In the Administration Console, expand the **Security** node.
2. Click the **Users** node. The list of all user accounts displays in the right pane.
3. Review the user information, including the **Name**, **Full Name**, **Description**, **Alias**, and **Privileged**.

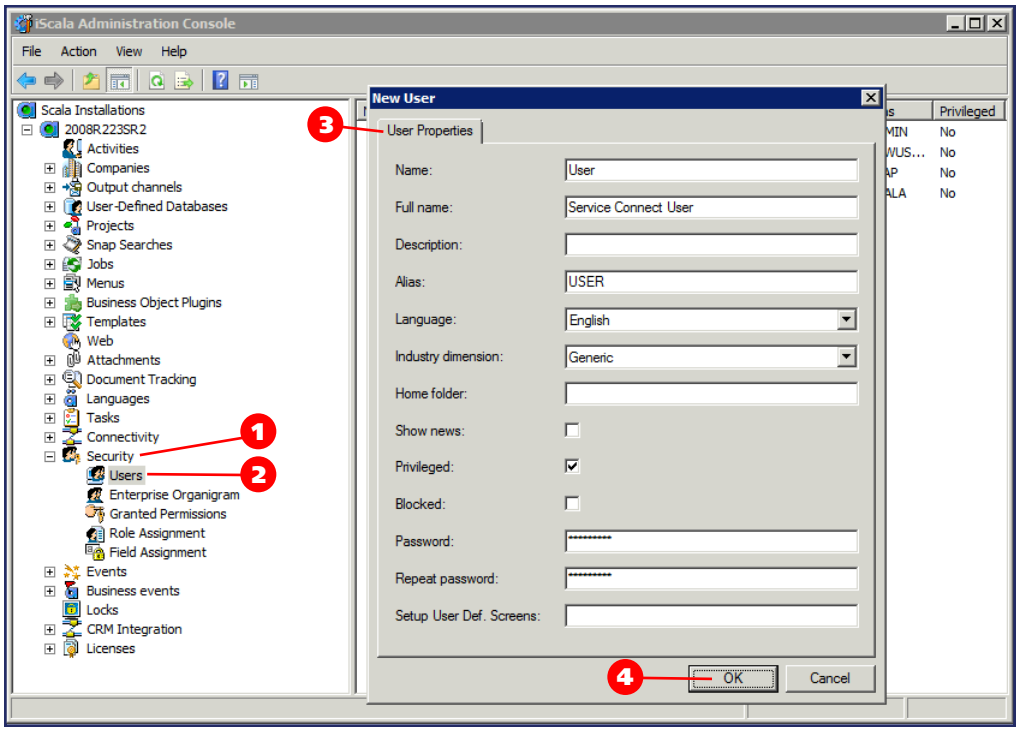
The Privileged property enables a user to access iScala anytime no matter how many users are currently working with the same iScala feature.



Create a New User

Use the following steps to create a new user:

1. In the Administration Console, expand the **Security** node.
2. Right-click the **Users** node and select **New User**.
3. Set the **User Properties**.
4. Click **OK**.



Import a Windows User

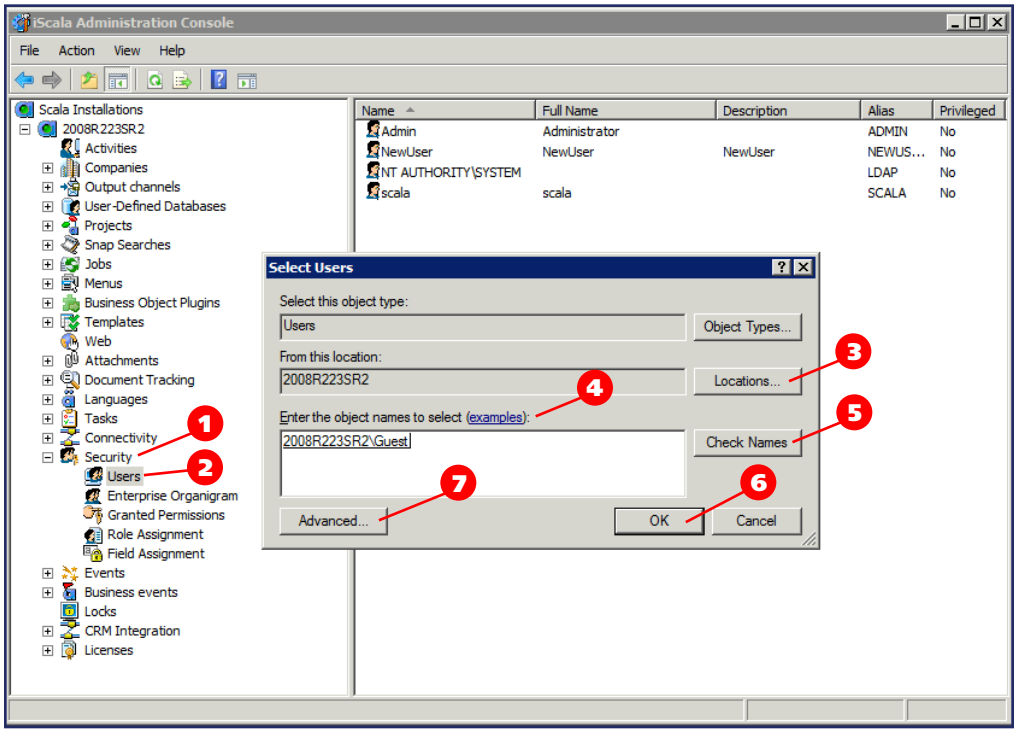
Use the following steps to import a windows user:

1. In the Administration Console, expand the **Security** node.
2. Right-click the **Users** node and select **Import Windows User**.
3. In the **Select Users** dialog box, click the **Locations** button to select a domain.
4. Type the user names in the **Enter the object names to select** field.

   You can type several names separated by semicolon.

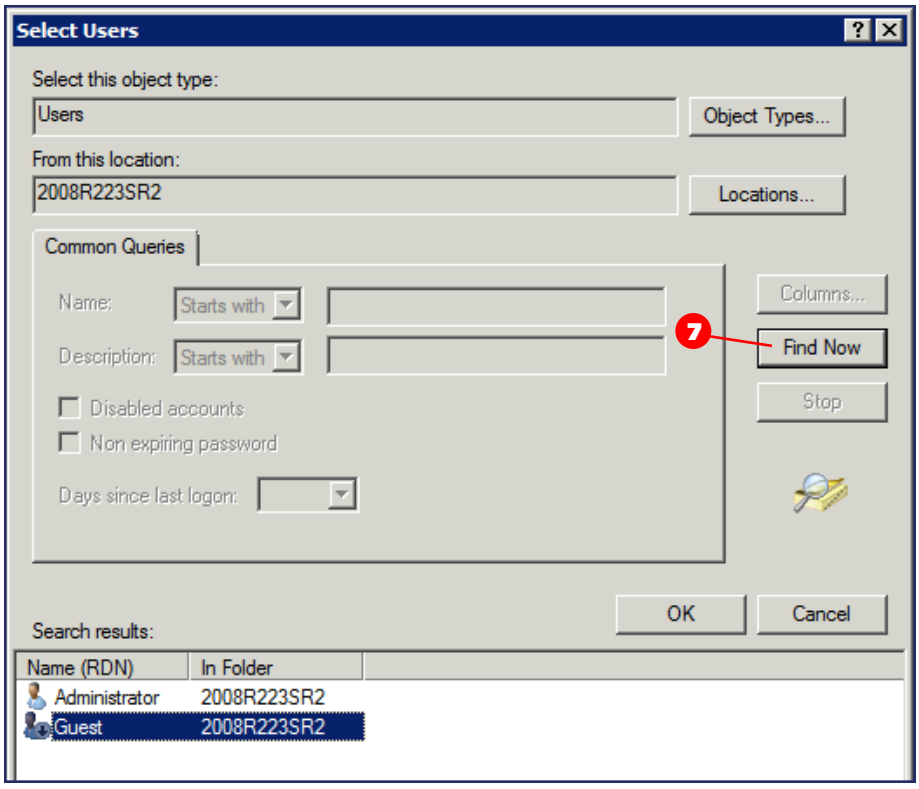
   > To view the syntax examples, click on examples.

5. Click the **Check Names** button.
6. Click **OK** to import the selected user or group.



7. If you prefer to view the list of all users within the domain you specified, click the **Advanced** button, and then click the **Find Now** button.

