

A black and white photograph showing the lower half of two people walking away from the camera. They are dressed in business attire; one person is wearing a dark suit and the other is in a light-colored dress. They are walking on a polished floor in what appears to be a modern office building's lobby, with large windows and architectural grid patterns visible in the background.

Epicor Service Connect User Guide

Disclaimer

This document and its contents, including the viewpoints, dates and functional content expressed herein are the proprietary copyrighted property of Epicor Software Corporation, are intended for informational purposes only and are believed to be accurate as of its date of publication. However, Epicor Software Corporation makes no guarantee, representations or warranties with regard to the enclosed information and specifically disclaims any applicable implied warranties, such as fitness for a particular purpose, merchantability, satisfactory quality or reasonable skill and care. As each user of Epicor software is likely to be unique in their requirements in the use of such software and their business processes, users of this document are always advised to discuss the content of this document with their Epicor support representative, account manager and/or consulting personnel. All information contained herein is subject to change without notice and changes to this document since printing and other important information about the software product are made or published in release notes, and you are urged to obtain the current release notes for the software product. The usage of any Epicor software shall be pursuant to an Epicor end user license agreement and the performance of any consulting services by Epicor personnel shall be pursuant to Epicor's services terms and conditions. Usage of the solution(s) described in this document with other Epicor software or third party products may require the purchase of licenses for such other products. Where any software is expressed to be compliant with applicable laws or other statutory or regulatory requirements in this document, such compliance is not a warranty and is based solely on Epicor's current understanding of such laws and requirements. All laws and requirements are subject to varying interpretations as well as to change and accordingly, Epicor cannot guarantee that the software will be compliant and up to date with such changes. All statements of platform and product compatibility in this document shall be considered individually in relation to the products referred to in the relevant statement, i.e., where any Epicor software is stated to be compatible with one product and also stated to be compatible with another product, it should not be interpreted that such Epicor software is compatible with both of the products running at the same time on the same platform or environment. Additionally platform or product compatibility may require the application of Epicor or third-party updates, patches and/or service packs and Epicor has no responsibility for compatibility issues which may be caused by updates, patches and/or service packs released by third parties after the date of publication of this document. Epicor, Business Inspired and the Epicor logo are trademarks of Epicor Software Corporation, registered in the United States, certain other countries and/or the EU. All other trademarks mentioned are the property of their respective owners. Copyright © Epicor Software Corporation 2015. All rights reserved. No part of this publication may be reproduced in any form without the prior written consent of Epicor Software Corporation.

10.1.400

Revision: December 18, 2015 12:21 p.m.

Total pages: 675

sys.ditaval

Contents

Introduction.....	11
-------------------	----

Chapter 1: Epicor Service Connect Overview.....	12
--	-----------

The Service Connect Environment.....	12
Service Connect Solutions.....	13
Epicor Service Connect Administration Console.....	13
Service Connect Workflow Designer.....	14
Service Connect Document Transformation.....	15
Document Submission.....	15
Document Conversion.....	18
Summary.....	21

Chapter 2: Epicor Service Connect Administration.....	22
--	-----------

User Management.....	22
Log in to Service Connect.....	22
Change the Admin and Epicor User Passwords.....	23
Set Up a User.....	24
Setup User Accounts Using Roles.....	26
Grant Permissions.....	29
Add Multiple Users.....	31
Import User Lists.....	35
Service Connect Installation Management.....	38
Register Servers.....	38
Manage User Sessions.....	40
Manage Service Connect Services.....	41
View Installation Properties.....	43
Use Event Logs.....	48
Filter Log Entries.....	50
Manage Licenses.....	52
View the License ID and Location.....	53
Import License.....	55
Connectivity Administration.....	55
Create a Backup.....	55
Restore from a Backup.....	61
Communication Setup Items, Workflows, Service references, .NET References or REST References	
Restore.....	68
Restore Service References.....	71
Restore .NET References.....	73
Restore RESTful References.....	74
Restore a Workflow with a DBOperation.....	74

Setup References Re-import.....	78
Summary.....	79
Chapter 3: Connectivity Components.....	80
Connectivity Components.....	80
Workflows.....	80
Create a Workflow Package.....	82
Move a Workflow between Workflow Packages.....	83
Expose Processes as Web Services.....	86
Message Attributes and Message Maps.....	88
Message Attributes.....	90
Add a Message Type.....	90
Add a Sender.....	92
Message Maps.....	94
Add a Message Map (Routing).....	94
Service References.....	97
Add a Service Reference.....	97
Finalize Service Reference Import.....	107
Add Service Methods.....	108
Re-Import Service References.....	110
Re-Import an Individual Web Service.....	113
Test Connectivity.....	115
Logging Web Service Calls.....	119
.NET References.....	119
Add a .NET Reference.....	120
Epicor 9 .NET Reference Setup.....	125
Epicor 10 .NET Reference Setup.....	126
.NET Import Finalization.....	128
Add Multiple .NET References.....	134
Multiple .NET References Import Finalization.....	140
Add .NET Methods.....	144
Re-Import .NET References.....	146
Test Methods.....	150
Epicor Session Usage Policy.....	153
RESTful References.....	153
Add a RESTful Reference Manually.....	153
Add Resource Group.....	163
Add Resource.....	164
Add Operation.....	166
Edit Resource Group, Resource, or Operation.....	169
Import a RESTful Reference Automatically.....	171
Schemas.....	177
Asynchronous Pools	178
Manage Asynchronous Pools.....	180

Create an Asynchronous Pool.....	180
Edit an Asynchronous Pool.....	181
Pause an Asynchronous Pool.....	182
Resume a paused Asynchronous Pool.....	185
Clear an Asynchronous Pool.....	187
Remove an Asynchronous Pool.....	188
Channels.....	188
Add a Channel.....	190
Set Up an Input Channel.....	191
Set Up Channel Schedule.....	196
Configure COM Channel Options.....	198
Configure FILE Channel Options.....	199
Configure FTP Channel Options.....	201
Configure SFTP Channel Options.....	202
Configure FTP (.NET) Channel Options.....	204
Configure HTTP Channel Options.....	207
Configure Schedule Channel Options.....	208
Configure IBM MQ and IBM MQ (.NET) Channel Options.....	213
IBM MQ Channel Options.....	214
IBM MQ (.NET) Channel Options.....	215
Configure MSMQ Channel Options.....	216
Configure POP3 Channel Options.....	217
Configure IMAP Channel Options	218
Configure SMTP Channel Options.....	220
Configure SonicMQ Channel Options.....	221
Configure Windows Service Bus Channel Options.....	222
Conversions.....	223
Set up Conversion Plug-In Configuration Options.....	226
Manage Custom Conversions.....	230
Failed Input.....	232
Failed Output.....	234
Set Number of Attempts for Failed Output.....	235
Workflow Converter.....	236
Convert Workflow.....	238
SharePoint Integration.....	247
Set Up the Input Channel.....	248
Create a Workflow.....	251
Add a Message Map.....	256
Define a SharePoint Workflow.....	258
Add an Output Channel.....	259
Use the Output Channel from a Poster.....	261
Tasks.....	263
Process a Task.....	263
Document Tracking.....	267
Set Up Document Tracking.....	268

Archiving Options.....	270
View Activities.....	271
Activity Views.....	279
Add a New View.....	280
Filter an Existing View.....	285
Use a Document Type to Filter a View.....	290
Filter a View Using Business Attributes.....	295
Apply an Action to All Activities in a View.....	301
Import to and Export from Document Tracking.....	302
Import Activities into ESC Administration Console.....	303
Summary.....	305
Chapter 4: Workflow Designer.....	306
Workflow Designer Window.....	306
Menu Bar.....	307
File Menu.....	307
Edit Menu.....	307
Tools Menu.....	308
View Menu.....	308
Window Menu.....	308
Help Menu.....	309
Commands Toolbar.....	309
Standard Toolbar.....	310
Items Toolbar.....	311
Splitter.....	311
Set Up a Splitter.....	312
Task.....	316
Task Example.....	316
Set Up a Task.....	316
Set Up Task General Properties.....	320
Set Up Task Exits.....	321
Set Up Task Dynamic Properties.....	323
Task Dynamic Configuration Example.....	329
Poster.....	335
Poster Example.....	335
Set Up a Poster.....	335
Build Email Template.....	339
Set Up a Poster Appearance.....	347
Set Up Poster Dynamic Properties.....	347
Sub-Workflow.....	353
Sub-Workflow Example.....	353
Set Up a Sub-Workflow.....	353
Set Up Sub-Workflow Cycling.....	357
Create a Subroutine Workflow.....	362

Windows Workflow Foundation® Call.....	363
Set Up the Windows WF Integration.....	364
Create a Project in Visual Studio.....	364
Add Service Connect Activities to the Visual Studio Toolbox.....	365
Register the WF Workflow in Service Connect.....	366
Call the WF Workflow from a Service Connect Workflow.....	366
Requester.....	367
Requester Example.....	367
Set Up a Requester.....	368
Set Up Requester Dynamic Properties.....	371
Conversion.....	377
Conversion Example.....	377
Set Up a Conversion.....	377
Condition.....	381
Use Condition Rule.....	381
Condition Example.....	381
Set Up a Condition.....	381
Choice.....	386
Use Choice Rules.....	387
Choice Example.....	387
Set Up a Choice.....	387
Set up Choice Rules.....	391
Web Method.....	395
Web Method Example.....	396
Set Up a Web Method.....	396
Response Processing.....	397
Epicor Credentials Setup for Epicor 10 WCF Web Methods.....	400
InfoMessage Setup for Epicor 10 WCF Web Methods.....	401
UpdateExt Setup.....	404
.NET Call.....	404
.NET Call Example.....	405
Set Up a .NET Call.....	405
REST Operation.....	408
Set Up a REST Operation.....	408
DB Operation.....	413
DB Operation Example.....	414
Set Up a DBOperation.....	414
Build a SQL Statement.....	418
Finalize DBOperation Setup.....	424
Break.....	427
Break Example.....	428
Set Up a Break.....	428
XML Mapper.....	435
XML Mapper Graphical Interface.....	437
Document Nodes.....	439

Simple Nodes.....	439
Complex Nodes.....	439
Attribute Nodes.....	440
Fake Nodes.....	441
Node Characteristics.....	441
Node Annotations and Data Types.....	441
Apply a Literal Value.....	445
Set Up a Node Condition.....	447
Node Mappings.....	450
View Mapped Nodes Only	451
Set Up a Link Condition.....	453
Functoids.....	456
Add a Functoid to a Conversion.....	457
String Functoids.....	461
Boolean Functoids.....	462
Mathematical Functoids.....	463
Operators.....	465
Nodeset Functoids.....	466
Special Functoids.....	466
Date-time Functoids.....	467
Date-time Functoids Format String.....	470
Conditional Functoid Example.....	471
Set Up a Conditional Functoid.....	471
Build XPath.....	474
Conversion Value Functoids.....	477
Define Conversion Rules In Place.....	477
Pre-Defined Conversion Rules.....	479
Create Rules.....	479
Add a Reference to the Rules.....	480
Use the Rules File for the Conversion Functoid.....	482
DB Lookup Functoid Example.....	483
Use DB Lookup Functoid.....	483
Generate the WHERE Expression.....	488
Epicor Log Converter.....	492
Convert an Epicor Log File.....	493
Generating Schema from Sample Data.....	494
Sample Data Example.....	495
Configure the Input Channel.....	495
Generate a Schema.....	498
Process Properties.....	505
General Process Settings.....	505
Document Tracking Process Settings.....	507
Message Extensions.....	509
Message Extensions Example.....	509
Create a Message Extension.....	509

Use the Container.....	515
Combine the Data.....	519
Process Variables.....	521
Process Variables Example.....	521
Set Up a Process Variable.....	521
Drop Nodes to the USR Section.....	525
Epicor Logon.....	531
Process Validation.....	532
DES Poster.....	535
Summary.....	535

Chapter 5: Epicor Service Connect Integration.....536

Epicor ERP.....	536
Integration Overview.....	536
Epicor ERP .NET Business Objects Setup.....	536
Import .NET Reference into Service Connect.....	537
Add a Message Type.....	544
Add a Sender.....	546
Create Input and Output Channels.....	548
Create the Message Map.....	555
Create Document Schemas.....	557
Epicor ERP Tracing.....	565
Start a Trace Log.....	565
Select Log Options.....	569
Create an XML Trace Report.....	572
Epicor ERP Business Process Management.....	575
Locate a Business Object Method.....	575
Create a Directive that Calls a Service Connect Workflow.....	576
BPM Integration Completion.....	580
Example.....	580
Exposing Info Prompt.....	583
Epicor Enterprise.....	584
Integration Overview.....	584
Add a Message Type.....	584
Add a Sender.....	586
Create a Message Map.....	588
Import Service References.....	590
Add an Input Channel.....	594
Add an Output Channel.....	599
Generate a Schema for an Inbound Document.....	602
Special Technique: How to Use the Epicor for Service Enterprises Debugger.....	610
iScala.....	611
Security.....	611
Grant Access Rights.....	612

User Parameters.....	618
View a List of All Users.....	619
Create a New User.....	619
Import a Windows User.....	621
Import a List of Users from a File.....	625
Export Users.....	626
Check Windows User Names.....	629
Delete a User.....	631
Manage User Properties.....	632
Assign a User to a User Group.....	636
Remove a User from a User Group.....	637
Edit Company-Specific User Properties.....	638
Company Parameters.....	641
View General Company Properties.....	642
Block a Company	643
Set an Administrator Logon to a Company Database.....	644
View the User Logon to a Company Database.....	645
Set the Company Default Menu and Override MSRS Properties.....	646
Define Company Audit Settings.....	647
Set Logging of Action Reauthentication Attempts.....	648
Manage the Financial Years for the Company.....	651
View Lists of Installed Modules and System Objects.....	656
Run the Consistency Report.....	657
Channel Parameters.....	659
Create a Message Type.....	659
Create a Sender.....	661
Create a Message Map.....	663
Create File Input Channel.....	665
Summary.....	669

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.

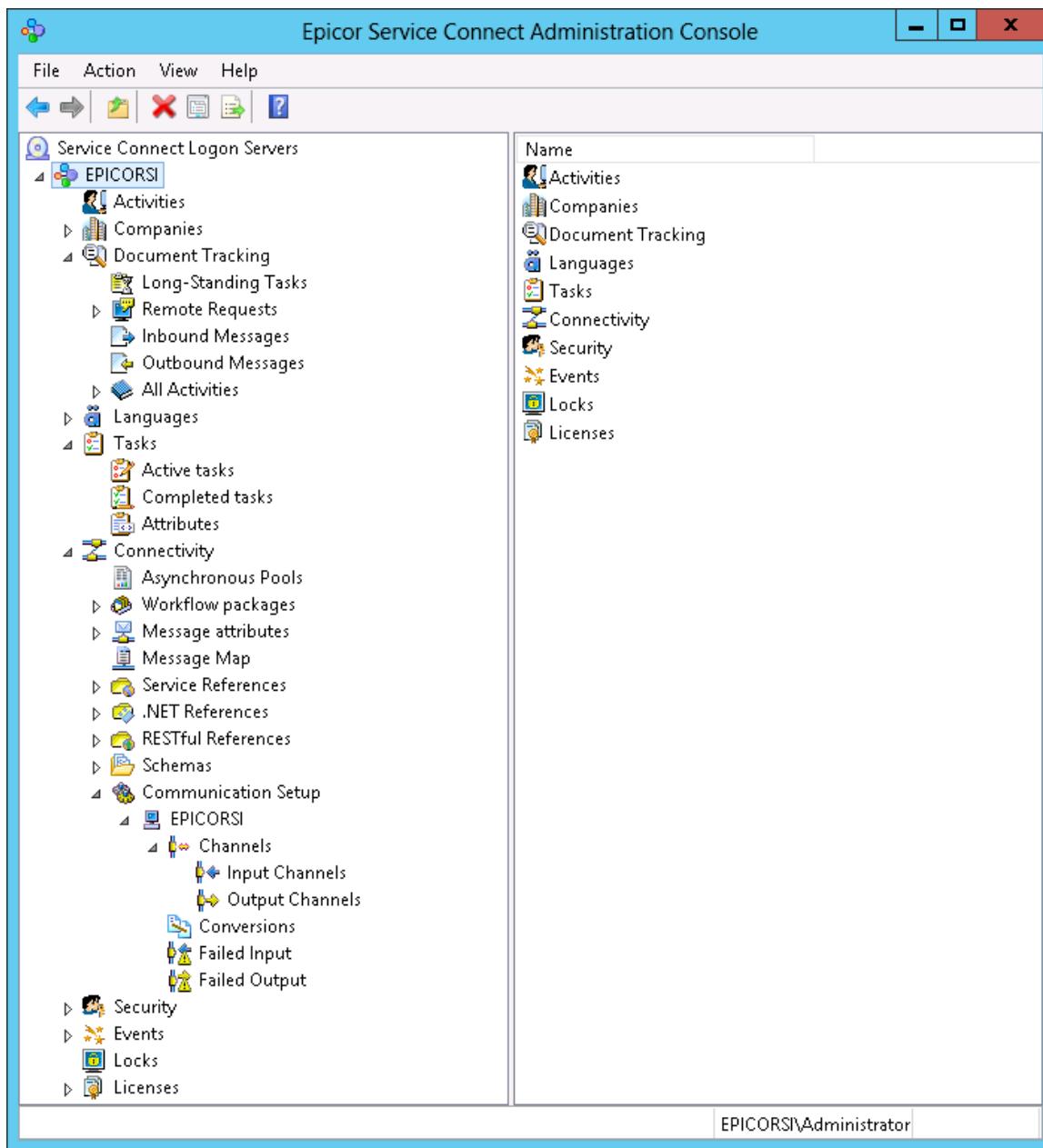
Service Connect Solutions

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

- **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

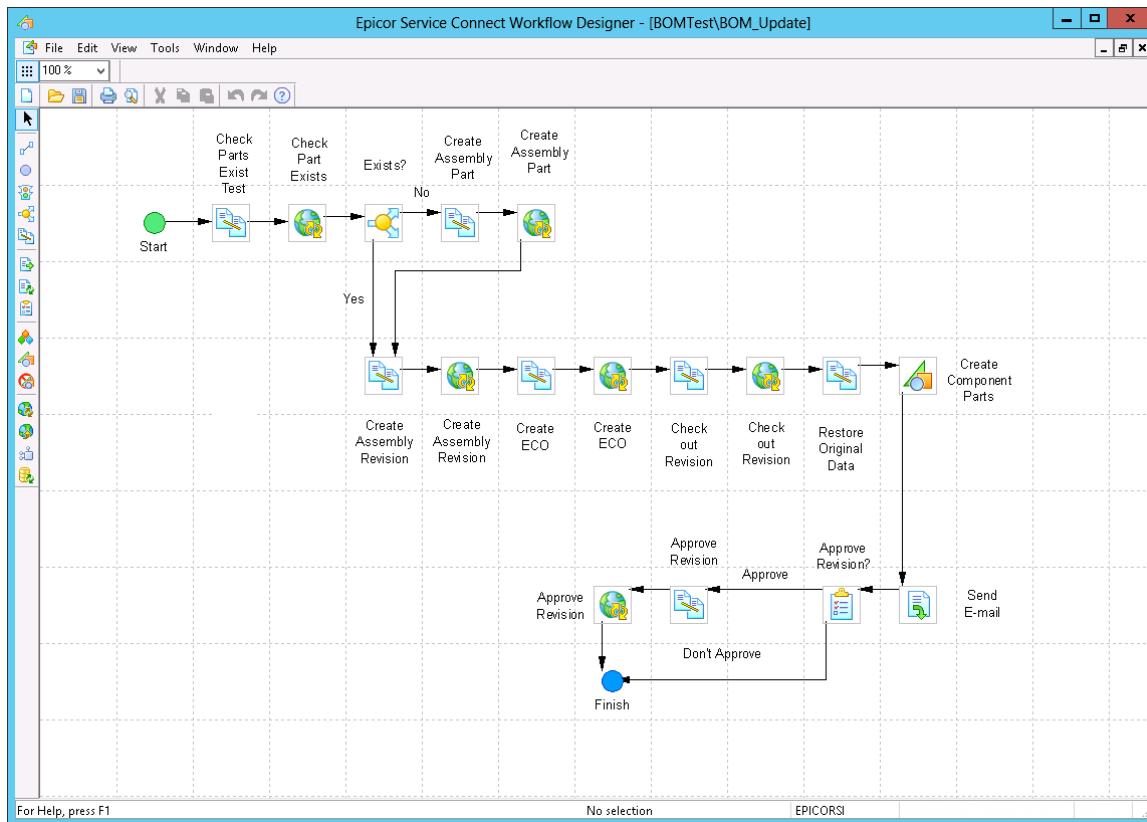
All system administration tasks can be performed from within the ESC Administration Console. This screen shows a single Service Connect installation managed on a server named EPICORSI.



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.

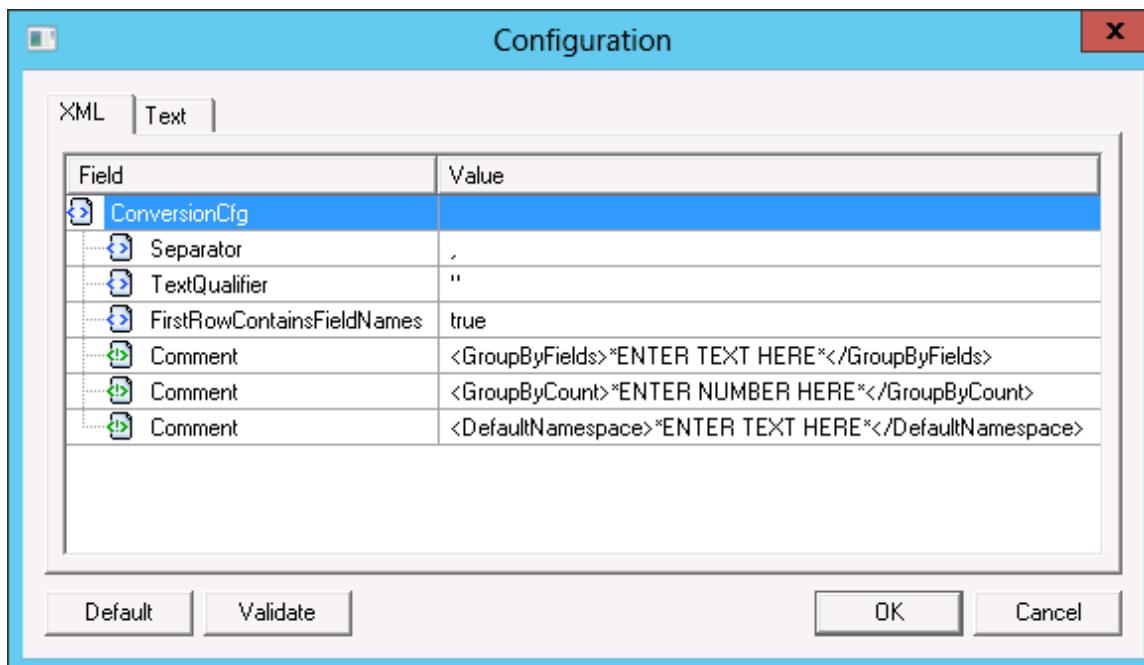
Documents are wrapped in the internal envelope at one of the entry points - an input channel or a workflow exposed as a web service. In general, most integration scenarios use input channels because they are less complex than calling workflows directly or calling workflows exposed as web services.

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.

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.

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 csv2xml conversion plug-in.



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 , cfg and trc 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 is represented as child nodes.
ers	The error messages node. If Service Connect or a web service returns an error, the error number and description are stored 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 is represented as a child node. Message extensions and process

Node	Description
	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 the Workflow Designer chapter 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 ERP version 10, Epicor ERP version 9, 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.

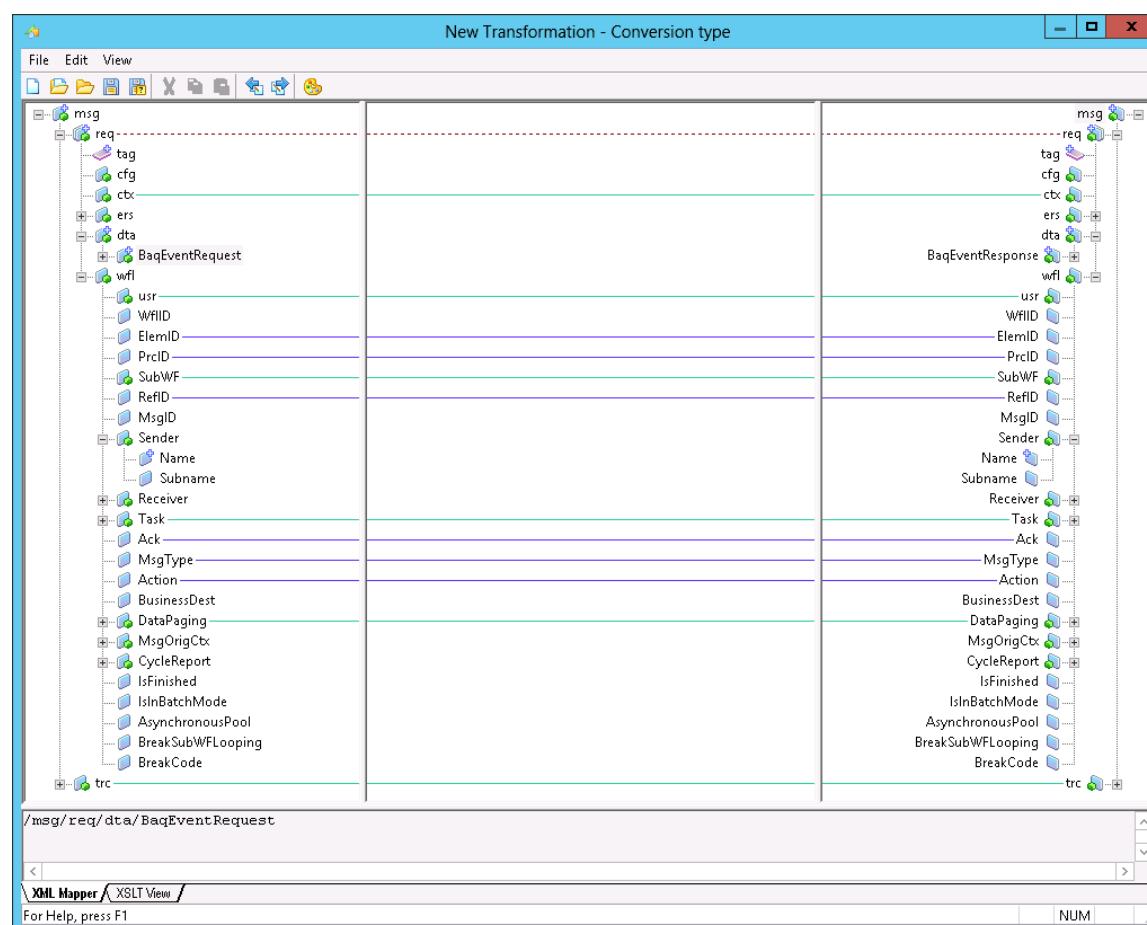
The Workflow is the key component of Service Connect. A workflow organizes a sequence of automatic activities or user-performed tasks to perform on a Service Connect internal message. In general, the goal of a workflow is to transform data via conversion, web services, .NET assemblies, and database operations 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 workflow element uses Extensible Style sheet 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 displays the XML Mapper utility.



Calls to business or custom logic are also key components of a workflow. These are represented as .NET Call, Web Method, and RESTful operation workflow elements. Use the .NET Call, Web Method and RESTful operation activity to send and receive business data to .NET Business Object methods, Web Service methods or methods exposed by a REST web service that have been configured in the ESC Administration Console. Review the Workflow Designer chapter for detailed descriptions of these workflow activities.

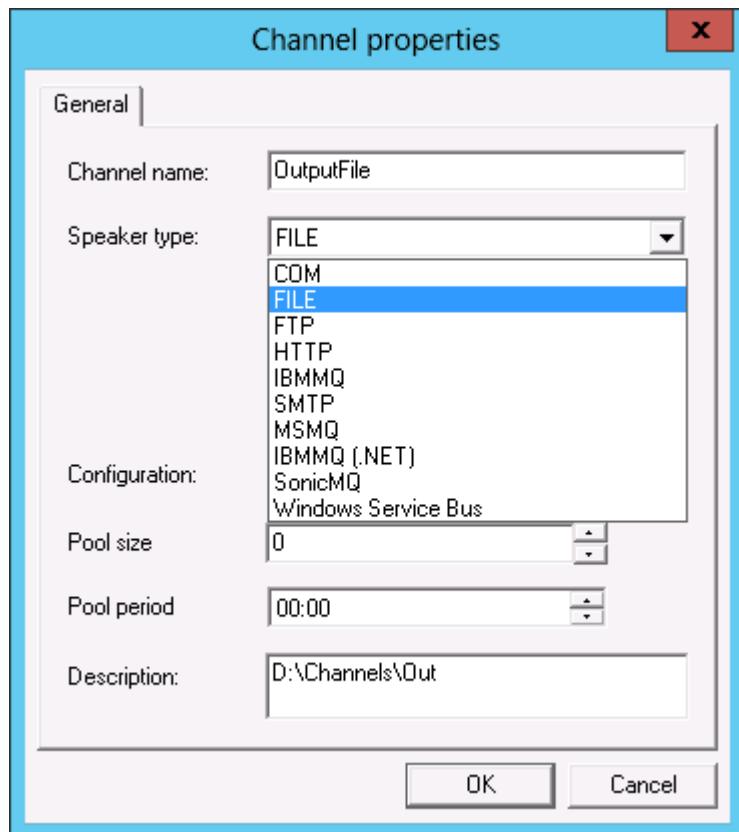
The Task activity is also commonly used. The Task activity pauses a process and alerts particular users so they can decide how the workflow should proceed. All Tasks are queued in the browser-based Task Monitor.

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 the internal 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.

This screen displays the protocols available for an output channel speaker type.



- **Requester activity** – This is similar to the Poster activity, except the Requester pauses workflow execution, and when ESC receives the response message through one of its input channels, the workflow execution is resumed so additional actions can be taken based on the response.
- **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 managing workflows. 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.

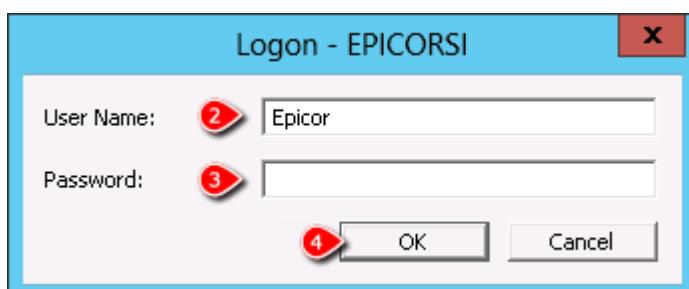
Online Service Connect documentation does not have the Company Specific User properties help topic.

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**.



3. In the **Password** field, enter the password set during the installation process.

4. Click **OK**.

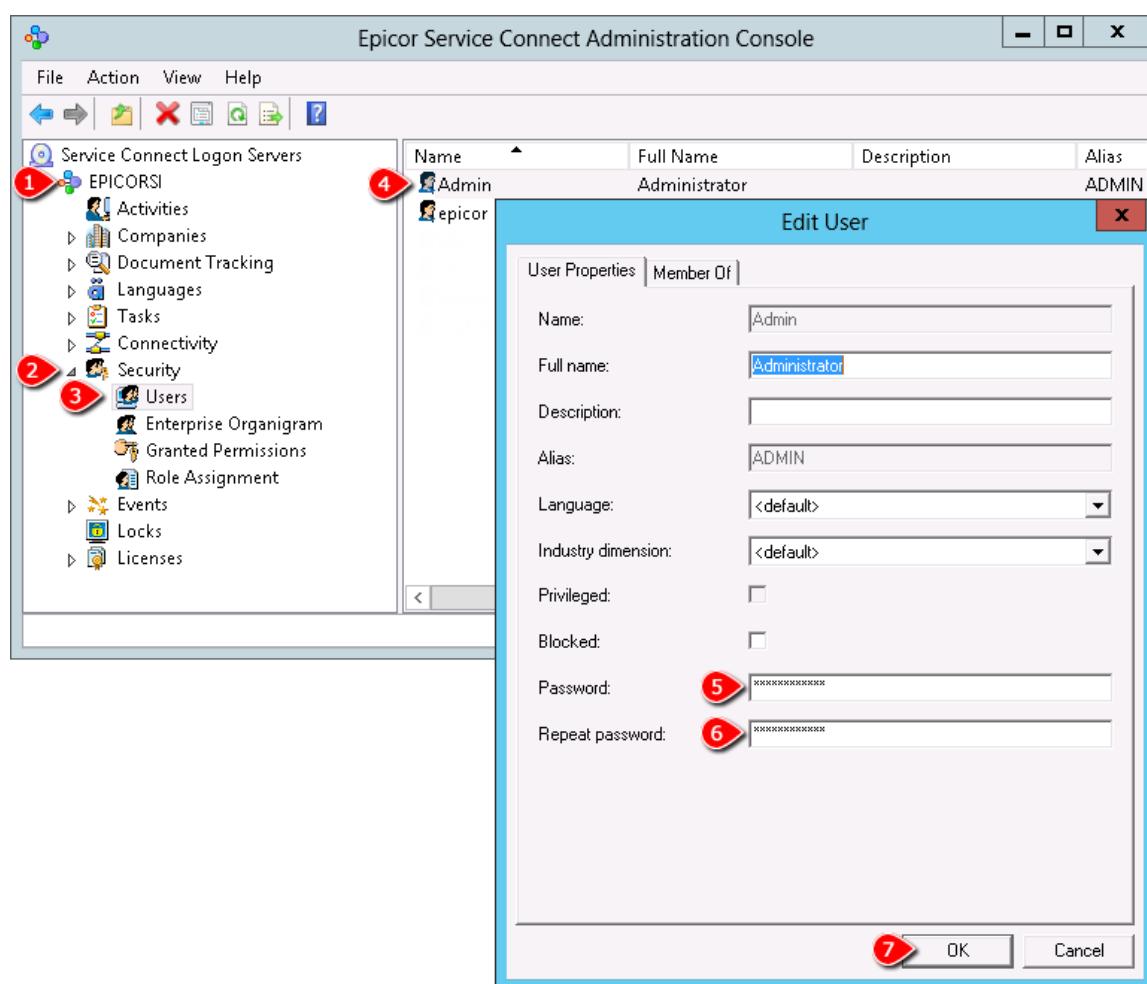
You can also log in with the Admin account.

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**.
6. Enter it again the **Retype password** field.
7. Click **OK**.

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

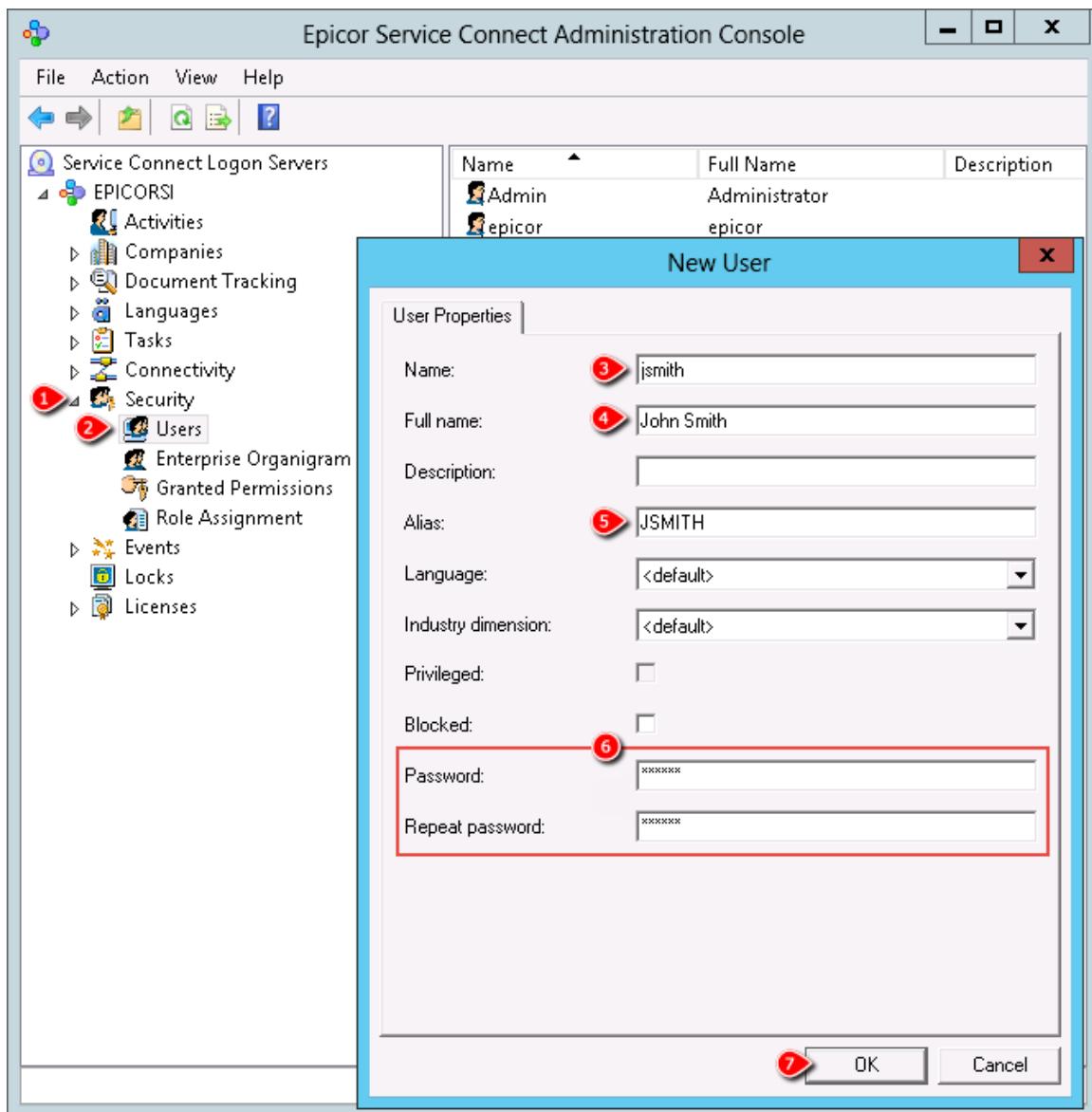
Set Up a User

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 user 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 the **Full name**.
5. 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.

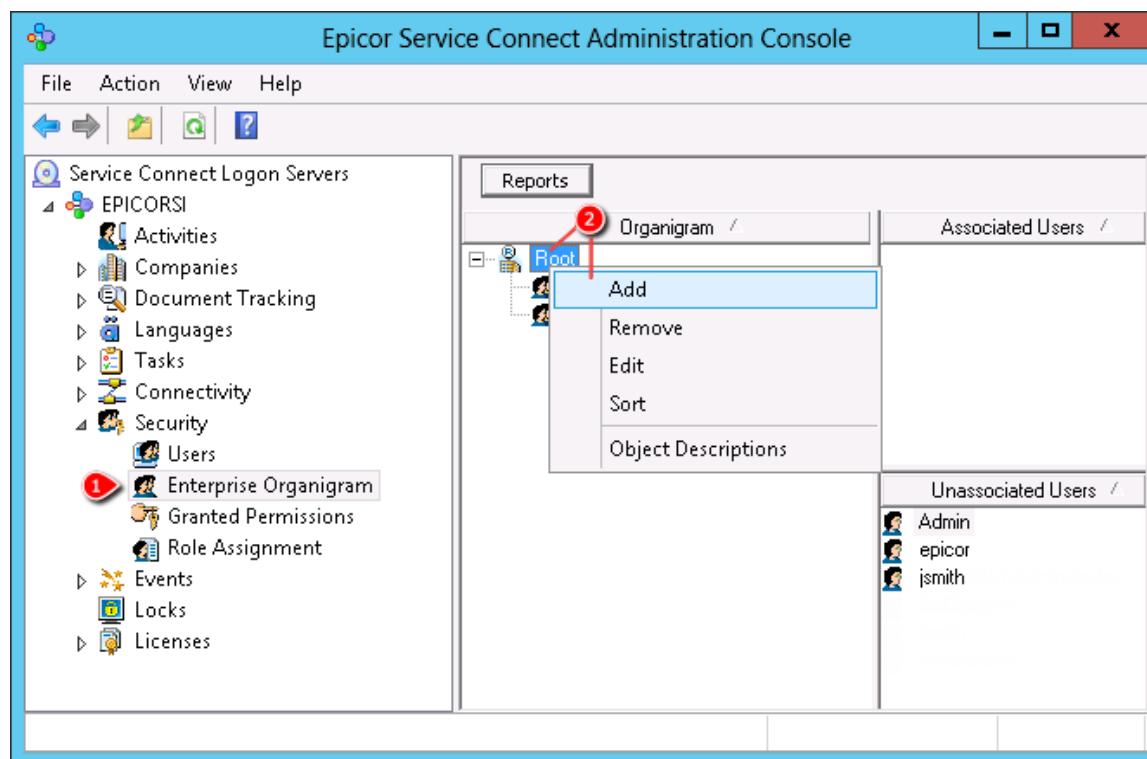
6. Enter your new **Password** and enter it again in the **Retype password** field.
7. When you finish entering the user properties, click **OK**.

Repeat these steps to add other users.

Setup User Accounts Using Roles

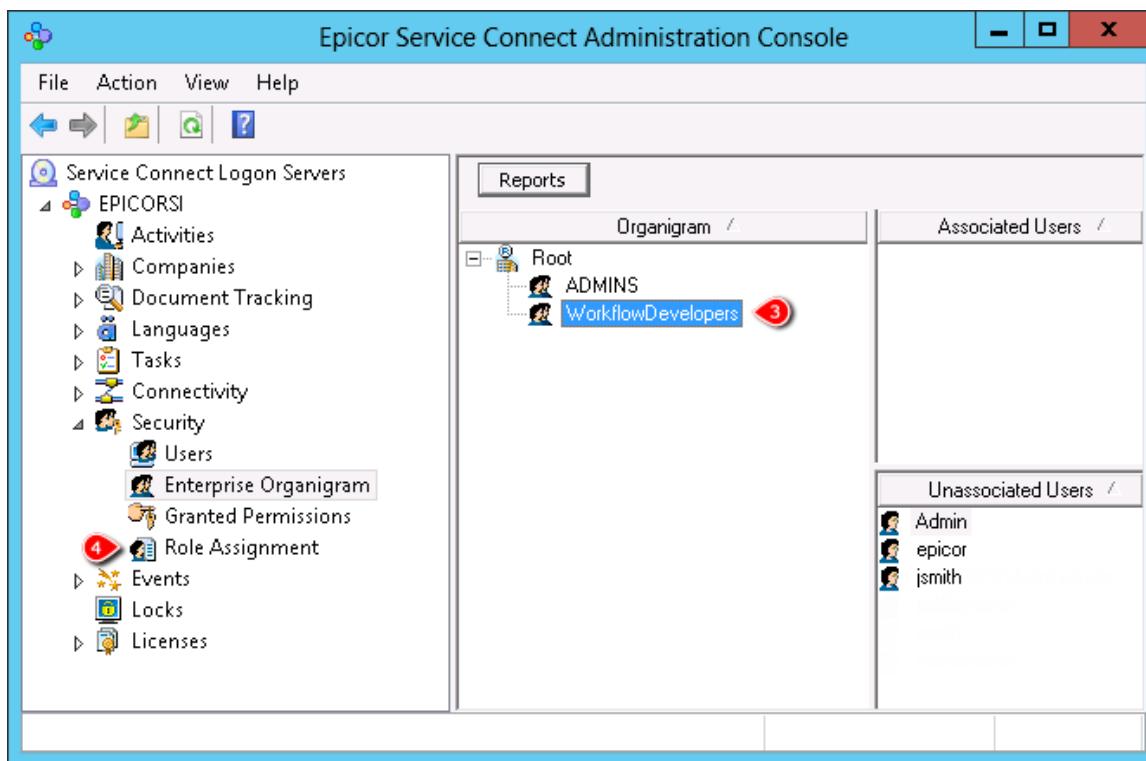
To set up user accounts using built-in roles:

1. In the Tree View, click **Enterprise Organigram**.



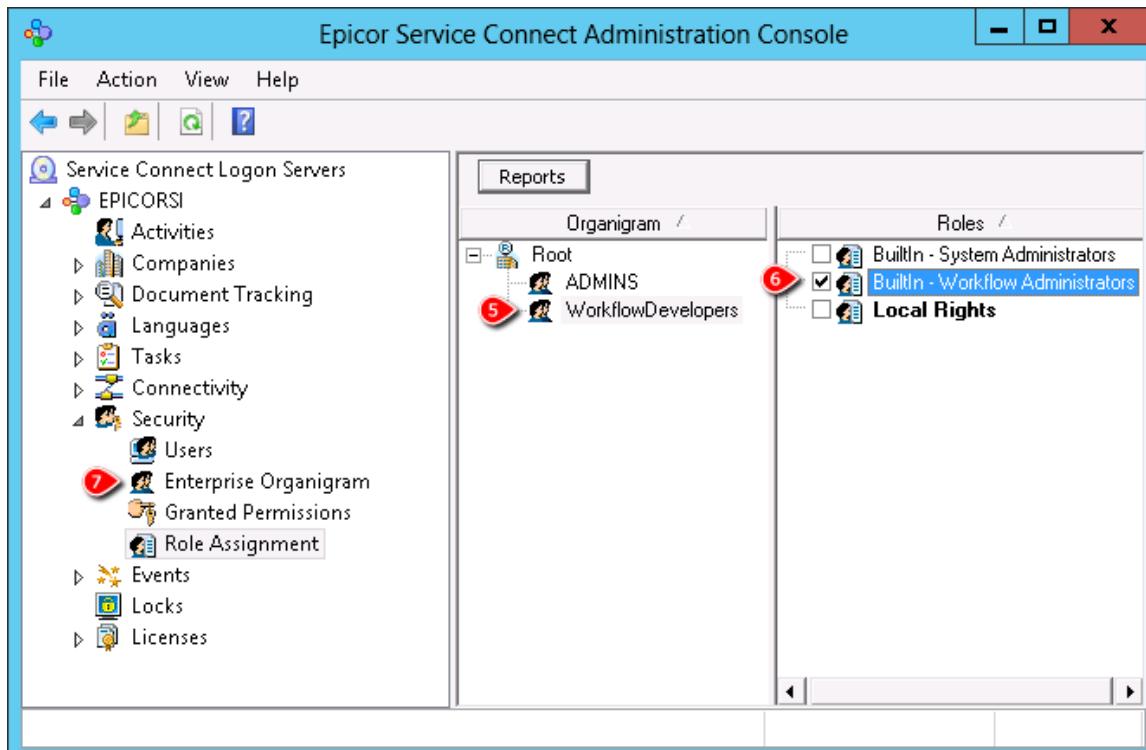
2. Right-click **Root** and select **Add**.

3. Enter **WorkflowDevelopers** to replace the default NewGroup1 text.

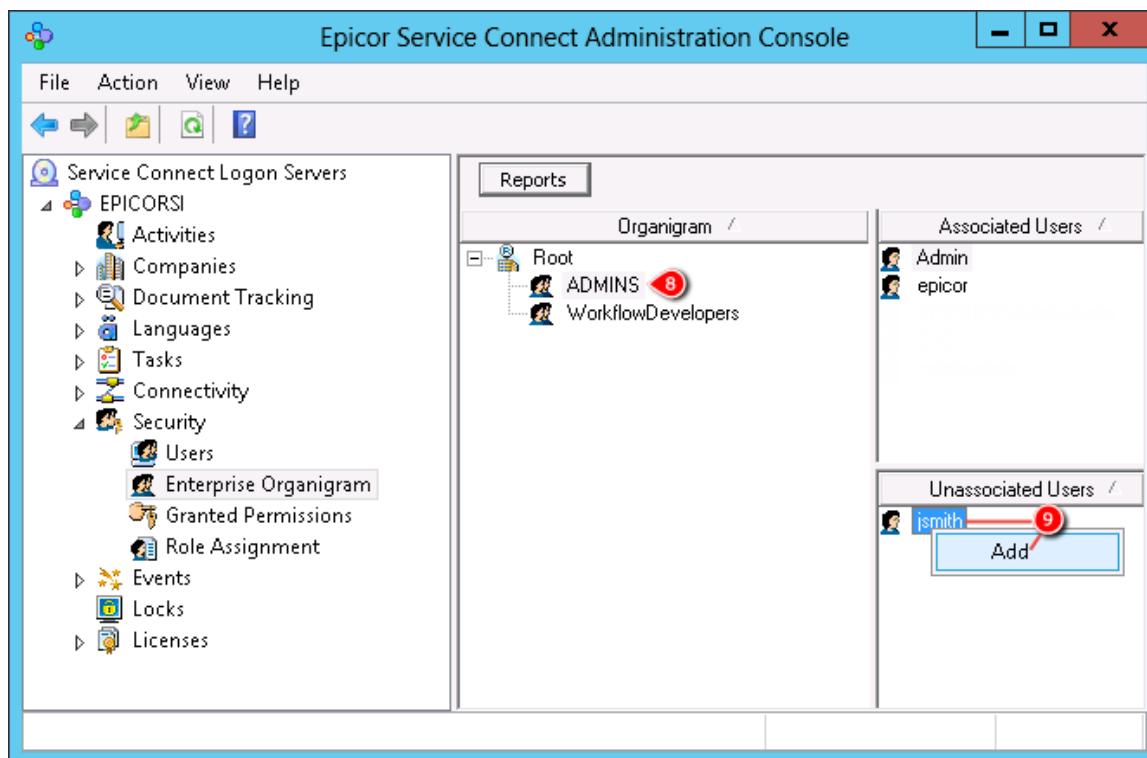


4. Click **Role Assignment**.

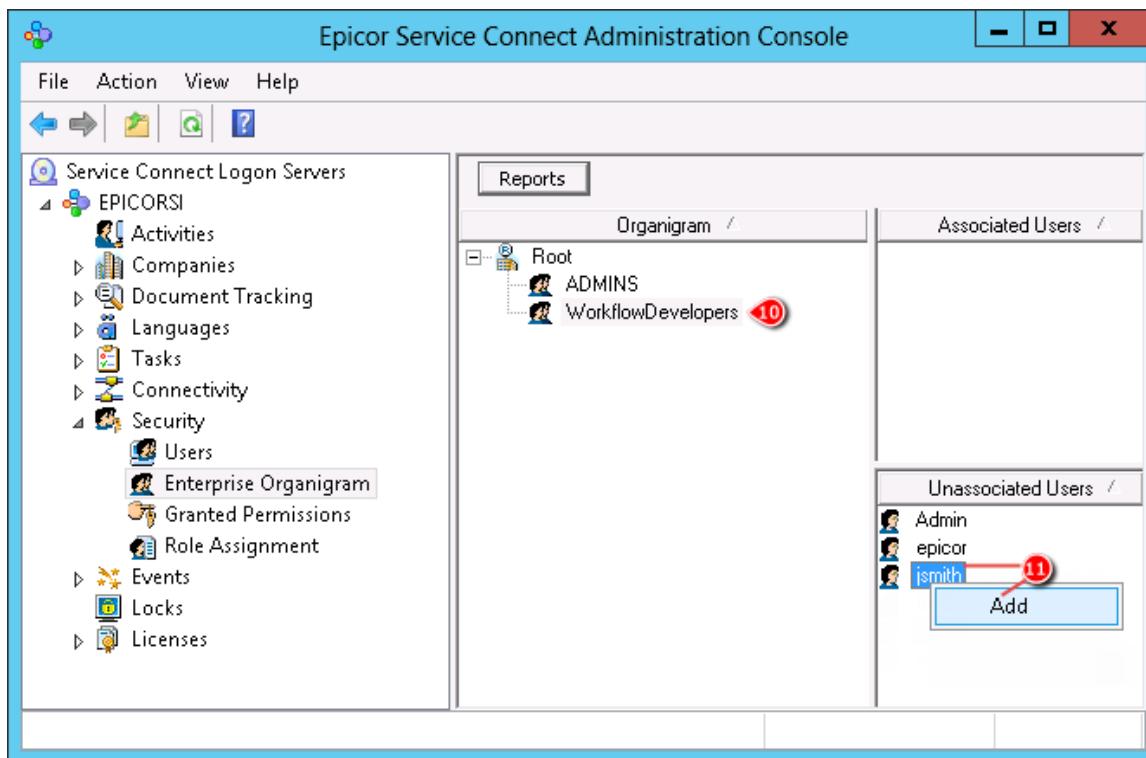
5. In the **Organigram** column, select the **WorkflowDevelopers**.



6. In the **Roles** column, select the **BuiltIn – Workflow Administrators** role.
7. Click **Enterprise Organigram** again.
8. Under the **Root** node, click **ADMINS**.



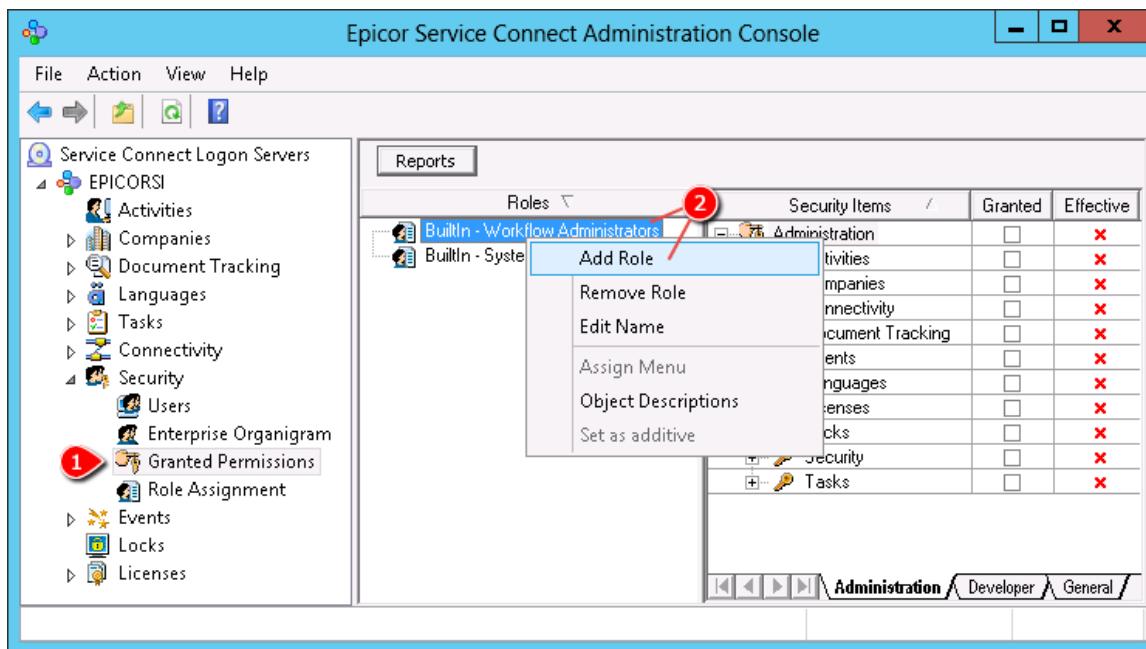
9. Under **Unassociated Users**, right-click the user you want as a system administrator and select **Add**. Repeat this step for all users you want as system administrators. These users are able to do anything in both the ESC Administration Console and the Workflow Designer.
10. Under **Root**, click **WorkflowDevelopers**.



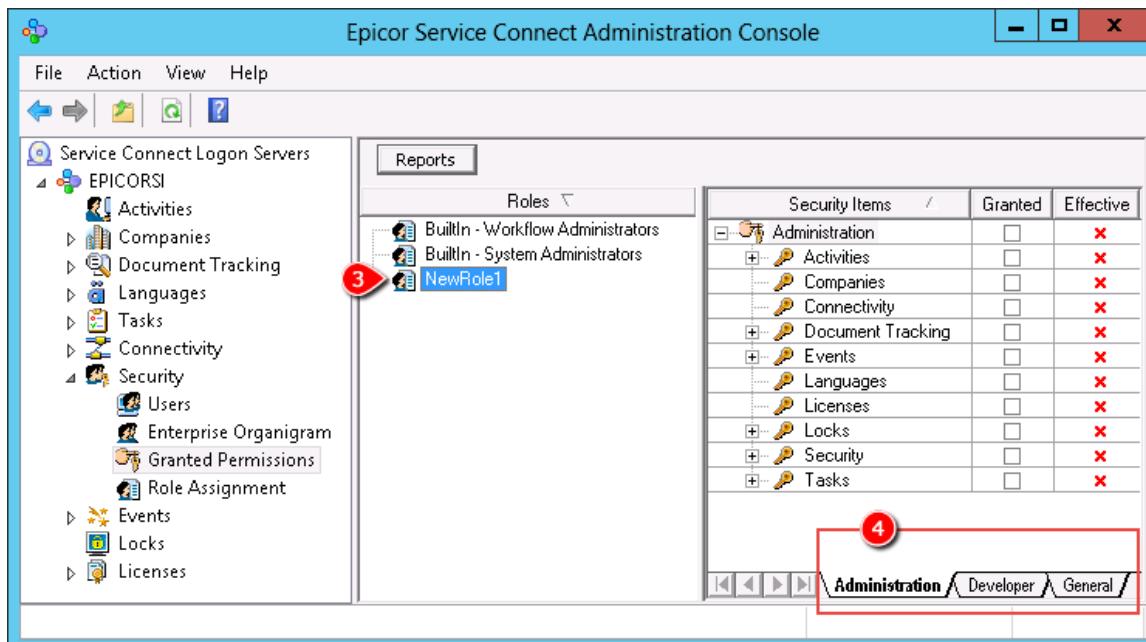
11. Under **Unassociated Users**, right-click the user you want as a workflow developer and select **Add**. Repeat this step for all user you want as workflow developers. These users are able to do anything in the Workflow Designer.

Grant Permissions

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

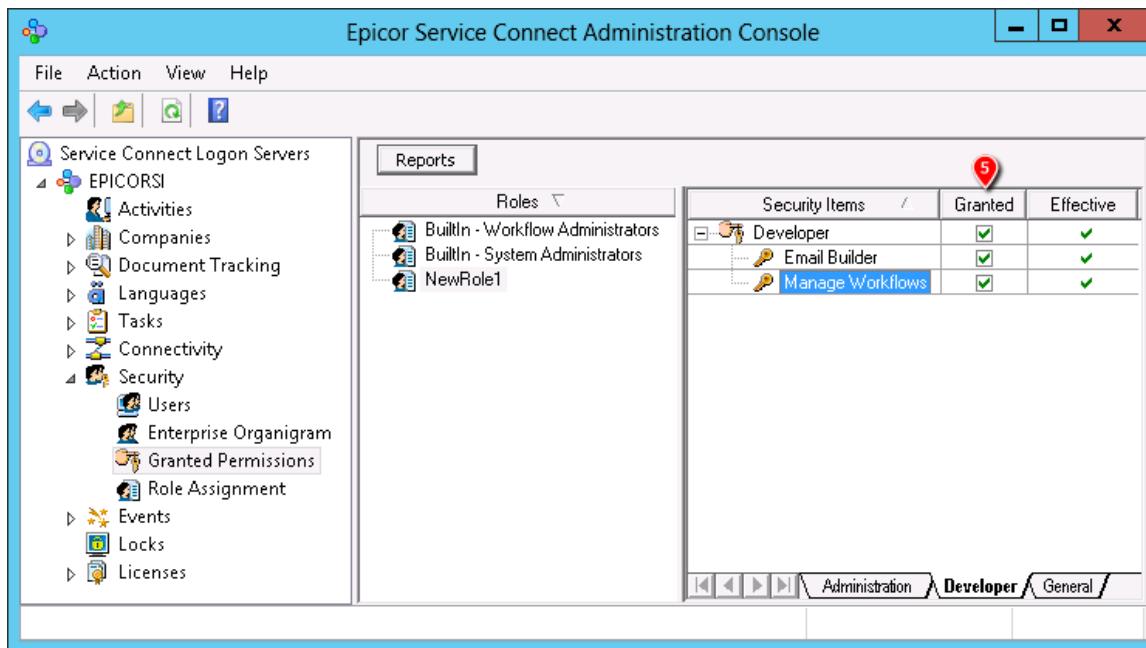


2. In the **Roles** column, right-click a role and select **Add Role**.
3. Highlight the new role.



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.

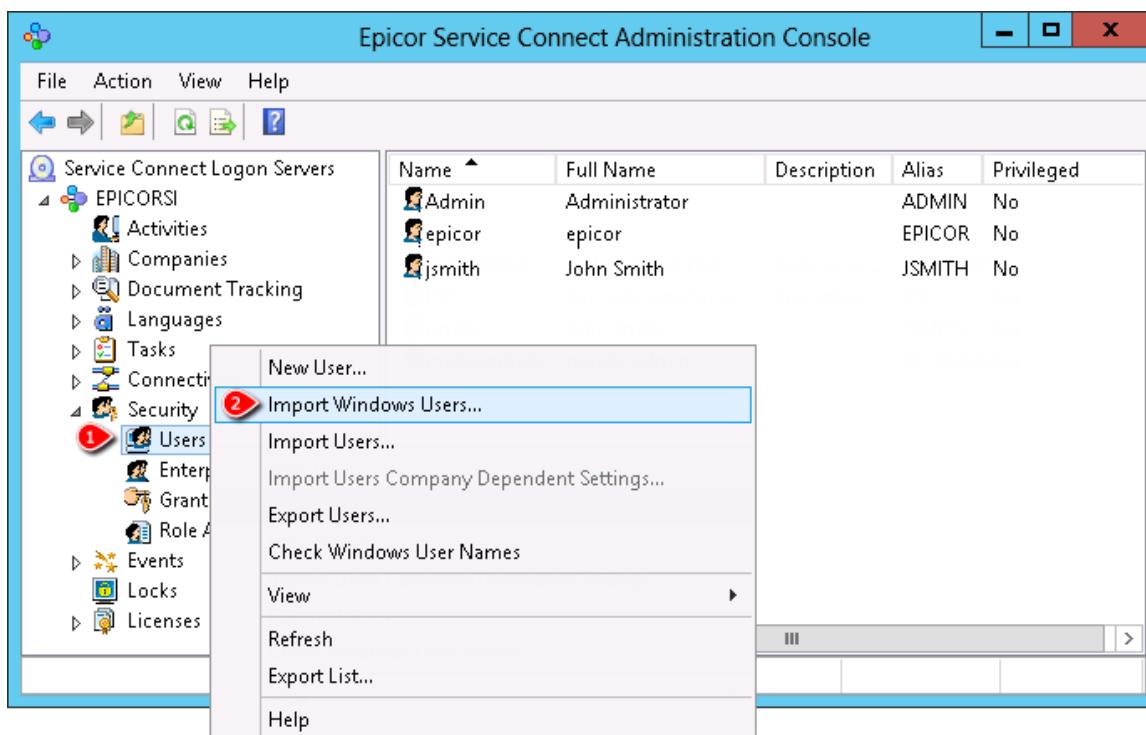


Add Multiple Users

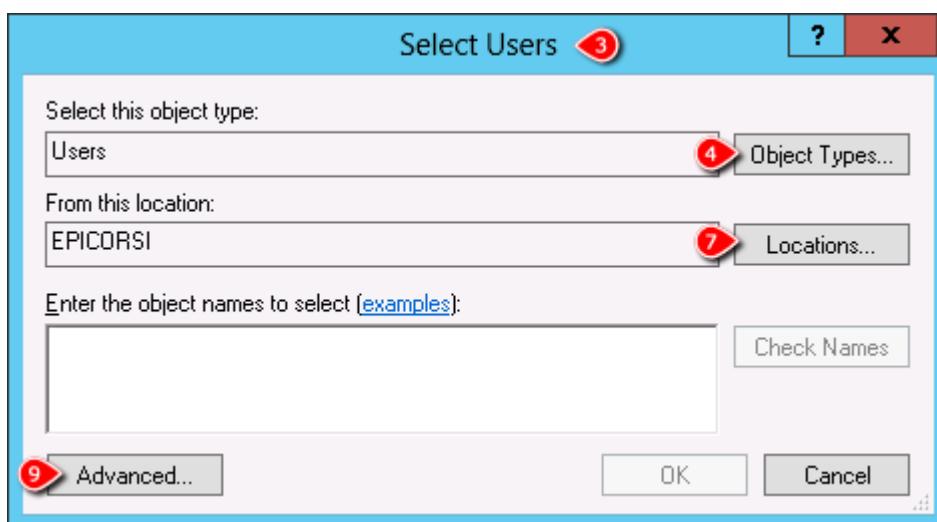
When you set up Service Connect, you can add multiple users at the same time. Use this function to import users from a file or to import Windows users. Windows users are able to log on automatically, that is, they are not prompted for credentials based on a Windows account. Automatic logons only work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

To add multiple Windows users:

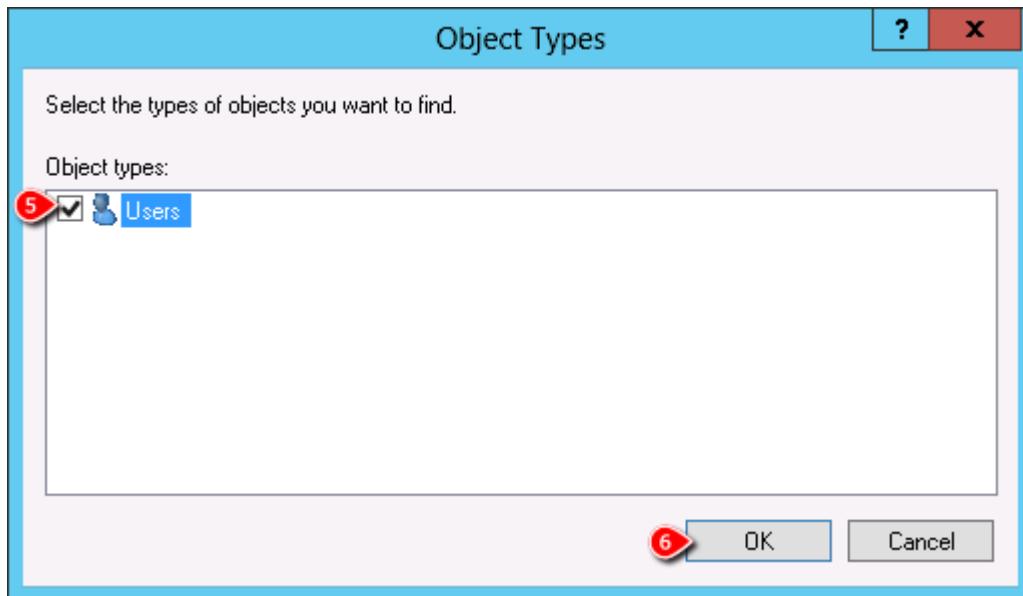
1. In the **Tree View**, right-click **Users**.



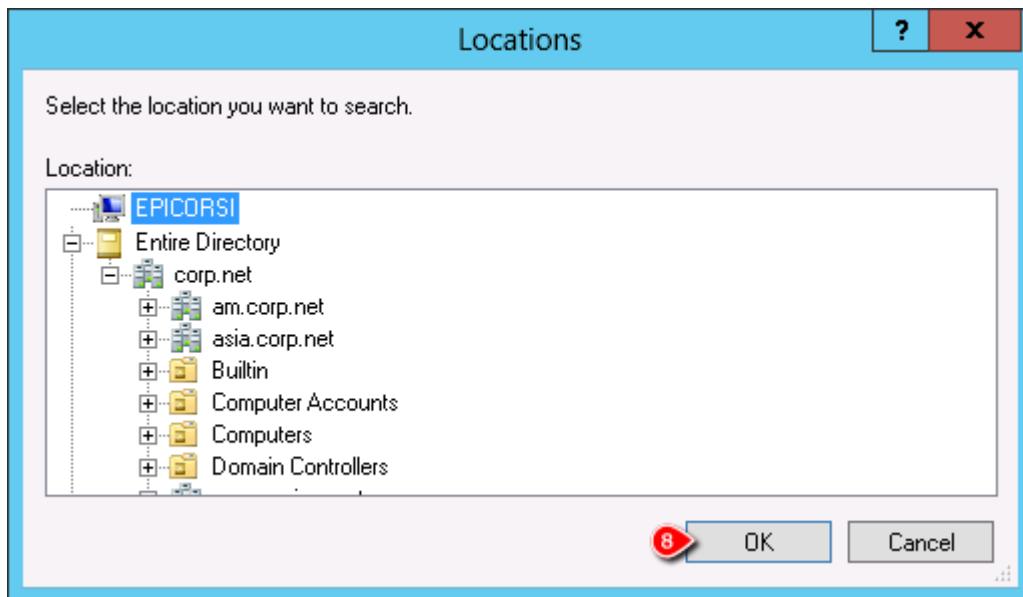
2. Select **Import Windows Users**.
3. The **Select Users** window displays.



4. Click the **Object Types** button to find and select users.

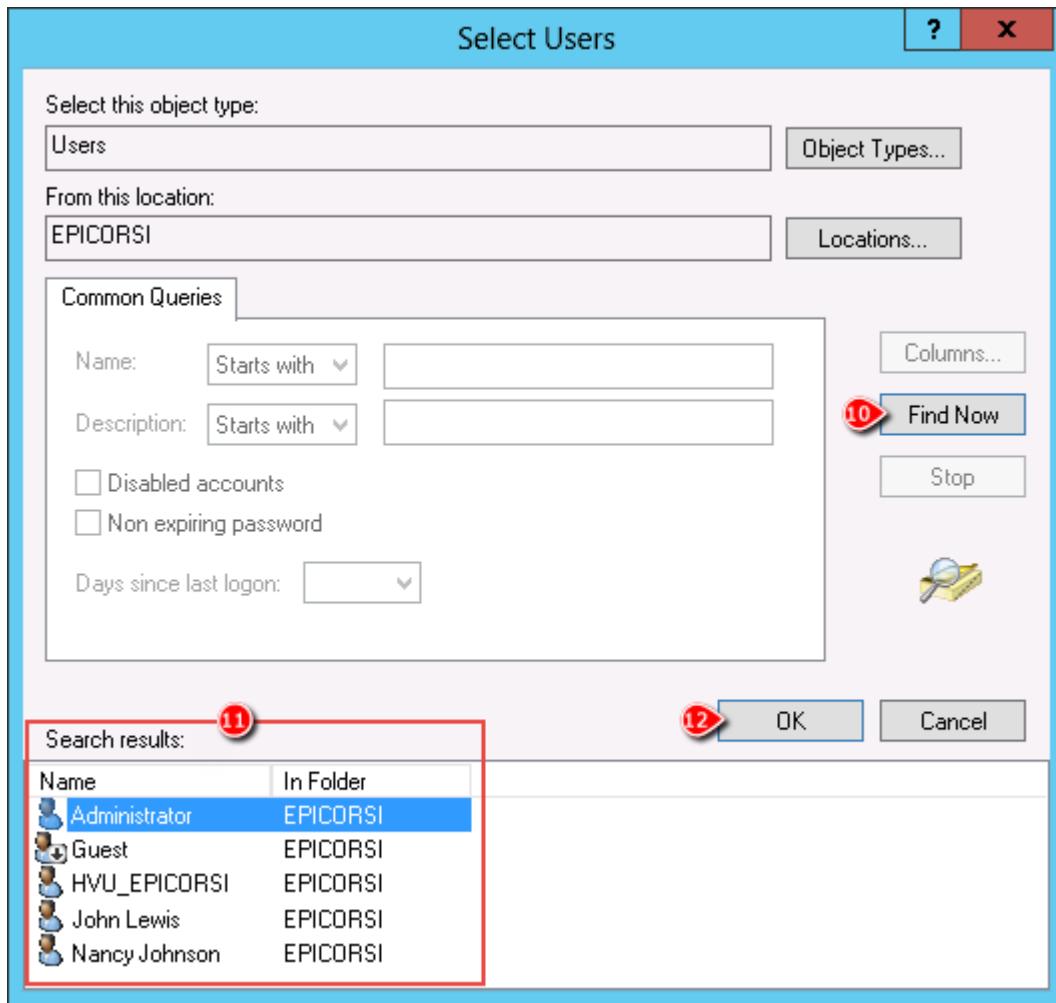


5. In the **Object Types** window, verify the **Users** check box is selected.
6. Click **OK**.
7. Click the **Locations** button to find and select the object type location.

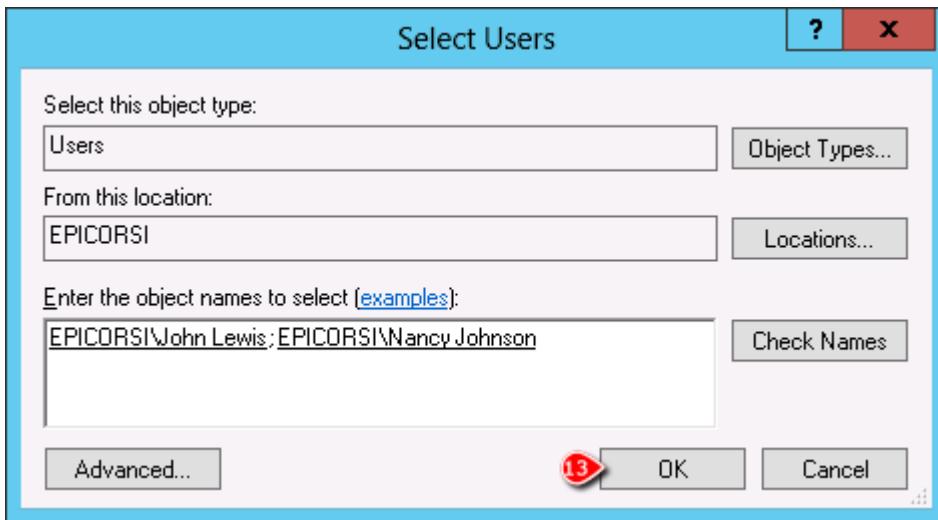


8. In the **Locations** window, select the location and click **OK**.
9. Click the **Advanced** button to find and select the object names.
You can also **Enter the object names to select** in the field.

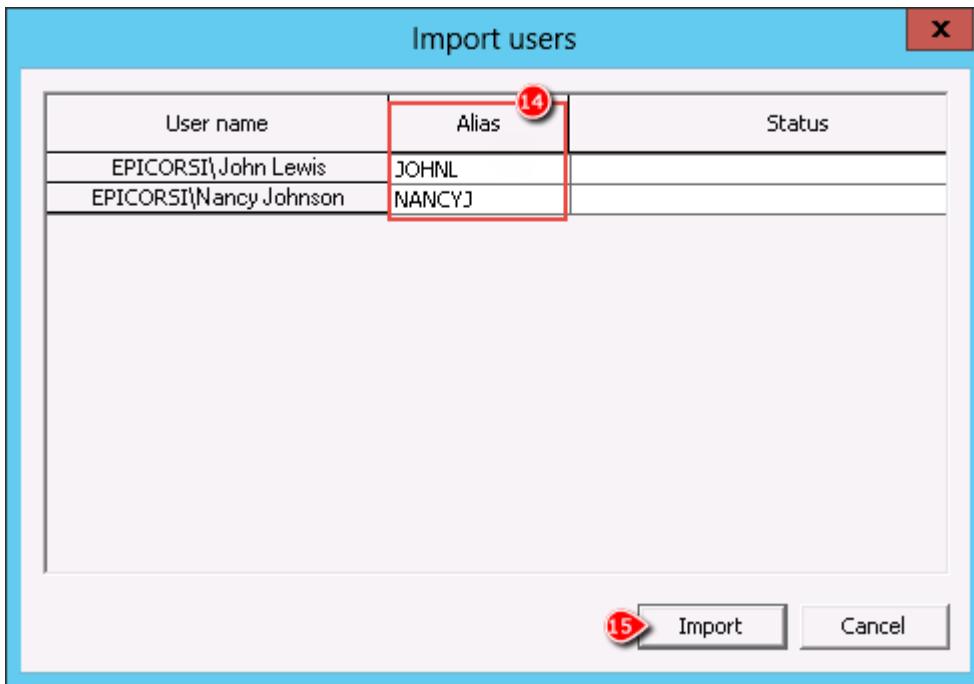
10. After you click the **Advanced** button, on the **Select Users** search form, click the **Find Now** button to find available users.



11. From the **Search results** grid, select the users you need.
12. Click **OK**.
13. Your selected users display. To confirm your selection, click **OK**.



14. In the **Import users** window, enter an **Alias** for the users.

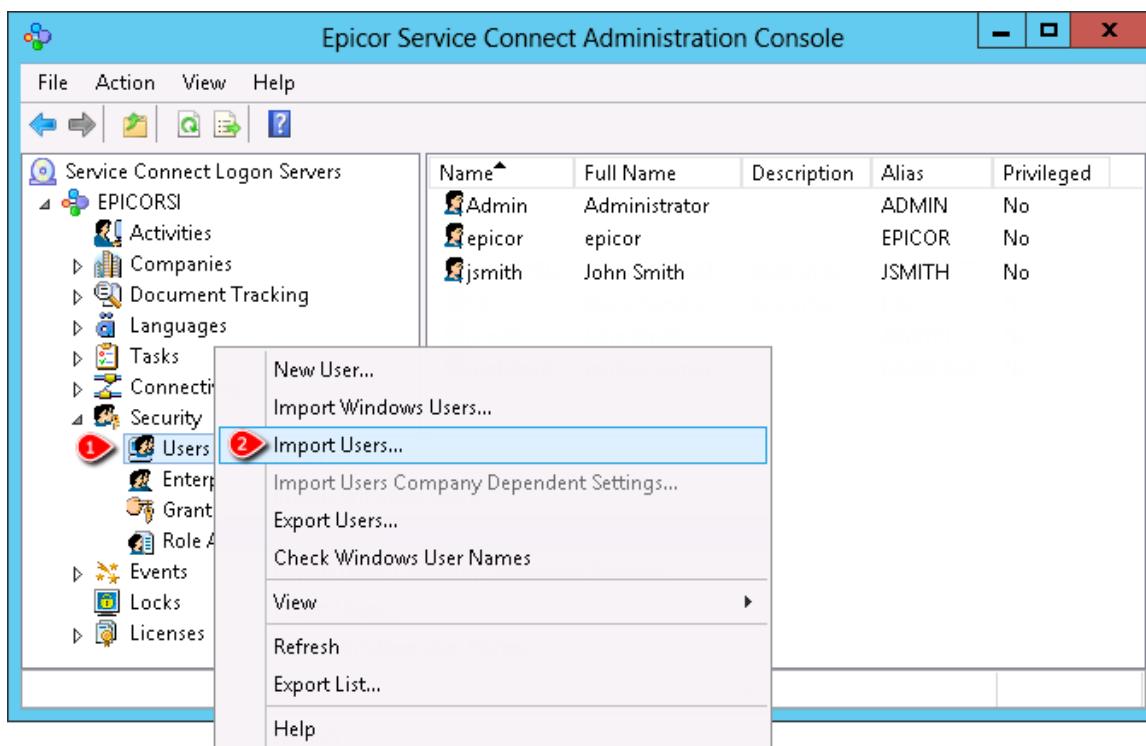


15. Click **Import**.

Import User Lists

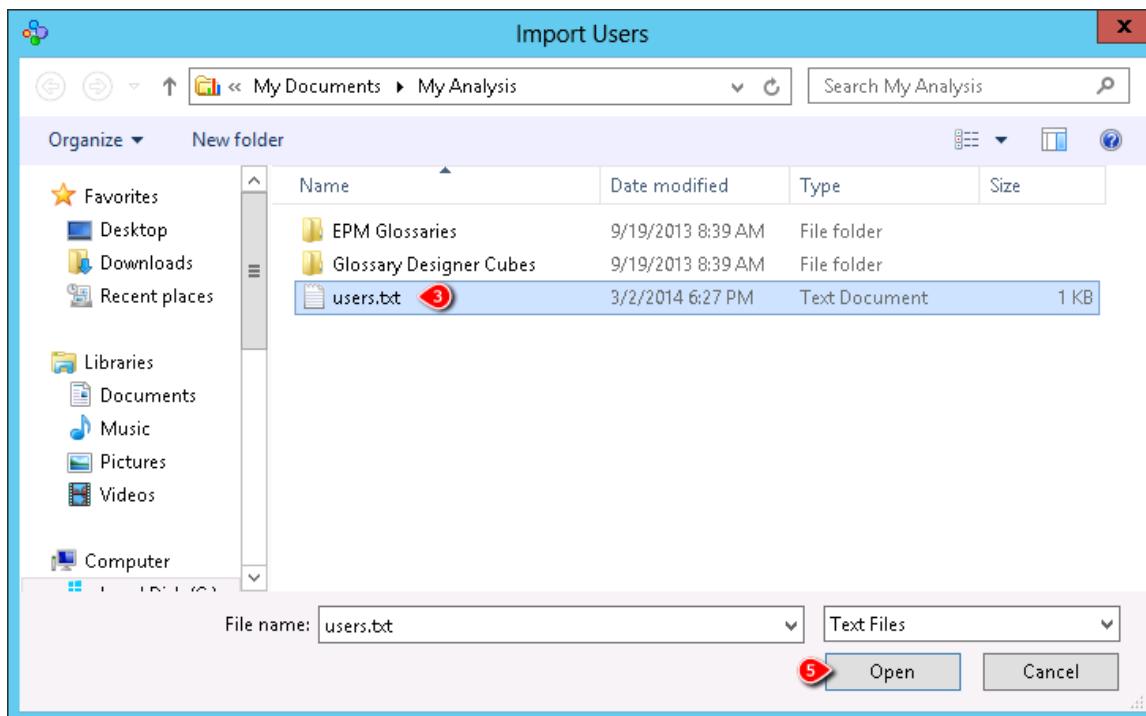
To import a list of users from a file:

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

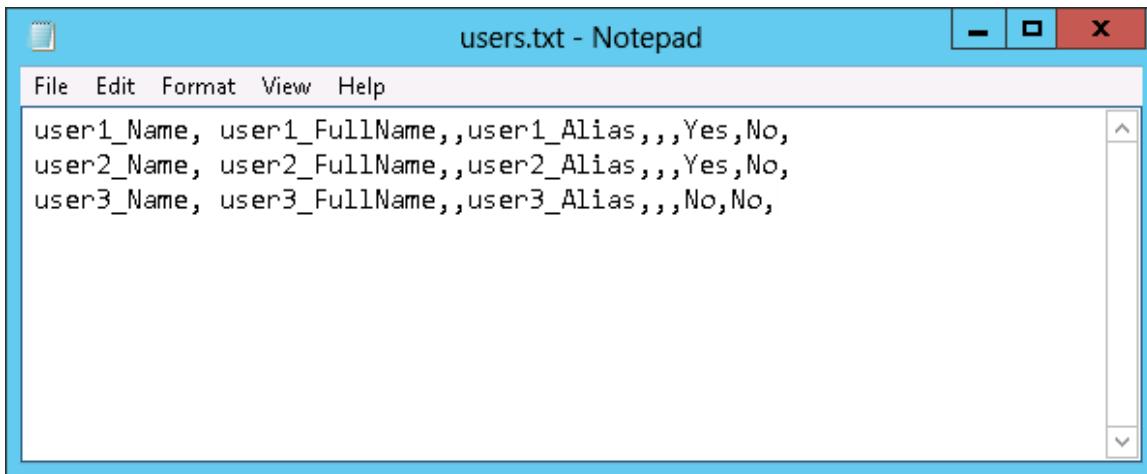


2. Select Import Users

3. Select the text file that contains the user list you want to import.



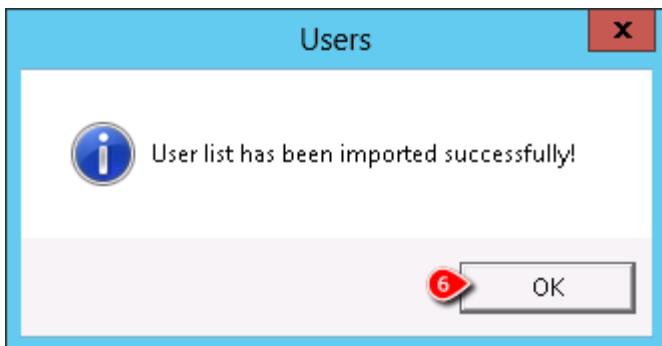
4. This example shows the imported file.



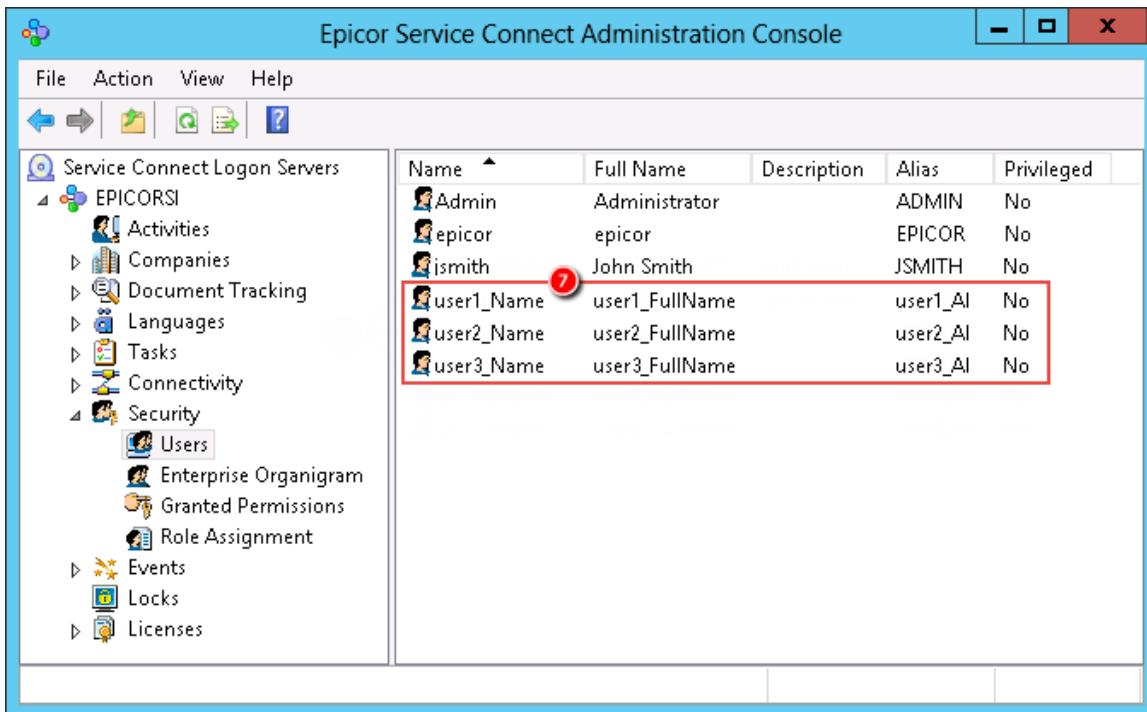
The screenshot shows a Windows Notepad window titled "users.txt - Notepad". The window contains the following text:

```
user1_Name, user1_FullName,,user1_Alias,,,Yes,No,  
user2_Name, user2_FullName,,user2_Alias,,,Yes,No,  
user3_Name, user3_FullName,,user3_Alias,,,No,No,
```

5. Click **Open**.
6. To the successful import notification message, click **OK**.



7. The imported users display in the right pane.



Service Connect Installation Management

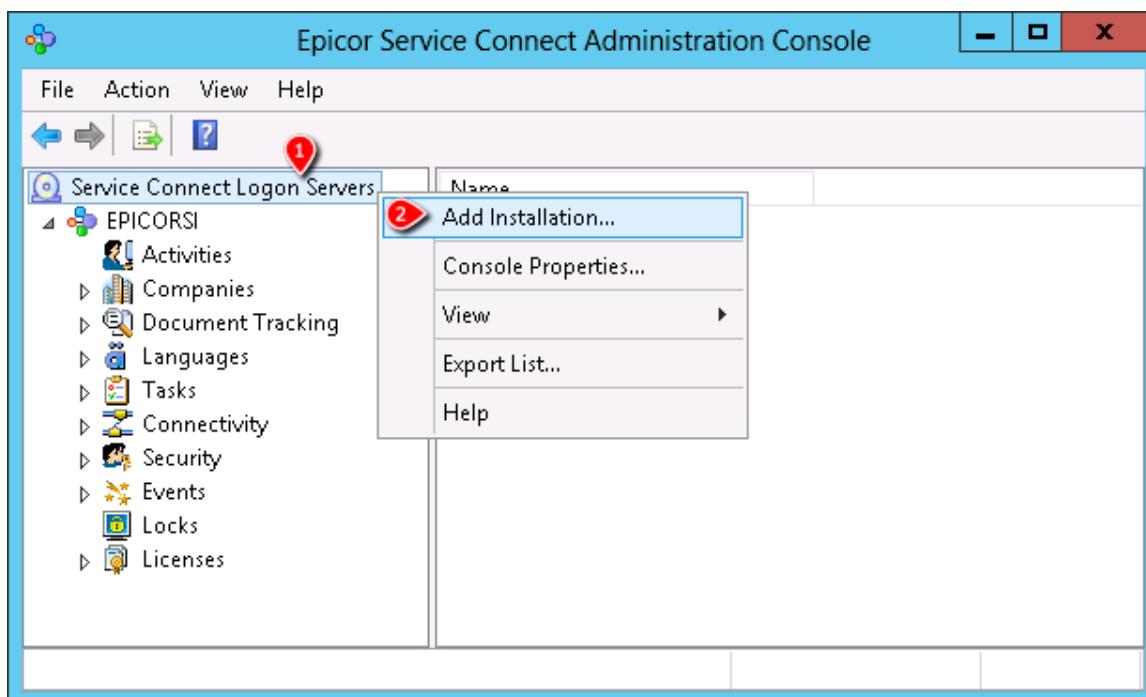
This section describes common installation-wide settings and procedures.

Register Servers

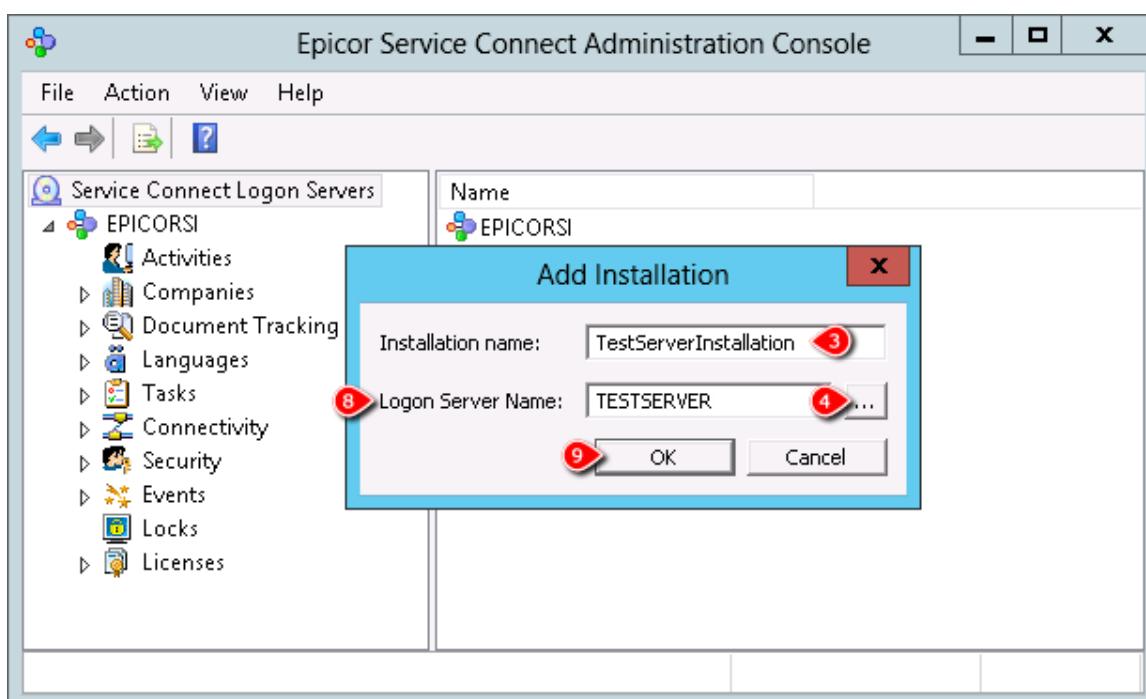
You can install Epicor Service Connect Administration Console on computers that are not the Service Connect server. You can then register one or more Service Connect installations to administer.

To register a Service Connect server:

1. In the Epicor Service Connect Administration Console **tree view**, right-click **Service Connect Logon Servers**.
2. Select **Add Installation**.