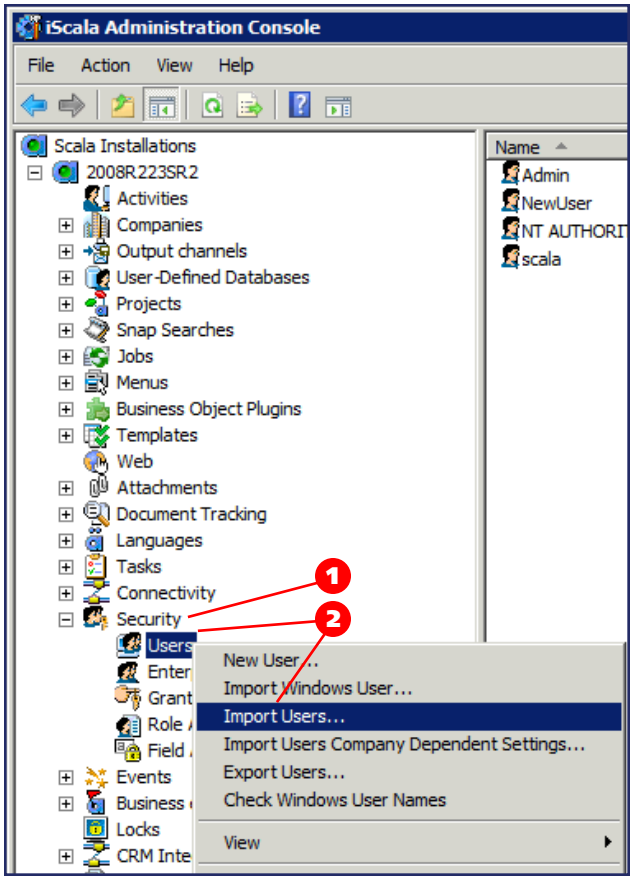
If you use Active Directory in your network, you must run iScala services under an account with the trusted for delegation check box selected in the user's properties in order to import Windows users. This property cannot be set to Domain Administrator since the corresponding property page is disabled, so the Windows users import will fail if iScala services work under the Domain Administrator account. To make the import successful, add a new user to the Domain Administrators group, give the user Domain Administrator rights, and select the trusted for delegation option. Also, the machine that runs iScala services must be trusted for delegation.



Import a List of Users from a File

Use the following steps to import a single user or multiple users from a text file:

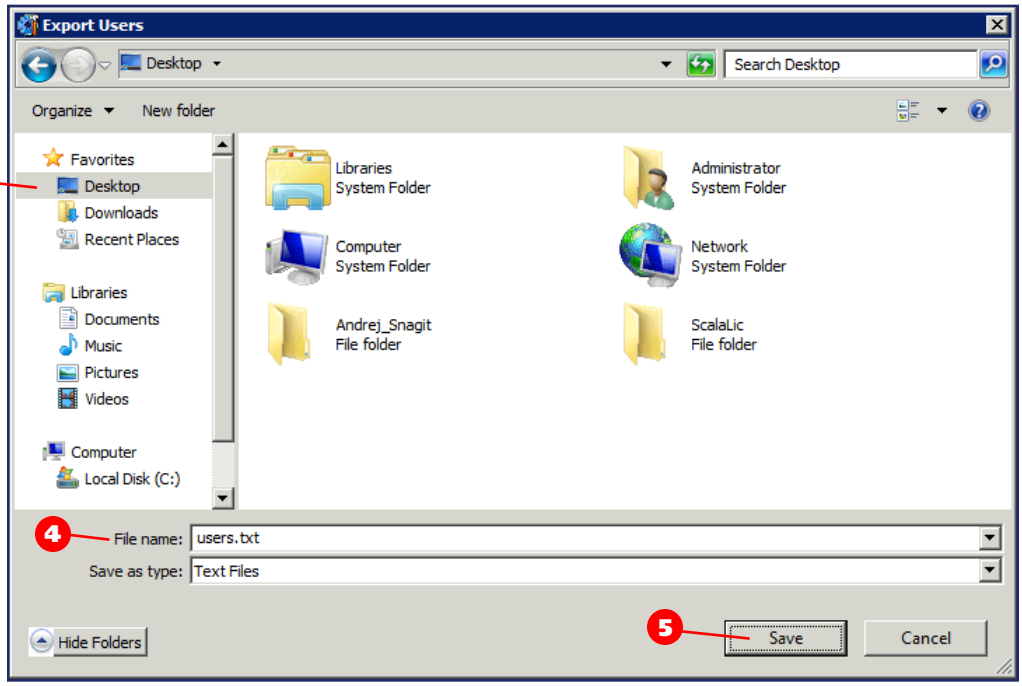
1. In the Administration Console, expand the **Security** node.
2. Right-click **Users** and select **Import Users**.
3. Locate the text file that contains the user list you want to import.
4. Click **Open**.



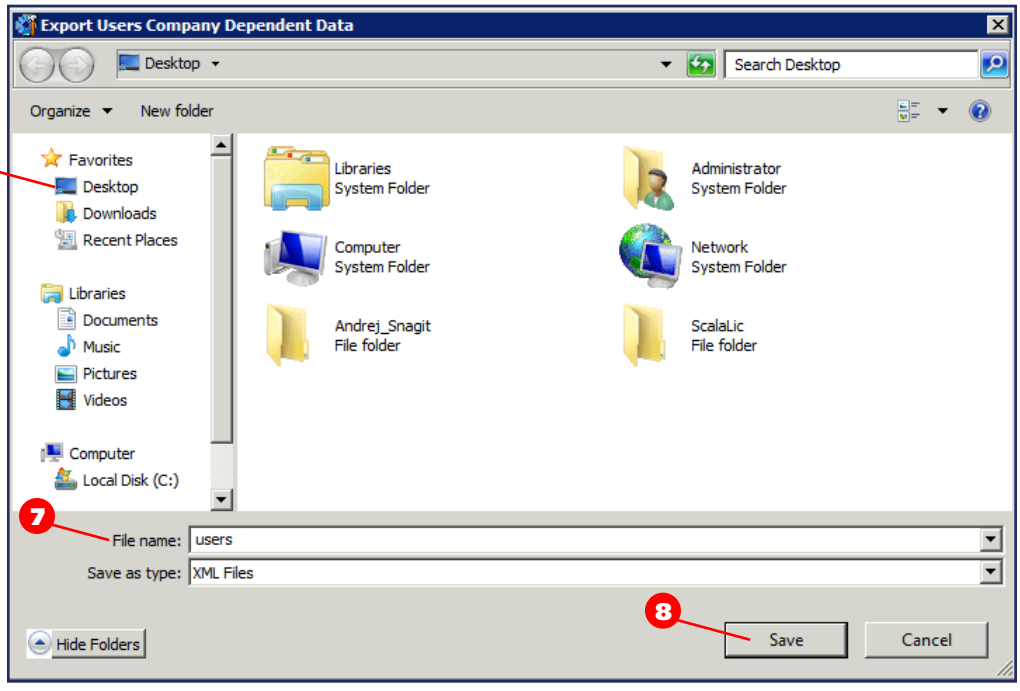
Export Users

Use the following steps to export a users list from Service Connect.

1. In the Administration Console, expand the **Security** node.
2. Right-click **Users** and select **Export Users**.
3. Select the appropriate location for the user text file.
4. Enter the **File name**.
5. Click **Save**.



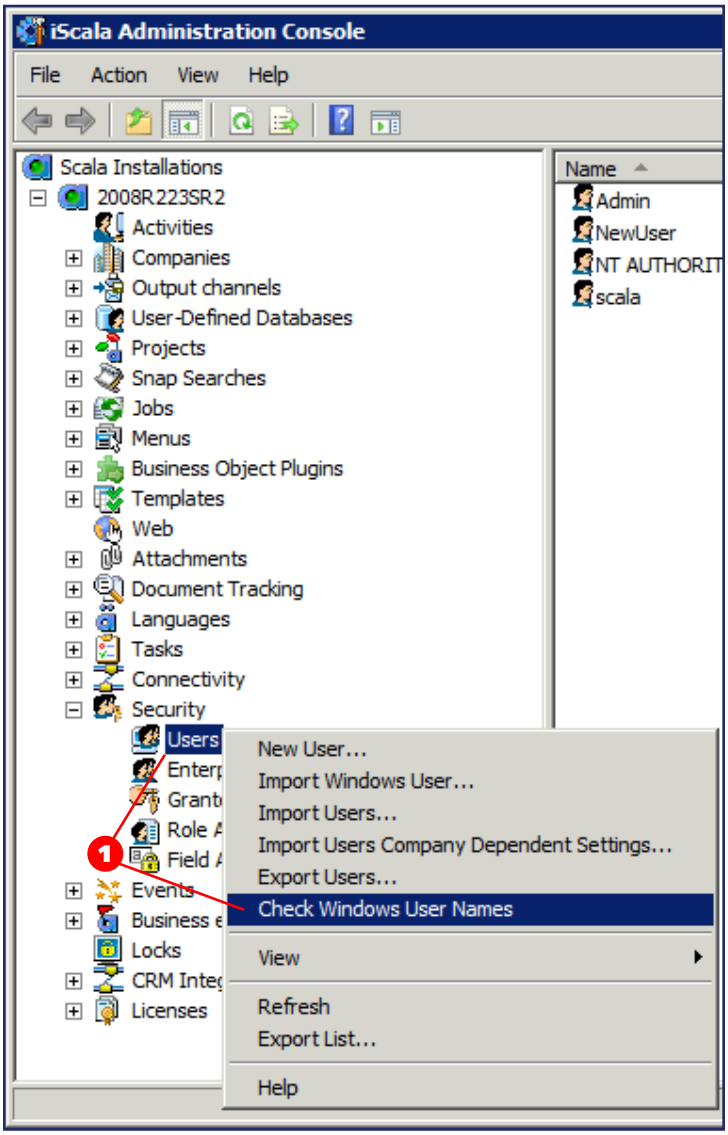
6. Select the appropriate location to export the users company dependent data file.
7. Enter the **File name**.
8. Click **Save**.



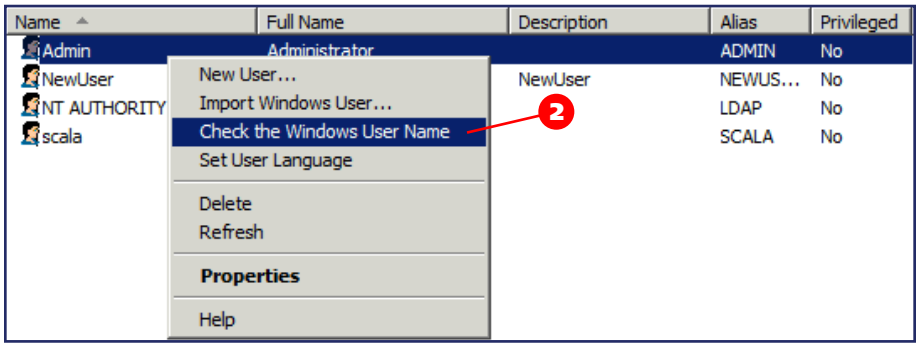
Check Windows User Names

Use the following steps to check windows user names.

1. To check all user names, right-click the **Users** node under the Security node in the Administration Console tree and select **Check Windows User Names**.



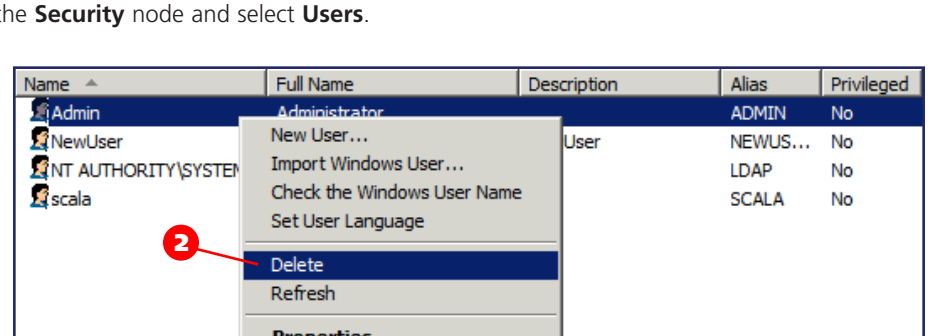
2. To check a particular user name, right-click that user in the list of users in the right pane and select **Check the Windows User Name**.



Delete a User

Use the following steps to delete a user.

1. In the Administration Console, expand the **Security** node and select **Users**.
2. In the right pane, right-click the user you want to delete and select **Delete** from the menu.



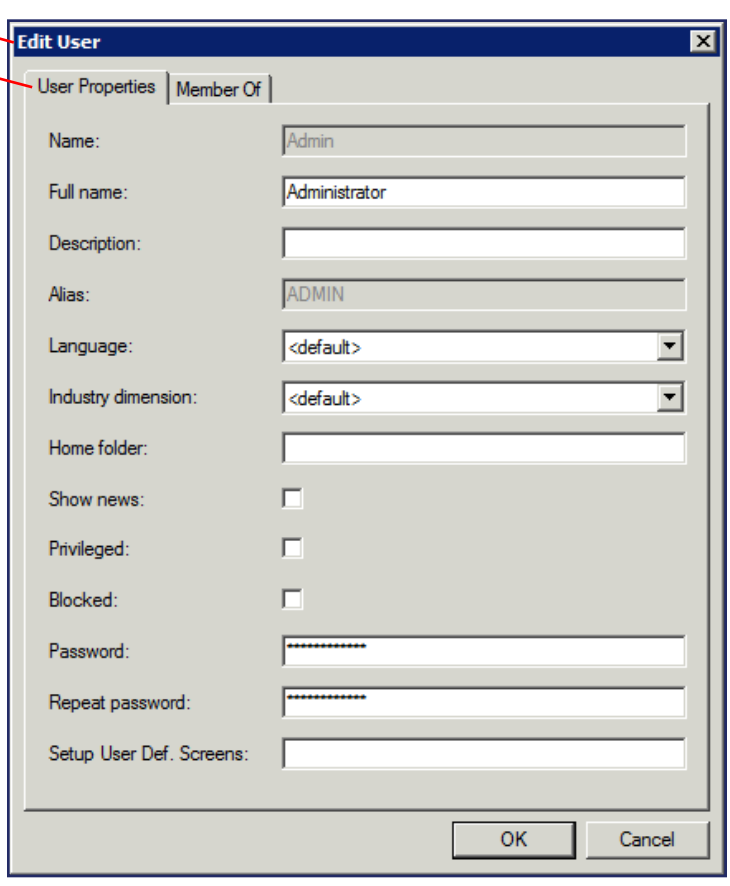
Use the User Properties Tab

To view or edit User Properties:

1. In the Administration Console, expand the **Security** node and select **Users**.
2. In the user list in the right pane, right-click the user whose properties you want to view and select **Properties**.
3. The **Edit User** window displays.

   These user properties are general for a user within all companies. You can also set company-specific user properties. Review the Company-Specific User Properties section for more details.
4. On the **User Properties** tab, you can view and specify the following user properties:

   - **Name** – This is the name of the user.
   - **Full name** – This is the full name of the user.
   - **Description** – This is a textual description of the user and can contain comments; up to two lines of 30 characters each.
   - **Alias** – This is the code used in some areas of iScala business logic. For example, the alias can be the user's initials. User Aliases are also used in audit reporting. You can only set the user Alias when you create a new user. The Alias is not editable as changes expose security risks when security and audit reporting is based on Aliases.
   - **Language** – This is the iScala user interface language.
   - **Industry dimension** – This defines the display of industry-specific terminology throughout iScala modules. The industry dimension assigned to the user is an indication of which functionality is being used and which role is being acted upon.

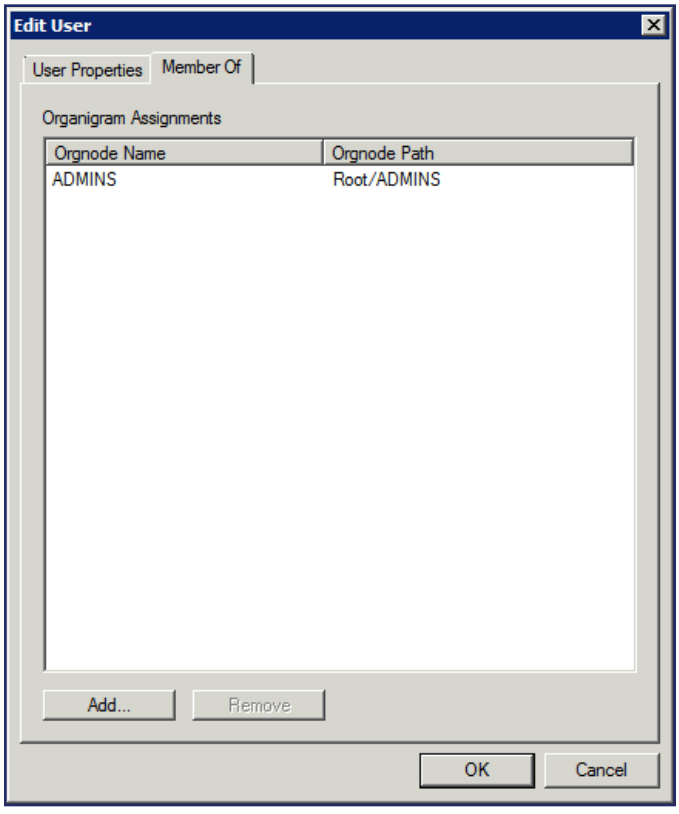


- **Home folder** – This is the folder for storing user-specific iScala files, such as local VBA projects. This is also the default out put folder for reports. If the Home Folder field is blank, the default home folder is used. The default home folder, Default User Profile, is set in installation properties. If the specified folder does not exist, it is created. If you cannot create the specified folder, the My Documents folder on the local computer is used. User environment settings, such as the iScala color scheme or the last open menu, are not saved in this folder but in the User Profile Cash folder.
- **Show news** - If selected, the news screen displays each time the user logs in. If cleared, the news screen never displays.
- **Privileged** - Select this check box to enable the user to access iScala anytime, no matter how many users are currently working with the same iScala feature.
- **Blocked** - To block a user from logging into iScala, select this check box. A blocked user exists in the list of users but cannot log into the application.
- **Password** – This is the user password for logging into iScala WinDS.
- **Setup User Def. Screens** - If you use Screen Definition Files (.SDF) to introduce modifications into standard iScala forms, specify the letter that identifies the custom .SDF files to be linked for this particular user; for example, 1 for sl0010x.sdf, A for sl0010a.sdf, and so on. You can also assign a custom .SDF file to a particular iScala routine for all users.