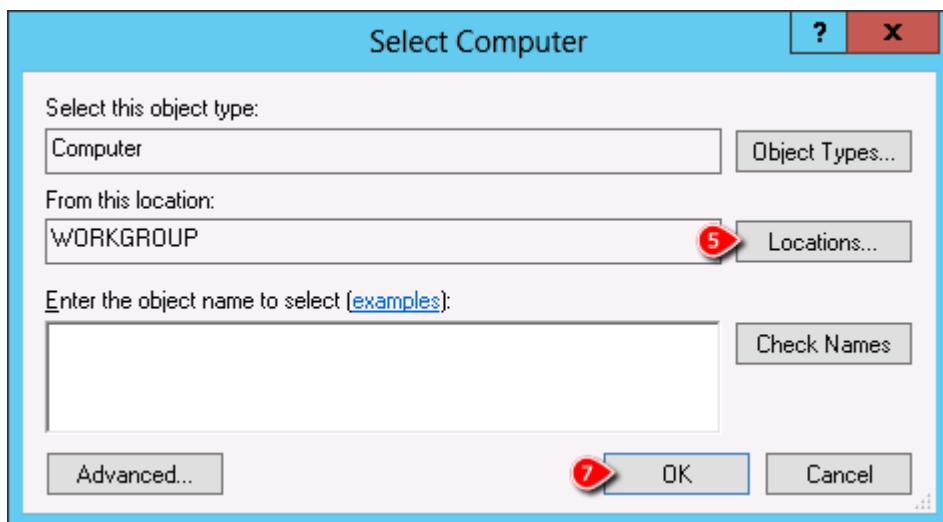


3. Enter an **Installation name.**

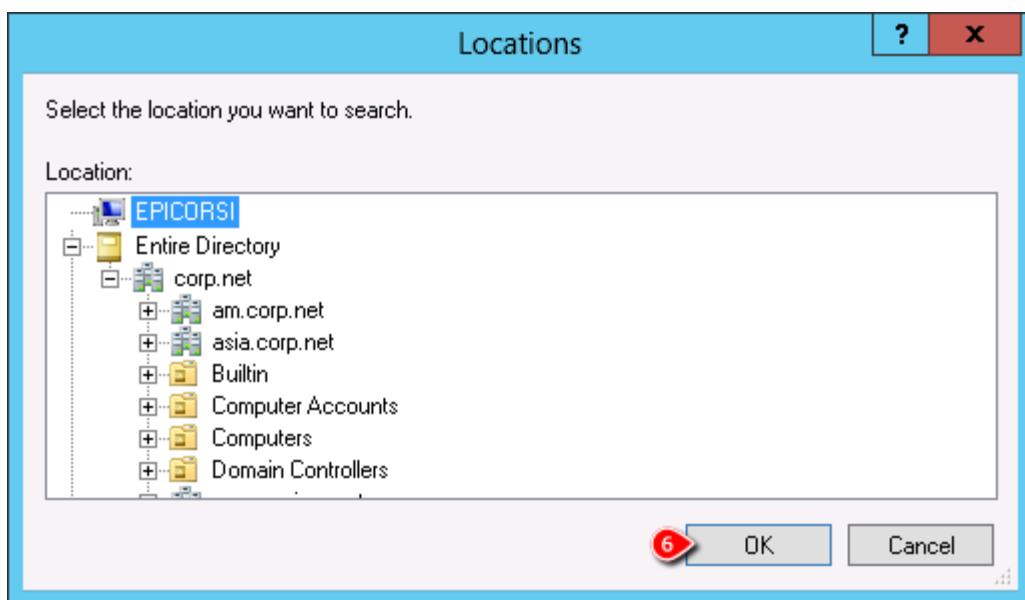


4. Next to the **Logon Server Name field, click the browse button.**

5. In the **Select Computer** dialog, click the **Locations** button to find and select the computer location.



6. Click **OK**.



7. In the **Select Computer** window, click **OK**.

8. Verify the **Logon Server Name** field displays the name of the server you selected.

You can alternatively enter the name of the computer where the Service Connect server is installed in the **Logon Server Name** field.

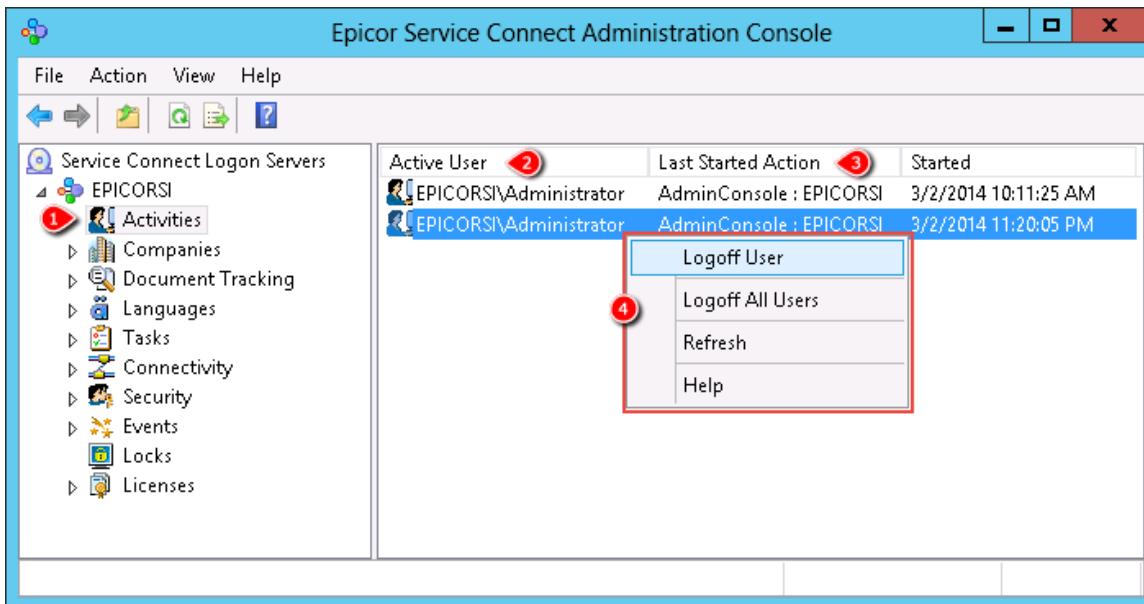
9. Click **OK**.

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. In the **Epicor Service Connect Administration Console** tree view, click **Activities**.
2. In the right pane, review **Active User**s.



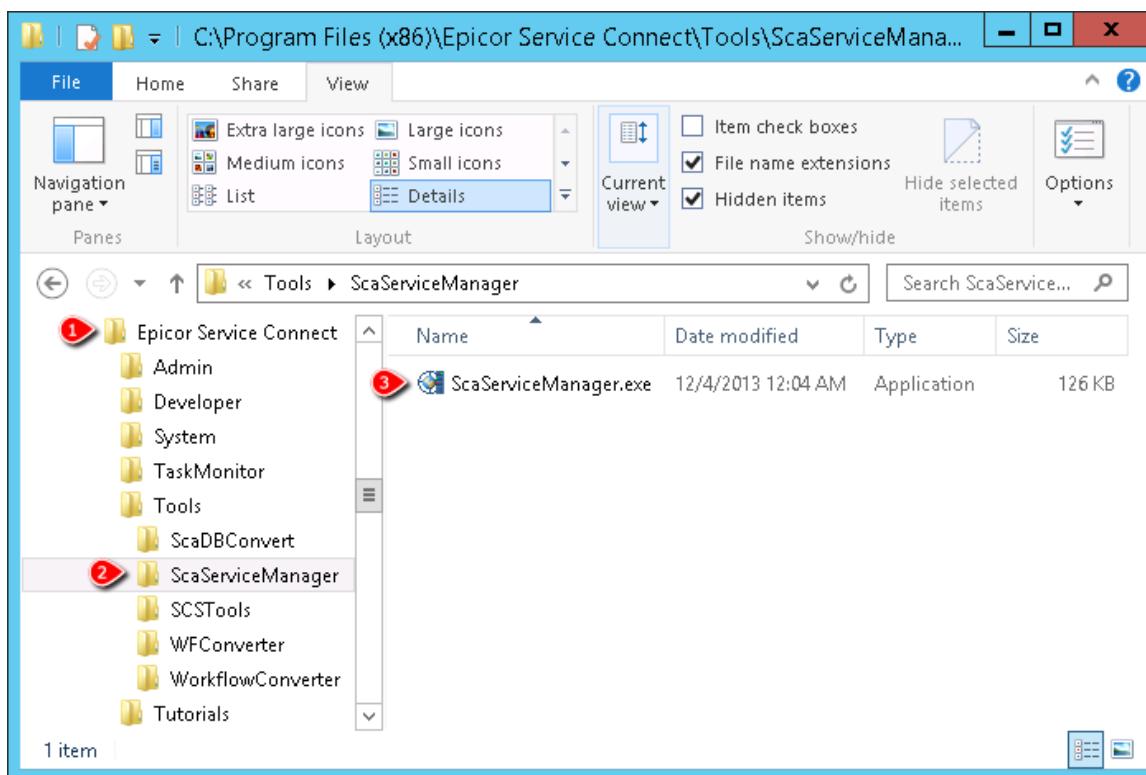
3. For each user, their **Last Started Actions** is displayed.
4. Right-click a user to display the user session management options:
 - **Logoff User**
 - **Logoff All Users**
 - **Refresh**
 - **Help**

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.



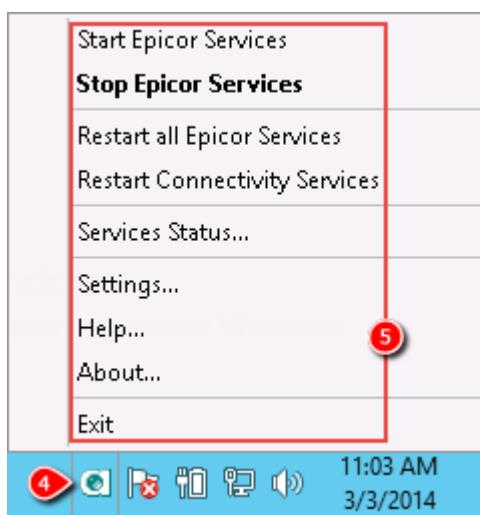
2. Browse to the **ScaServiceManager folder.**

The default path is: C:\Program Files (x86)\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.**

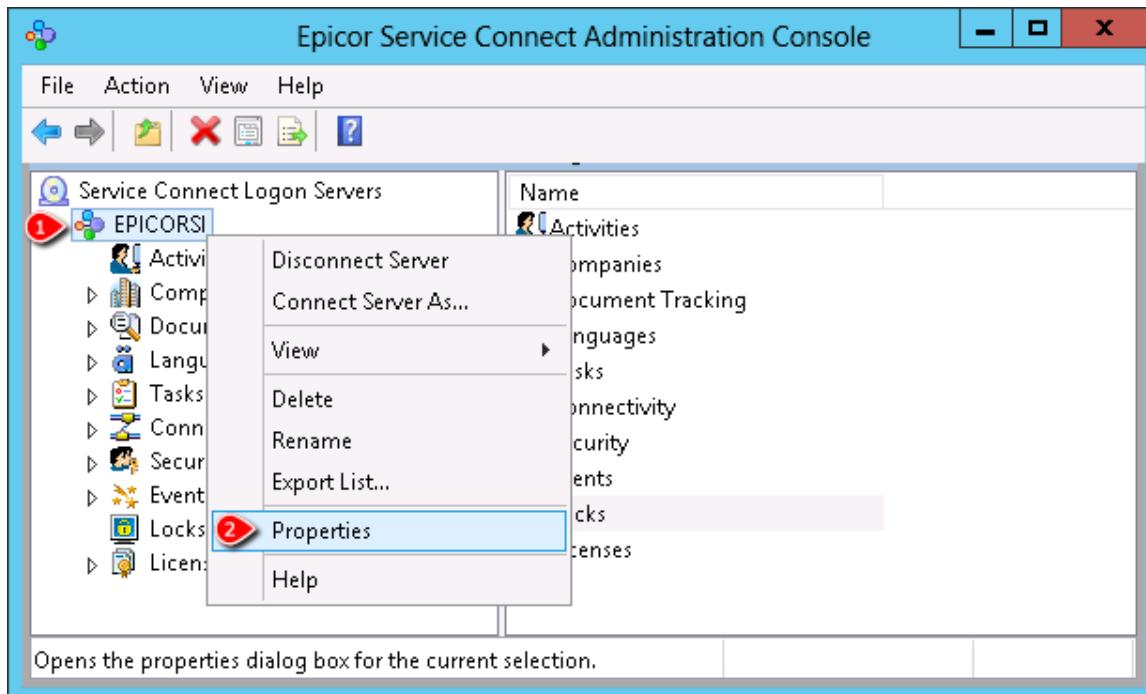
5. To access the menu, right-click the icon and select from the options.



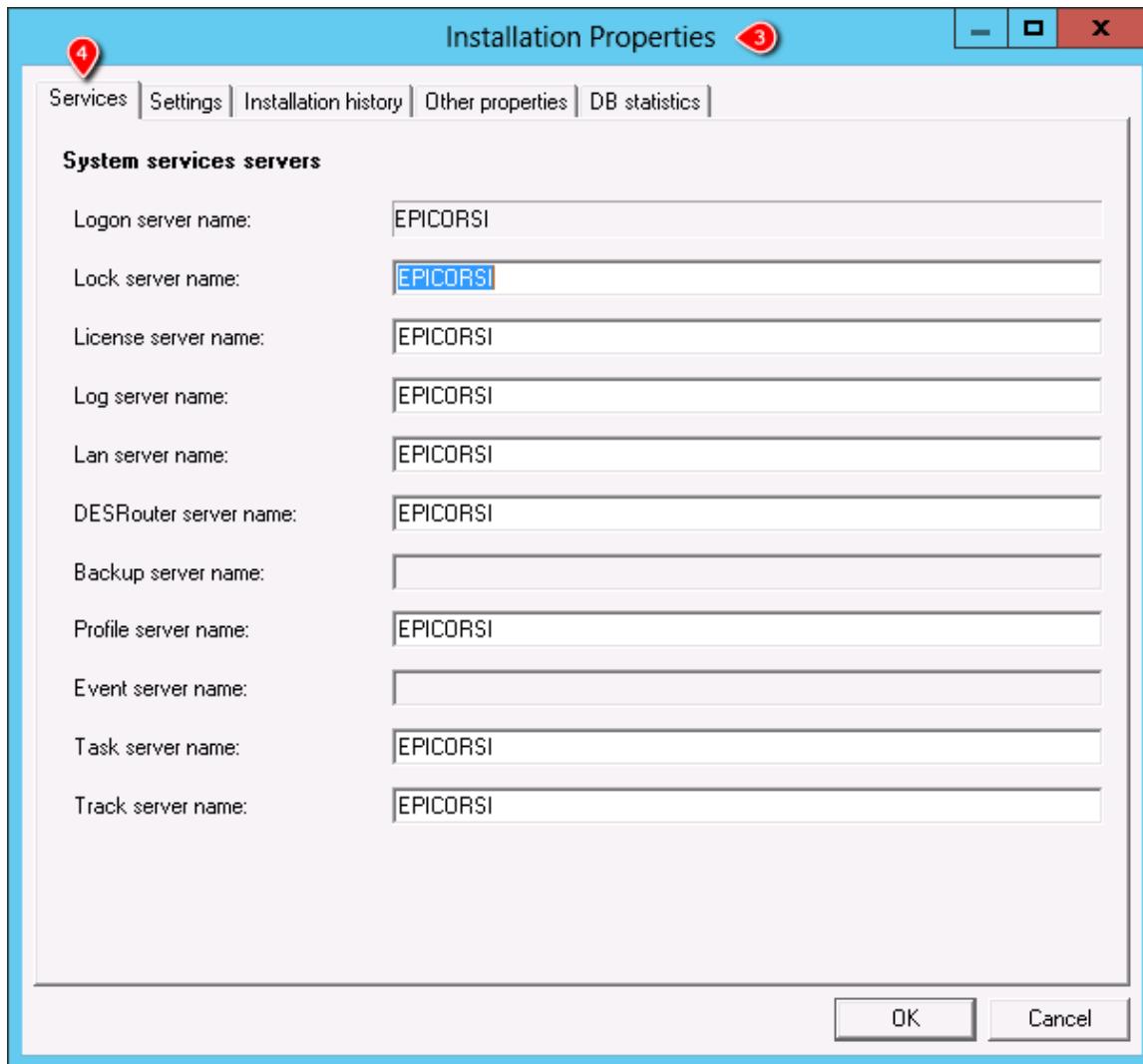
You can check statuses, stop Epicor Services, start Epicor Services, or restart the Service Connect Windows services. You can also select the Settings menu option to set options for the Service Manager utility.

View Installation Properties

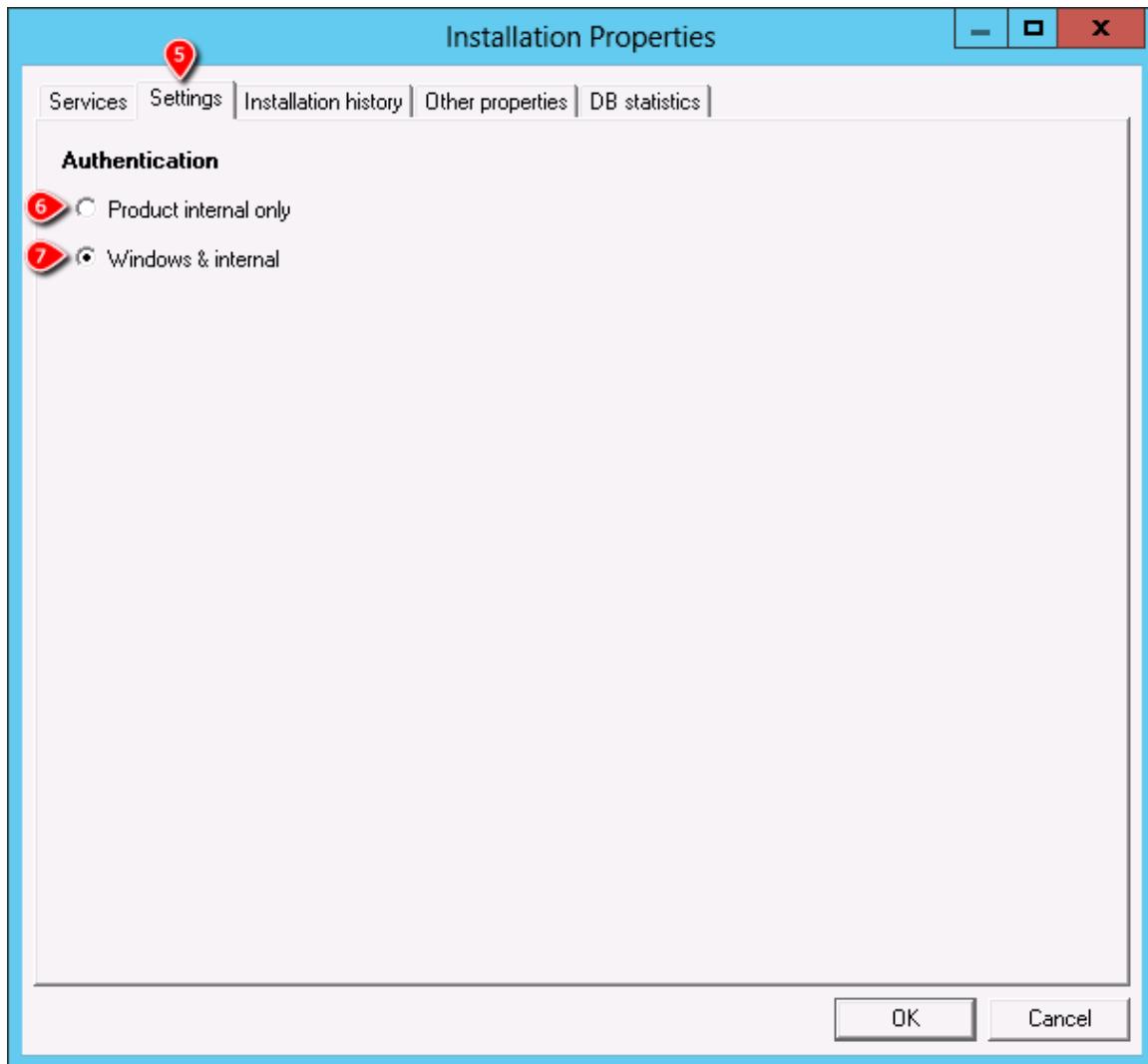
1. To display the installation properties, in the **tree view**, right-click the main node (server name).



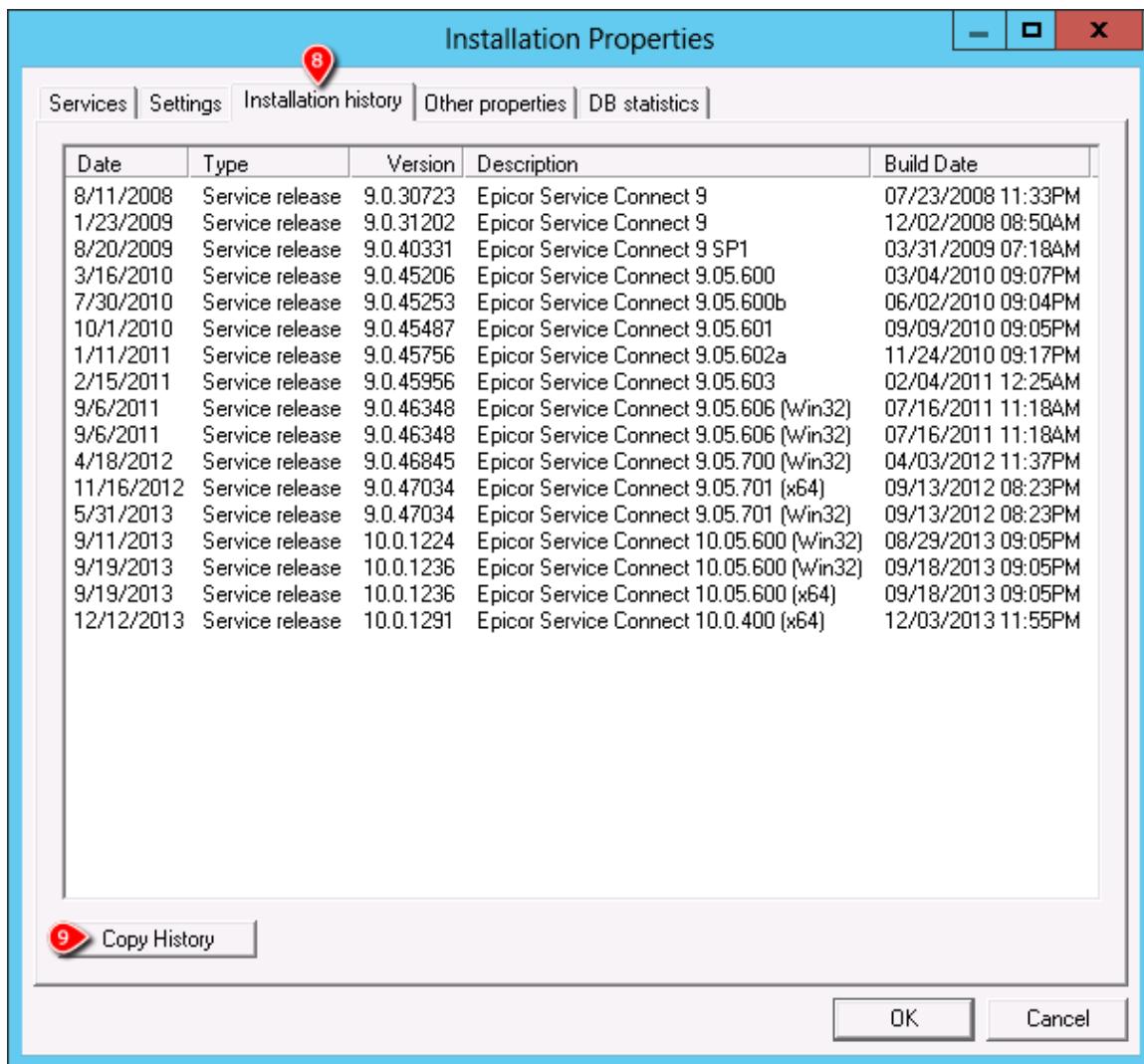
2. Select **Properties**.
3. The **Installation Properties** window displays.



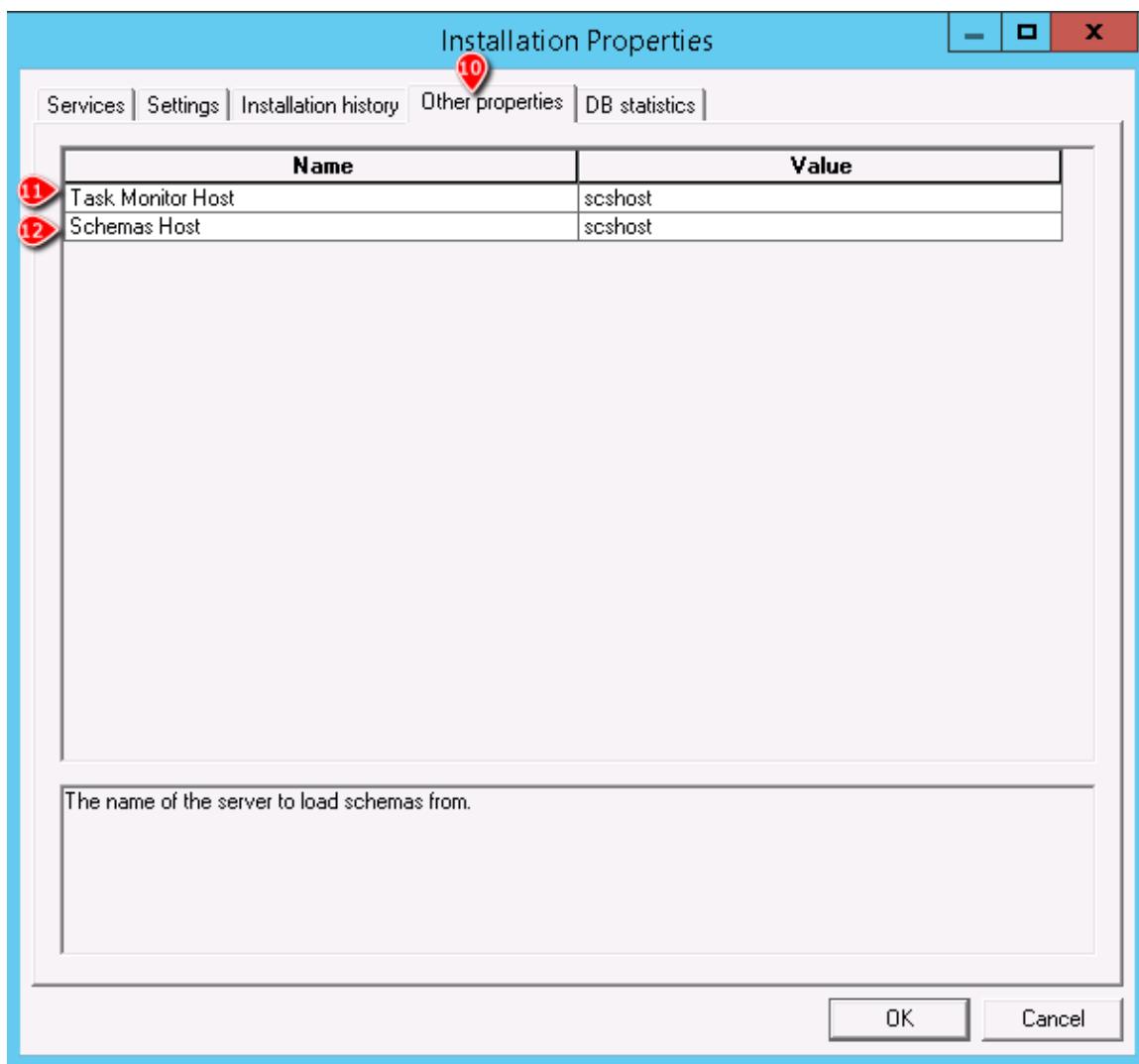
4. Use the **Services** tab to set system services servers to different server names.
5. Use the second tab, **Settings**, to select the **Authentication** type.



6. If you want users to enter their credentials to log in to ESC Administration Console and Workflow Designer, select the **Product internal only**.
7. 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.
8. Use the **Installation history** tab to track the succession of your Service Connect installations.



9. To copy the grid content to the clipboard, click the **Copy History** button.
10. Use the **Other Properties** tab to view and adjust the Service Connect installation properties.



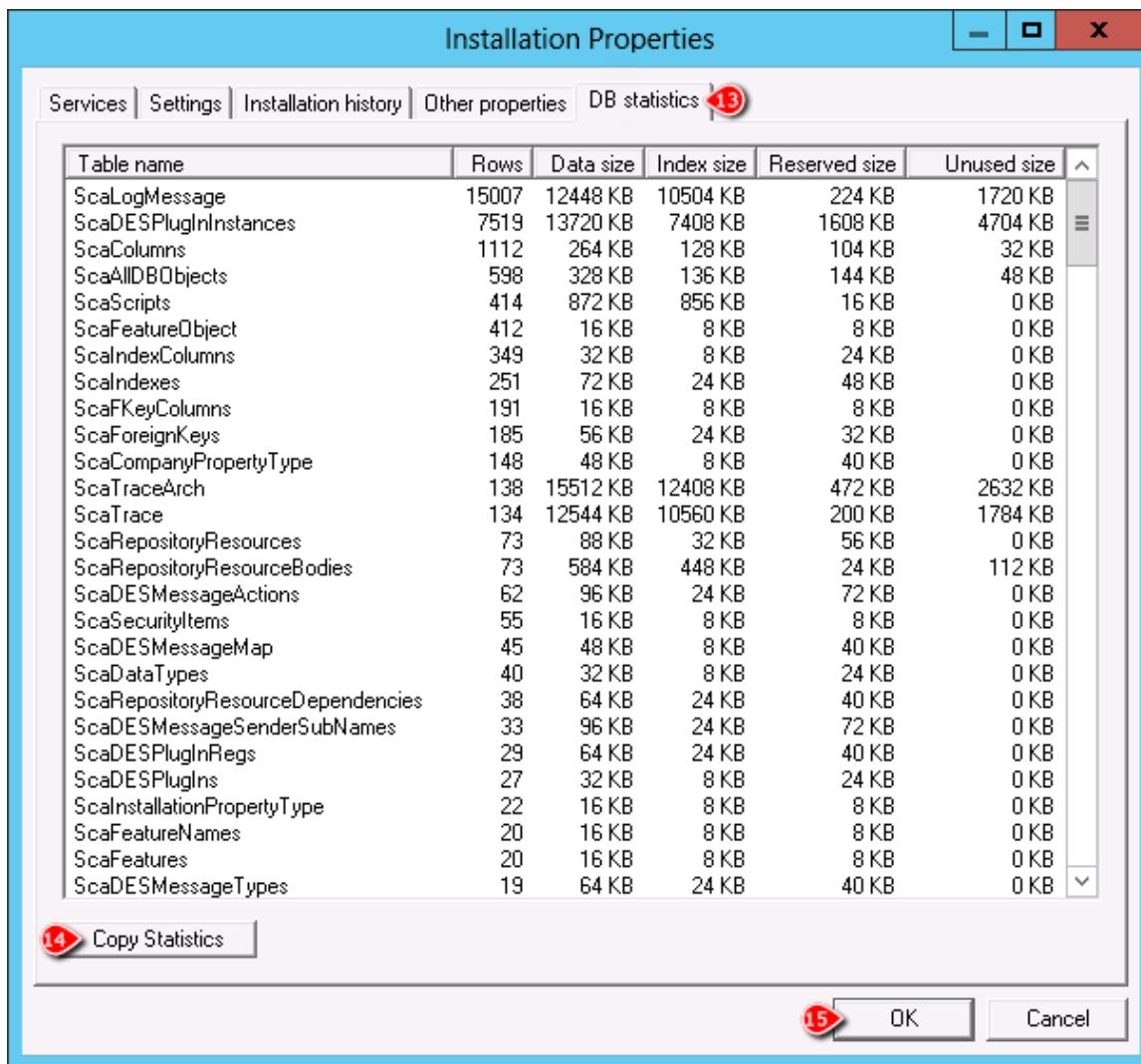
11. In the **Task Monitor Host** row, view the name of the server hosting Task Monitor. The default value is scshost.

The Task Monitor host must be recognized as one either belonging to the Local Intranet Zone by the Internet Explorer installed on the client workstation or to any zone where Automatic logon with the current username and password is enabled in the Internet Explorer on the client workstation.

When Task Monitor Host is defined, it is used to load Task Monitor from within Service Connect. If it is not defined, the default scshost reference is used. In the latter case, scshost must be resolvable from the client workstation where Task Monitor is attempted to be run either through DNS or hosts file.

Existing customized menus may need to be updated to refer to [http://<TASKMONITORHOST>/TaskMonitor/...](http://<TASKMONITORHOST>/TaskMonitor/) instead of [http://scshost/TaskMonitor/...](http://scshost/TaskMonitor/) for Task Monitor references.

12. In the **Schemas host** row, view the location from which schemes are loaded if scshost is not listed in the hosts file in the \\WINDOWS\system32\drivers\etc folder.
13. Use the **DB Statistics** tab to view Service Connect database statistics and analyze space allocation for all tables (Document Tracking, Task, Event) in ESC DB.



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

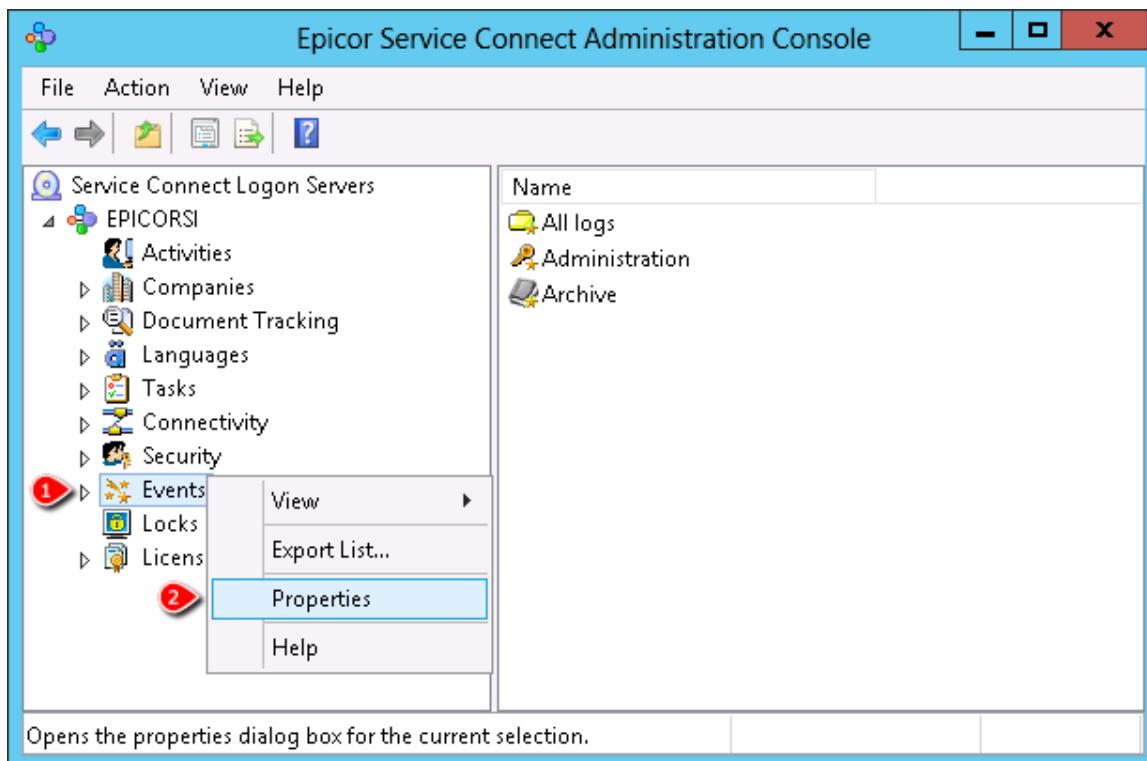
15. Click **OK**.

Use Event Logs

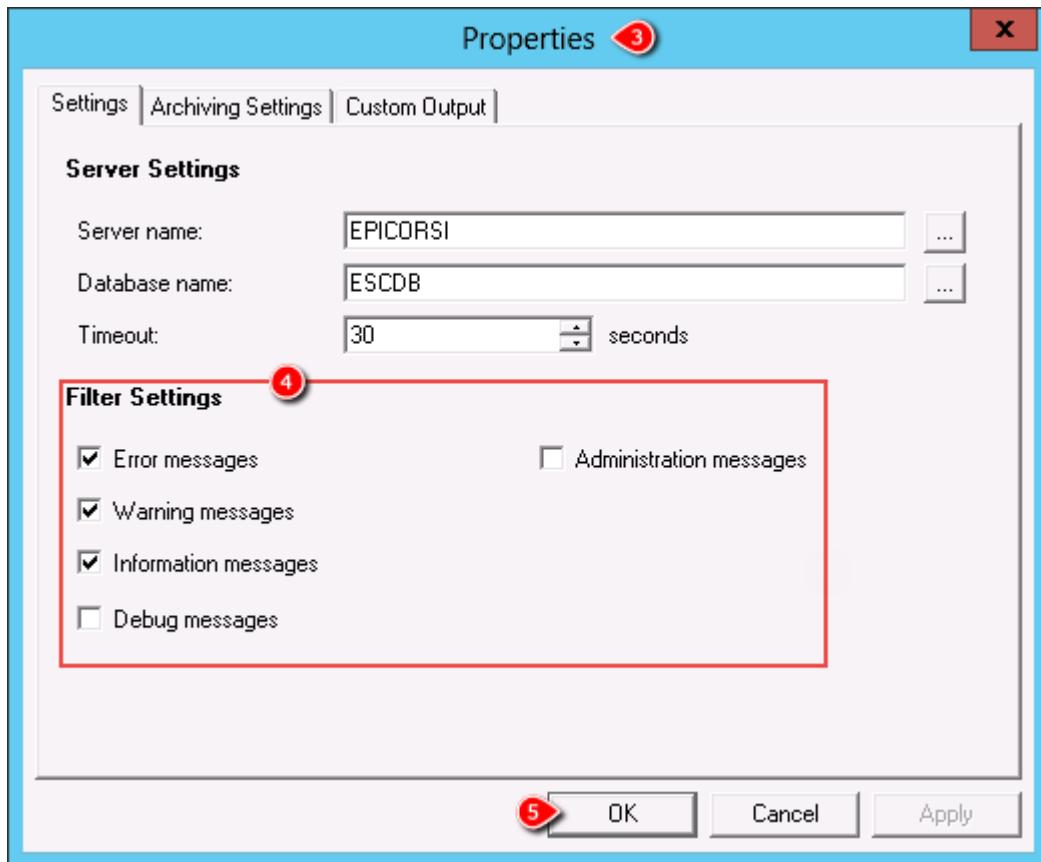
In Epicor Service Connect Administration Console, under the Events node, you can view a Service Connect events log.

To define which events are logged:

1. In the **tree view**, right-click the **Events** node.



2. Select **Properties**.
3. The **Properties** window displays.
4. In the **Filter Settings** section, select which type of messages to include in the log.
Selecting **Debug messages** can be helpful when you develop a workflow.

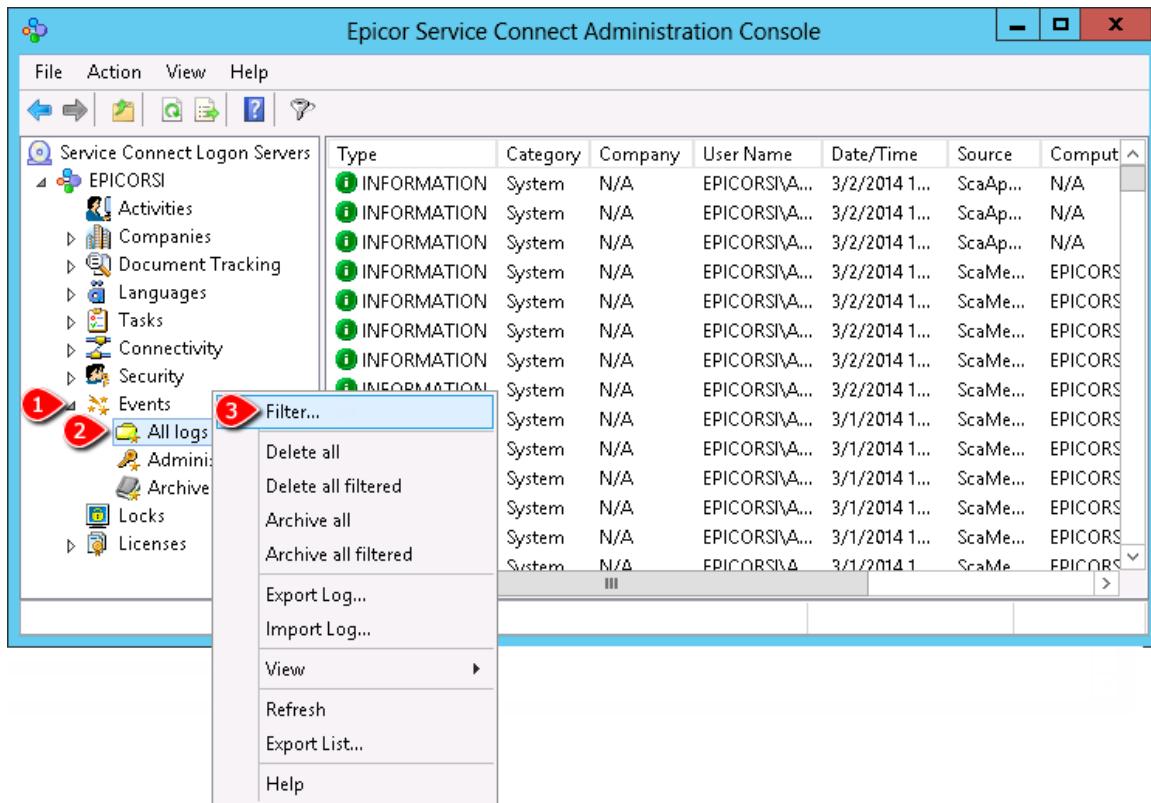


5. Click OK.

Filter Log Entries

To establish filter parameters:

1. In the **tree view**, expand the **Events** node.



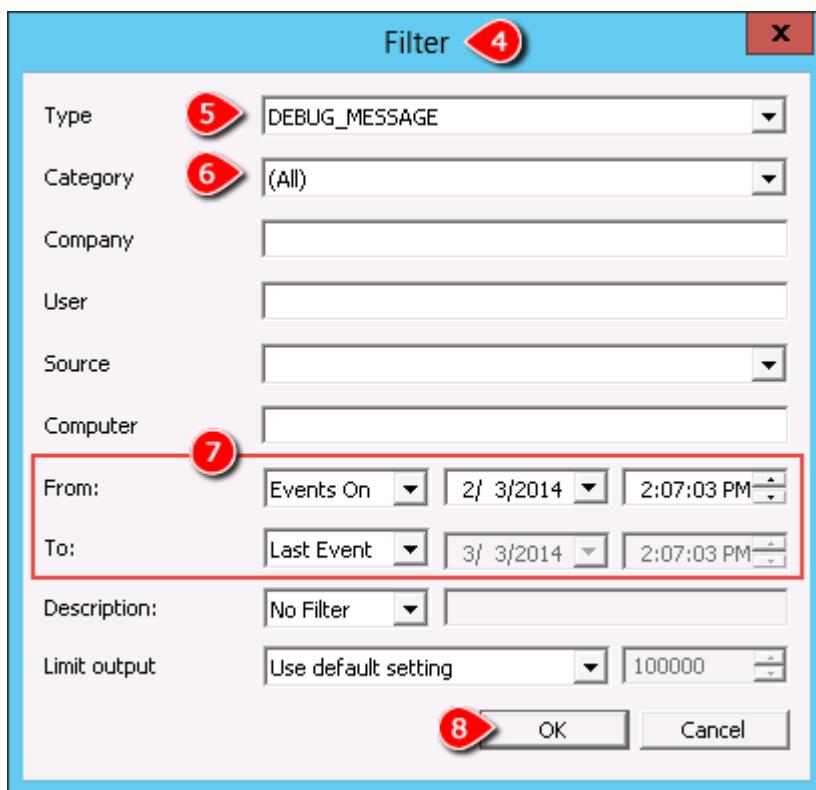
2. Right-click **All logs**.

3. Select **Filter**

4. The **Filter** window displays.

5. Select a **Type**.

For example, select DEBUG_MESSAGE to see only the corresponding entries.



6. Optionally, select a **Category**.
7. Use the **From** and **To** fields to set the time period when the debug messages were logged.
8. 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.

Manage Licenses

Before you start to use Epicor Service Connect, verify your license is valid and view the list of all licensed functionalities.

In Epicor Service Connect Administration Console, under the **Licenses** node, you can view data on licensed functionalities and their license files. You can also import and manage Epicor Service Connect licenses in this node.

Licenses include System and Business Features.

To easily see what functional items are licensed:

1. In the **tree view**, expand the **Licenses > Features > System Features** or **Licenses > Features > Business Features** node.

Epicor Service Connect Administration Console				
File Action View Help [Minimize] [Maximize] [Close]				
[Back] [Forward] [Home] [Search] [Print] [Help] [Sort Up] [Sort Down] [Filter]				
Service Connect Logon Servers	EPICORSI	Description	Value	Expiration Date
Activities		.NET References Support	True	4/3/2014 7:00:00...
Companies		Any Database Server	True	4/3/2014 7:00:00...
Document Tracking		Communication Channel: COM	True	4/3/2014 7:00:00...
Languages		Communication Channel: FTP	True	4/3/2014 7:00:00...
Tasks		Communication Channel: File	True	4/3/2014 7:00:00...
Connectivity		Communication Channel: HTTP	True	4/3/2014 7:00:00...
Security		Communication Channel: IBM MQ Series	True	4/3/2014 7:00:00...
Events		Communication Channel: MSMQ	True	4/3/2014 7:00:00...
Locks		Communication Channel: SMTP	True	4/3/2014 7:00:00...
Licenses		Communication Channel: Sonic MQ Series	True	4/3/2014 7:00:00...
Features		Communication Channel: Windows Service Bus	True	4/3/2014 7:00:00...
Business Features	1	Data Exchange Server	True	4/3/2014 7:00:00...
System Features		English (UK)	True	4/3/2014 7:00:00...
Licenses		Epicor Service Connect Standalone	True	4/3/2014 7:00:00...
License Usage		Integrated Workflow Designer	True	4/3/2014 7:00:00...
		Multiple Installation Support	True	4/3/2014 7:00:00...
		Multiple server support	True	4/3/2014 7:00:00...

2. Click the **Description** column title to sort the features by the functionality name.

For example, you need to find out if Workflow Element: DB Operation is licensed. If it is not licensed or is expired, it will not appear in the list or its **Expiration Date** would be in the past.

3. In the **Value** field, the **True** value confirms you can use the functionality or a number associated with the license item displays.

The meaning of a numeric value depends on the license item – this can be the number of users allowed to work with the functionality, modes allowed for specific functionality and so on.