If you change any user properties, click OK or Apply to save the changes.

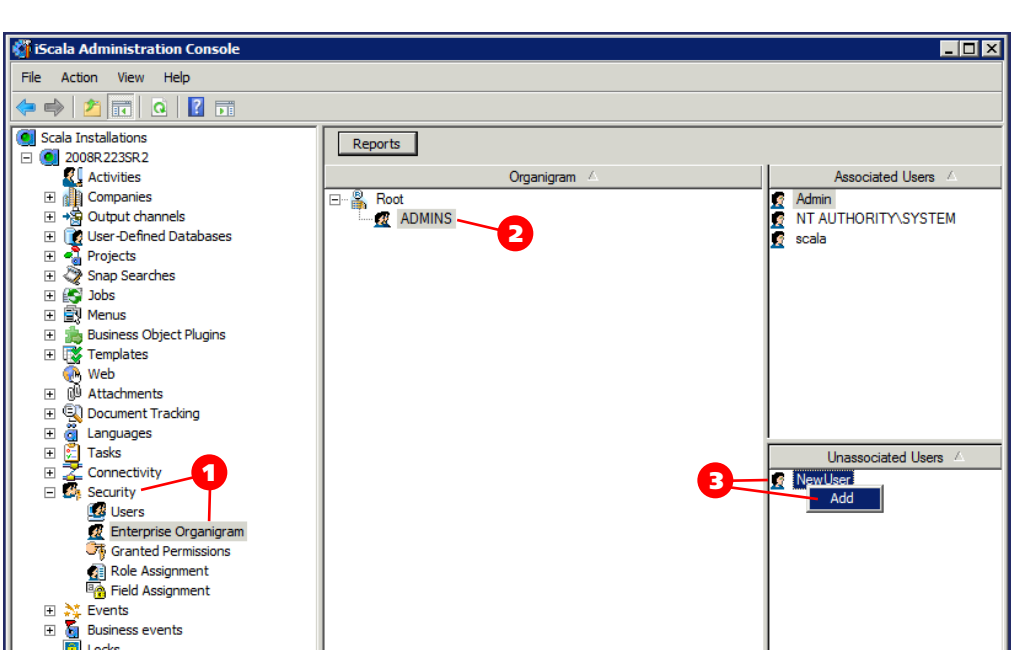
Use the Member Of Tab

On the Member Of tab in the user properties window, you can view the list of user groups to which the user belongs.



Assign a User to a User Group

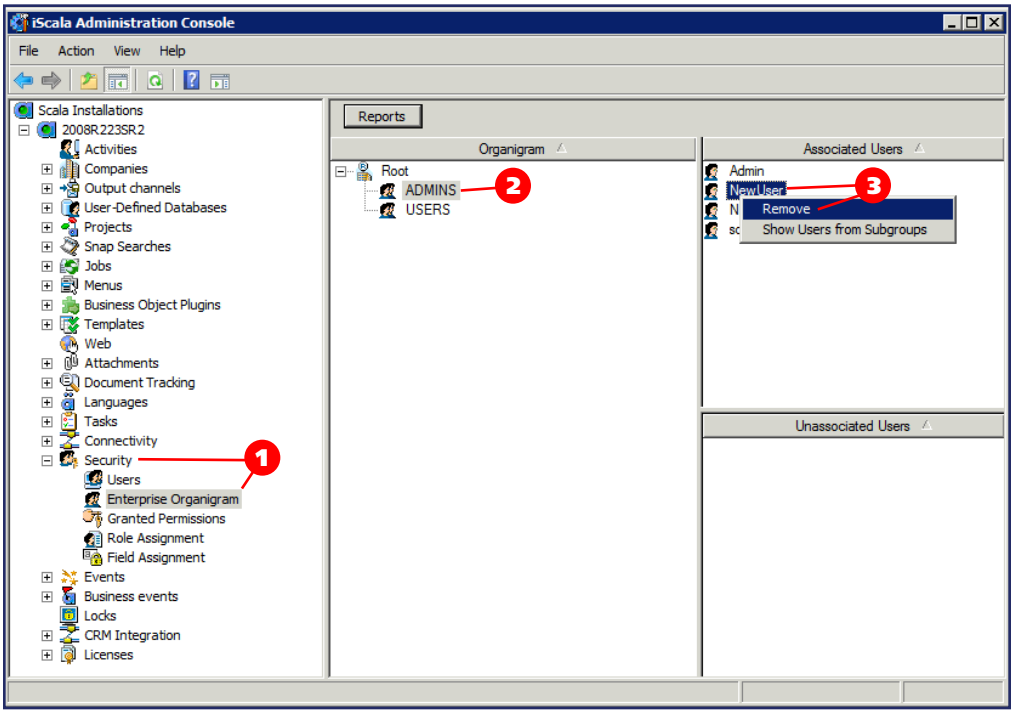
1. In the iScala Administration Console, expand the **Security** node and select **Enterprise Organigram**.
2. In the middle pane under **Organigram**, click the group to which you want to add the user.
3. In the **Unassociated Users** list, right-click the user you want to add to the group and select **Add**.



Remove a User from a User Group

1. In the iScala Administration Console, expand the **Security** node and select **Enterprise Organigram**.
2. In the middle pane under **Organigram**, click the group from which you want to remove the user.
3. In the **Associated Users** list, right-click the user you want to remove from the group and select **Remove**.

You can adjust several other iScala-specific user properties in WinDS, using **System Utilities > User Management > User Properties**.

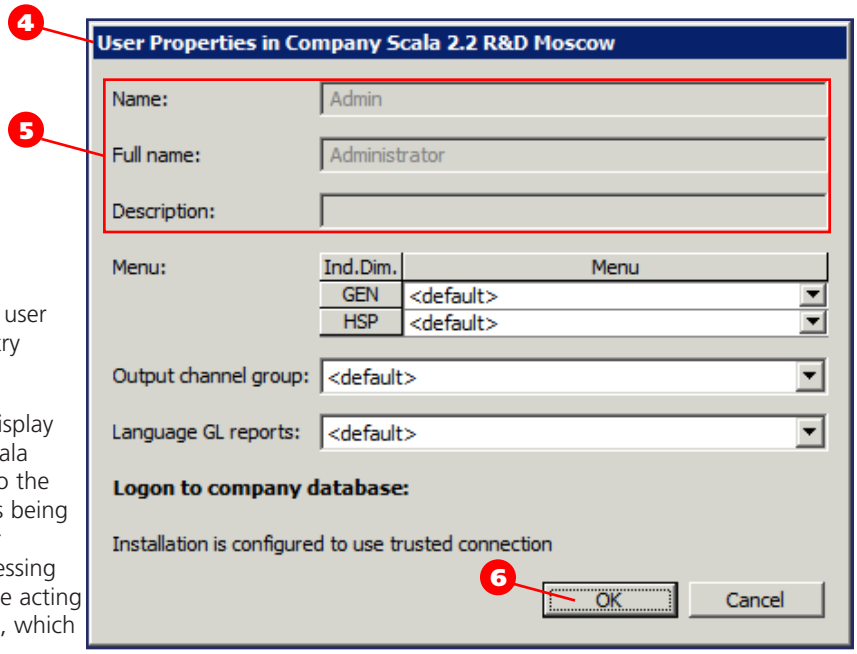


Edit Company-Specific User Properties

Company-specific user properties override the general properties set in Security > Users.

To view or edit company-specific User Properties:

1. In the iScala Administration Console, expand the **Companies** node and expand the node for a specific company
2. Expand the **Security** node and select **Users**.
3. In the list of users in the right pane, right-click the user whose company-specific properties you want to view or edit and select **Properties**.
4. The **User Properties in Company <Company Name>** window displays.
5. You cannot change the user's **Name**, **Full name**, or **Description**. These properties are read-only.



You can edit the following company-specific user properties:

- **Menu** – The iScala menu displays when the user logs into WinDS and is set for various industry dimensions.

  Use the industry dimension mechanism to display industry-specific terminology throughout iScala modules. The industry dimension assigned to the user is an indication of which functionality is being used and what role is being acted upon. For example, in one company, various users accessing the same Bill of Material functionality may be acting as generic iScala users or as hospitality users, which affects the user interface and reporting.

- **Output channel group** – An output channel group is assigned to a user. Only the output channels within the specified output channel group are available to the user.

  Output Channel Groups are created and managed in Output Channels > Groups. If no Output Channel Groups are assigned to the user, all output channels are available.

  If the user to which the output channel group is assigned has output channel administrator rights, the user automatically bypasses the output channel group restrictions.

  You can assign a menu to a user; within this menu, you can assign an output channel to a menu item. If the output channel assigned to a menu does not belong to the output channel group assigned to the user, the output channel from the assigned group is used.

- **Language GL reports** – This is the language for the General Ledger reports.
- **Logon to company database** – You can either inherit company settings to connect to the iScala Company database where all the business data is located, or, you can set up new user logon properties.

  When you set up new user logon properties, you must choose between two possible authentication procedures: SQL Server authentication or Trusted Connection.

  When SQL Server Authentication is used, you can enter a SQL username and password iScala will use when it makes SQL calls for the current user. This SQL username and password should not be known to the end-user; only to the system administrator. If the end-user knows the credentials iScala uses to make SQL calls, then the user can misuse those credentials to access the SQL database directly.

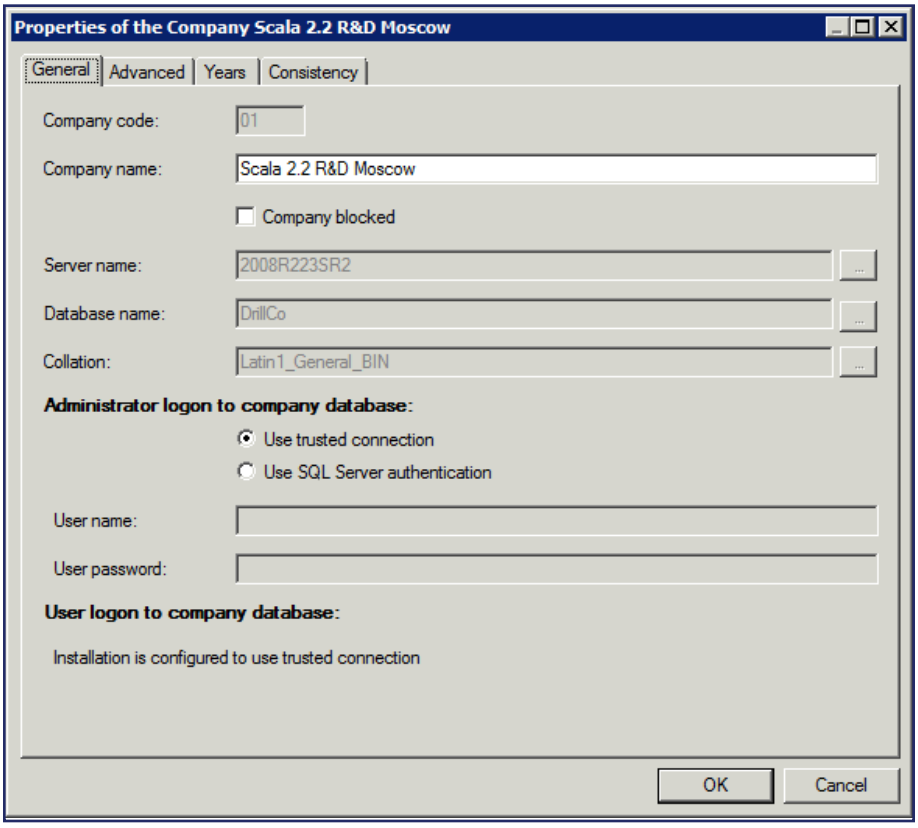
In the case of selecting a trusted connection to the SQL Server, Windows Authentication is used. The SQL Server will revalidate the account name and password by calling back to Windows for information. It is not recommended that you use trusted connections because it requires users to have SQL access rights connected to their Windows account. Users may abuse these rights by using other tools than iScala, such as Access or Excel, to read or change data in the iScala database.

6. Click **OK** to save the changes, if any.

Company Parameters

Company parameters are accessed from the Company node. You can perform the following actions using the Properties of the Company dialog box:

- View general company properties.
- Block a company from logging in.
- Set the administrator logon and user logon to the company database.
- Set the company default menu.
- Specify company login parameters.
- Add and delete company financial years.
- Set a financial period for an existing year.
- View lists of installed modules and system objects for the selected financial year.



View General Company Properties

1. Expand the **Companies** node.
2. Right-click the company you want from the list and select **Properties**.
3. Click the **General** tab. The following company information is available:
   - **Company Code** - A code that identifies the company.
   - **Company Name** - The name that identifies the company.
   - **Server Name** and **Database Name** - Where the company is located.
   - **Collation** - Company database sorting rules which are set when the company is created.

Block a Company

1. Click the **General** tab of the Properties of the Company dialog box.
2. Select the **Company blocked** check box.

Set an Administrator Logon to a Company Database

1. Click the **General** tab of the Properties of the Company dialog box.
2. In the **Administrator logon to company database** section, specify the account the iScala Administration Console uses to access the company database. This account can be a trusted connection or SQL Server Authentication.

If you select **SQL Server authentication**, specify the **User Name** and **User Password**. The iScala Administration Console will use this account to make SQL calls to the company database.

If you select **trusted connection**, the iScala Administration Console will use the Windows account of the person logged on to make SQL calls to the company database. In this case, the iScala Administration Console user must have SQL privileges.

View the User Logon to a Company Database

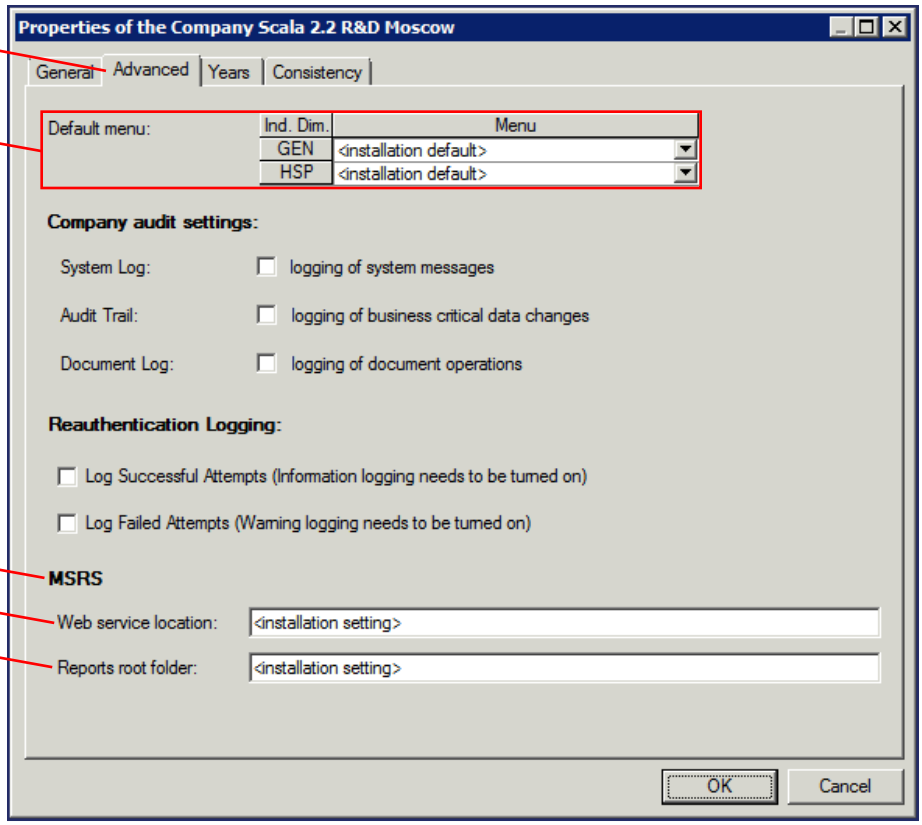
1. Click the **General** tab of the Properties of the Company dialog box.
2. In the **User logon to company database** section, specify whether users connect to the iScala Company database via a trusted connection or using SQL Server Authentication.

You can select an authentication procedure for a particular user when you create the user.