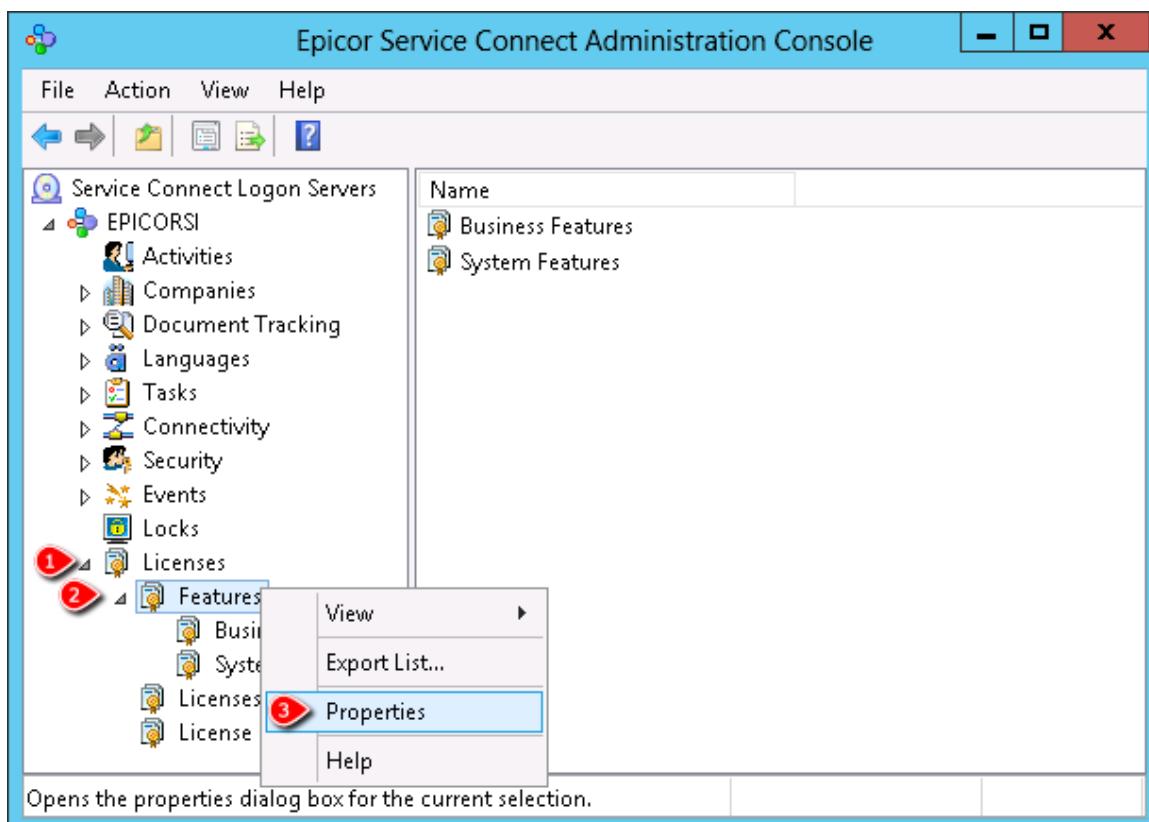
4. In the **Feature ID** field, view the unique identification number of the functionality.

5. In the **License File Name** field, view the name of the license file for the functionality.

View the License ID and Location

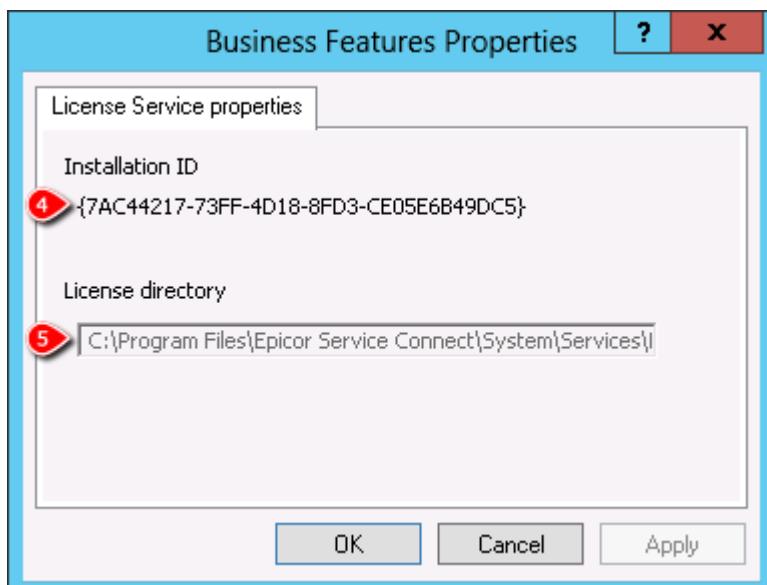
To view the identification number of the license file and the directory where it is located:

- In the **tree view**, expand the **Licenses** node.
- Right-click any folder under the **Licenses** node.



3. Select **Properties**.

4. In the **Feature Properties** window, view the **Installation ID**.

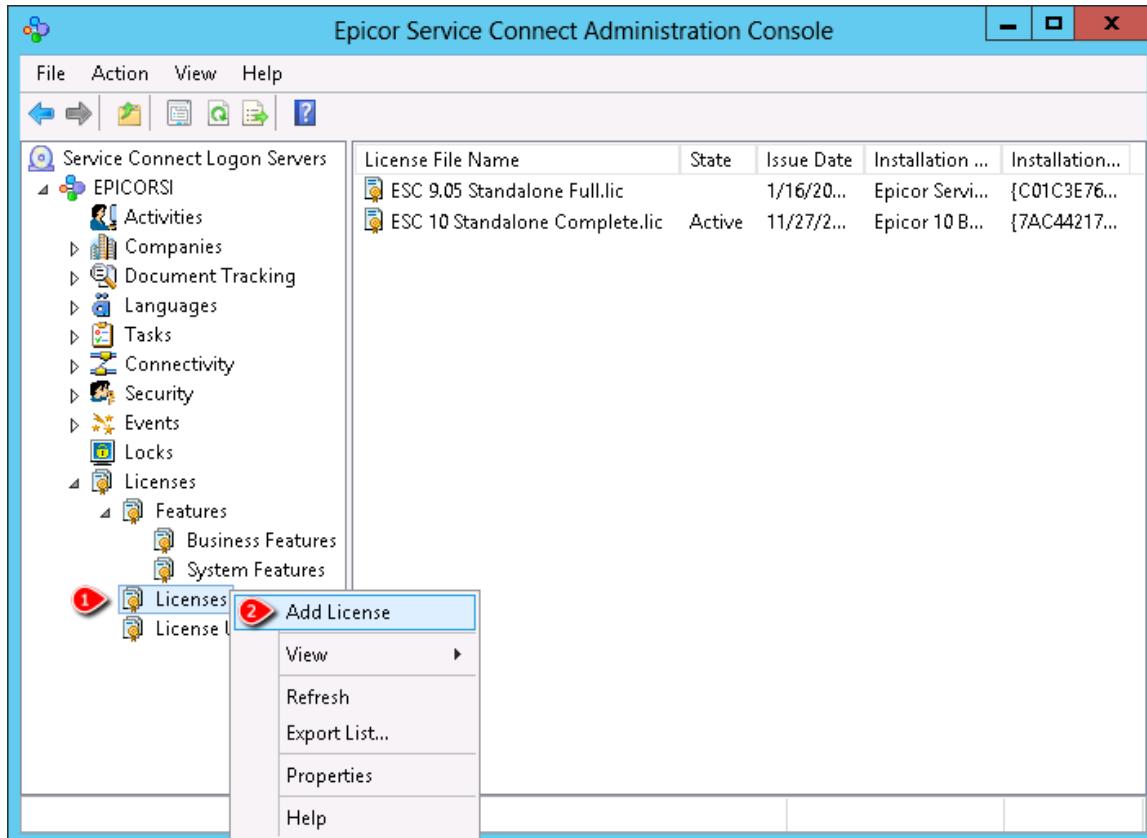


5. In the **License directory** field, view the license file location.

Import License

To import a license:

1. In the **tree view**, right-click the **Licenses > Licenses** node.



2. Select **Add License**.
3. In the standard **Open** window, browse to and select the license file.
4. Click **Open**.

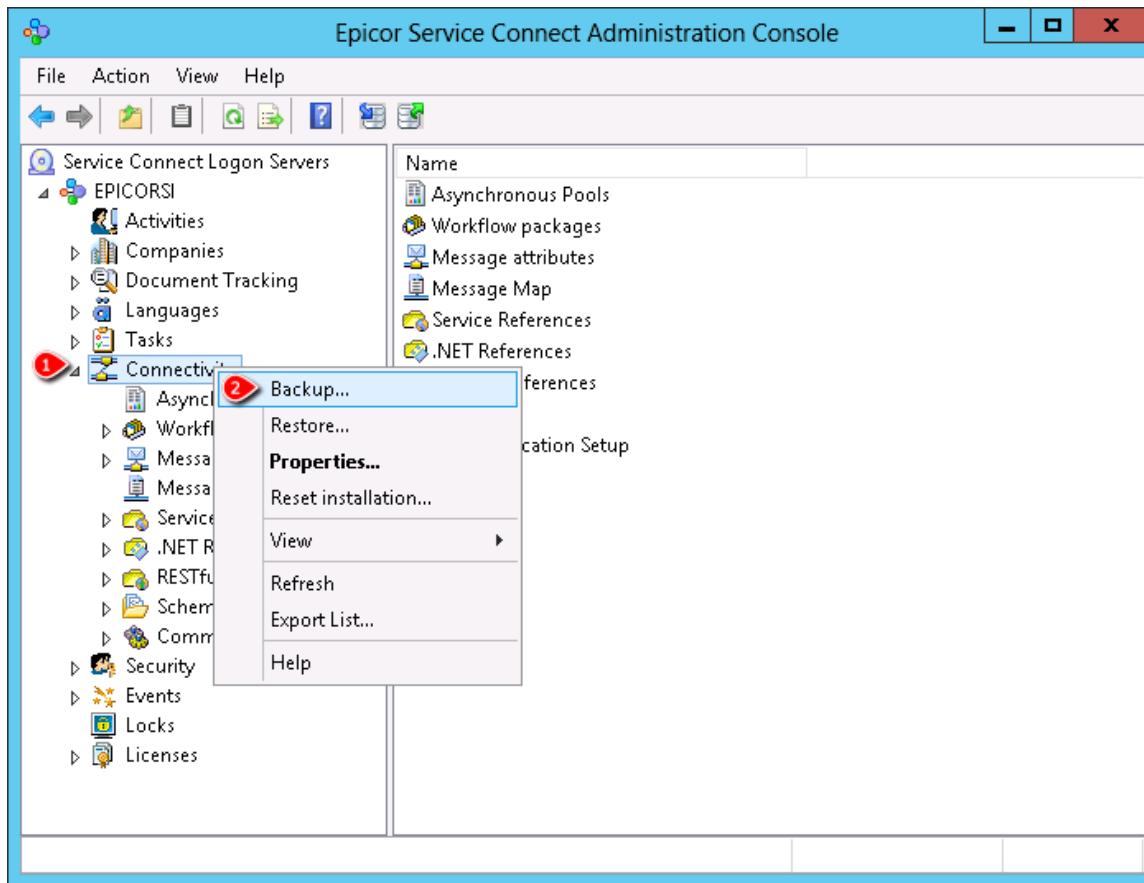
Connectivity Administration

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 the Connectivity Components chapter.

Create a Backup

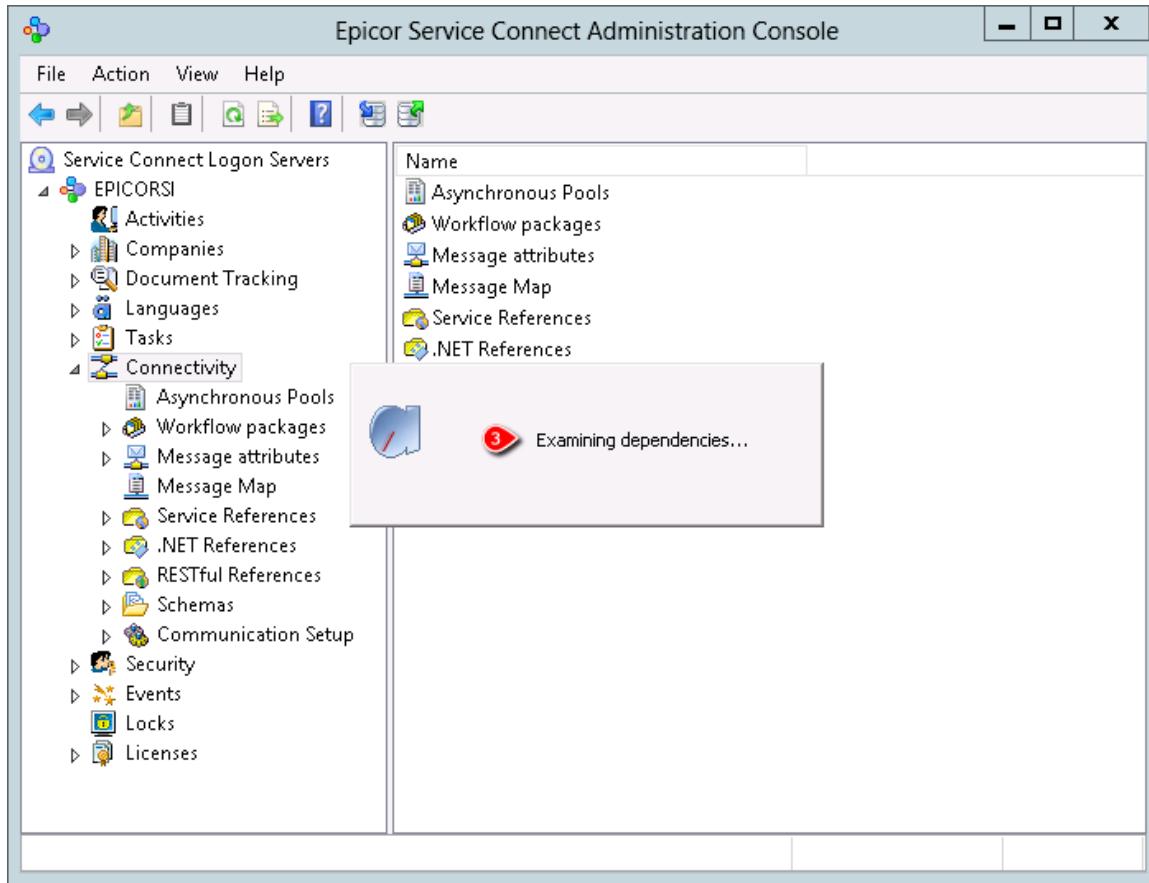
To backup connectivity settings:

1. In the **tree view**, right-click the **Connectivity** node.



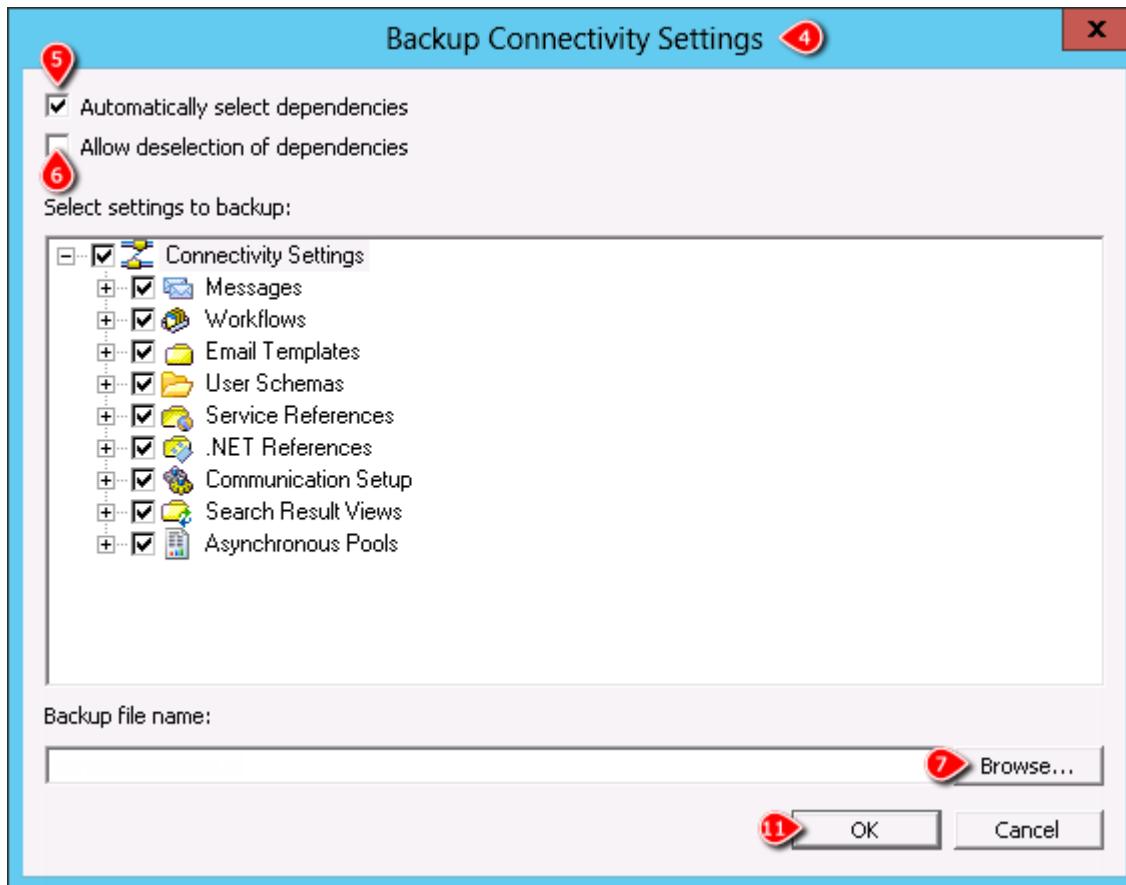
2. Select **Backup**.

3. When the **Examining dependencies** message is displayed, wait while Service Connect verifies that every item that can be backed up is backed up with all its references.



For example, if you backup a workflow process, which includes a Poster and a Web Method element, two more items will be selected for backup - Output Channel referenced by the Poster, and Web Reference referenced by the Web Method element.

4. The **Backup Connectivity Settings** window displays.



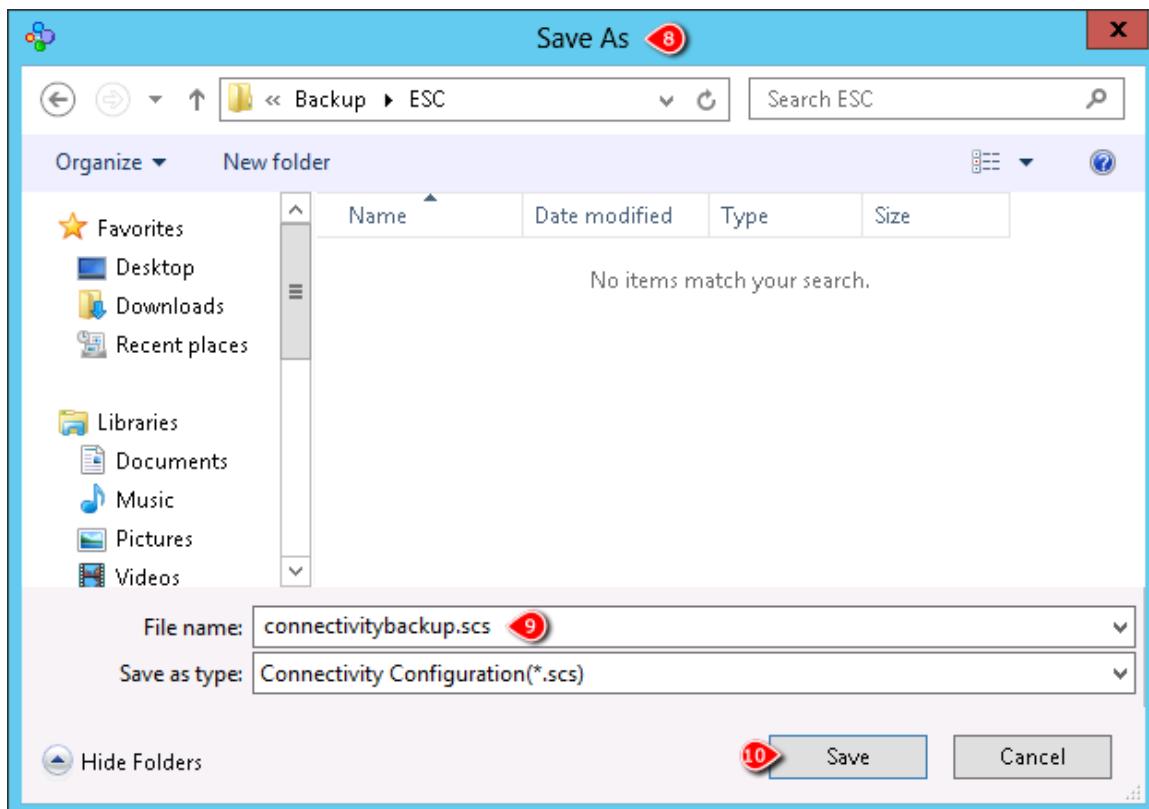
5. Leave the **Automatically select dependencies** check box selected.

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.

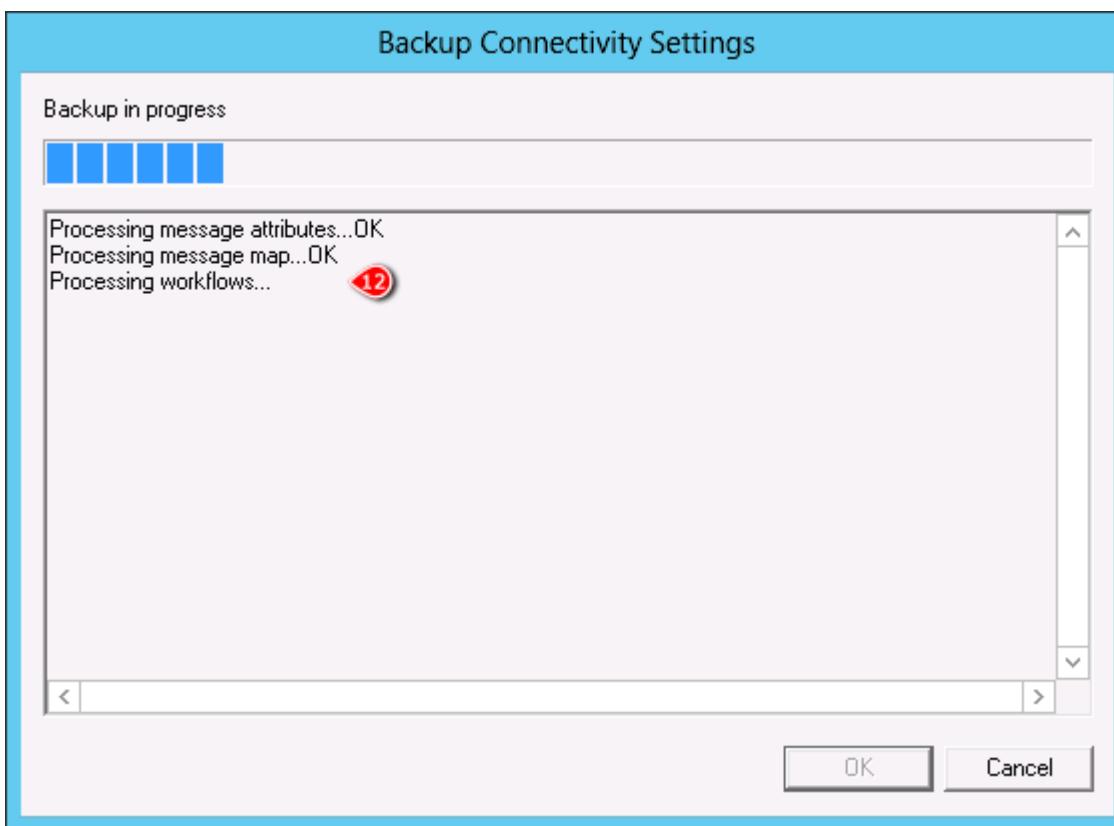
6. Select **Allow deselection of dependencies** if you want the ability to select or clear any item regardless of its dependencies.

If you select both options, when you select an object with a dependency, the dependency will also be selected until you manually clear it.

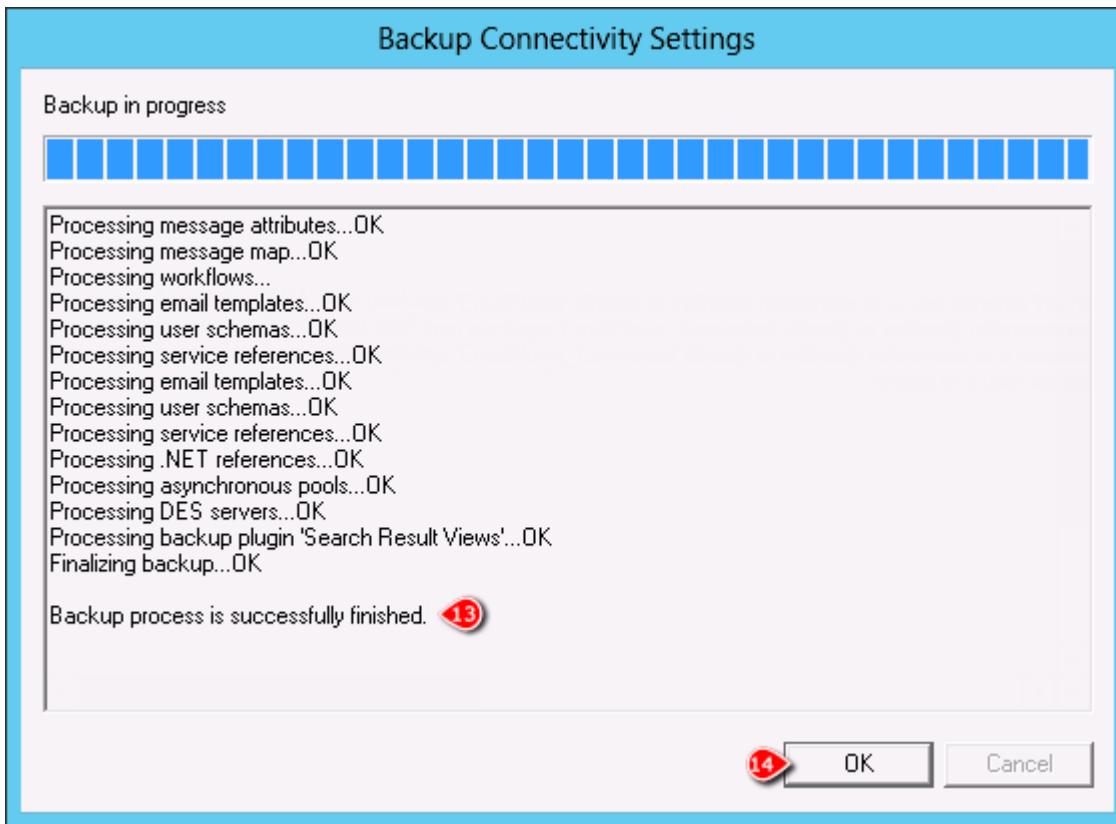
7. Click **Browse**.



8. In the standard **Save As** dialog, navigate to the folder where you want to save the backup file.
9. Enter a filename for the backup file.
10. Click **Save**.
11. Click **OK**.
12. The **Backup Connectivity Settings** window displays the backup progress.



13. Once the backup process is complete, the **Backup Connectivity Settings** window displays the **Backup process is successfully finished** message.



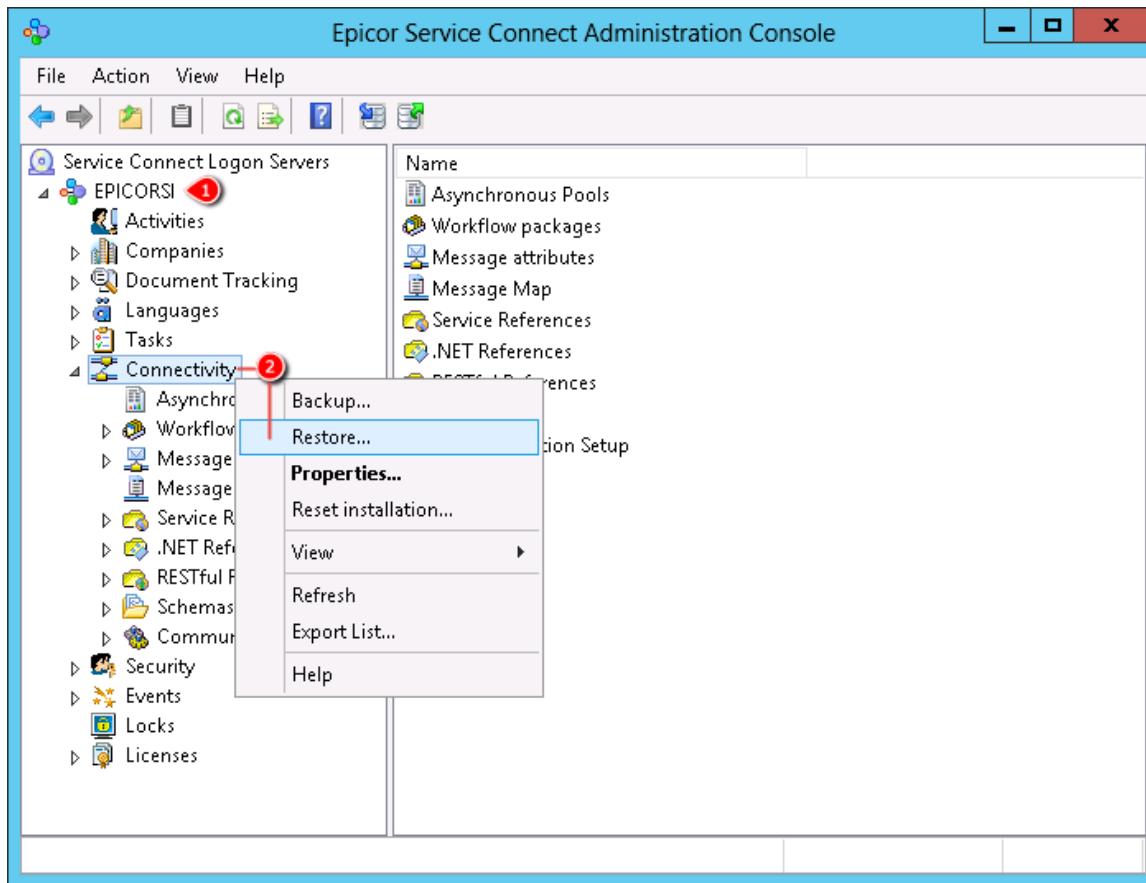
14. Click **OK**.

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

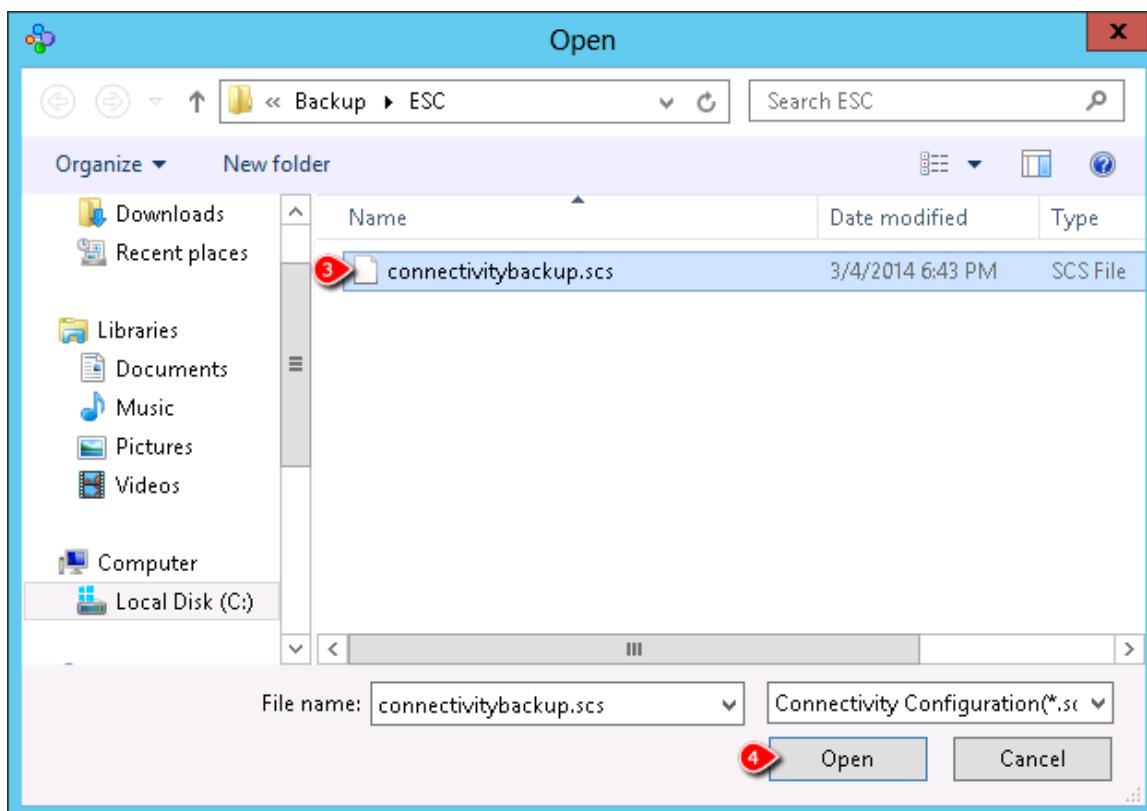
Restore from a Backup

To restore connectivity settings:

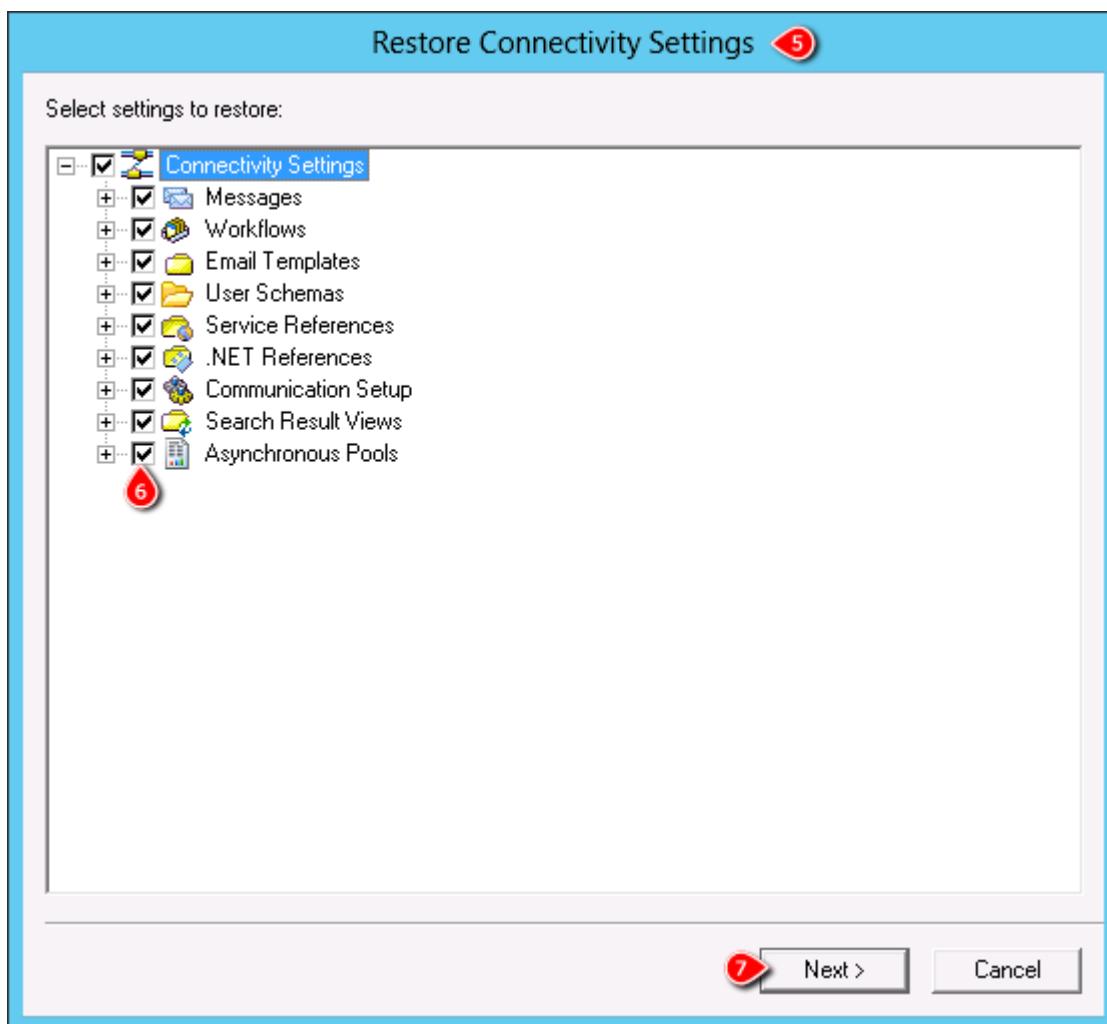
1. In the **tree view**, expand the logon server node.
This is where you want to import connectivity settings.



2. Right-click the **Connectivity** node and select **Restore**.
3. In the **Open** dialog, browse to and select your backup file.
By default, the backup file has .scs extension.



4. Click **Open**.
5. The **Restore Connectivity Settings** window displays the connectivity settings that were backed up.
All the backup items are selected by default.



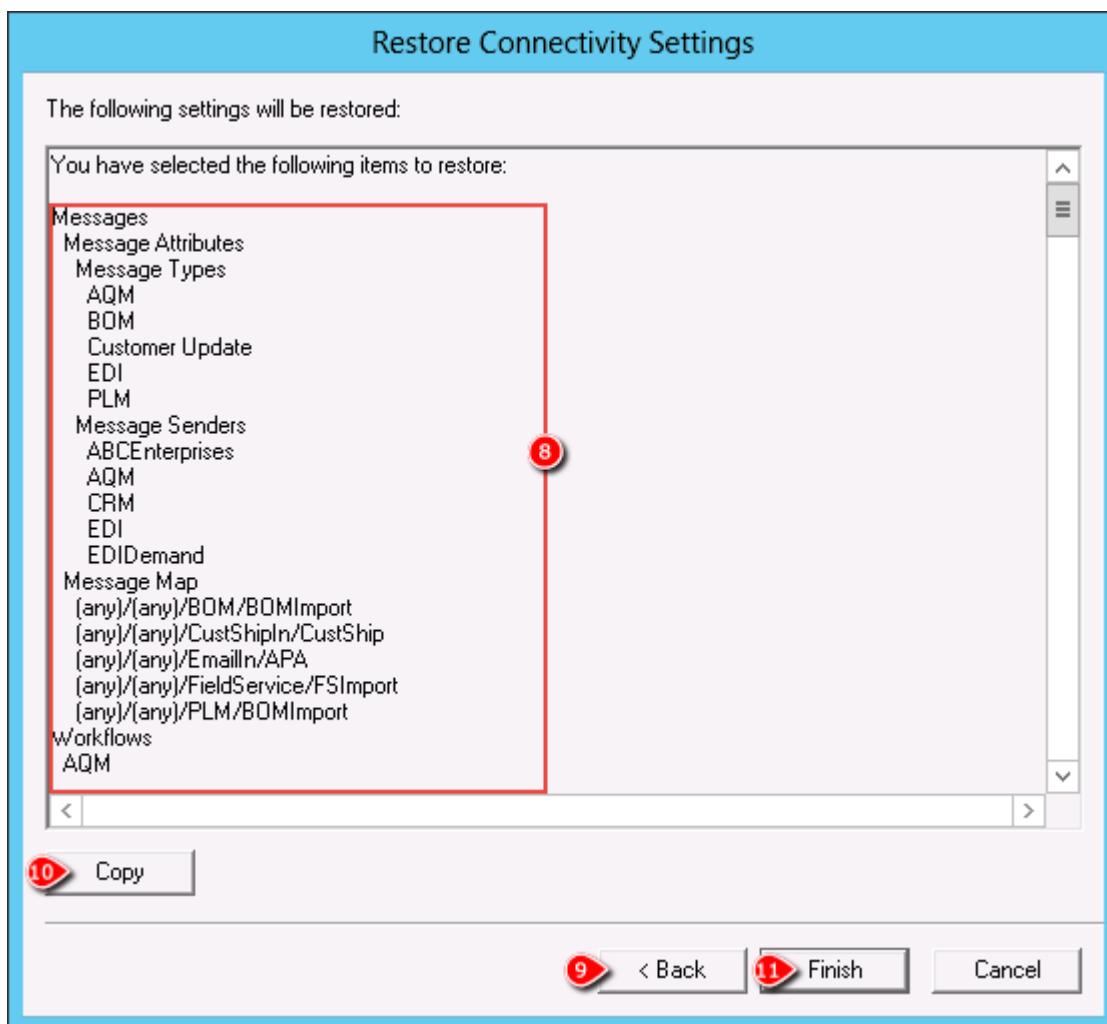
6. Clear the check box next to the items you do not want to restore.

The dependencies and integrity are not checked during connectivity settings restore, so be careful when selecting or deselecting individual nodes.

7. Click **Next**.

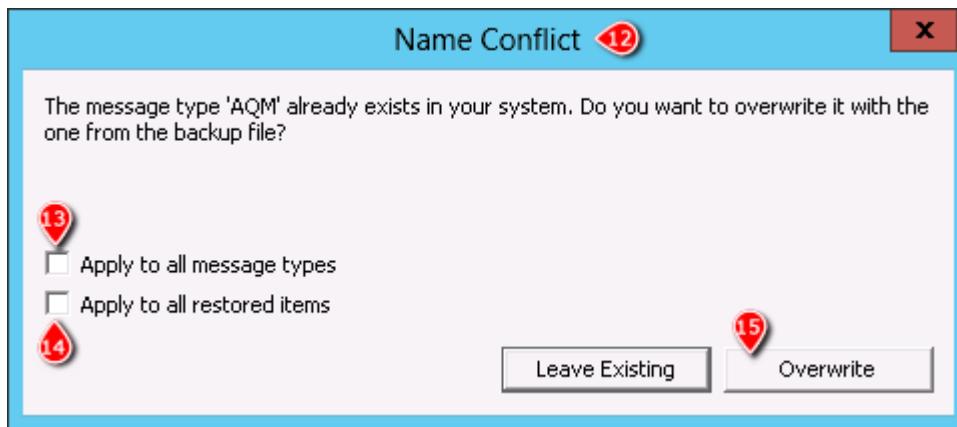
If you select to restore Communication Setup items, Workflows, Service references, .NET references or REST references, you must perform specific additional steps. Refer to the Communication Setup Items, Workflows, Service references, .NET References or REST References Restore procedure for details.

8. Review the list of restore settings.



9. If you need to return to the previous screens, click **Back**.
10. If you want to copy the list of all restored items to the clipboard, click the **Copy** button.
11. To start the restore procedure, click **Finish**.

12. If an item which already exists in the system is found among the restored items, the name conflict message displays the details.

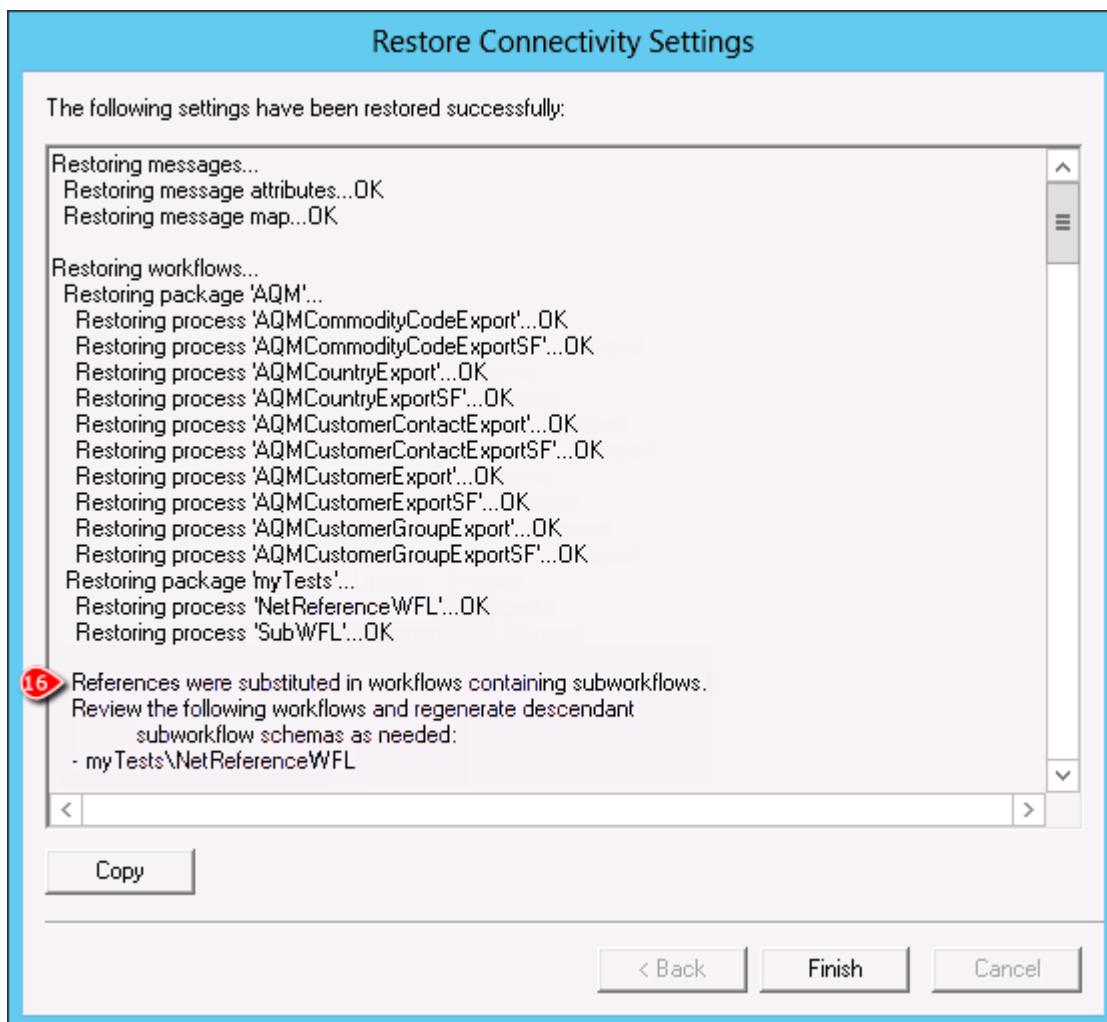


Select one of the restore options.

To perform the action you choose (either **Leave Existing** or **Overwrite**) on the current restored object only, do not select any check box.

13. To perform the action you choose (either **Leave Existing** or **Overwrite**) on all the restored items of the current type (messages, workflows, input or output channels, web references, and so on), select the **Apply to all message types** (workflows, channels, web-references) check box.
14. To perform the action you choose (either **Leave Existing** or **Overwrite**) on all the restored items of all the types, select the **Apply to all restored items** check box. The **Apply to all message types** (workflows, channels, web-references) check box is automatically selected in this case.
15. To replace the current item or items with an item or items from the backup, click **Overwrite**. Alternatively, to skip the backup item and leave the item that is already present, click **Leave Existing**.
16. If you restore a workflow that meets the following conditions:
- the workflow contains .Net Calls, REST Operation or Web Methods elements;
 - the workflow contains Sub-Workflow element that uses Cycling with XPath that defines input schema for sub-workflow element;
 - during restore operation, you changed the reference to .Net reference, RESTful reference or Service Reference element;

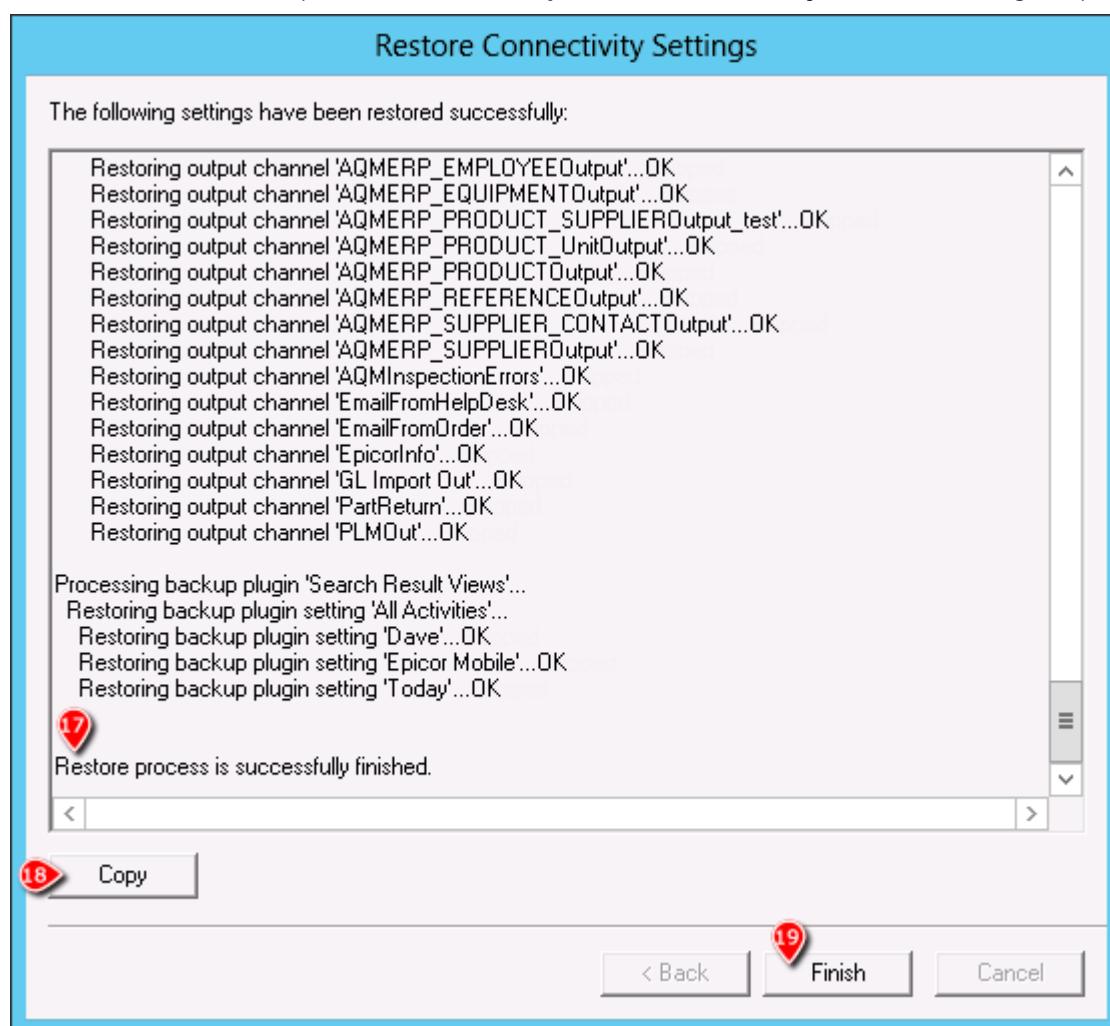
a warning message with the list of affected workflows is displayed.



You must open these workflows after restore operation is complete and regenerate the descendant sub-workflow schemas.

17. In the **Restore Connectivity Settings** window, view the restore progress.

At the end of the restore process, the **Restore process is successfully finished.** message displays.



18. Click the **Copy** button to copy the complete export log to the Windows clipboard.

19. Click **Finish**.

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

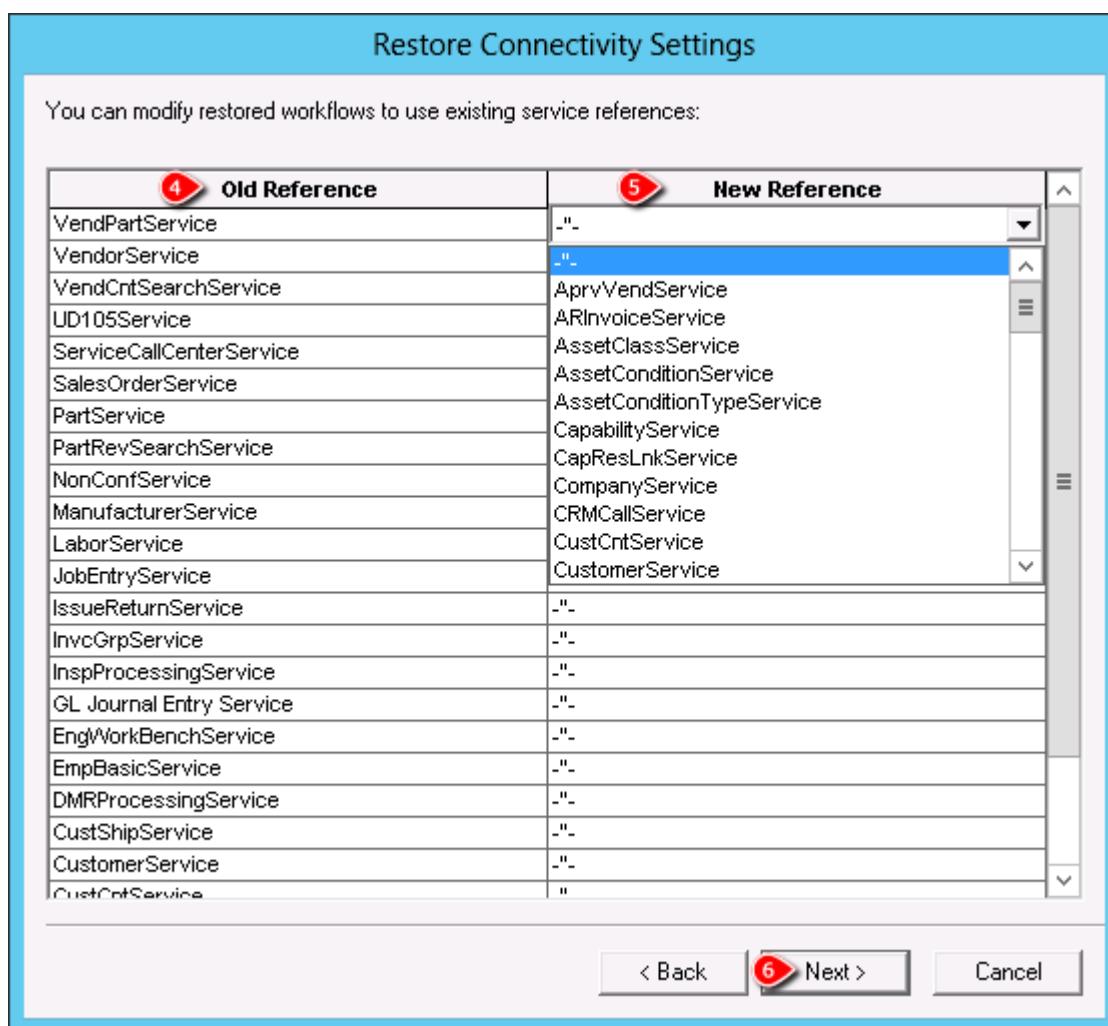
Communication Setup Items, Workflows, Service references, .NET References or REST References Restore

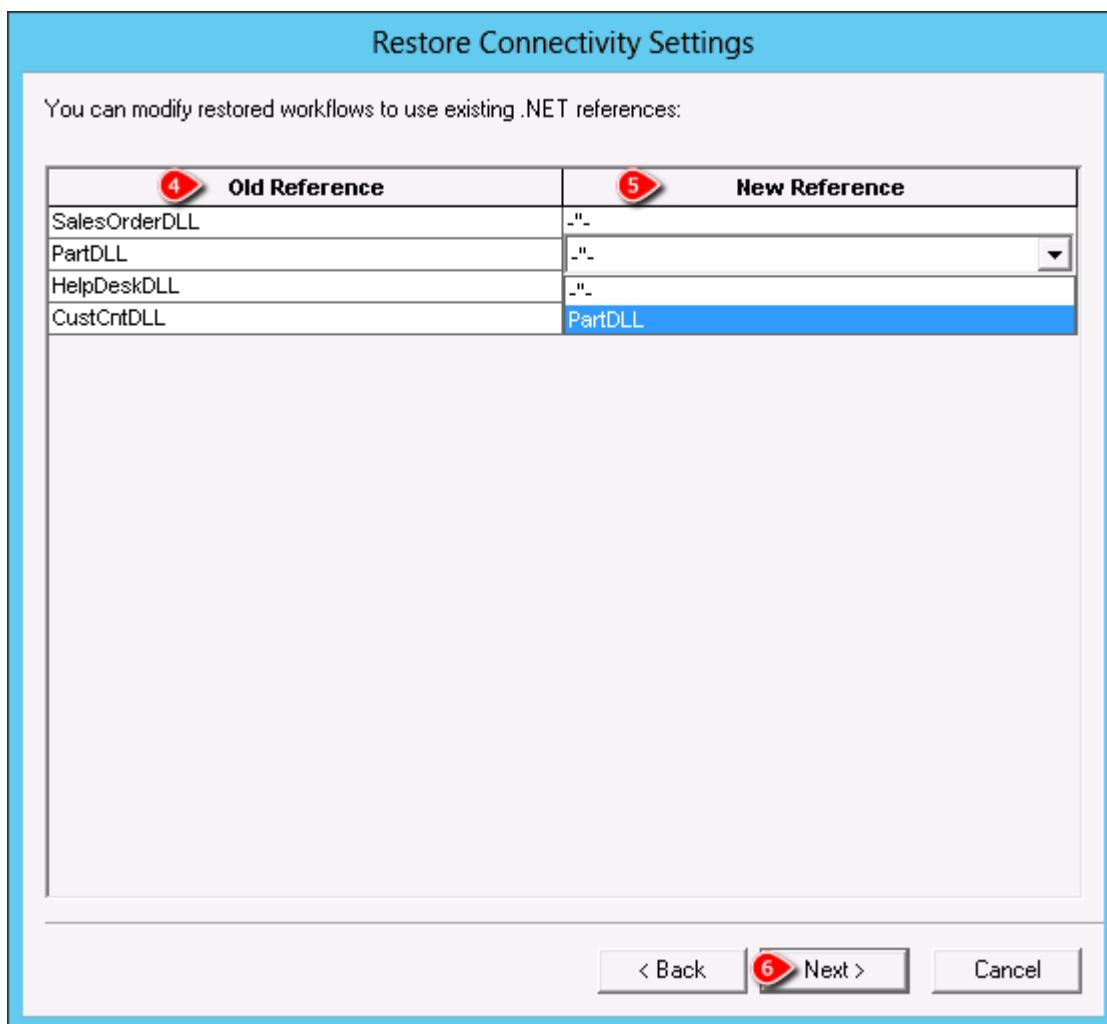
If you select to restore Communication Setup items, Workflows, Service references, .NET references or REST references, perform the following steps:

1. If you selected to restore communication setup, the servers associated with the restored settings are retrieved. The **Old Names** column displays the list of servers saved in the backup file.

2. If the list of servers in the backup file differs from that of the existing servers, map the imported servers to the existing ones. For each old server name, select the existing server name in the **New Names** column.
3. Click **Next**.
4. If you selected to restore a workflow that uses Web-methods, or .NET references, or REST references, the **Old Reference** column displays the list of references saved in the backup file.
5. In the **New Reference** column, select the Web-methods, or .NET references, or REST references to use instead of the imported one.

The **New Reference** column displays only the web references that exist in the system and have the same web-methods.



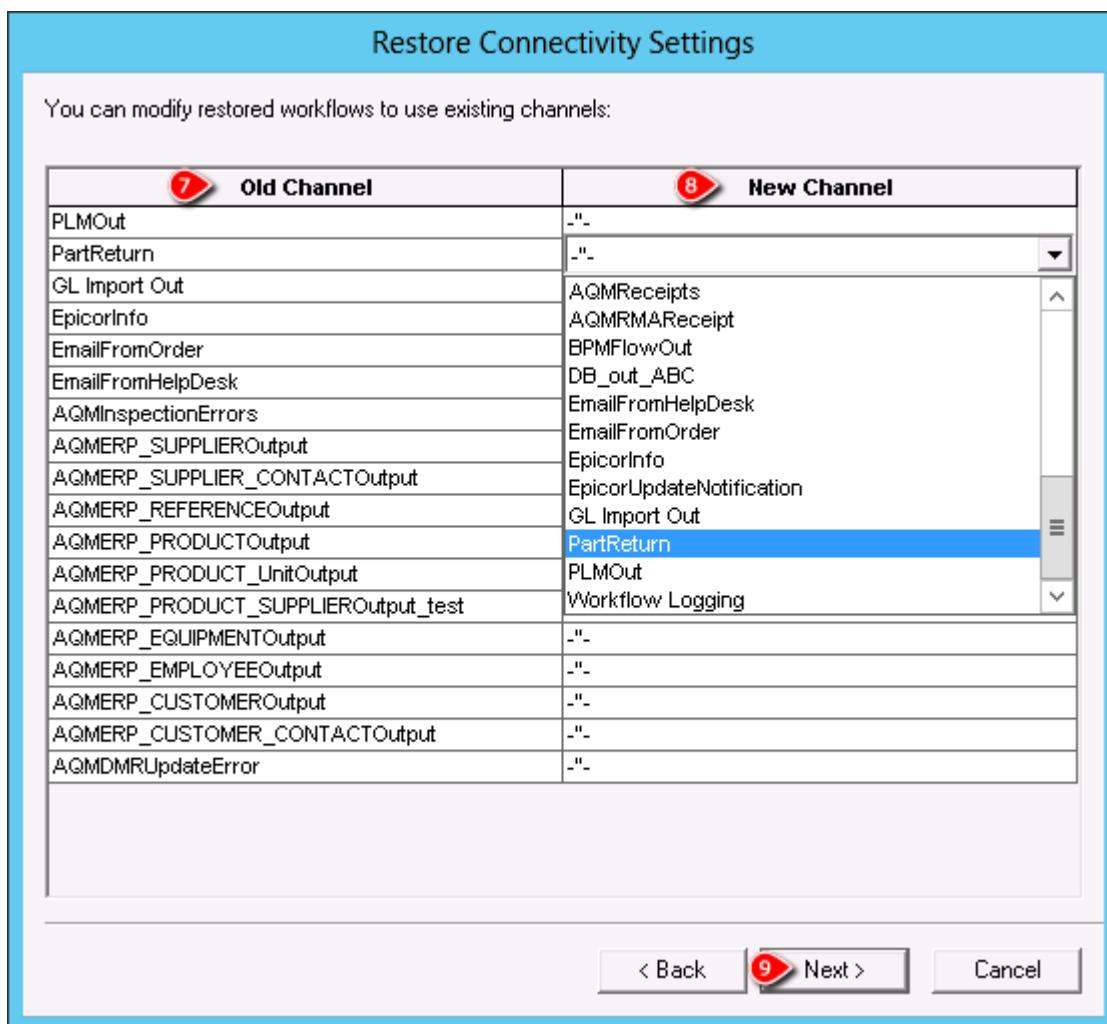


When you change a .NET, REST, or Web reference during restore from backup, the related Conversions and schema properties are changed automatically.

6. Click Next.

If you selected to restore a workflow you may need to adjust the communication channels it uses, as these channels may not exist on the computer, on which you perform the restore. You don't have to manually correct the channel settings or correct the workflow to point it to other channels. Instead, map and edit the channel during the restore process.

7. The Old Channel column displays the list of channels registered in the system.

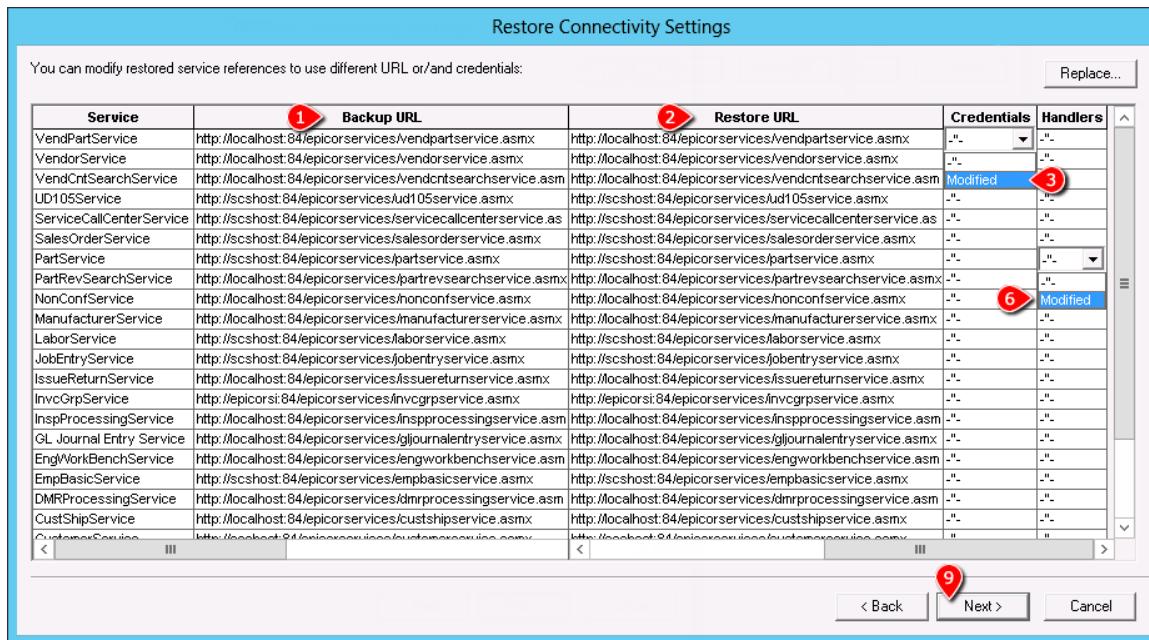


8. In the **New Channel** column, select the channel saved in the backup file that should be used instead of the one which displays in the **Old Channel** column.
9. Click **Next**.

Restore Service References

If you selected to restore service references, you can change service reference URL and windows credentials to gain access to the specified server. Also changes in service reference handlers can be required.

1. To modify service reference URL, on the service reference settings modification screen, locate the reference in the list and review the **Backup URL** for this reference.
This information is read-only.

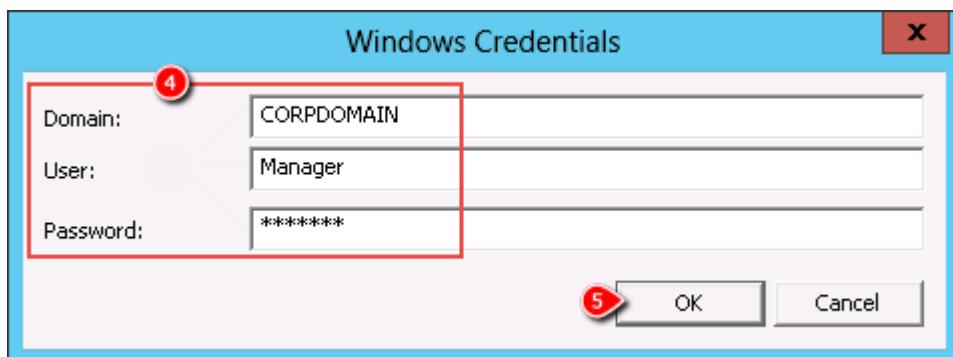


2. Adjust the corresponding **Restore URL**

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

3. To modify Windows credentials, in the **Credentials** column, select **Modified**.

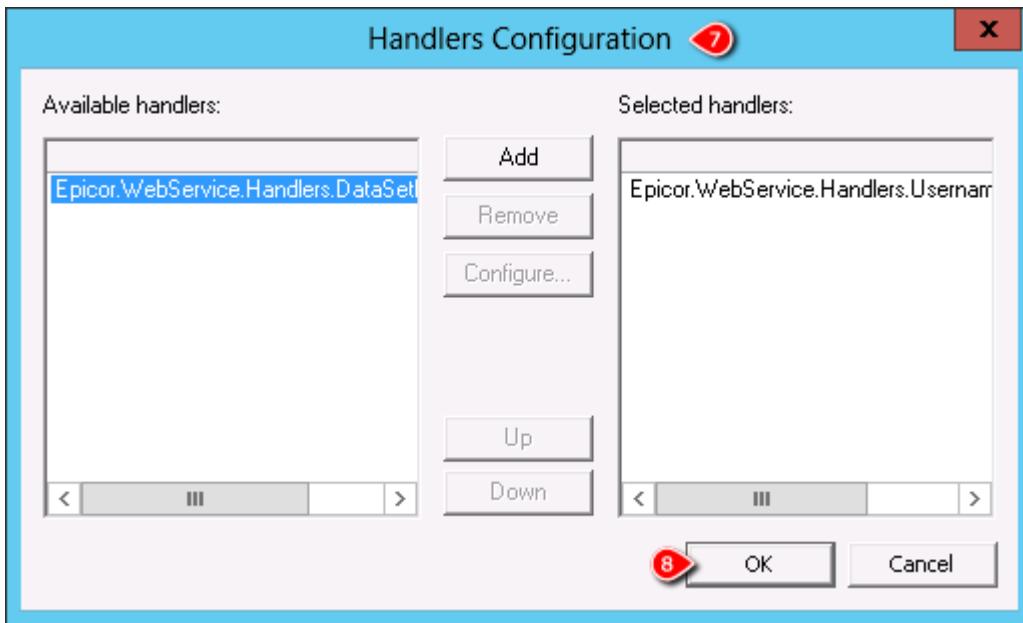
4. In the **Windows Credentials** dialog, specify **Domain**, **User** and **Password**.



5. Click **OK**.

6. To modify service reference handlers, click in the **Handlers** column, and select **Modified**.

7. In the **Handlers Configuration** dialog, select and adjust handlers.



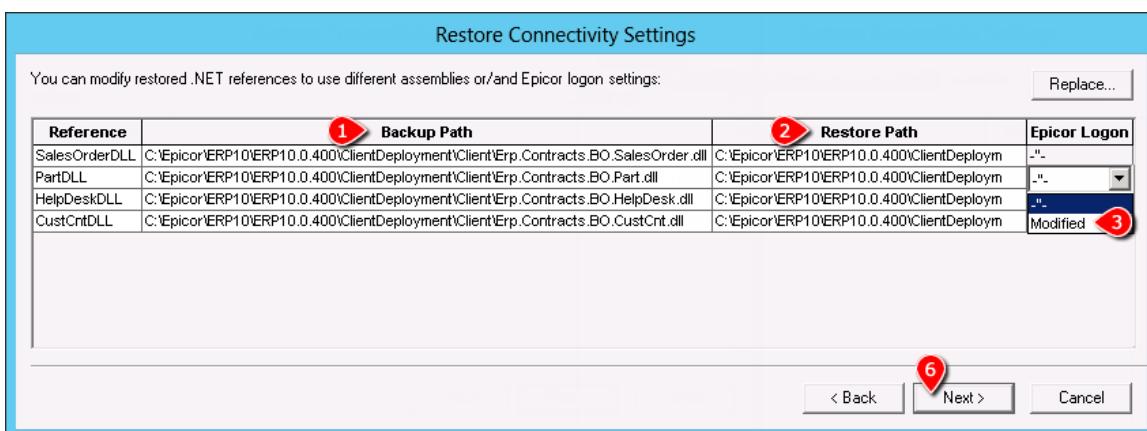
8. Click **OK**.

9. Click **Next**.

Restore .NET References

If you selected to restore .NET References, you may need to modify the **Restore Paths** to assemblies.

1. To modify .NET reference folder, on the .NET reference settings modification screen, locate the reference in the list and review the **Backup Path** for this reference.



This information is read-only.

2. Adjust the corresponding **Restore Path** for the restored reference.

3. If you restore Epicor ERP Assembly .NET Reference, you may need to modify Epicor ERP configuration settings (logon credentials). To do so, in the **Epicor Logon** column, select **Modified**.
4. In the **Logon to Epicor** window, enter the new information to access the application server. Specify **AppServer** and **Port**. In the **User** and **Password** fields, provide the Epicor logon credentials. In the **Company** and **Plant** fields, specify company and Plant.
5. Click **OK**.
6. Click **Next**.

Restore RESTful References

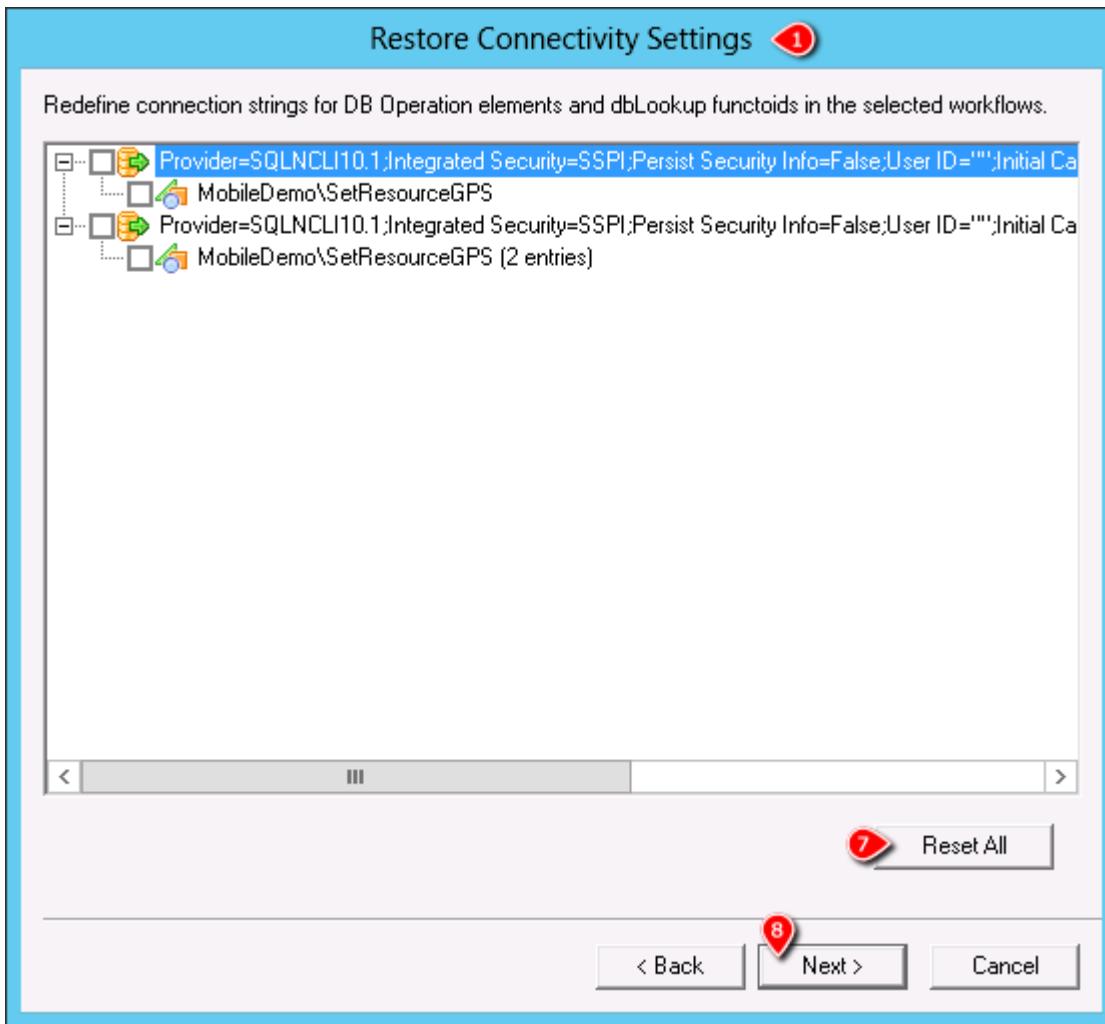
If you selected to restore RESTful References, you may need to modify their service URL (base URI), authentication mode and credentials to gain access to the specified server.

1. To modify REST web service URL, on the RESTful reference settings modification screen, locate the reference in the list and review the **Backup URL** for this reference.
This information is read-only.
2. Edit the corresponding **Restore URL**.
Restored URL for RESTful references is checked on validity: empty URL or URL with incorrect syntax is not accepted.
3. To modify REST web service authentication mode and credentials, click in the **Authentication** column, and select **Modified**.
4. In the **Authentication** dialog, in the **Authentication** field, select the authentication mode. The following options are available:
 - **None**
 - **Basic**
Specify **User** and **Password**.
 - **Windows**
Specify **Domain**, **User** and **Password**.
Alternatively, select the **Use DES service account** check box.
 - **OAuth**
Specify **Token server URL**, **User** and **Password**
5. In the **Authentication** dialog, click **OK**.
6. Click **Next**.

Restore a Workflow with a DBOperation

If you restore a workflow with a DBOperation workflow element or dblookup functoid, you are prompted to adjust connection strings for the restored DBOperation elements and dblookup functoids.

1. View the connection strings that 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. For each workflow, if the number of the connection string entries is more than one it is displayed after the workflow name.

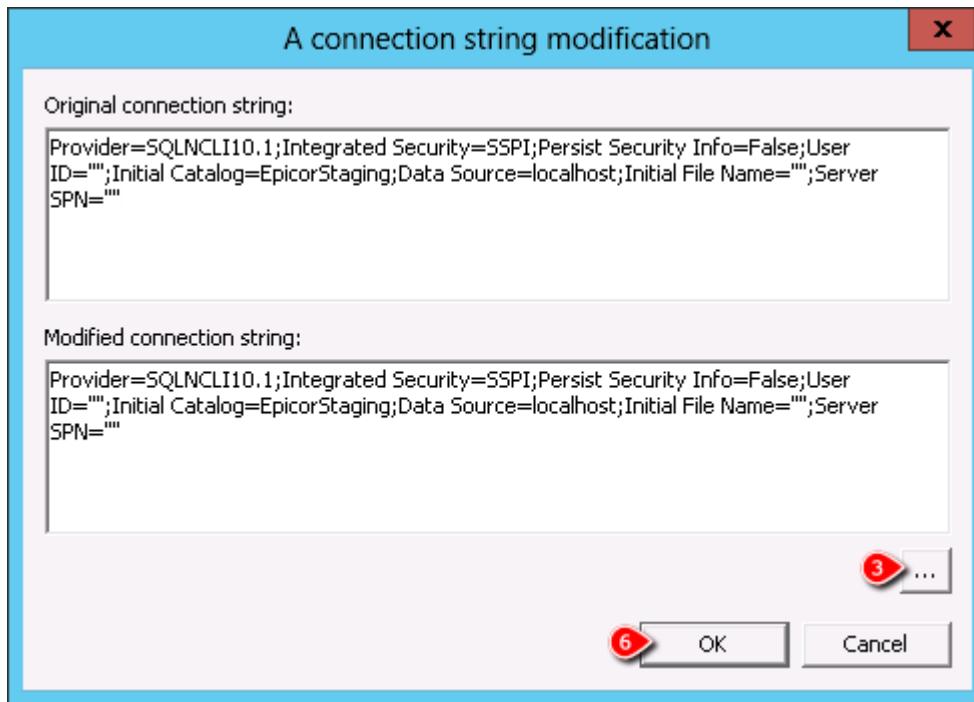


2. Select the connection string or strings to edit.

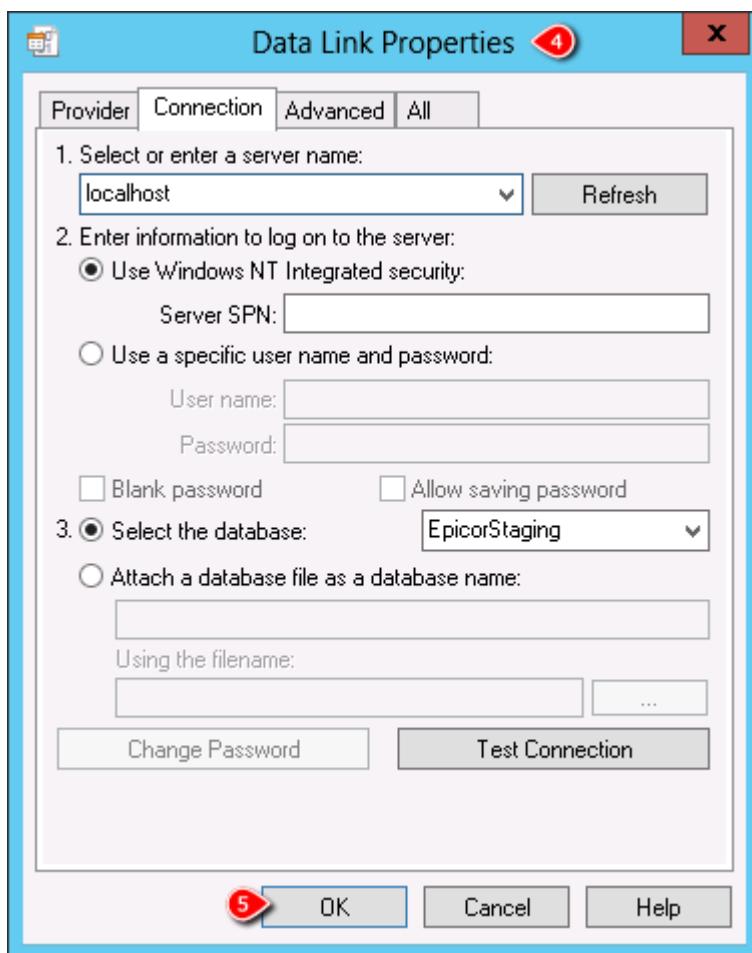
To edit a connection string for all the workflows that use it, select the root node in the tree. All the subordinate nodes are selected automatically.

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.

3. To adjust the connection string, on the **A connection string modification** screen, click the **ellipse** button to display.



4. On the **Data Link Properties** screen, make the required changes.



5. Click **OK**.

6. On the **A connection string modification** screen, click **OK**.

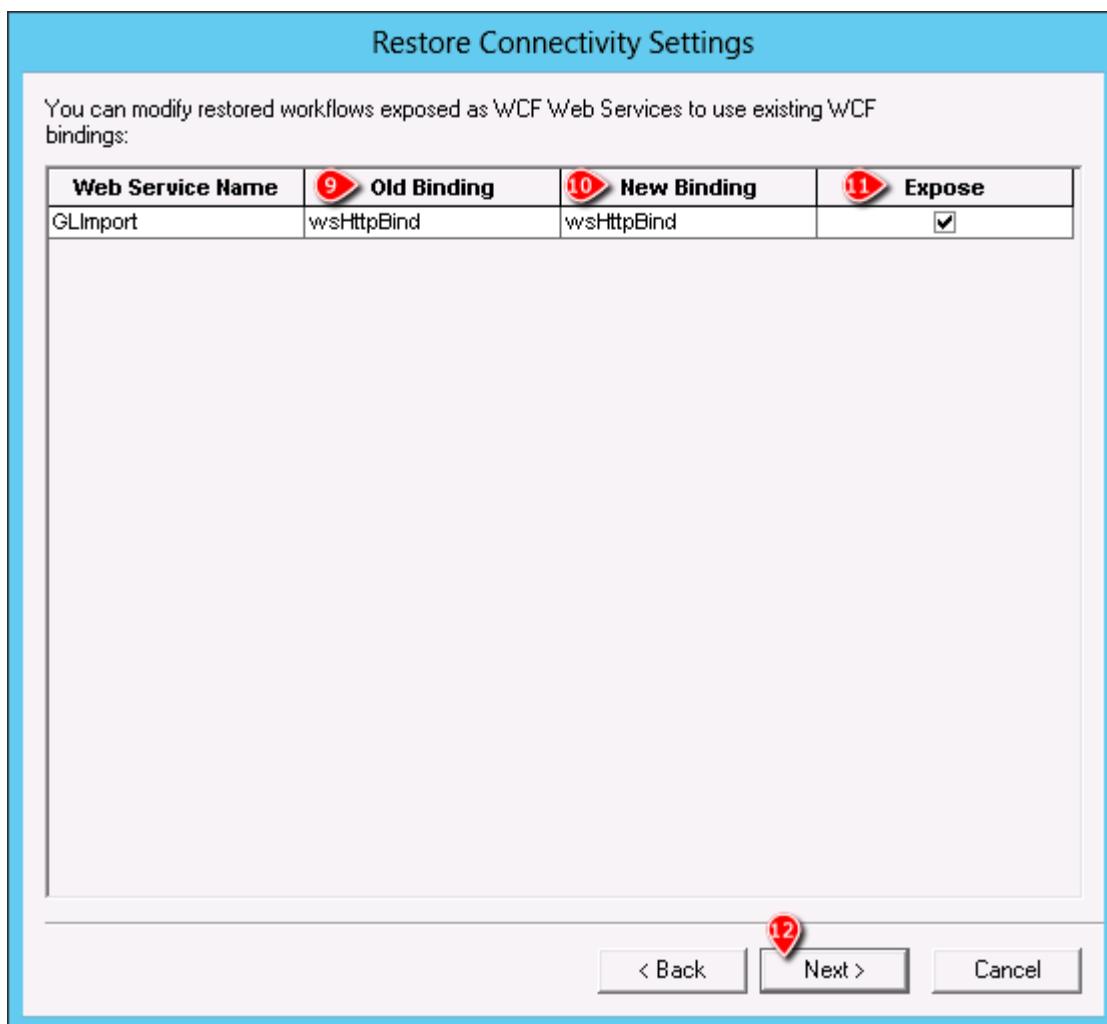
7. To reset all changes you have made to the connection string, click the **Reset All** button.

Use this button if you changed some connection strings and want to bring back the original values.

8. Click **Next**.

9. If you restore a workflow exposed as WCF Web Service, you are prompted to modify its binding.

To modify the binding, locate the WCF Web Service name in the list and review the **Old Binding** for this Web Service.



10. To modify restored workflows exposed as WCF Web Services to use existing WCF bindings, in the **New Binding** column, select a binding.

If the original binding of the restored Web service cannot be found among the existing ones the keyword **Undefined** is used and the **Next** wizard button is disabled. To enable the button select existing binding or clear the check box in the **Expose** column for this web service.

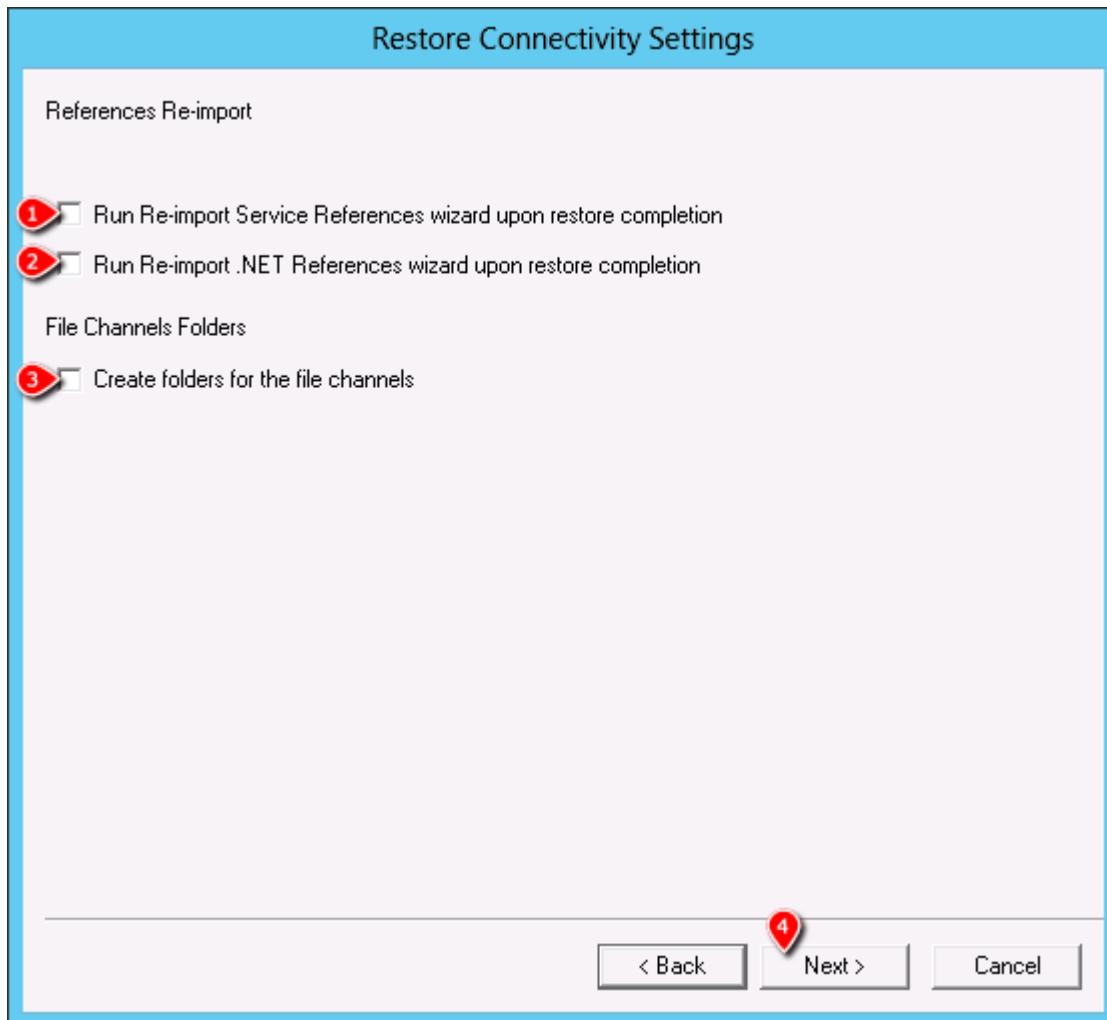
11. In the **Expose** column, select the check box to expose the workflow as WCF Web Service.

12. Click **Next**.

Setup References Re-import

If you selected to restore Service References, .NET References you can setup automatic reimport of these references.

- To automatically re-import service references, select the **Run Re-import Service References wizard upon restore completion** check box.



2. To automatically re-import .NET References, select the **Run Re-import .NET References wizard upon restore completion** check boxes.
3. If you selected to restore Input or Output channels, under the **File Channels Folders** title, select the **Create folders for the file channels** check box to restore channel folders during restore process.
If it is impossible to create a folder with a path from backup, a warning displays.
4. Click **Next** and return to the general restore procedure.

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

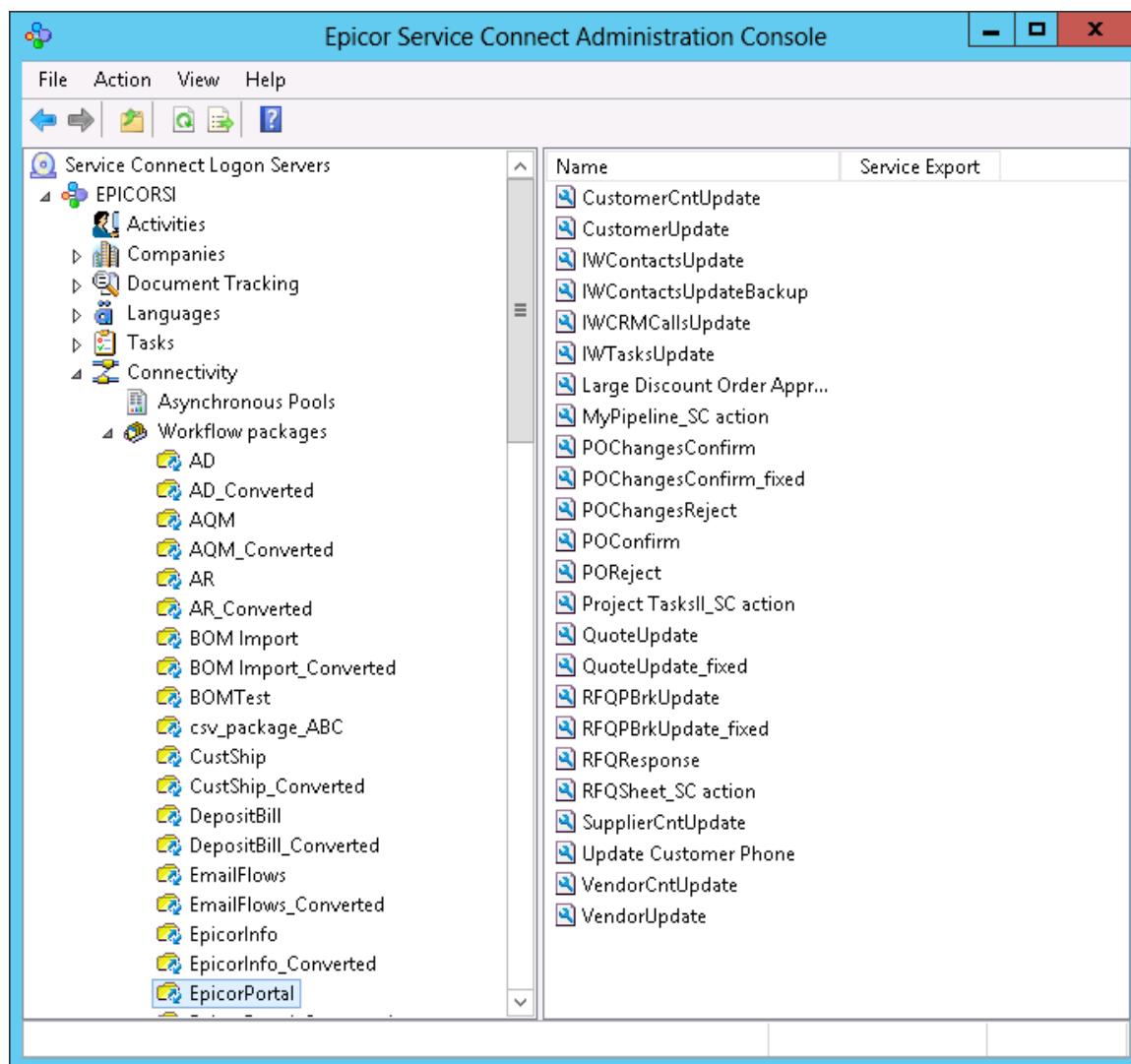
The components under the Connectivity node in the ESC Administration Console configure, route, and process messages. Refer to Epicor Service Connect Administration chapter for information on how to back up and restore all items under the Connectivity node.