Set the Company Default Menu and Override MSRS Properties

1. Click the **Advanced** tab on the Properties of the Company dialog box.
2. Under the **Default menu** section, view or edit the default menus for the company for various industry dimensions.
3. You can override iScala **MSRS** installation-level properties on the Company basis by defining Microsoft Reporting Services Service settings in the MSRS group.

   If you have several Companies and need to create MSRS reports of various designs for each of them, you can specify different Reports root folder, where the report templates are stored. You can also utilize this company-level override to use different MSRS services for different companies.
4. Specify the path to Microsoft Reporting Services Service in the **Web service location** field.

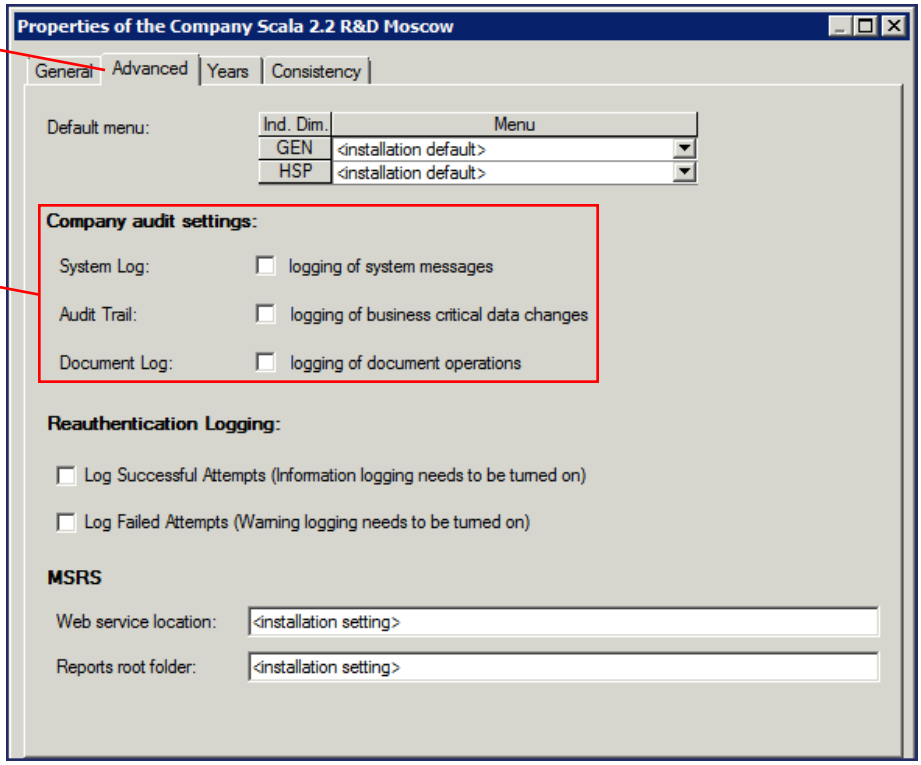


5. Specify the path to the folder, where the MSRS Report templates are stored, in the **Reports root folder** field.

If you leave these two fields blank, the installation-wide setting will be used.

Define Company Audit Settings

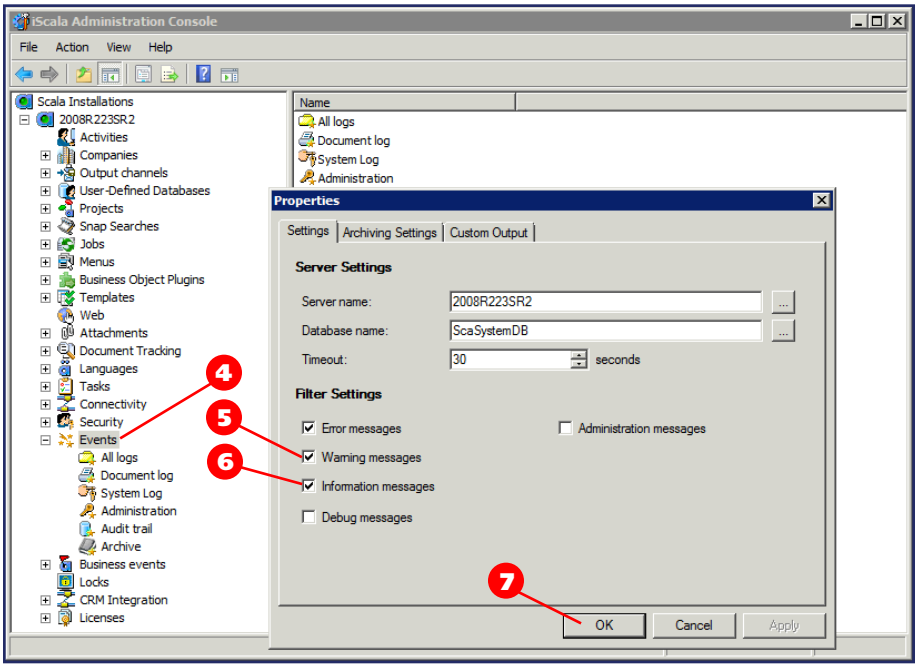
1. Click the **Advanced** tab on the Properties of the Company dialog box.
2. In the **Company audit settings** section, view or change the log settings by selecting or clearing the check boxes. Check boxes include:
   - **System Log** - Enables the logging of system messages.
   - **Audit Trail** - Enables the logging of business-critical data changes.
   - **Document Log** - Enables the logging of document operations.



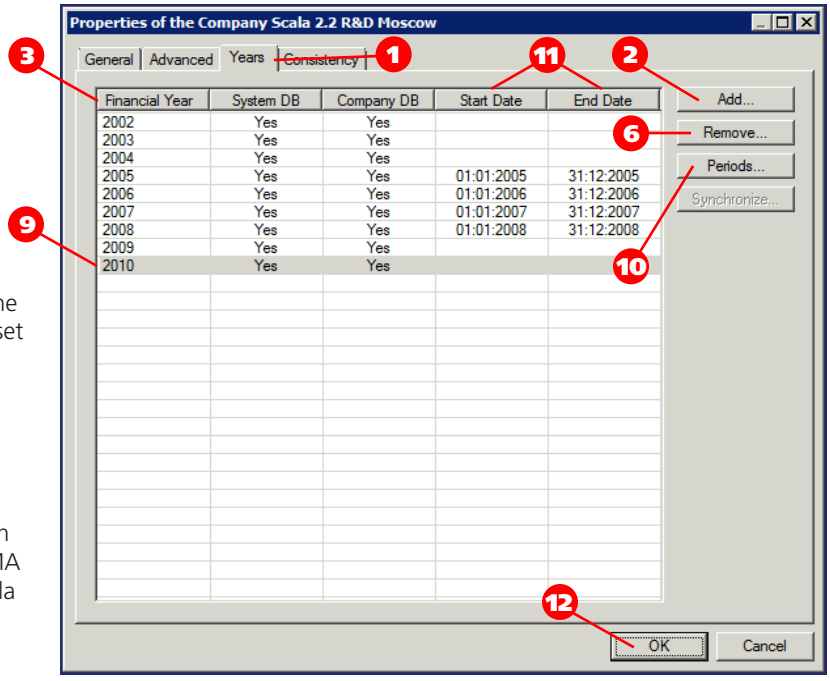
Set Logging of Action Reauthentication Attempts

iScala Actions can be password-protected, requiring logged-on users to re-authenticate prior to starting an action, thus preventing misuse of iScala sessions that are left running without supervision. When a user starts the protected Action, the User Reauthentication dialog displays with the information that the Action is protected and prompts the user to enter a password in order to run the Action. On the Advanced tab, under Reauthentication logging title, you can enable or disable logging of successful and failed reauthentication attempts.

1. To enable failed Reauthentication attempts logging, select the **Log Failed Attempts** check box.
2. To enable successful Reauthentication attempts logging, select the **Log Successful Attempts** check box.
3. Click **OK**.
4. In the iScala Administration Console, right-click the **Events** node and select **Properties**.
5. For failed Reauthentication attempts logging, in the **Properties** window, under **Filter Settings**, select the **Warning messages** check box.
6. For successful Reauthentication attempts logging, select the **Information messages** check box.
7. Click **OK**.