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.

The workflows provided with the Epicor Service Connect are referred to as **Standard**. The workflows created manually or derived from a Standard Process by the user in Workflow Designer are referred to as **Custom**.

A workflow package is the Service Connect equivalent of a physical folder. It is used to store workflows in logical groups. A list of all existing workflow packages is displayed in the **Workflow Packages** node under the **Connectivity** node of the Epicor Service Connect Administration Console tree.



In Epicor Service Connect Administration Console, you can perform the following actions with workflows and workflow packages:

- Create new workflow packages.
- Customize a Standard workflow to modify it.
- Edit a Custom workflow.
- Expose Custom workflow as Web Service.
- Edit several workflows simultaneously using mass update feature.
- Backup and restore workflows and packages.
- Rename and remove Custom workflows and packages
- Copy workflow packages
- Copy and move workflows from one package to another

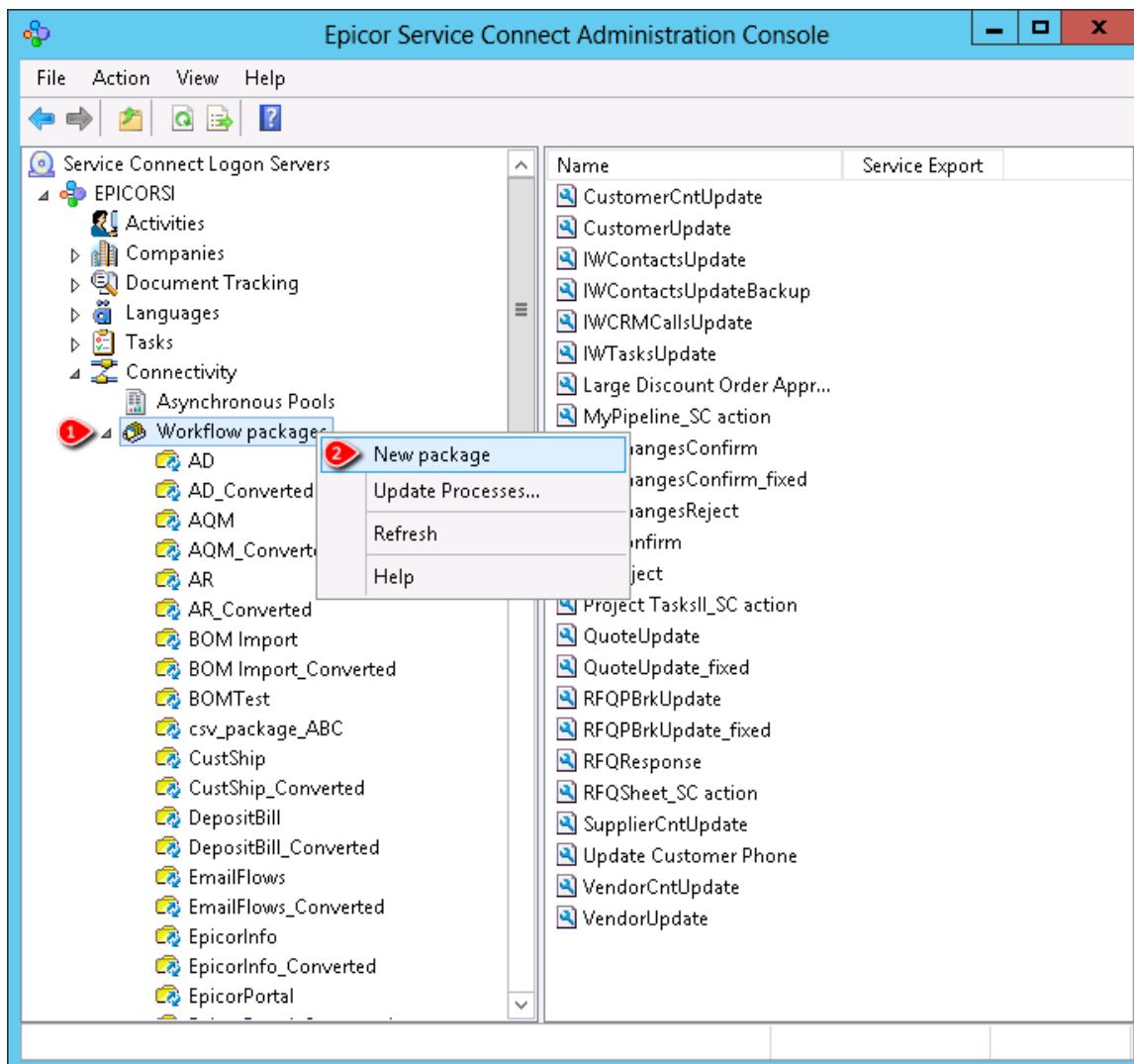
To learn how workflows interact and how to develop workflows refer to Workflow Designer chapter and to the product-specific chapters available on EPICweb. This section describes the

connectivity components that route messages to workflows and interact with workflows.

Create a Workflow Package

To create a workflow package:

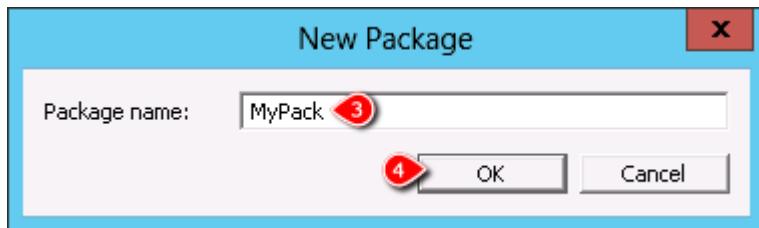
1. In the Epicor Service Connect Administration Console tree, expand the **Connectivity > Workflow Packages** node.



2. Right-click **Workflow Packages** and select **New package**.

You can alternatively click **Action** in the standard menu, and select **New package**.

3. In the **New Package** dialog enter the package name.



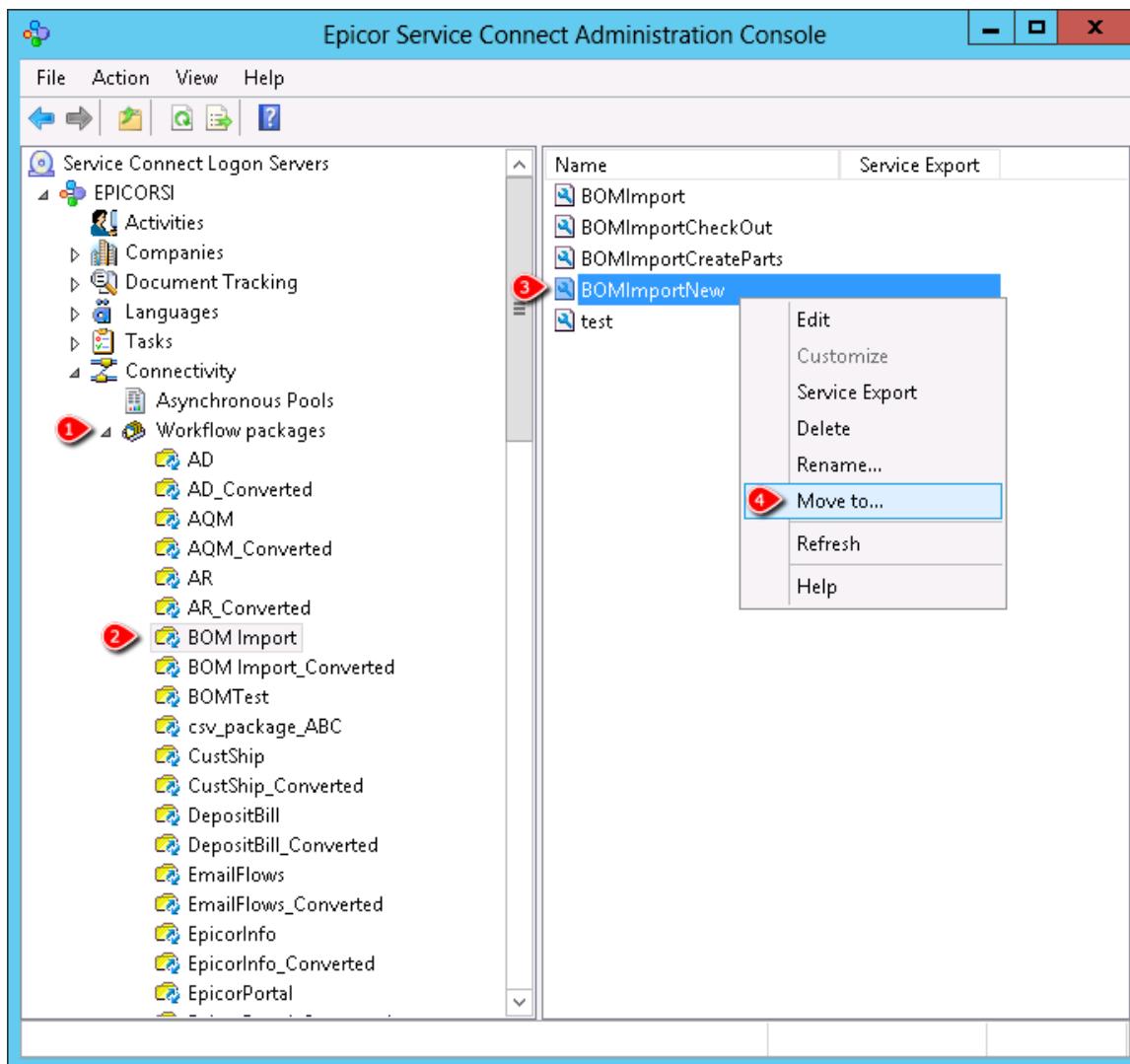
4. Click **OK**.

Move a Workflow between Workflow Packages

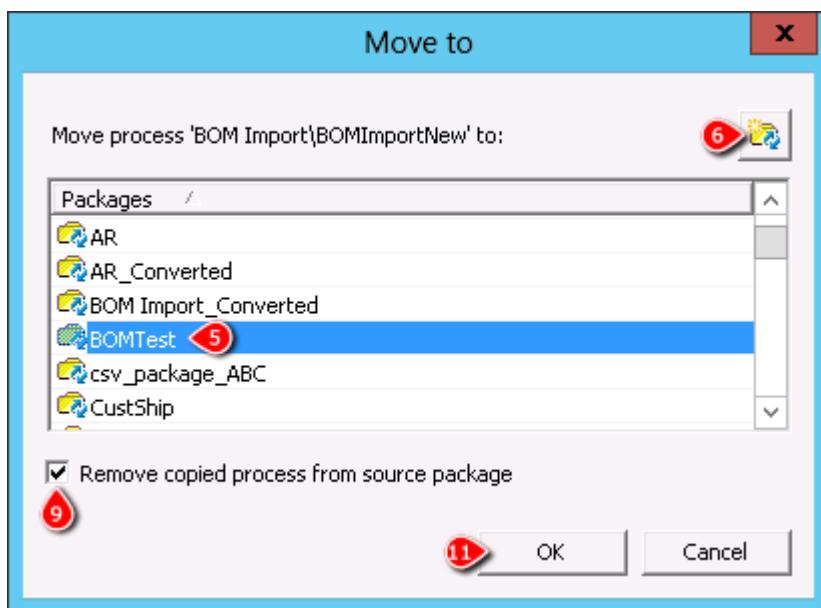
Use the **Move to** functionality to create a copy of the selected workflow in a different package. You can select to delete the original workflow after it is copied.

To copy or move a workflow:

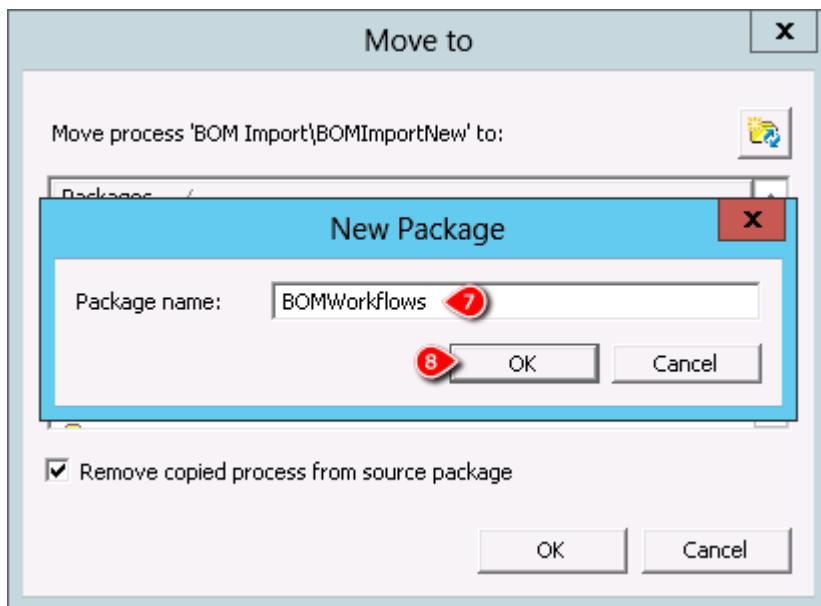
1. In the Epicor Service Connect Administration Console tree, expand the **Connectivity > Workflow Packages** nodes.



2. Click the package that contains the workflow you want to move or copy.
3. In the list in the right pane, right-click the workflow you want to copy or move.
4. Select **Move to...**.
5. In the **Move to** window, select the destination workflow package.



6. If the destination package does not exist yet, click the button in the dialog upper right corner to create a new package.



7. In the **New Package** dialog, enter the new package name.
8. Click **OK**.
9. If you want to delete the original workflow after its copy in the destination workflow package is created, select the **Remove copied process from the source package** check box.
Use this option to move a workflow from one package to another.

10. To copy the workflow from one package to another, clear the **Remove copied process from the source package** check box.
11. Click **OK**.
12. If you selected to move a workflow that is referenced from some other workflows, a warning is displayed. To update all the links to the workflow automatically, click **OK**.

When you move a workflow that is referenced by external callers like BAQs or BPMs directly, make sure to update settings in those callers as well.

After copying or moving a workflow that was exposed as a web or WCF service, the destination workflow is not exposed.

Expose Processes as Web Services

You can expose any Epicor Service Connect workflow (also referred to as process) as a web service. For each process registration the system can create a web service with a single method with the help of the .NET Framework. .NET classes are generated on the basis of the process XSD schemas. Thus web service method argument types structure is identical to that of the XML messages passed to and received from the process. All generated source code for the web service is placed into the ASP.NET application directory. After the first request to the exposed web service, ASP.NET services compile and execute the source code. The next time the compiled code is used unless the source code is changed.

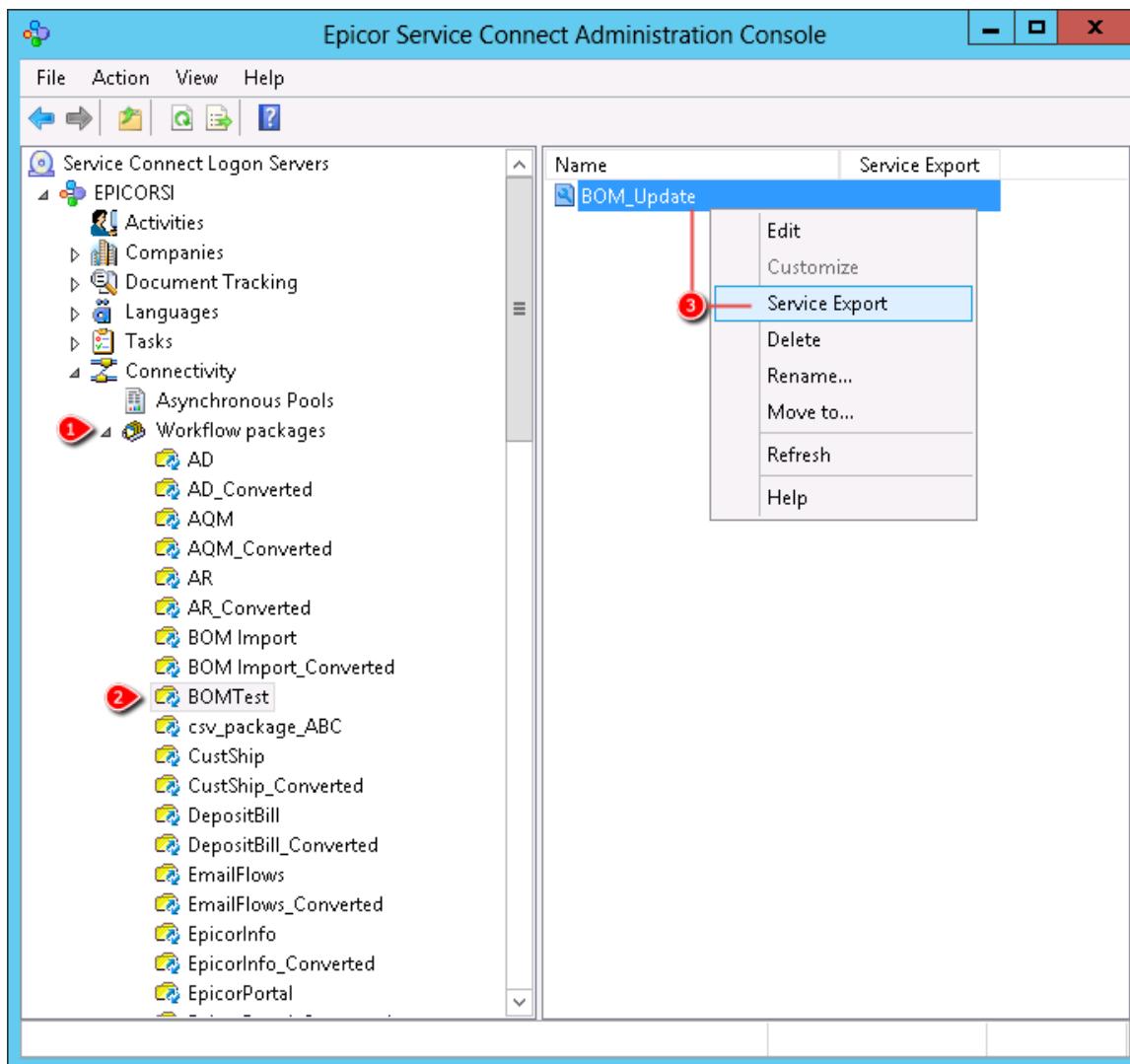
To activate Document Tracking for the workflow which is exposed as web service, set the **Tracing status** to **Required** on the **Document Tracking** tab of Process Properties dialog.

Also, schemas on the Start and Finish elements should not contain root elements with the same name and namespace, but with different content. If the Start and Finish workflow elements have the same name and namespace, they must be the same type and the content should be completely equal.

Refer to Workflow Designer chapter for details.

To expose a process as web service:

1. In Epicor Service Connect Administration Console, expand the **Connectivity > Workflow packages** node.



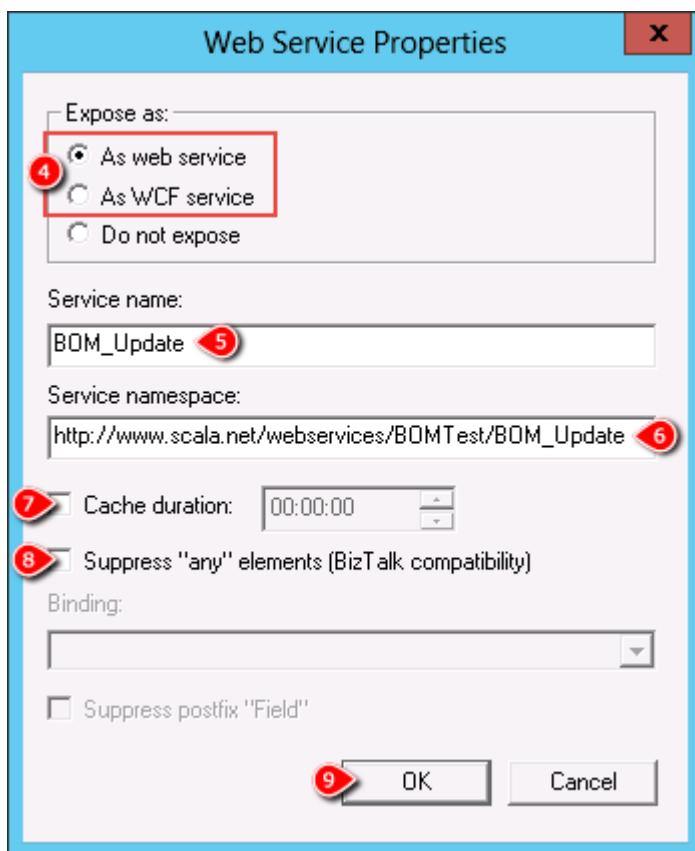
2. Click the package that contains the process you want to expose as a web service.

The list of processes is displayed in the right pane.

3. On the processes list in the right pane, right-click the process, and select **Service Export**.

You can only expose a Custom process as a web service.
To expose a Standard process, you must customize it first.

4. In the **Web Service Properties** dialog, select if you want to expose the process **As web service** or **As WCF service**.



5. In the **Service name** field, a web service name derived from the process name displays. You can change the web service name.

6. In the **Service namespace** field, define namespace for the web service.

By default, Service Connect uses the following pattern to generate namespace:

`http://www.scala.net/webservices/{package name}/{process name}`

7. To enable caching and specify the cache duration, select the **Cache duration** check-box and enter cache duration. This option is available for web services only.

8. For web services, you can select the **Suppress "any" element (BizTalk compatibility)** check box, to remove elements typed as "any" during exposure wrapper generation.

In the Workflow Designer, you can specify XSD schemas for the outgoing messages for Processes which include Task elements. But if the control flow reaches a Task element, it initializes the Task and returns the current XML message, which may not correspond to the specified output XSD schemas.

9. Click **OK**.

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

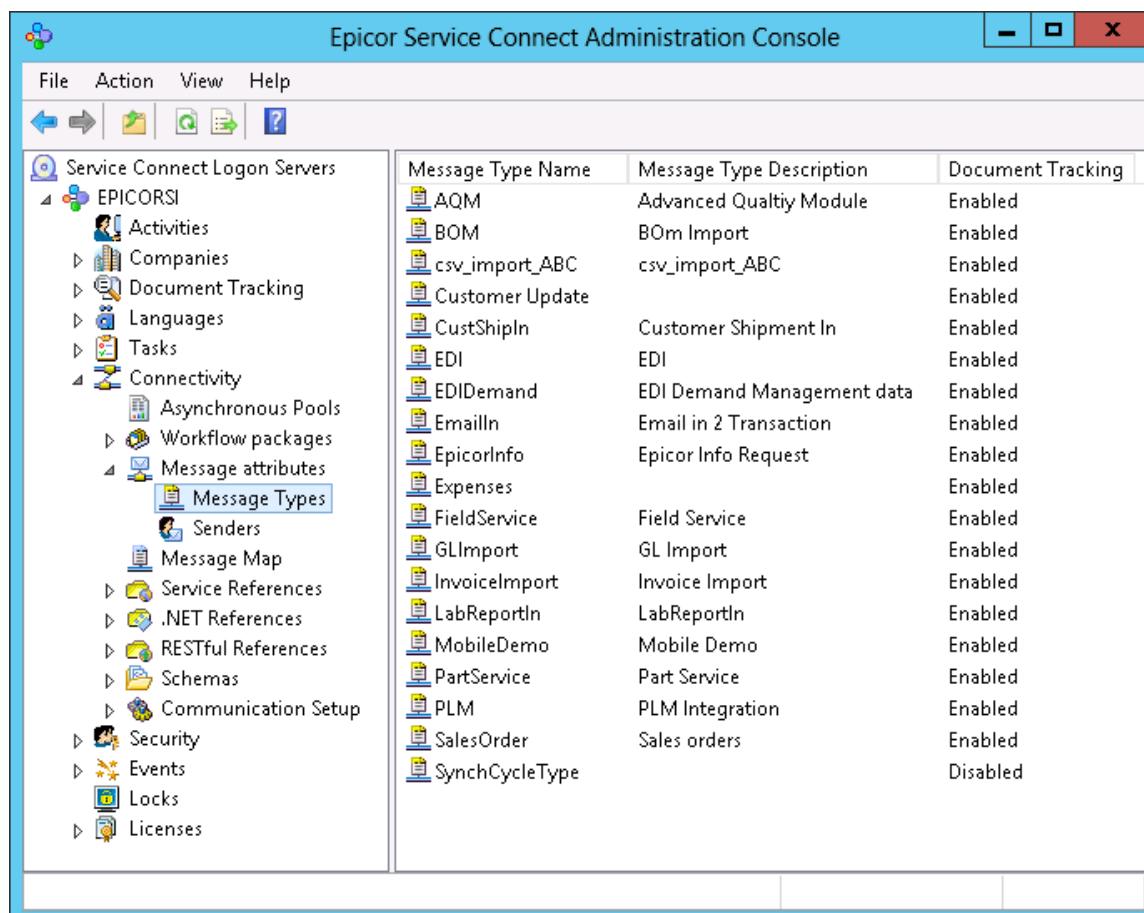
Message attributes are used to help route documents submitted to an input channel. A document that drops into an input channel has one of two formats:

- 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 Epicor Service Connect Overview chapter 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.

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.



Message Attributes

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

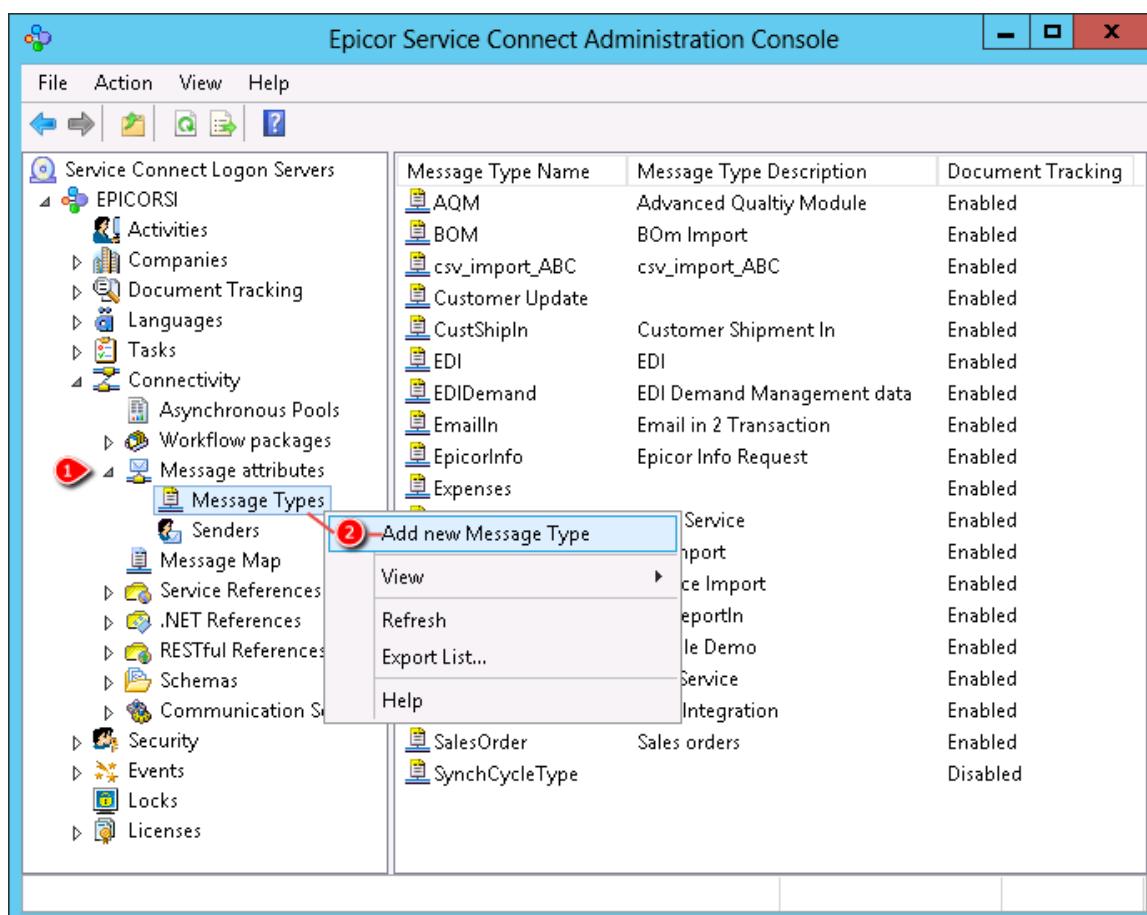
Use message types to classify the documents you plan to process in Service Connect. For example, if you plan to send documents that represent orders to Service Connect, you might create a message type called Order. Within a message type, you can have sub-classifications that define message actions, such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

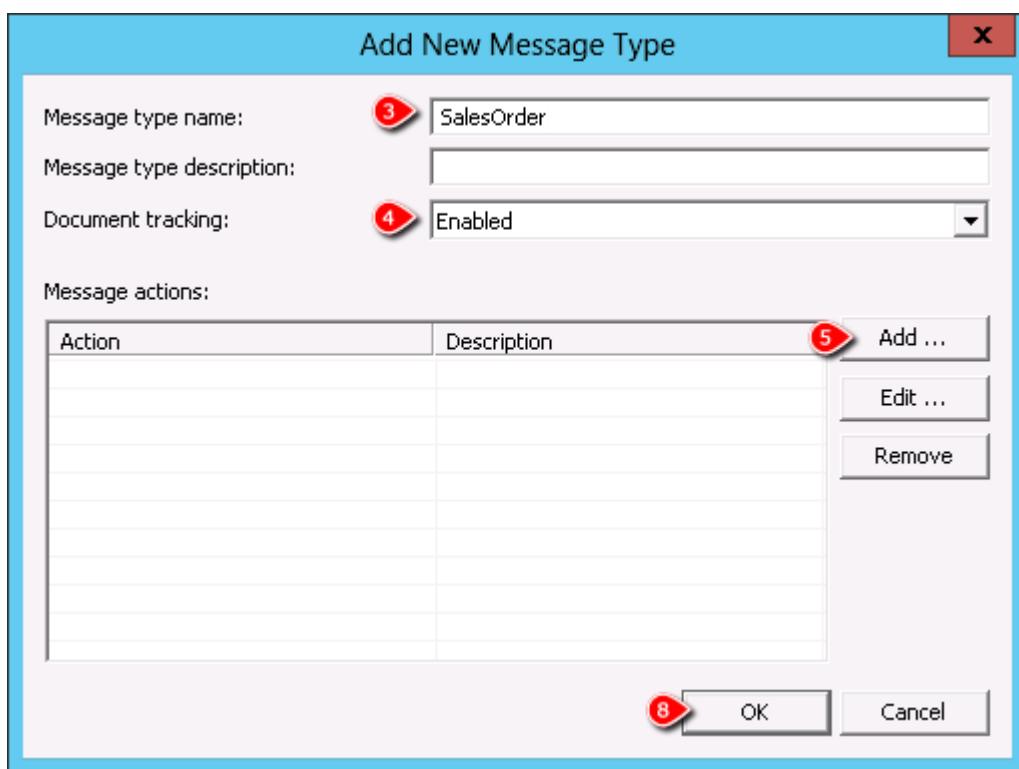
Add a Message Type

To add a new message type:

1. In the **tree view**, expand the **Connectivity > Message attributes** node.

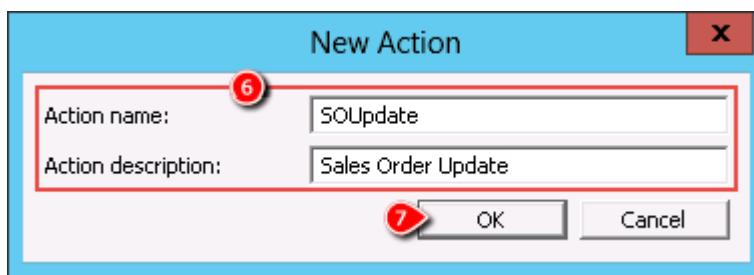


2. Right-click **Message Types** and select **Add new Message Type**.



The **Add New Message Type** window displays.

3. 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.



The **New Action** window is displayed.

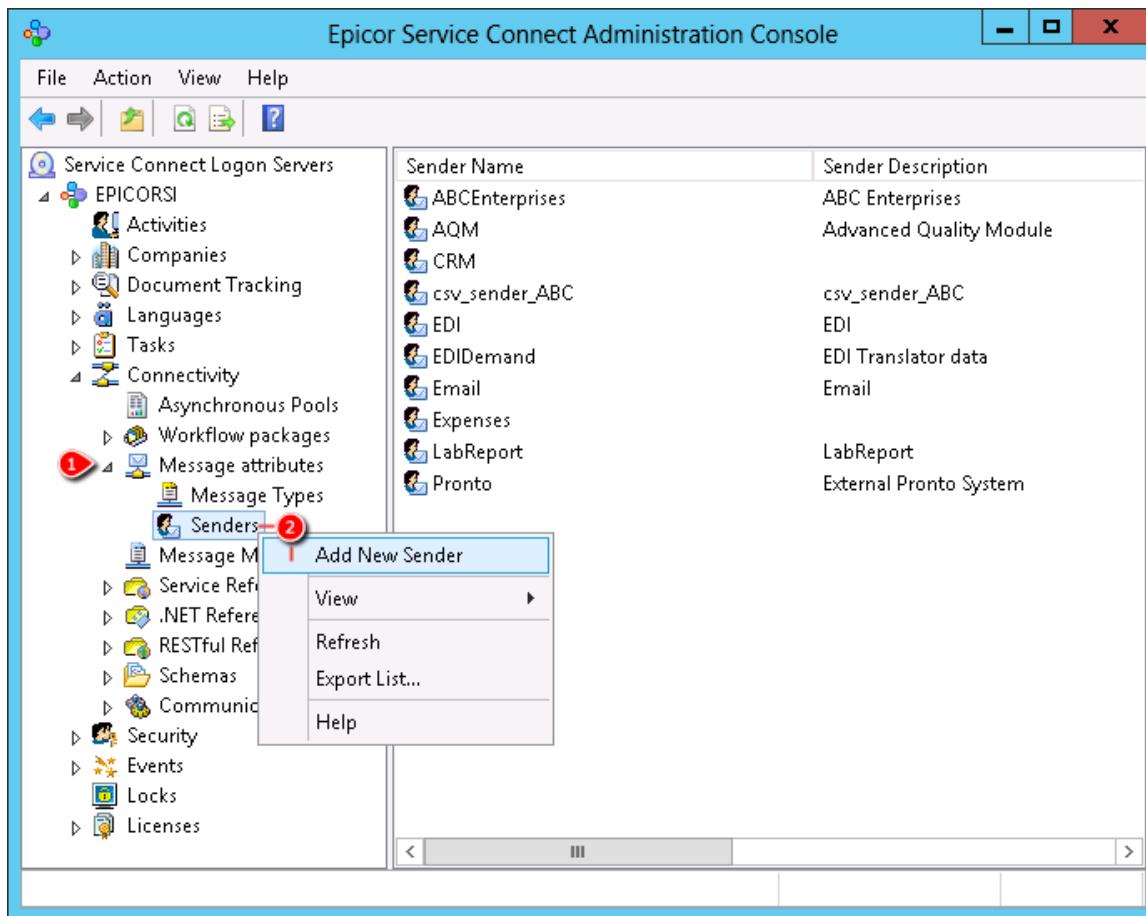
6. Enter an **Action name** and an **Action description**.
In general, action names are verbs that describe typical database actions, such as create, retrieve, update, or delete.
7. Click **OK**.

8. In the **Add New Message Type** window, click **OK**.

Add a Sender

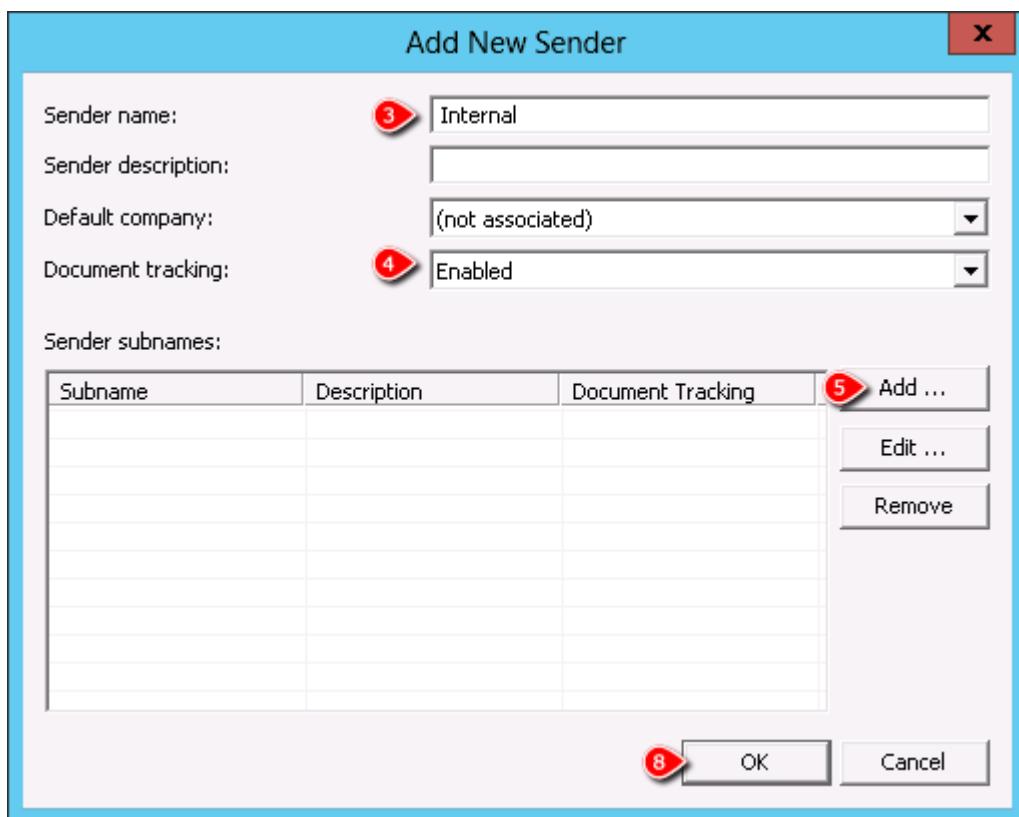
To add a new sender:

1. In the **tree view**, expand the **Connectivity > Message attributes** node.



2. Right-click **Senders** and select **Add New Sender**.

The **Add New Message Type** window displays.



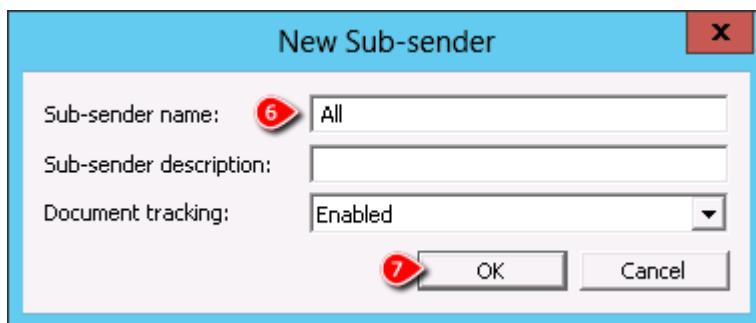
3. Enter a **Sender name** that describes the origin of a document.

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.

The **New Sub-sender** dialog is displayed.



6. Enter a **Sub-sender name** that represents a smaller category or use a catch all name like All.

7. Click **OK**.

8. In the **Add New Sender** window, click **OK**.

Message Maps

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.

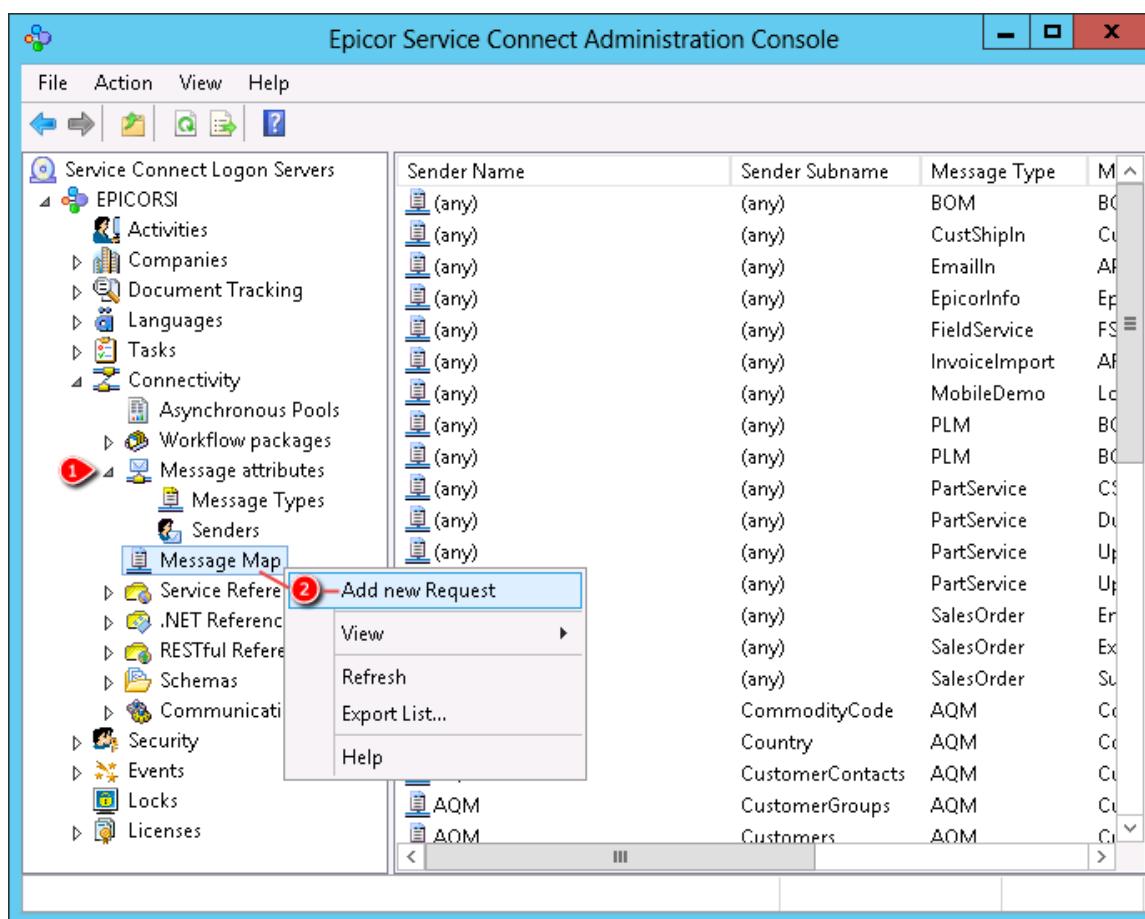
If you choose not to call a workflow directly from an Epicor application or not to call a workflow exposed as a web service, you will need to set up message maps to route incoming documents.

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.

Add a Message Map (Routing)

To add a Request ID to a message map:

1. In the **tree view**, expand the **Connectivity > Message attributes** node.

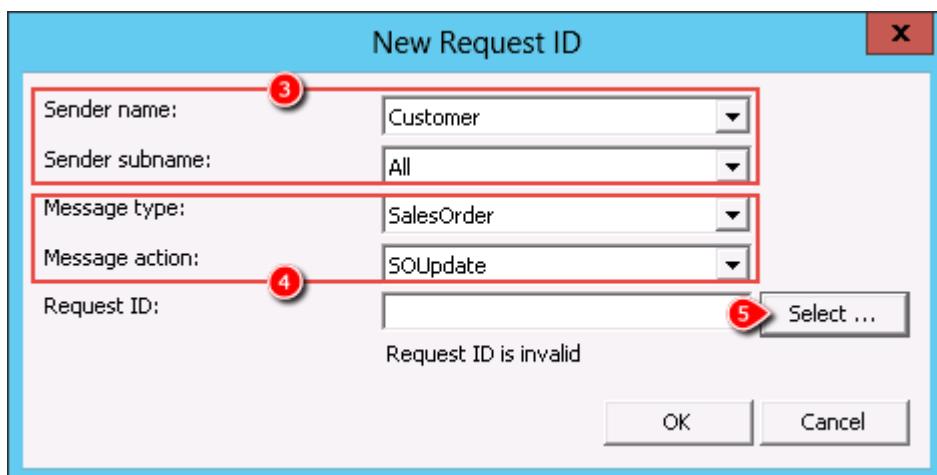


2. Right-click **Message Map** and select **Add new Request**.

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

- In the **New Request ID** window, select the appropriate **Sender name** and **Sender subname** for the intended Request ID.

In this example, select **Customer** and **All**.

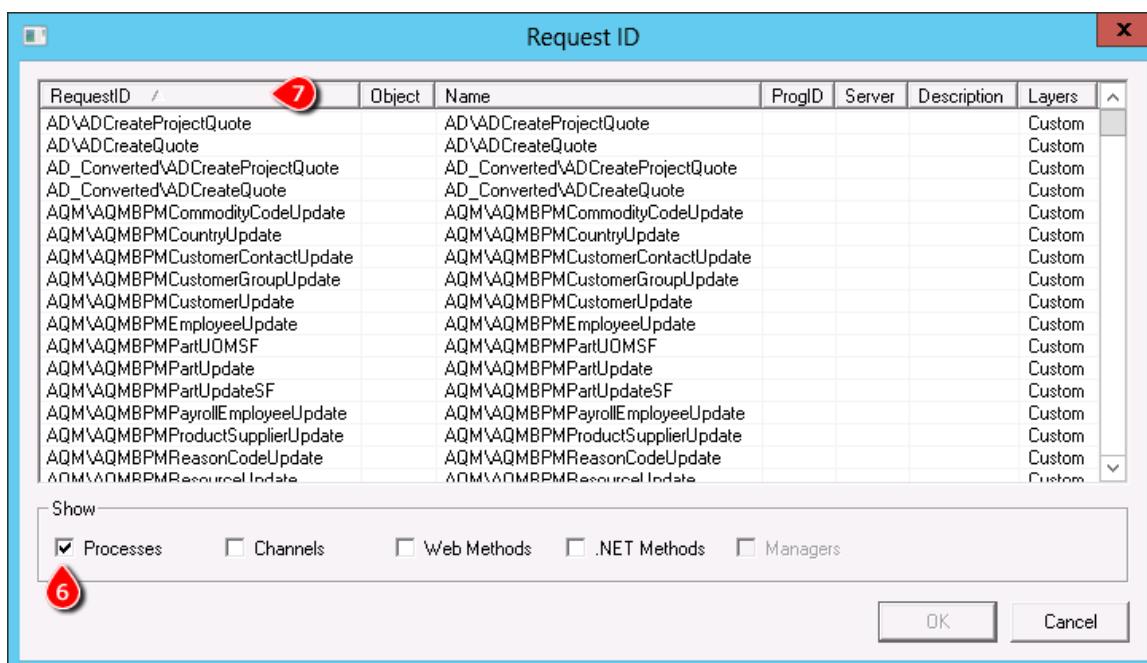


- Select the appropriate **Message type** and **Message action** for the intended Request ID.

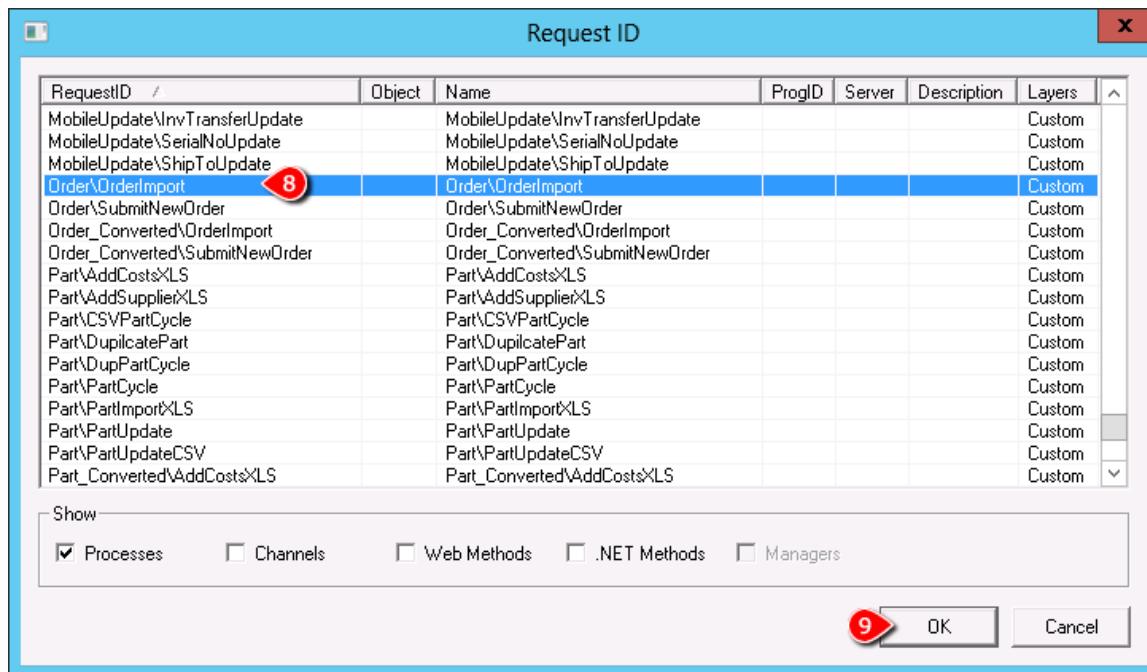
In this example, select **SalesOrder** and **SOUpdate**.

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

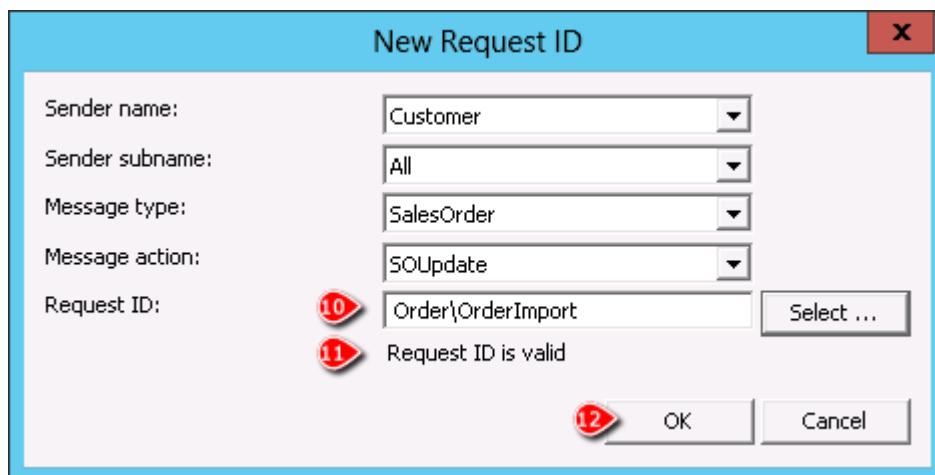
The **Request ID** window displays.



6. 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.
7. The **RequestID** column lists workflows in the <workflow package>/<workflow name> format.
8. Select the workflow that should process the incoming document.



9. Click **OK**.
10. Notice the **Request ID** field now displays the workflow package and workflow name.



11. Below the **Request ID** field, notice the **Request ID is valid** message.
12. 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 Workflow Designer chapter and the product-specific chapters for more details and examples on how to use service references in a workflow.

The following icons are used in Service Connect Administration Console to distinguish between different Service Reference types:

Icon	Web Service Reference Type
	Generic Web Service or other type of non-WCF Web Service
	Epicor 10 WCF Service
	Generic WCF Service or Epicor 9.05 WCF
	The stop sign marks the Service References that do not function. If you want to use this Service Reference, you must re-import it.

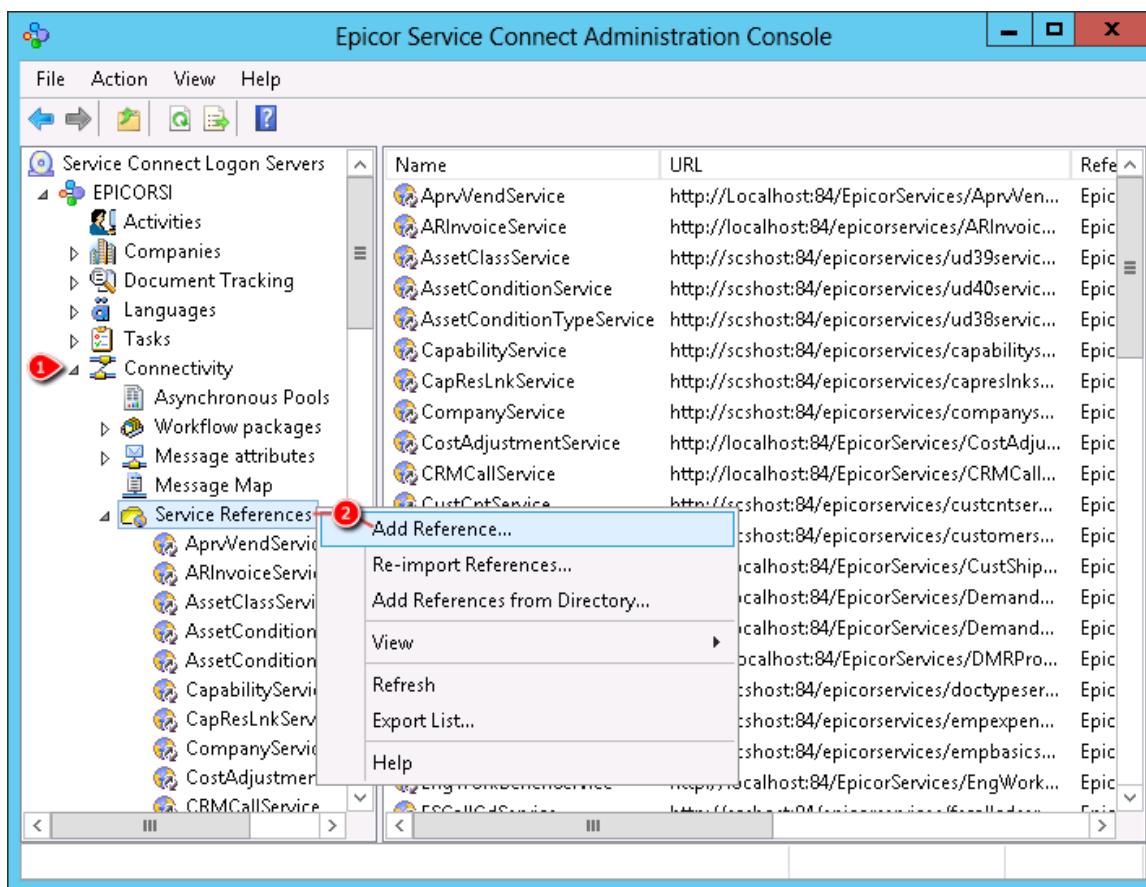
If a Web Service has custom SOAP header, it is visible in imported Web Service schema. You can use the custom SOAP header in XML Mapper and XPath Builder as other message elements.

For the detailed steps required to update the E10 web.config file to all SOAP references to connect to E10, refer to the Setting Up SSL and Enable HTTP endpoints sections in the Epicor WCF Services Developer Guide. You can also find the latest version of the Guide on [epicweb](#). You can find the same information in the Setting Up SSL and Enable HTTP Endpoints sections in Epicor ERP application help.

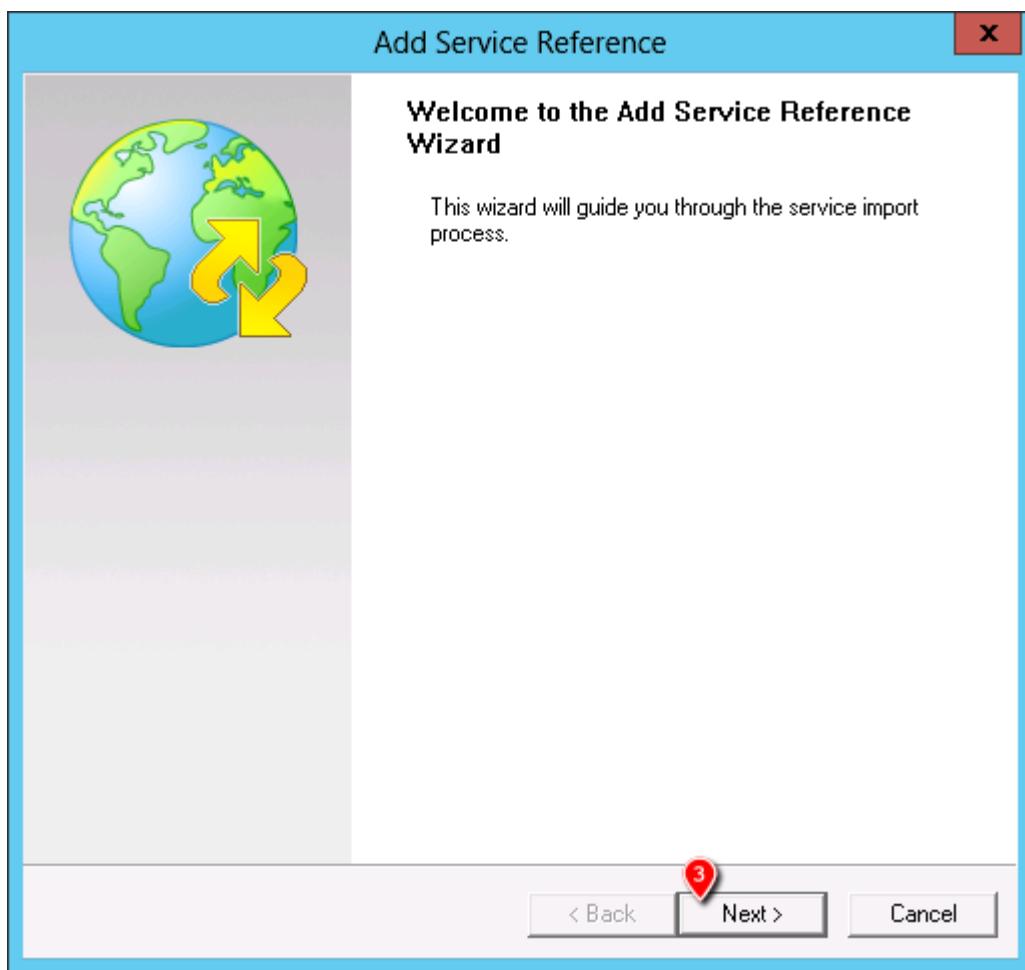
Add a Service Reference

To add a service reference:

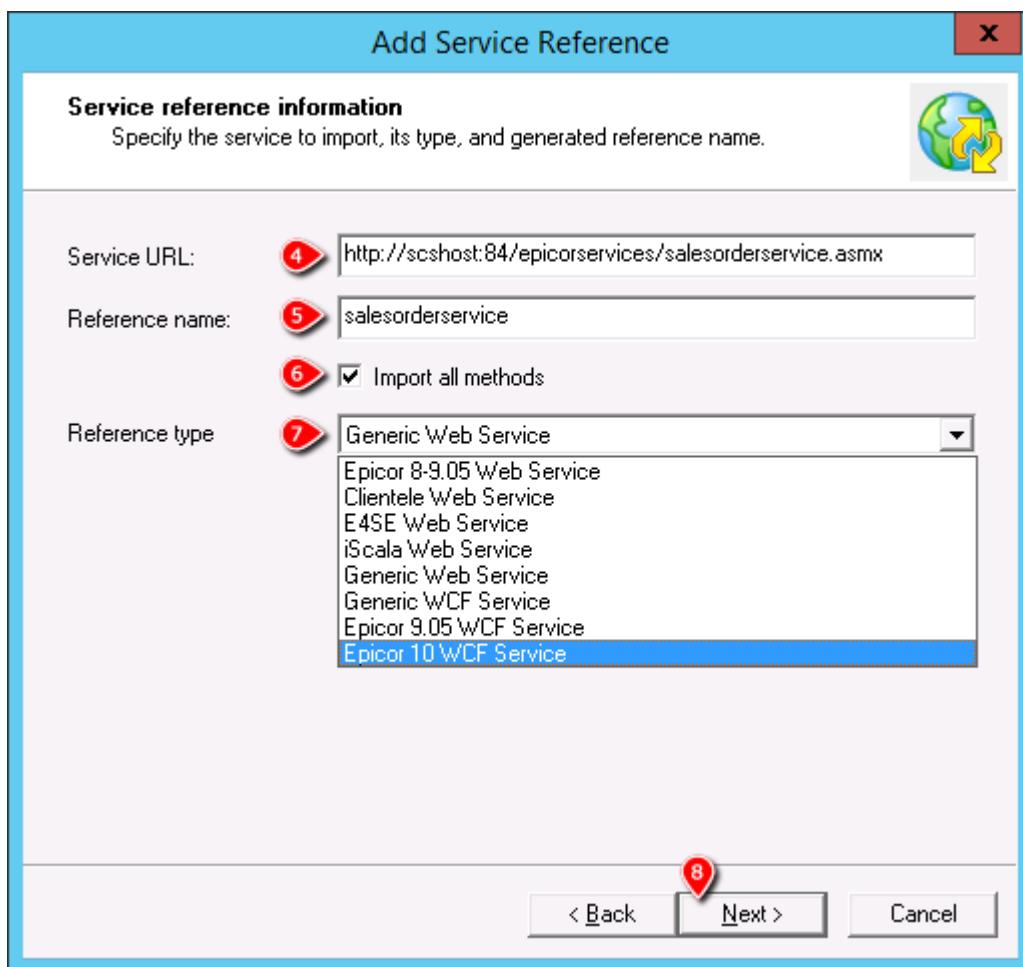
1. In the **tree view**, expand the **Connectivity** node.
2. Right-click **Service References** and select **Add Reference**.



3. On the **Add Service Reference** screen, click **Next**.



4. Enter the **Service URL**.



URL should point to a WSDL file.

5. Verify the **Reference name** field displays the default service reference name.

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

Available types are:

- Epicor 8-9.05 Web Service (applicable to Epicor ERP version 8 and 9 web services)
- Clientele Web Service
- E4SE Web Service
- iScala Web Service
- Generic Web Service
- Generic WCF Service
- Epicor 9.05 WCF Service (applicable to Epicor ERP version 9 WCF services)

- Epicor 10 WCF Service

You can select **Generic** for any service.

Windows Communication Foundation (WCF) is Microsoft's unified programming model to build service-oriented applications. You can use it to build secure, reliable, transacted solutions that integrate across platforms and interoperate with existing investments. For details on WCF type, refer to MSDN web-site.

8. Click **Next**.

Follow the wizard steps and set type-specific settings, such as user name, password and domain (if the Web Service requires windows authentication), and so on.

9. Complete the **Security settings**.

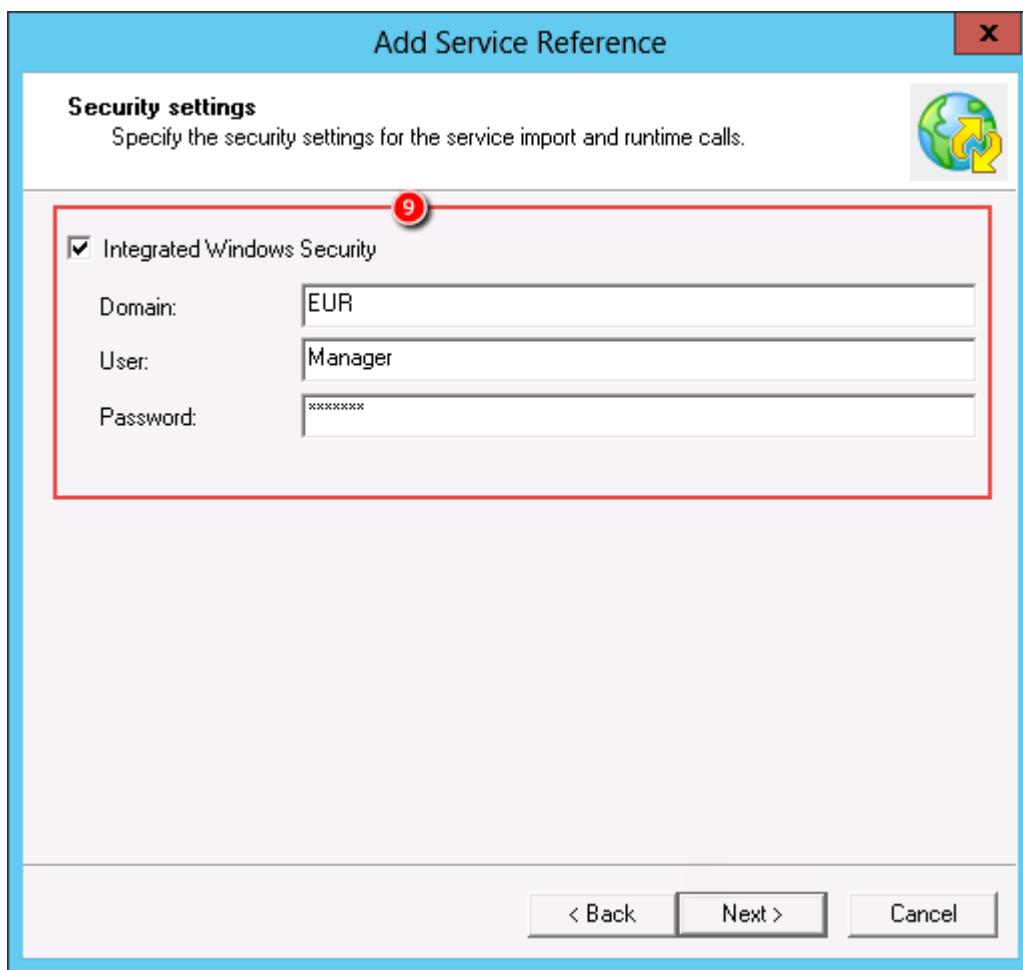
Depending on the Reference Type you selected, you may have different options on your Security settings screen. Refer to the product-specific chapters for details on how to set up security for your product.

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

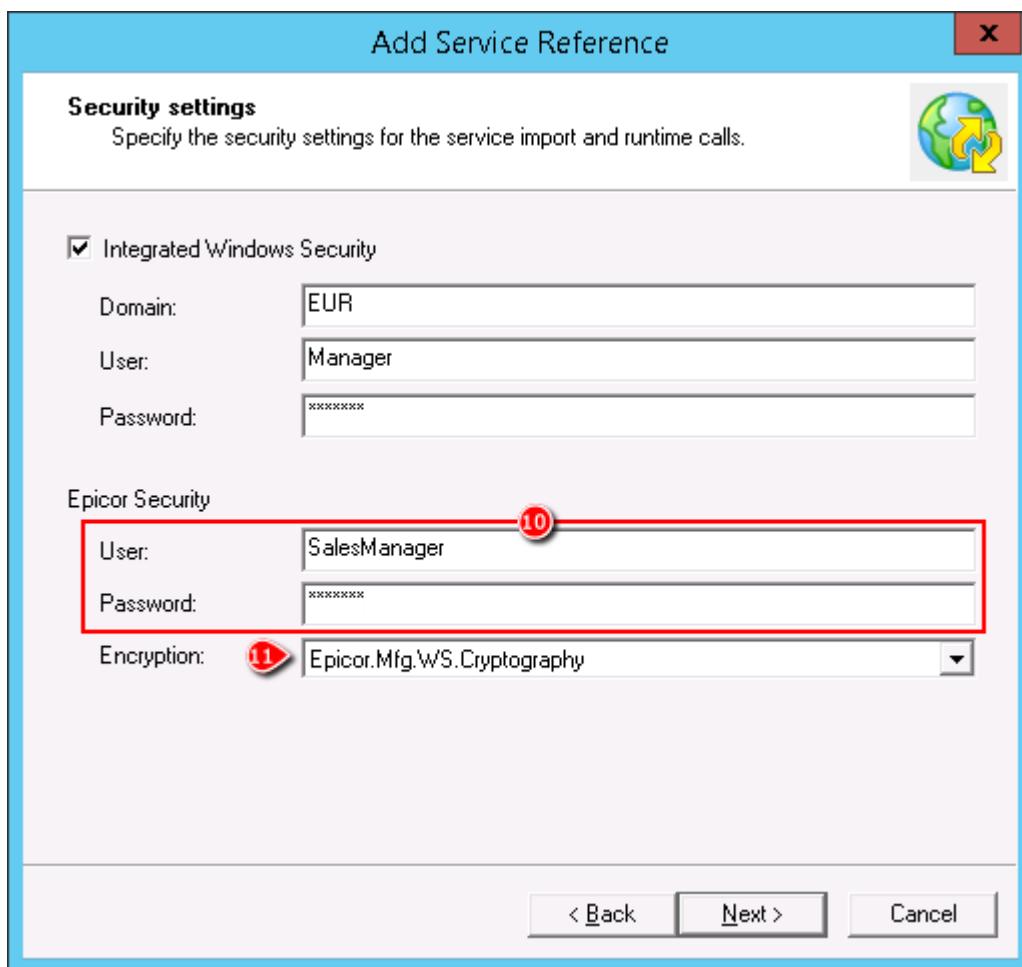
- Select **Integrated Windows Security** and enter a **Domain**, **User**, and **Password** to supply the Windows credentials needed to access the service.
- Select **Integrated Windows Security** and do not enter credentials. In this case, the Windows account used to run the ScaDESRouter service is used.

For Epicor ERP version 9 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 version 9 or Vantage server.

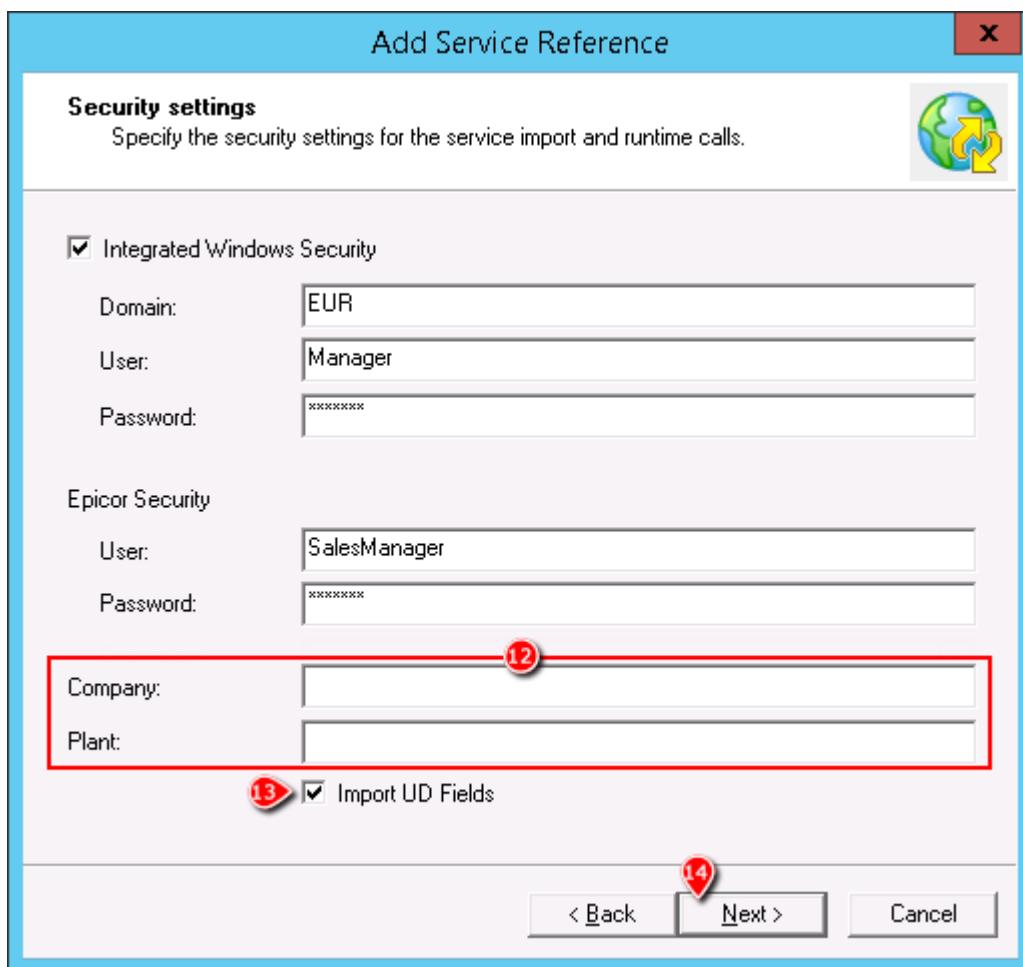
- Do not select Integrated Windows Security. In this case, the Windows account used to run the ScaDESRouter service is used.
- If you selected Generic Web Service type, and the service is configured to use a Web Services Enhancements (WSE) for Microsoft .NET policy file, click Next and on the Policy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



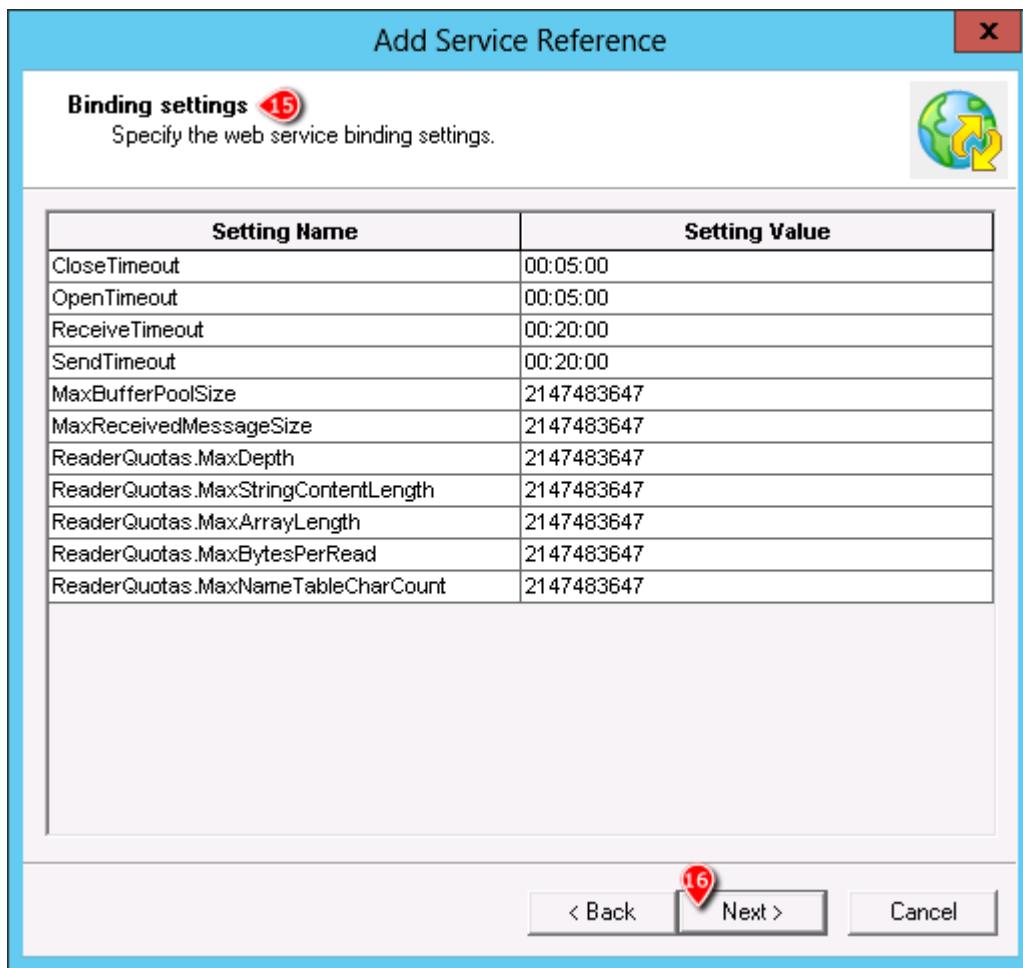
10. If you import one of the Epicor service types, under **Epicor security**, specify Epicor **User** (Epicor Security Name) and **Password**.



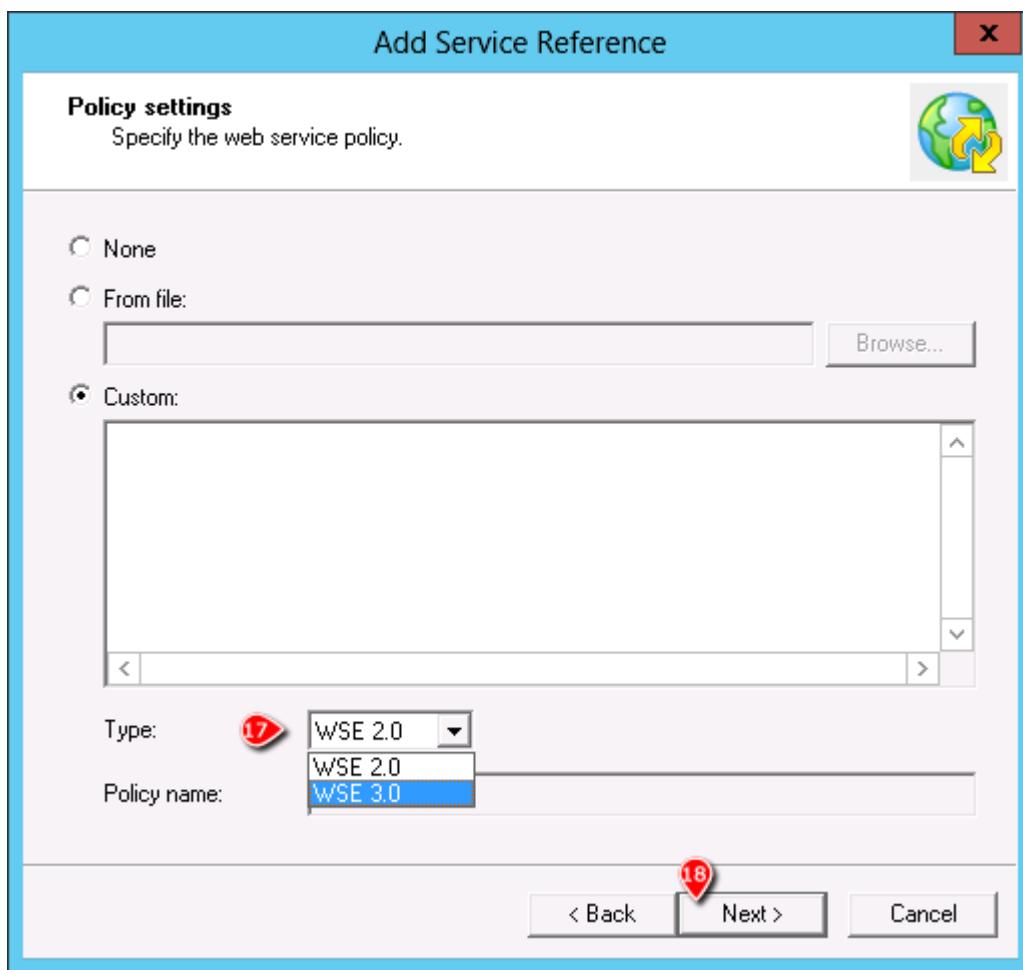
11. If you import Epicor 9.05 WCF Service, you can select the **Encryption**.
12. If you import Epicor 10 WCF Service, in the **Company** and **Plant** fields, specify Epicor Company and Plant.



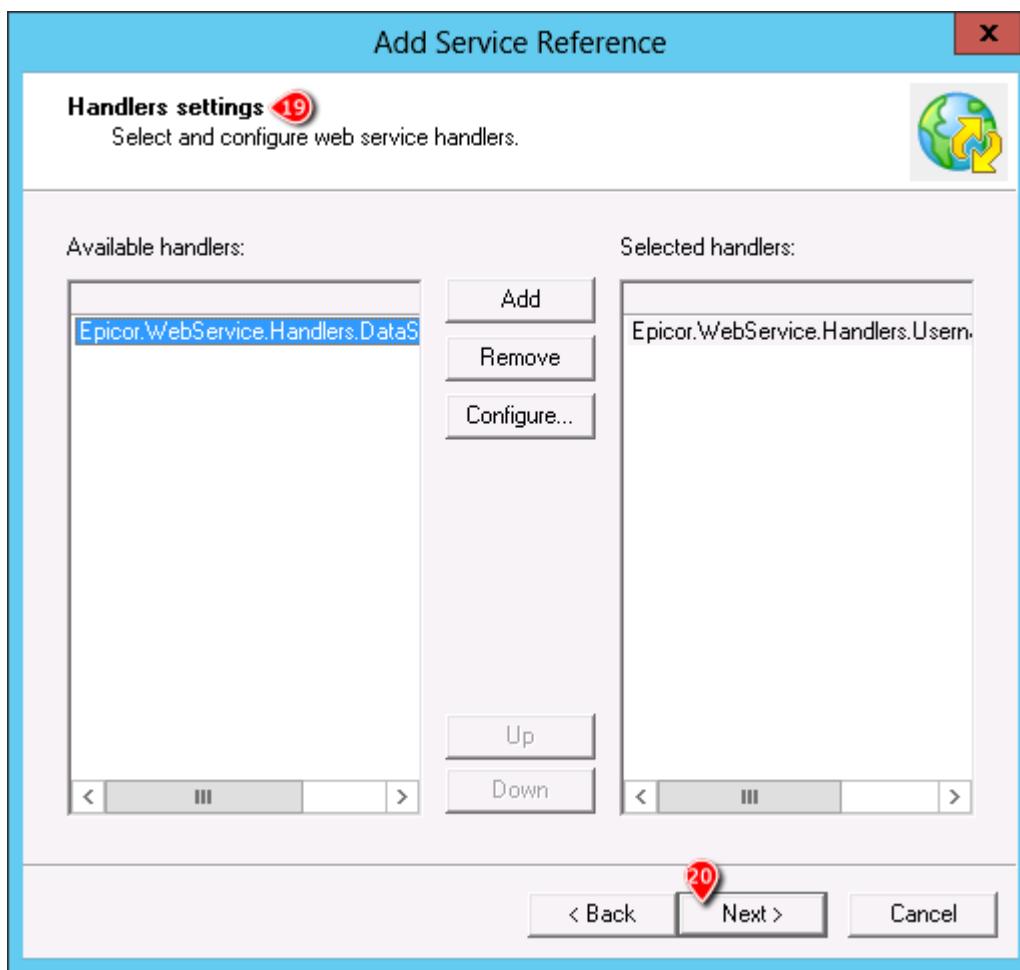
13. If you import Epicor 10 WCF Service, select the **Import UD fields** check box to import Epicor BOs with Extended user defined fields (stored in database).
If you select the Import UD fields check box, Service Connect identifies BOs with UD fields and imports them (generates schema, and so on) in such a way that UD fields are visible in BO dataset in Service Connect and can be mapped in XML Mapper.
14. Click **Next**.
15. If you selected one of the WCF Service types, specify the web service client binding settings on the **Binding Settings** screen.



16. Click **Next**.
17. If you selected Generic Web Service type, on the **Policy settings** screen, in the **Type** field, select WSE 2.0 or WSE 3.0. Review the Policy settings.
If you select WSE 3.0, in the **Policy name** field specify the Policy name.



18. Click **Next**.
19. If you selected Generic Web Service type, on the **Handlers settings** screen, select and configure handlers for your web service..

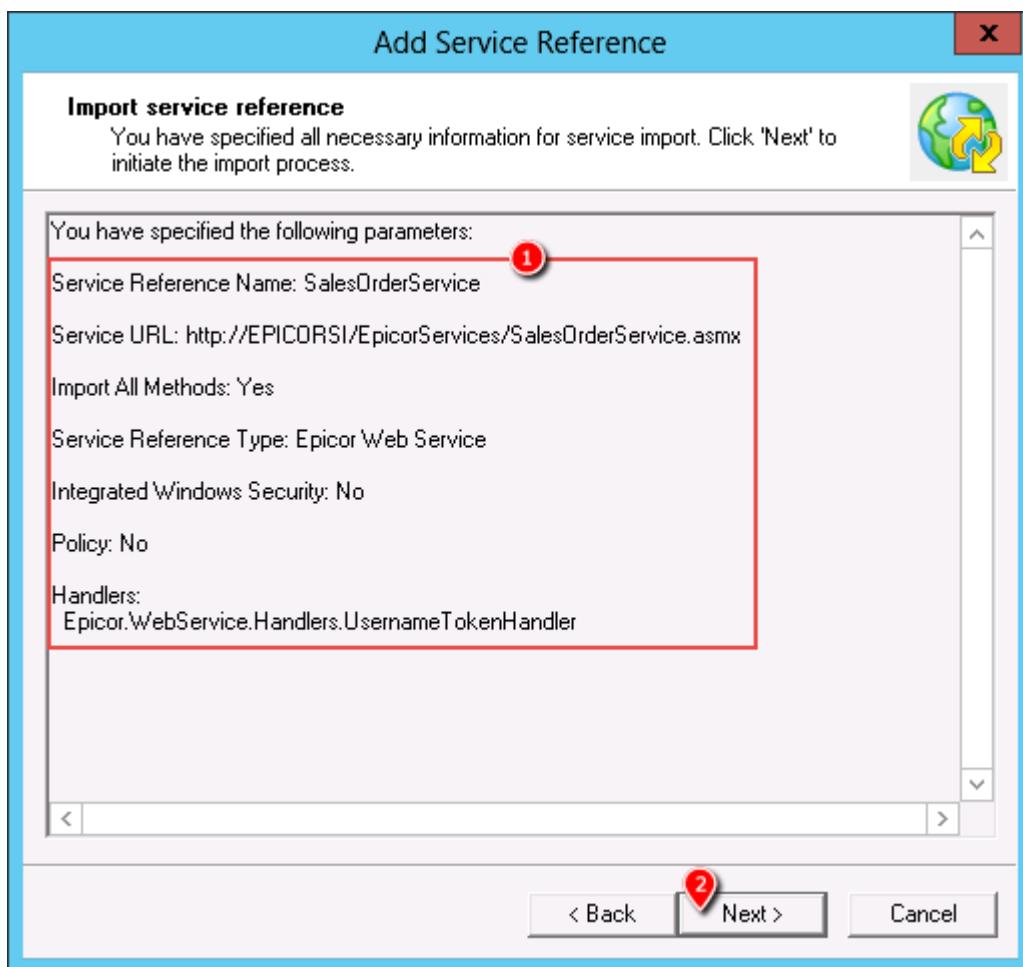


20. Click **Next**.

Finalize Service Reference Import

After you specify all the service reference import details, review the information you provided and start the import process.

1. Review all the information you provided about the Web Service that displays on the wizard screen.



2. Click **Next** to start import process, or **Back** to change the information.
3. When the import is complete, a detailed log displays on the **Review import log** screen.
You can use the **Copy** button to copy and save a copy of the log.

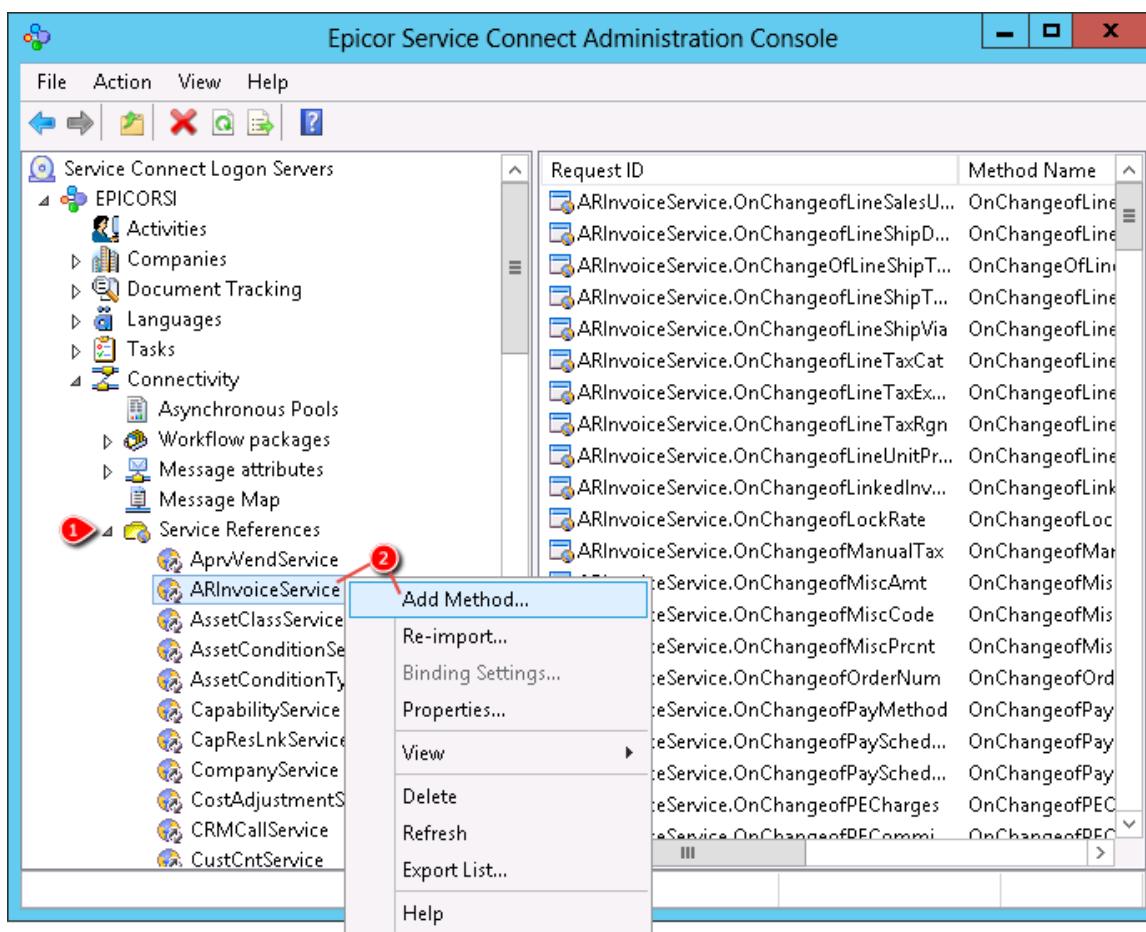
If the Web Service Reference import failed, you can click the **Back** button to go back and adjust the reference settings, or, you can select **Cancel** to abort the import operation.

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

Add Service Methods

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

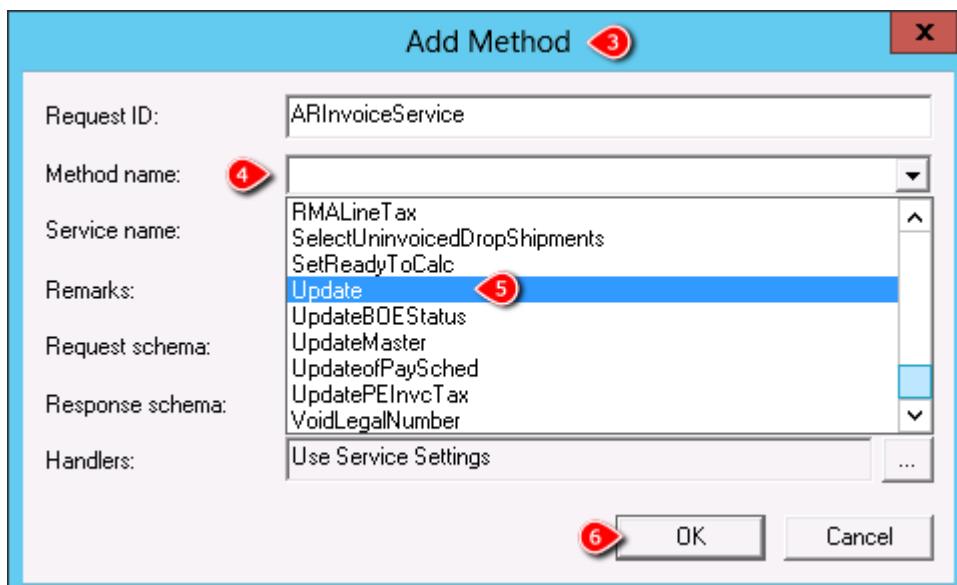
1. In the tree view, expand the **Connectivity > Service References** node.



2. Right-click the service reference to which you want to add a method and select **Add Method**.

In this example, select ARInvoiceService.