Manage the Financial Years for the Company

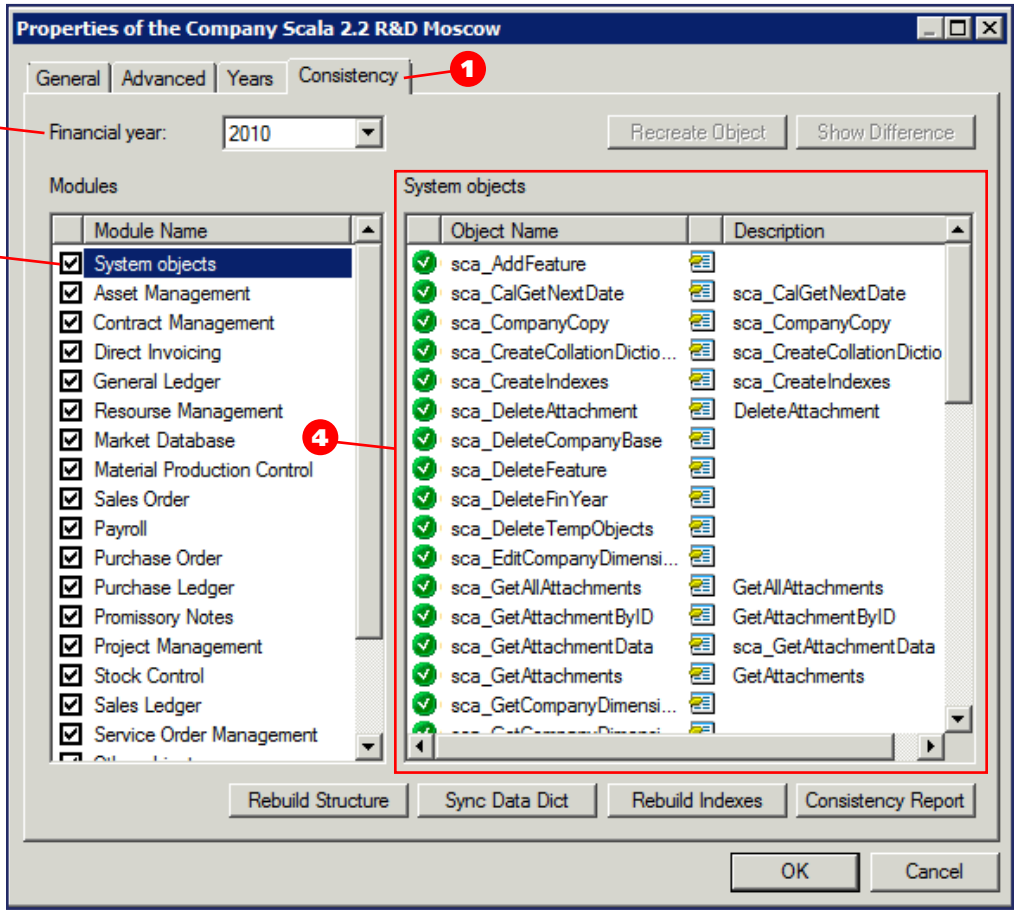


1. In the **Properties of the Company <company name>** window, click the **Years** tab. The tab displays the list of company financial years.
2. To add a company financial year, click the **Add** button.
3. In the **Financial Year** window, enter the year in the **Financial Year** field.

   To specify a financial year period different from the default, select the corresponding check box and set the period start and end dates.
4. In the **Module Name** section, view the list of installed iScala modules.

   When you add a Financial Year, all modules are installed and the necessary tables are created with one exception. For the Market Database of the MA module, several tables are not created in the iScala Administration Console. You should create these tables in the corresponding iScala module.
5. Click **OK**.
6. To remove a financial year from the **Years** list, select the year you want to remove and click the **Remove** button.
7. Specify whether you want to remove year-independent data.
8. Click **OK**.
9. To set a financial period for an existing year, select the year in the grid.
10. Click **Periods**.
11. Specify the period **Start Date** and **End Date**.
12. Click **OK**.

View Lists of Installed Modules and System Objects



1. In the **Properties of the Company <company name>** window, click the **Consistency** tab.
2. The tab displays the list of installed iScala modules and system objects for the year selected in the **Financial year** field.
3. Select a module from the **Modules** list.
4. The list of all **System objects** available for this module displays in the right pane. For each System object, the following information dis plays:

   - **Object status**

     Object does not exist.

     Structure or index of the Object differs from the system database data dictionary.

     Object does not fall into one of the above groups.

   - **Object type** - Procedure, Function, Table, or View
   - **Object description**

Run the Consistency Report

1. In the **Properties of the Company <company name>** window, click the **Consistency** tab.
2. Click the **Consistency** button.

   The **Check Consistency** window displays.

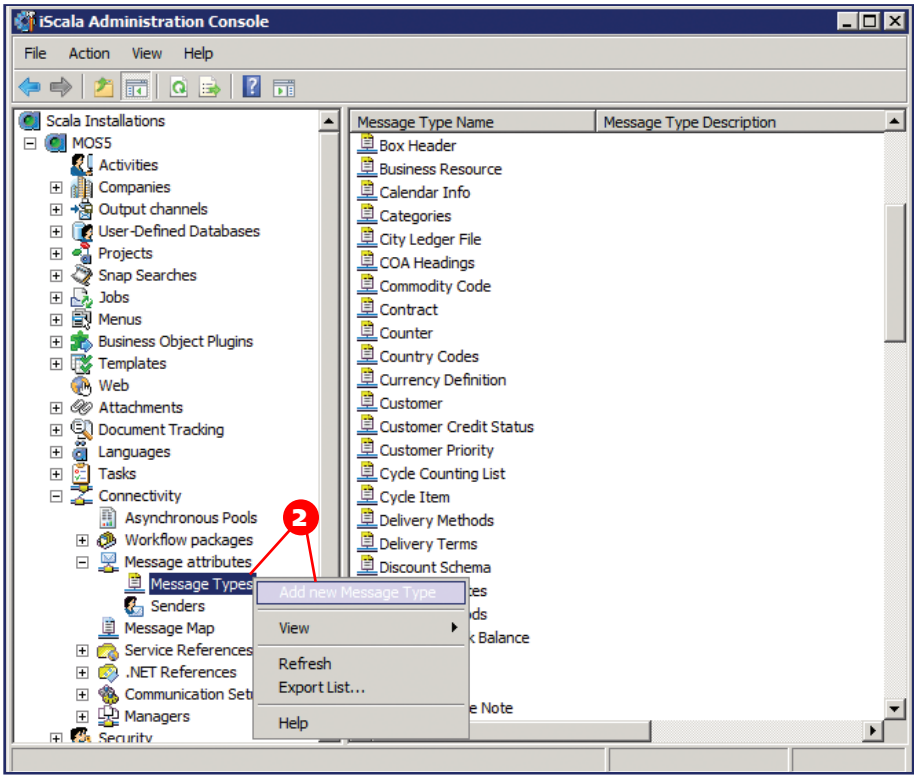
Channel Parameters

Channel parameters are accessed through the iScala Administration Console. You can use the Administration Console to create a Message Type, Sender, and Message Map which connects the message type and sender with the workflow. The iScala Administration Console is also used to define the input file channel.

Create a Message Type

1. In the iScala Administration Console, expand the **Connectivity, Message Attributes**, **Message Types** nodes.

2. Right-click **Message Types** and select **Add New Message Type**.

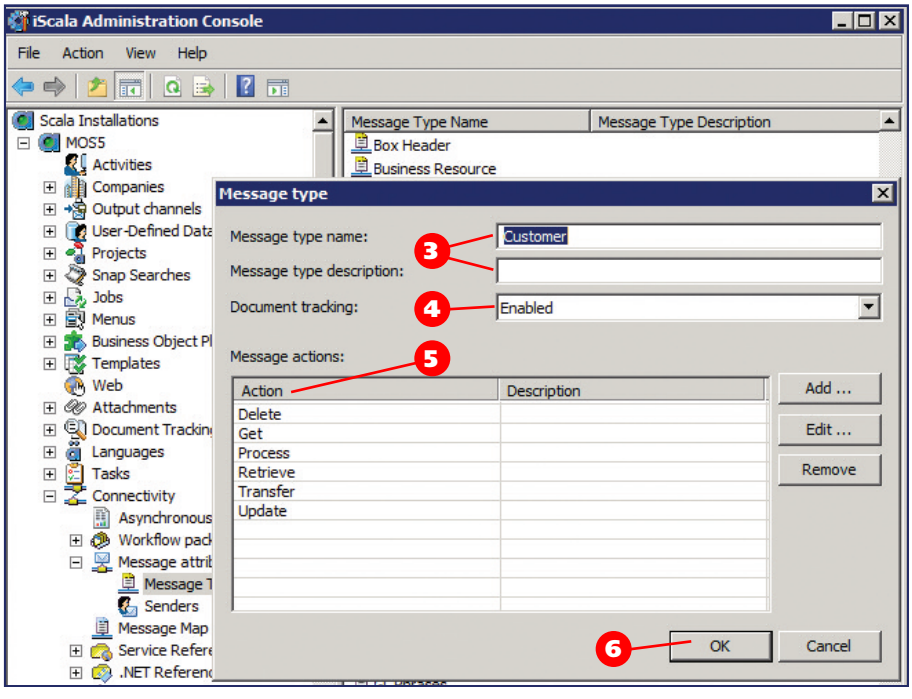


3. Enter a **Message type name** and **Message type description**.

4. Select **Enabled** in the **Document Tracking** field.

5. Add a **Message Action**. Available actions act as descriptions of mapped processes.

6. Click **OK**.



Create a Sender

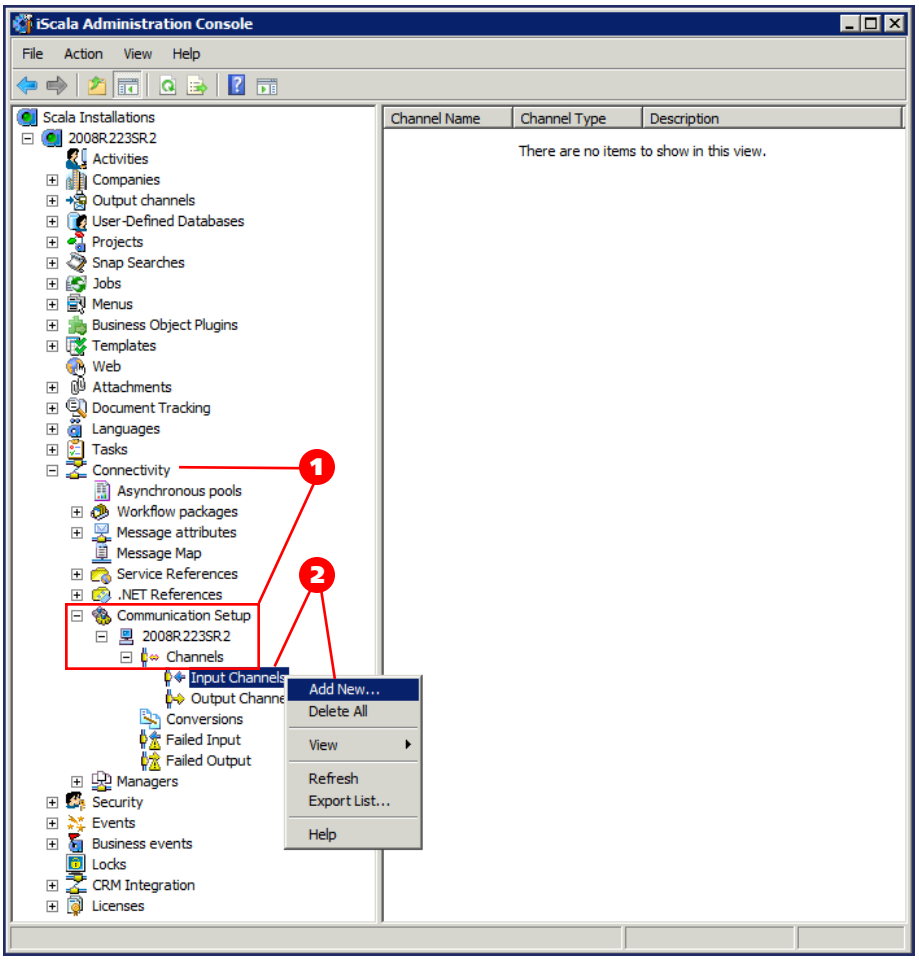
1. In the iScala Administration Console, expand the **Connectivity** node and the **Message Attributes** node.
2. Right-click **Senders** and select **Add New Sender**.
3. Enter a **Sender name** and **description**.
4. Select **Enabled** in the **Document tracking** field.
5. Click the **Add** button in the **Sender subnames** section.
6. Enter a **Sub-sender name** and **Sub-sender description**.
7. Click **OK**.

Create a Message Map

1. In the iScala Administration Console, expand the **Connectivity node**.
2. Right-click **Message Map** and select **Add New Request**.
3. Enter the **Message Type** and **Sender** information in the **New Request ID** dialog box.
4. If you have already created a workflow, click **Select** to select your workflow from the list.
5. Click **OK** until you exit all dialog boxes.

Create File Input Channel

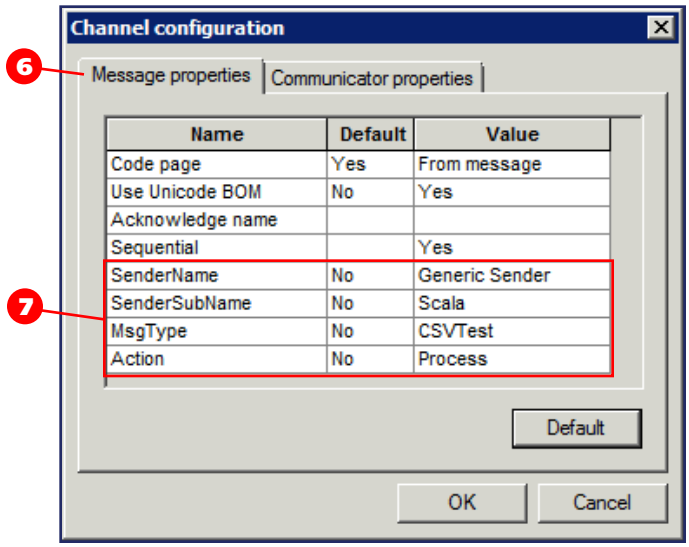
1. In the iScala Administration Console, expand the **Connectivity** node, **Communication Setup** node, **<server name>** node, and **Channels** node.
2. Right-click **Input Channels** and select **Add New**.
3. In the **Channel properties** window, enter a **Channel name** and select **FILE** as the **Listener type**.
4. Select the **Enable channel** check box.
5. To set channel properties, click **Configure**.



6. In the **Channel configuration** window, click the **Message properties** tab.

7. Enter the same information, listed below, as that entered in the Message Map. These settings link the Message Map to the Input Channel:

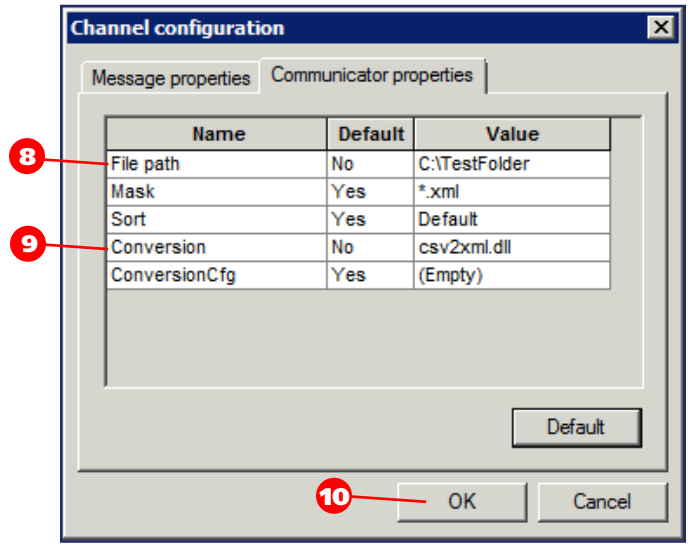
   - SenderName
   - SenderSubName
   - MsgType
   - Action



8. On the **Communicator properties** tab, specify the **File path** to a folder Service Connect can monitor, for example, C:\TestFolder.

9. Select the appropriate **Conversion**.

10. Click **OK**.



Summary

This chapter described how to integrate Epicor Service Connect with Epicor ERP, Epicor for Service Enterprises, and Epicor iScala. You can now create workflows. For more details about how to create workflows, refer to Chapter 4: Workflow Designer or to the appropriate sample workflow documentation for your Epicor solution, posted on EPICweb.

EPICOR®

Business Inspired™

Worldwide Headquarters
San Francisco Bay Area
4120 Dublin Boulevard, Suite 300
Dublin, CA 94568 USA
Toll Free: +1.888.448.2636
Direct: +1.925.361.9900
Fax: +1.925.361.9999

Latin America and Caribbean
Blvd. Antonio L. Rodriguez #1882 Int. 104
Plaza Central, Col. Santa Maria
Monterrey, Nuevo Leon, CP 64650
Mexico
Phone: +52.81.1551.7100
Fax: +52.81.1551.7117

Europe, Middle East and Africa
No. 1 The Arena
Downshire Way
Bracknell, Berkshire RG12 1PU
United Kingdom
Phone: +44.1344.468468
Fax: +44.1344.468010

Asia
238A Thomson Road #23-06
Novena Square Tower A
Singapore 307684
Singapore
Phone: +65.6333.8121
Fax: +65.6333.8131

Australia and New Zealand
Level 34
101 Miller Street
North Sydney NSW 2060
Australia
Phone: +61.2.9927.6200
Fax: +61.2.9927.6298