3. The **Add Method** window displays.



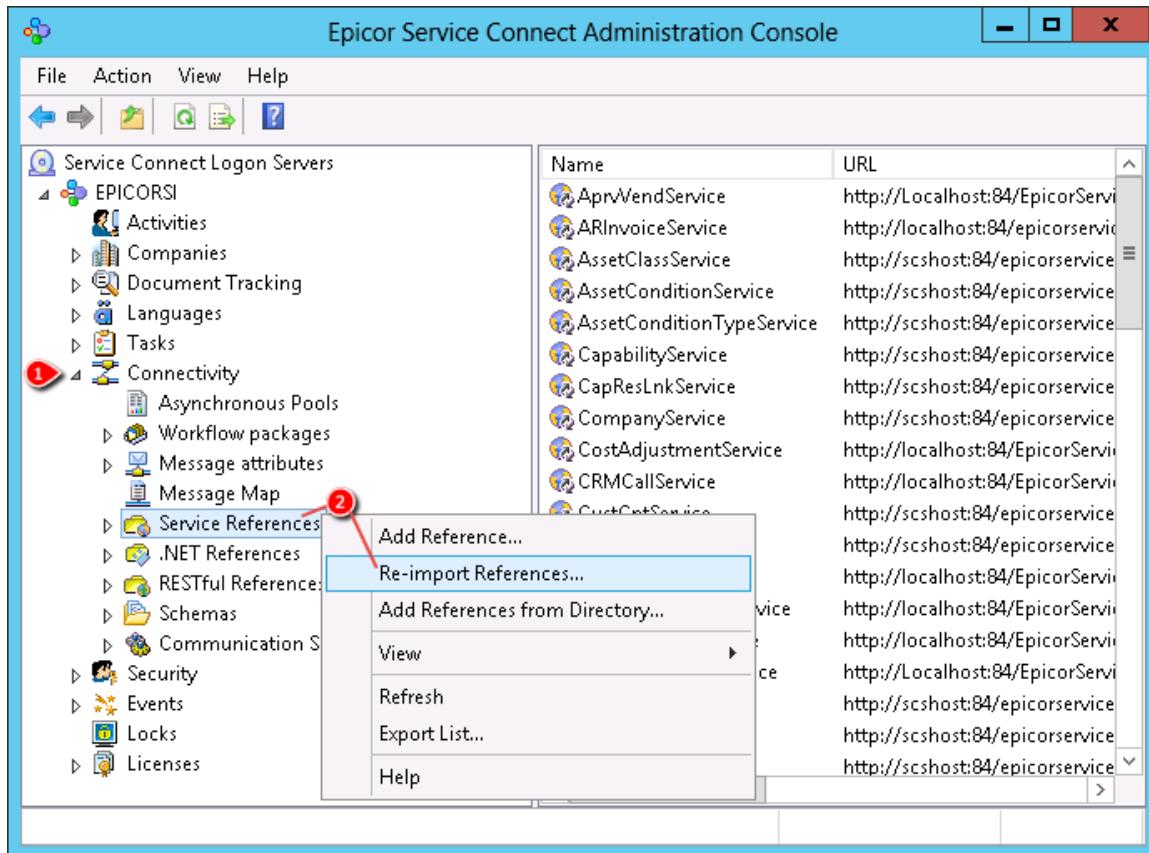
4. Click the **Method Name** field.
5. Select a method.
6. Click **OK**.

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.

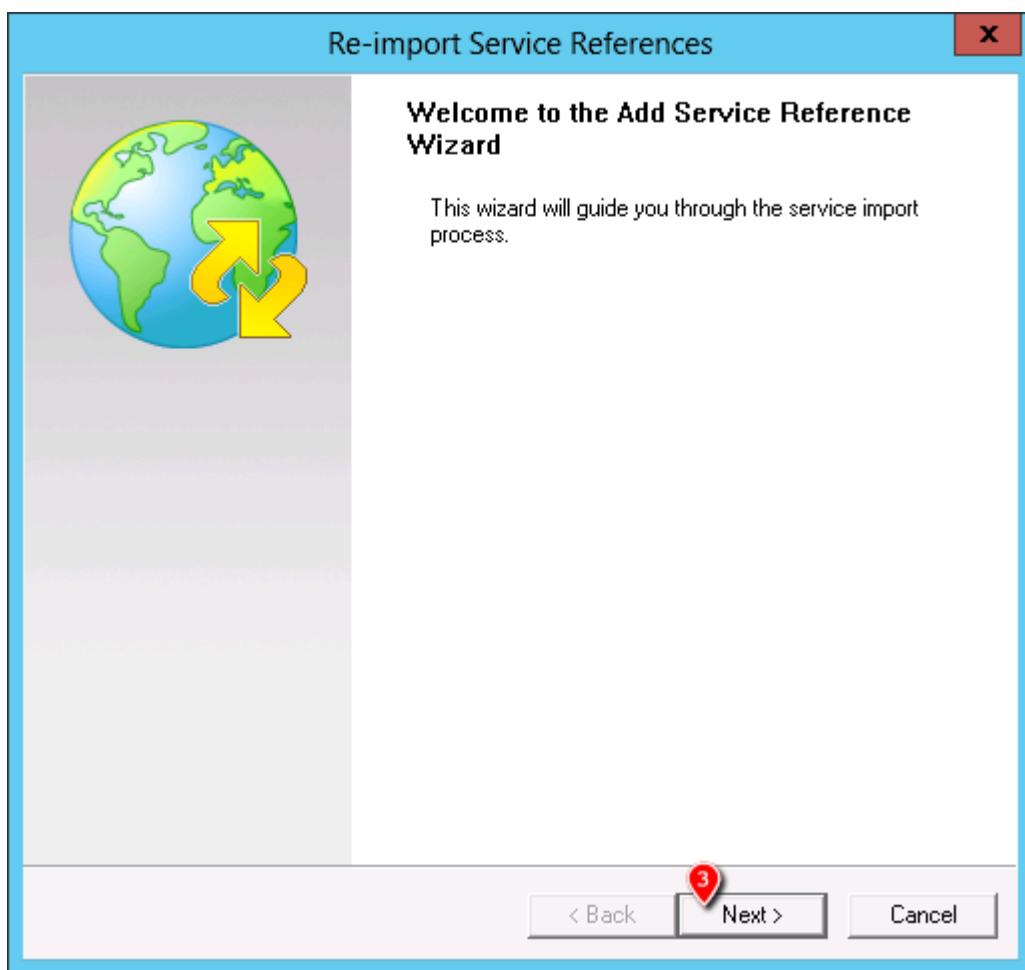
To re-import services:

1. In the **tree view**, expand the **Connectivity** node.



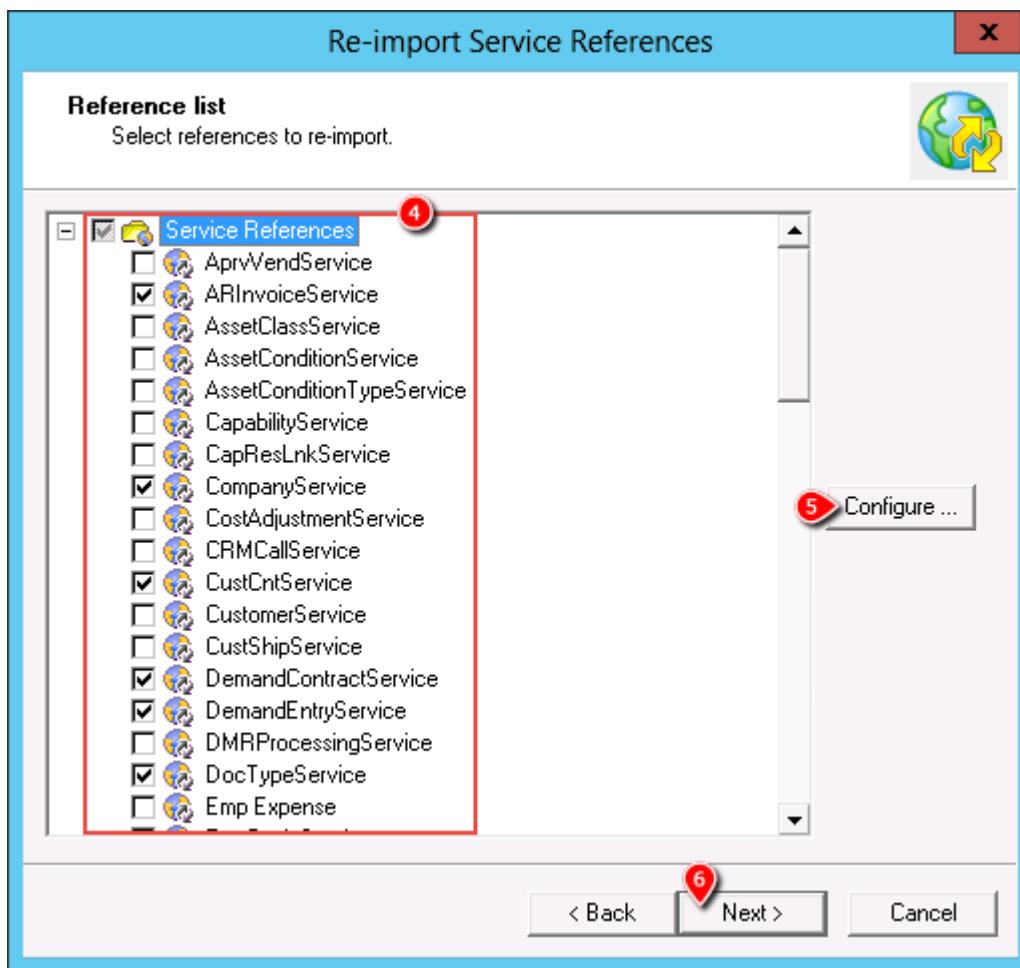
2. Right-click **Service References** and select **Re-import References**.

3. In the **Re-import Service References** window, click **Next**.



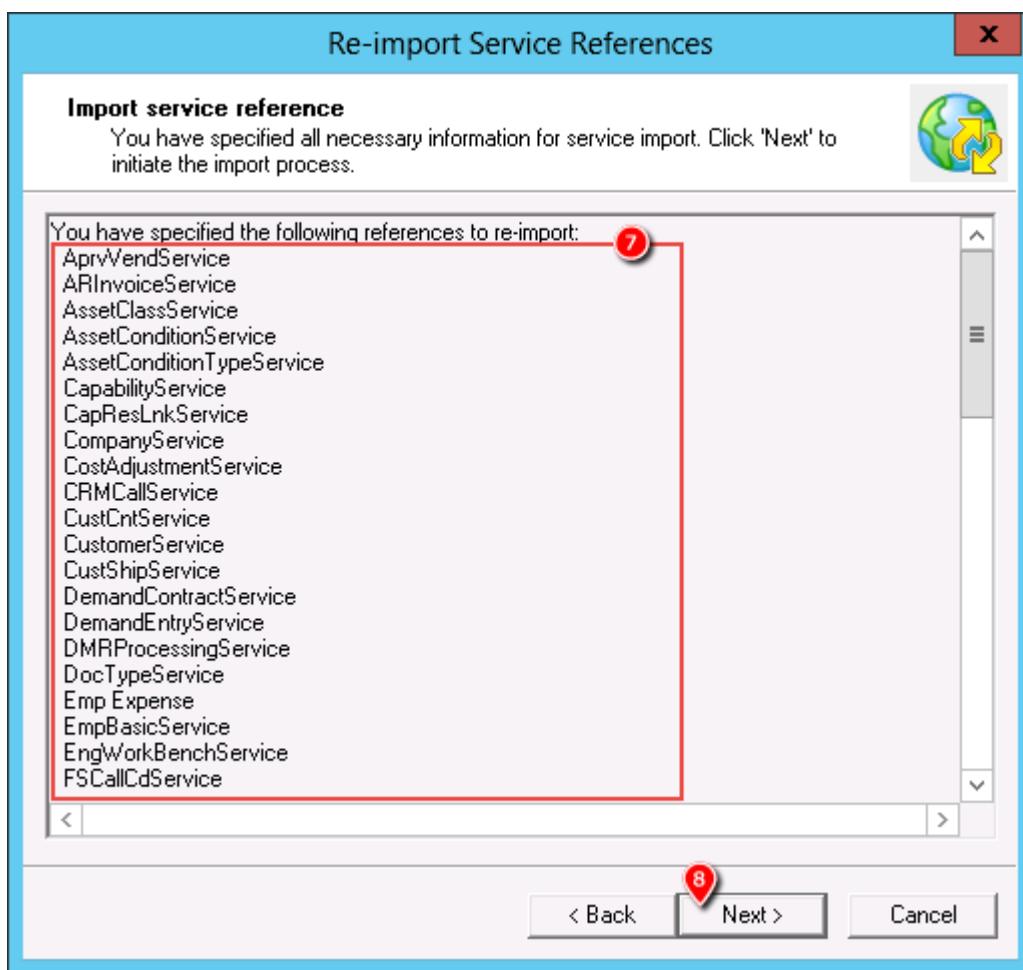
4. From the **Reference list**, select the references to re-import.

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. On the **Reference list** screen, 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.

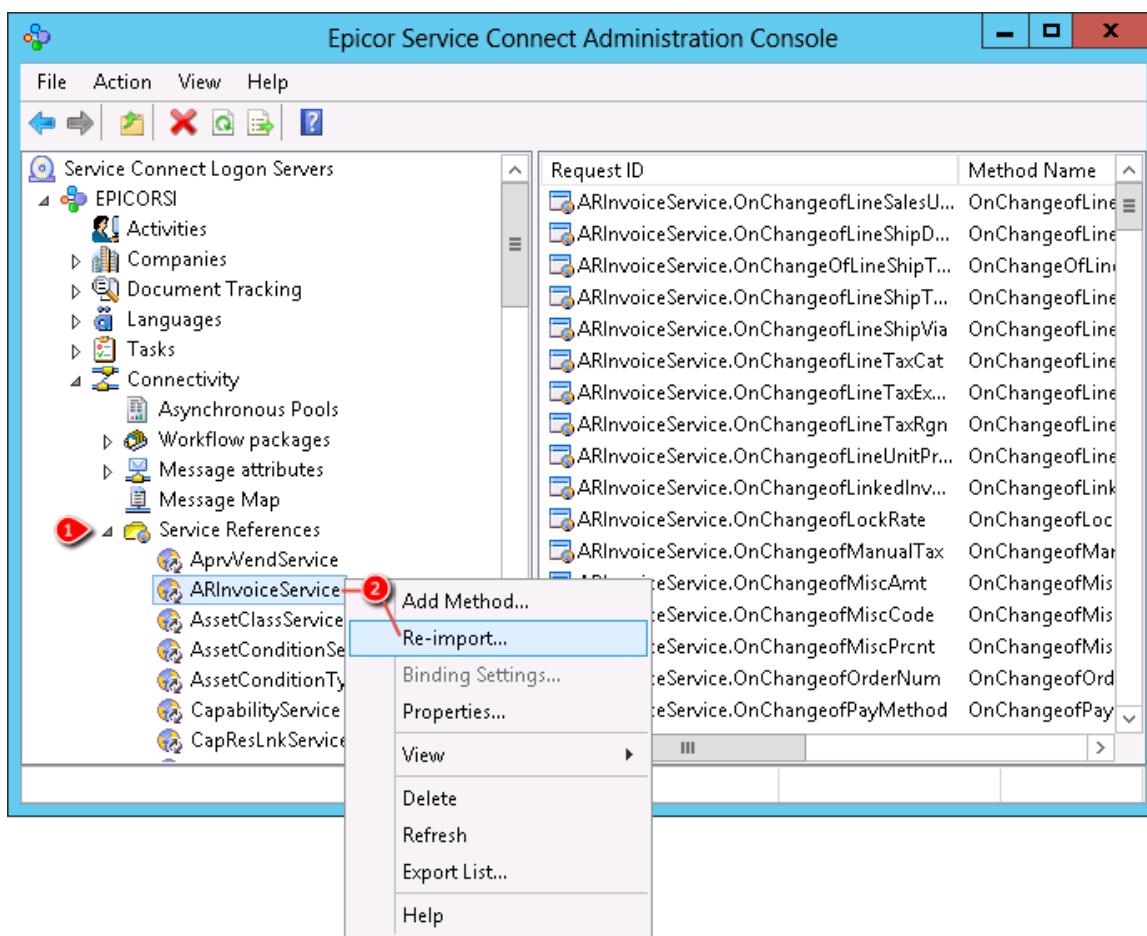
10. You can use the **Copy** button to copy the re-import log to the clipboard.

11. Click **Finish**.

Re-Import an Individual Web Service

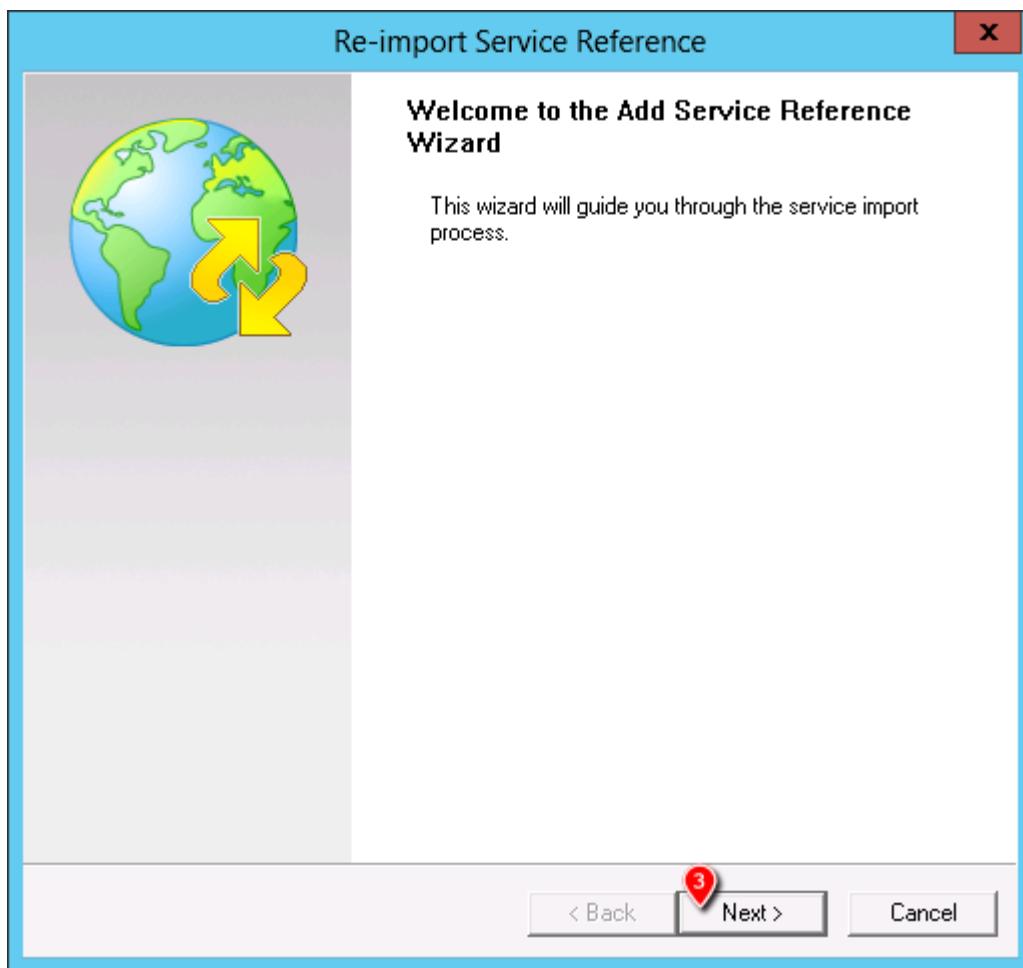
You can also re-import an individual web service.

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. On the **Re-import Service Reference** screen, 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-importing log to the clipboard.

5. Click **Finish**.

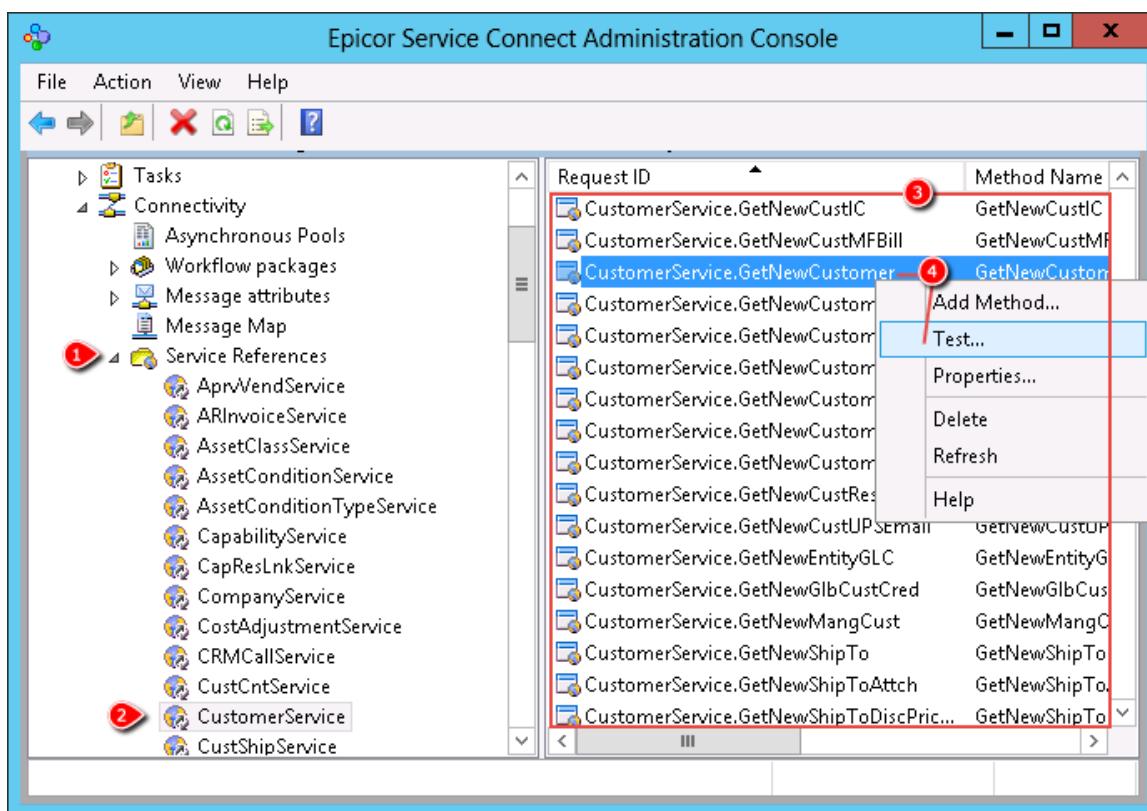
Depending on the number of services you re-import, the process can take several minutes. When the process is complete, the application displays a dialog box that shows the settings used to re-import each service plus a summary that shows how many services were successfully and unsuccessfully re-imported.

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 References** node.



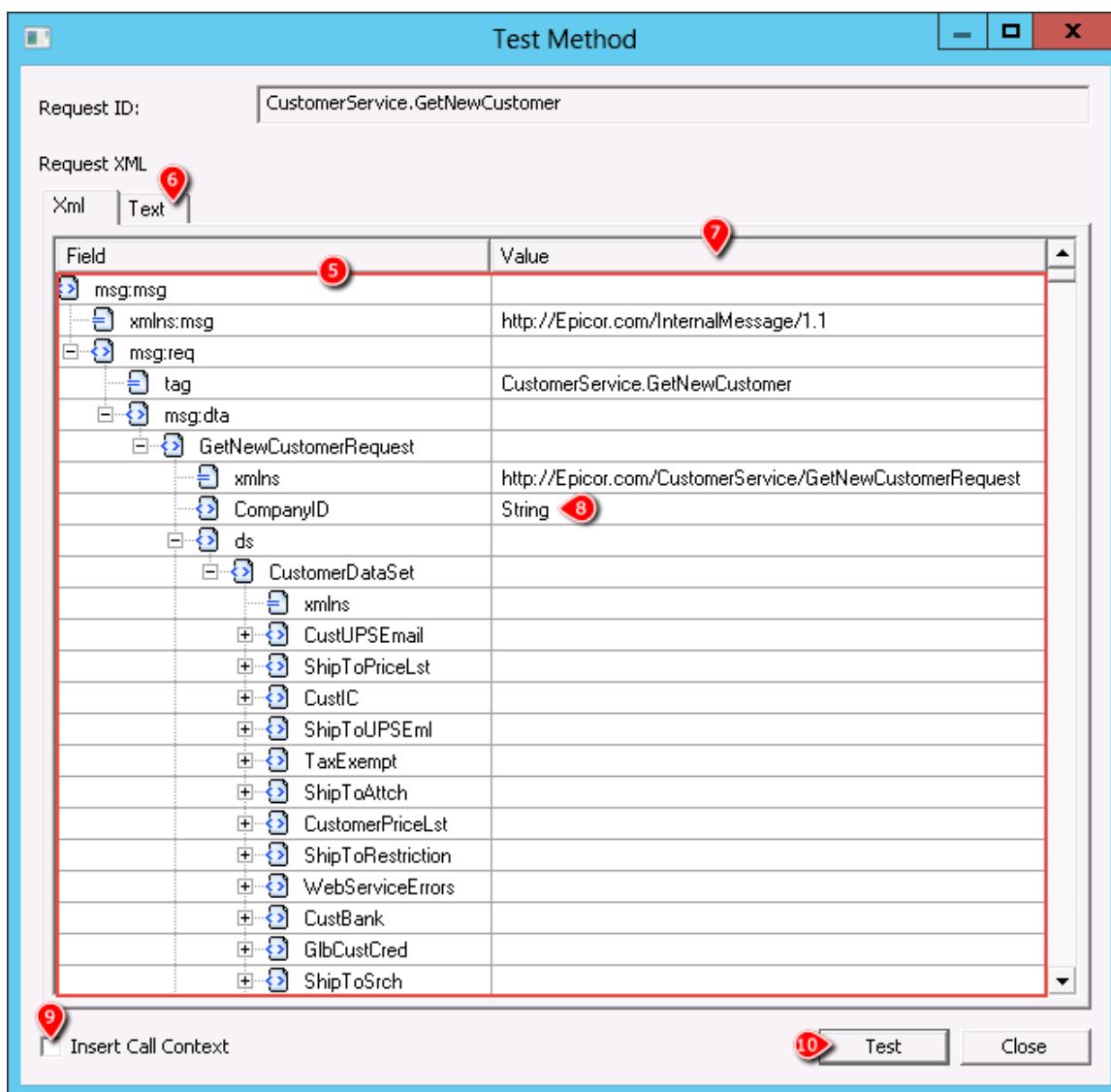
2. Select the Service Reference to test.

In this example, CustomerService is selected.

3. In the right pane, review the list of its methods.

4. In the right pane, right-click the method to test and select **Test**.

5. The **Test Method** dialog displays the request XML structure in **Xml** tab.



The request XML is a message template. You must replace the placeholders with meaningful data before you send the request.

6. Open the **Text** tab to view the xml text.
7. In the **Value** column, enter the criteria the tested method expects.
8. In this example, you specify the CompanyID for which you want to perform the connectivity check. In the **CompanyID** node, you must remove **String**, or replace it with a real company ID.
9. If the **Insert Call Context** check box is available, select it to be able to include call context into the request XML.

Use this option to specify additional data, for example BPM data.

The Insert Call Context option is available for web services which expose the setting, for example, Epicor E10 WCF Web services.

If you select the **Insert Call Context** check box, the CallContext node is added to the request XML, and you can fill its sub-nodes with Epicor-readable data.

The Insert Call Context option is available for web services which expose the setting, for example for Epicor E10 WCF Web services.

callContextIn	<p>This node is added if you test E10 WCF reference.</p> <p>Use callContextIn/BpmData/BpmDataRow node of the Request to pass data for the first Info Prompt.</p> <p>The data from callContextIn node are always used for Web service call. If Epicor requests additional information for other Info Prompts, the data from optionalContextIn node is used.</p> <p>If all InfoPrompt requests were processed with corresponding information from callContextIn and optionalContextIn nodes, then the response error section keeps Info messages, otherwise the error section keeps Logical error messages with Info Prompt form name.</p>
optionalContextIn	<p>An additional node Epicor ERP uses to return method execution context information.</p> <p>The structure of this node is defined by Epicor ERP. For example, additional BpmData fields for Info Message forms.</p>

If all data were provided to request, the Information message is logged to the Service Connect Events log. If there was an InfoPrompt without the corresponding information in callContextIn and optionalContextIn nodes, the Error message with details is logged.

Service Connect uses CallContextIn and optionalContextIn fields for InfoPrompt fields initialization, if they were assigned for the called Web service method.

For Epicor 10 WCF Service References, the callContextIn and optionalContextIn should be used to provide data for mandatory Info Prompts which are assigned to the Web method. Data from callContextIn is initially passed to Epicor with other parameters of Web method. If Epicor decides that information in callContextIn is not enough then it requests Service Connect for additional information passing Info Prompt Form ID. Service Connect looks through optionalContextIn/BpmDataContext/FormID nodes to find corresponding information by FormID and provides data from optionalContextIn/BpmDataContext/BpmData back to server. Note, at this step data from optionalContextIn overrides data from callContextIn if it is overlapped. During one WCF method execution Epicor 10 server can request Service Connect multiple times to get data for all Info Prompts which are assigned to the method using BPM Directive. The call is successful if all Info prompt requests were resolved. Service Connect logs each interaction with Epicor to Service Connect Events log.

10. Click **Test**.
11. The connectivity check is performed and the response message displays.
12. 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.
13. Click **Close**.

Logging Web Service Calls

After calling an Epicor 10 WCF Web Service, Service Connect checks whether Epicor provided an InfoMessage. If an InfoMessage was provided, a record is made in the ESC Events log for every InfoMessage. The record type depends on the InfoMessage Severity, for example Warning.

Examples:

```
Epicor BPM Info Message
```

```
BO: <BO>
Method: <Method>
UserIdent: <UserIdent>
Company: <Company>
Plant: <Plant>
Version: <Version>
Display Mode: <DisplayMode>
Severity: <Severity>
Message Text: <MessageText>
```

```
BO: ABCCode
Company: EPIC06
DisplayMode: 0
MessageText: EMK ABC Code Warning Message Body
Method: GetByID
Plant: MfgSys
Severity: 2 (Warning)
UserIdent: MANAGER
Version: 0
```

```
BO: ABCCode
Company: EPIC06
DisplayMode: 0
MessageText: ABCCode = A
Method: GetByID
Plant: MfgSys
Severity: 0 (Information)
UserIdent: MANAGER
Version: 0
```

For E10 WCF Service references, the Information message is logged in the Service Connect Events log if all data were provided to request. If there was an InfoPrompt without the corresponding information in callContextIn and optionalContextIn nodes the Error message with details is logged and logic errors are added to es section. In this case, Web service does not return response and workflow continues processing.

.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.

Only the following Epicor 9.05 assembly types can be imported:

- Epicor.Mfg.BO
- Epicor.Mfg.Proc
- Epicor.Mfg.Rpt

Only the following Epicor 10 assembly types can be imported:

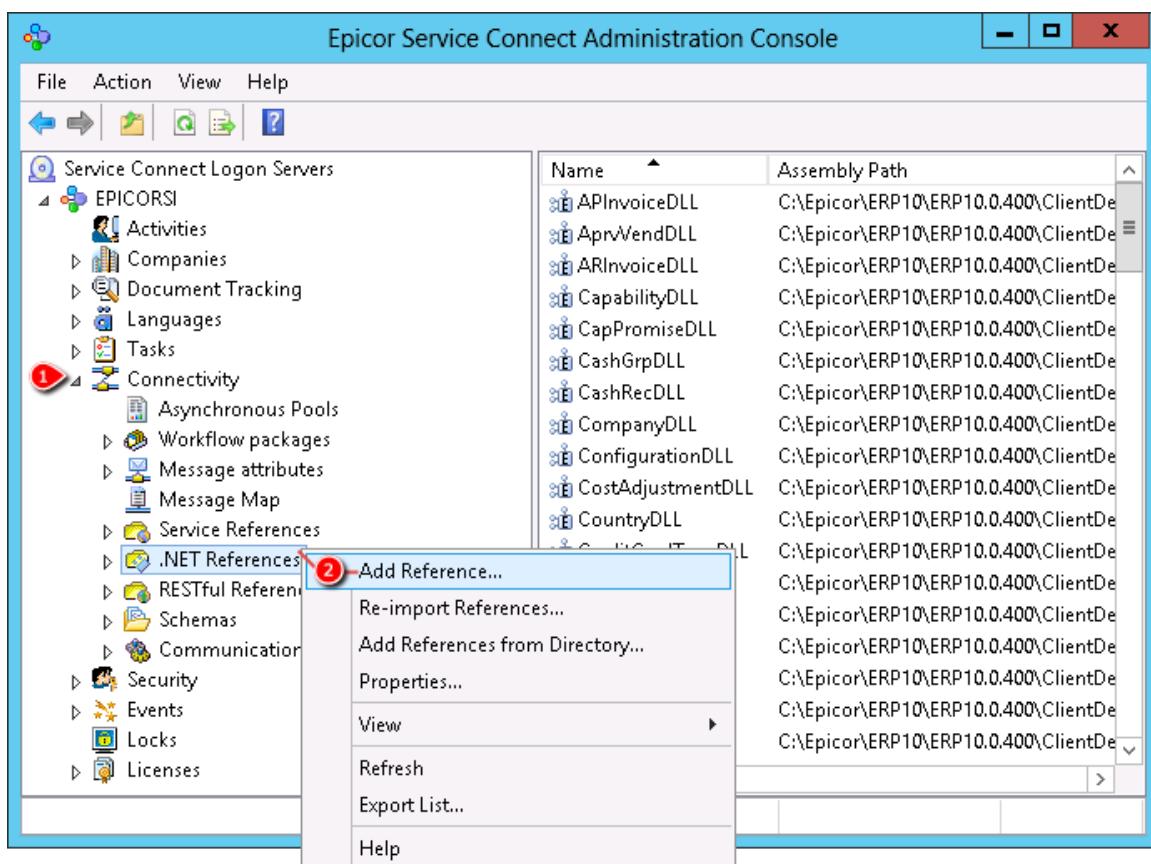
- Erp.Contracts.BO
- Erp.Contract.lib
- Erp.Contracts.Proc
- Erp.Contracts.Rpt
- Ice.Contracts.BO
- Ice.Contract.lib
- Ice.Contracts.Proc
- Ice.Contracts.Rpt

For Generic assembly import, there are no restrictions on class names.

Add a .NET Reference

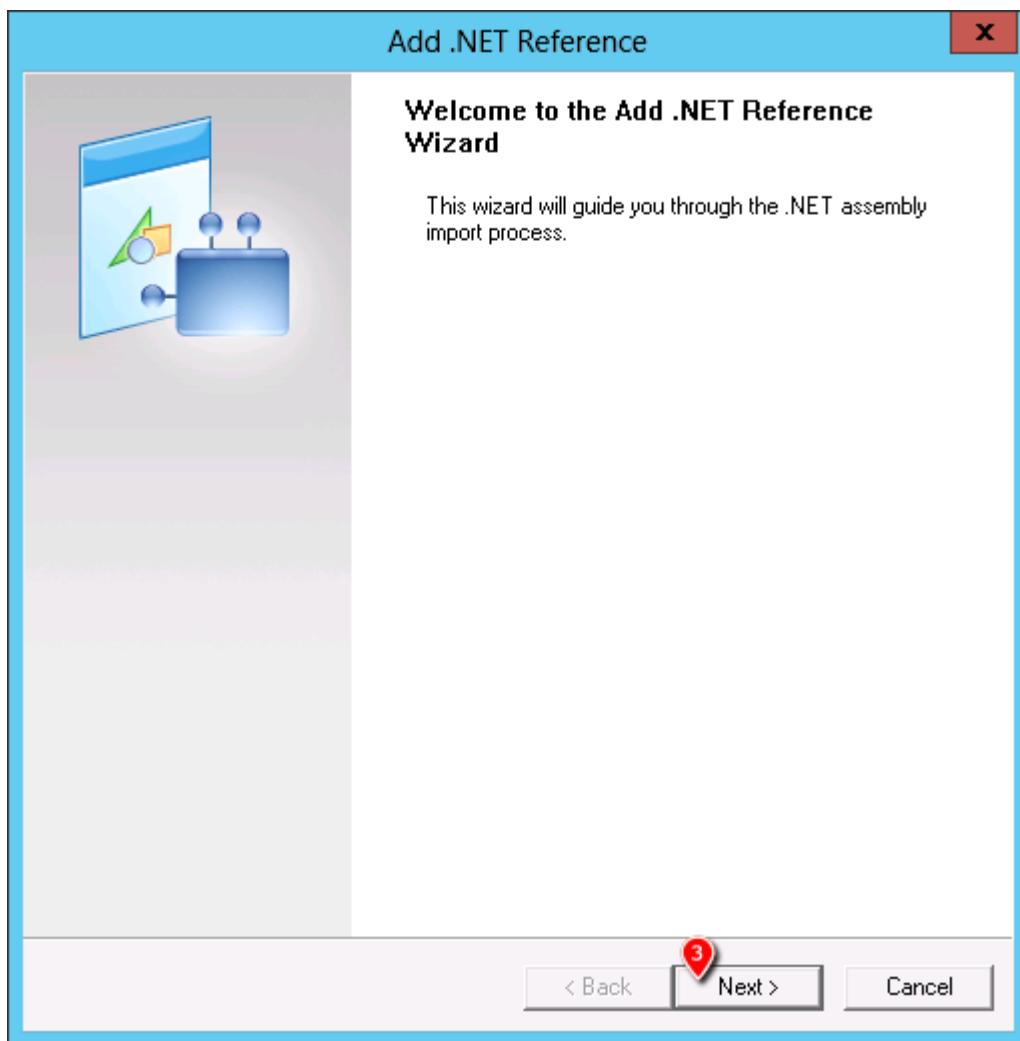
To add a .NET reference:

1. In the **tree view**, expand the **Connectivity** node.



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

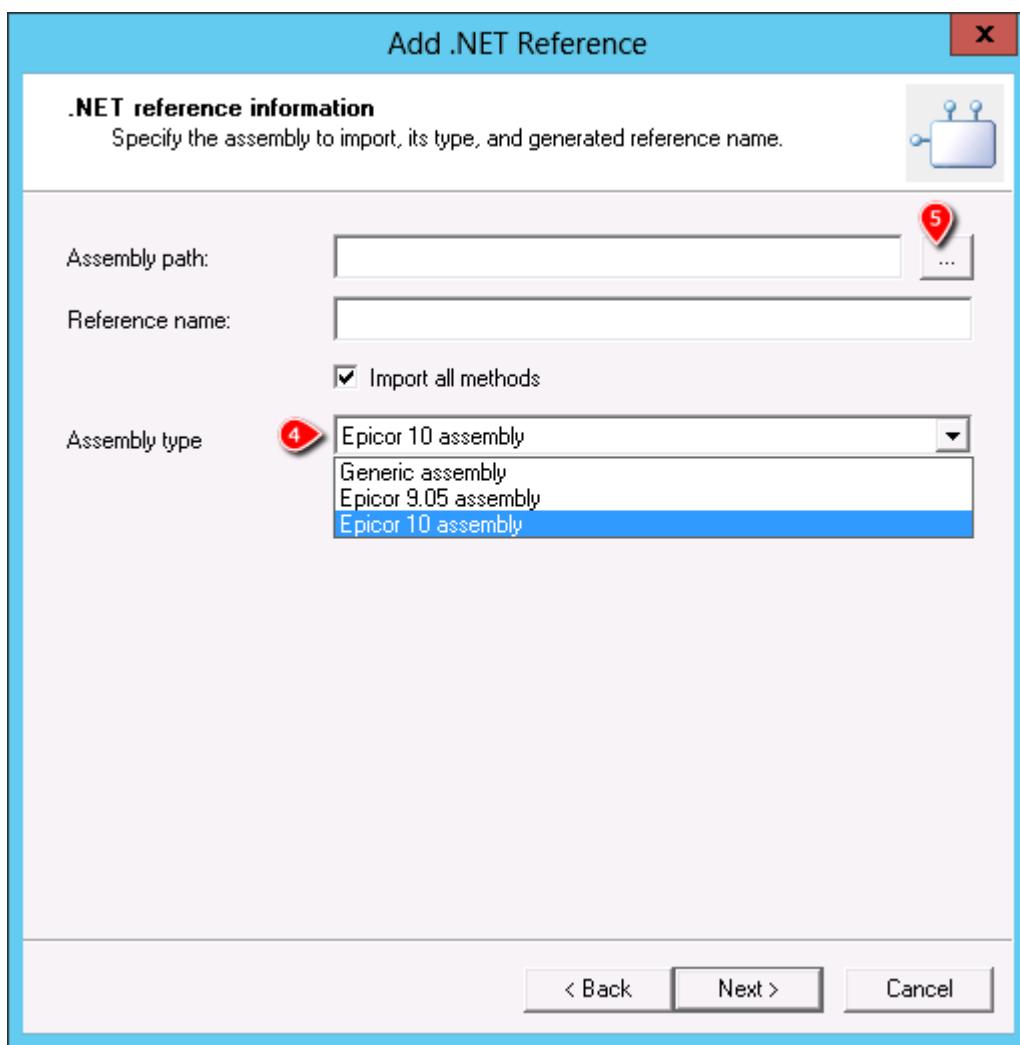
3. On the **Add .NET Reference** wizard welcome page, click **Next**.



4. On the **.NET reference information** page, click the **Assembly type** drop-down list and select the .NET reference type. The following types are available:

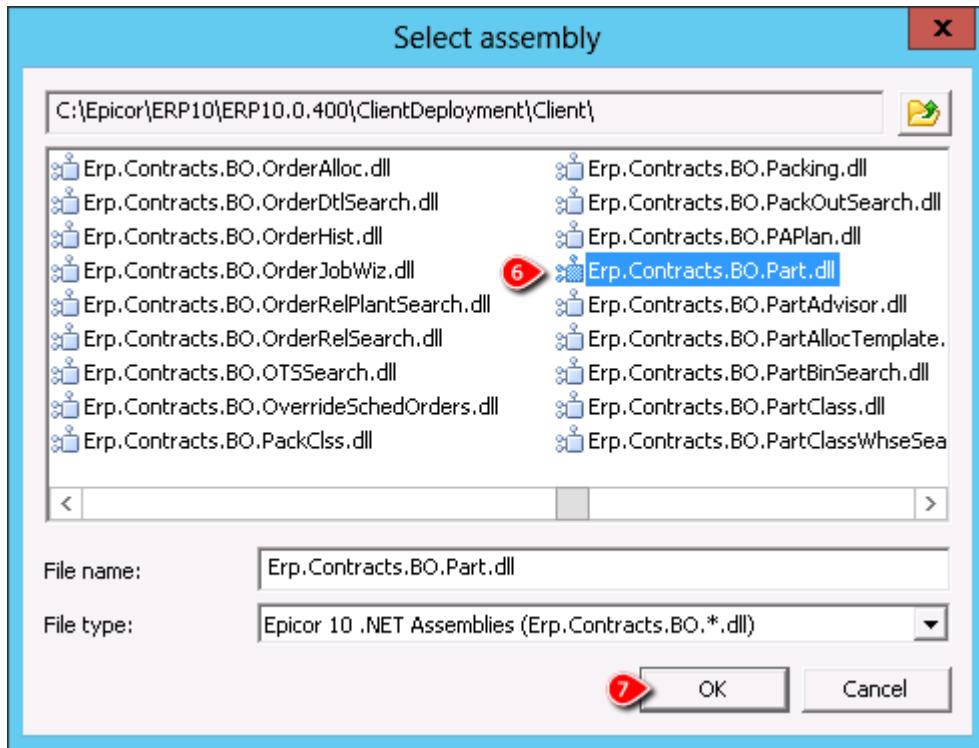
- Generic assembly
- Epicor 9.05 assembly
- Epicor 10 assembly

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



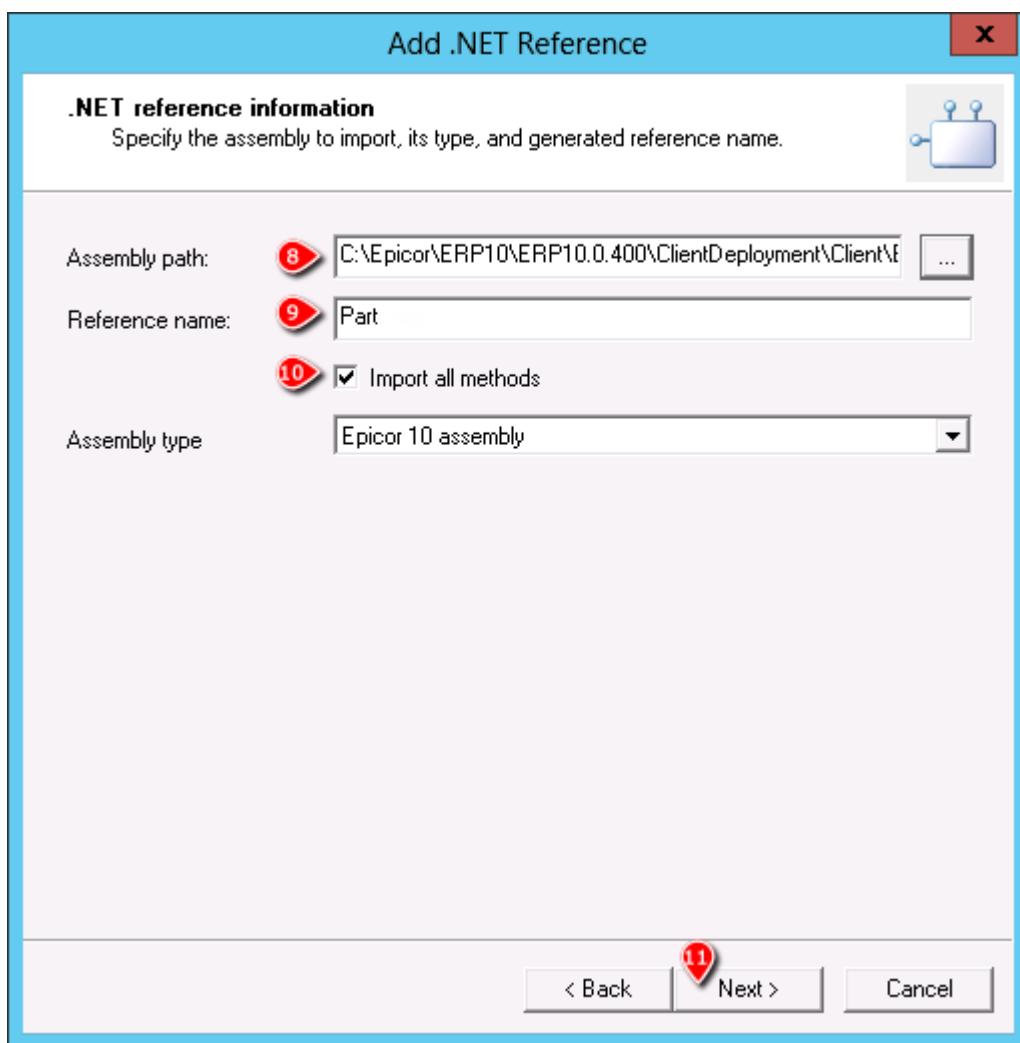
5. Next to the **Assembly path** field, click the ... (Ellipse) button.

6. In the **Select assembly** window, find and select the .NET assembly file.



7. Click **OK**.

8. On the **.NET reference information** page, notice, the **Assembly path** field displays the path to the assembly now.



This is a path on the Service Connect server.

You can enter the path to the assembly directly to this field. You can specify assembly quoted path as well.

9. In the **Reference name** field, view the default .NET reference name.

You can assign any name to a reference, but it is recommended that you use the default name. This name will display in the imported .NET references list.

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.

10. 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.

11. Click **Next**.

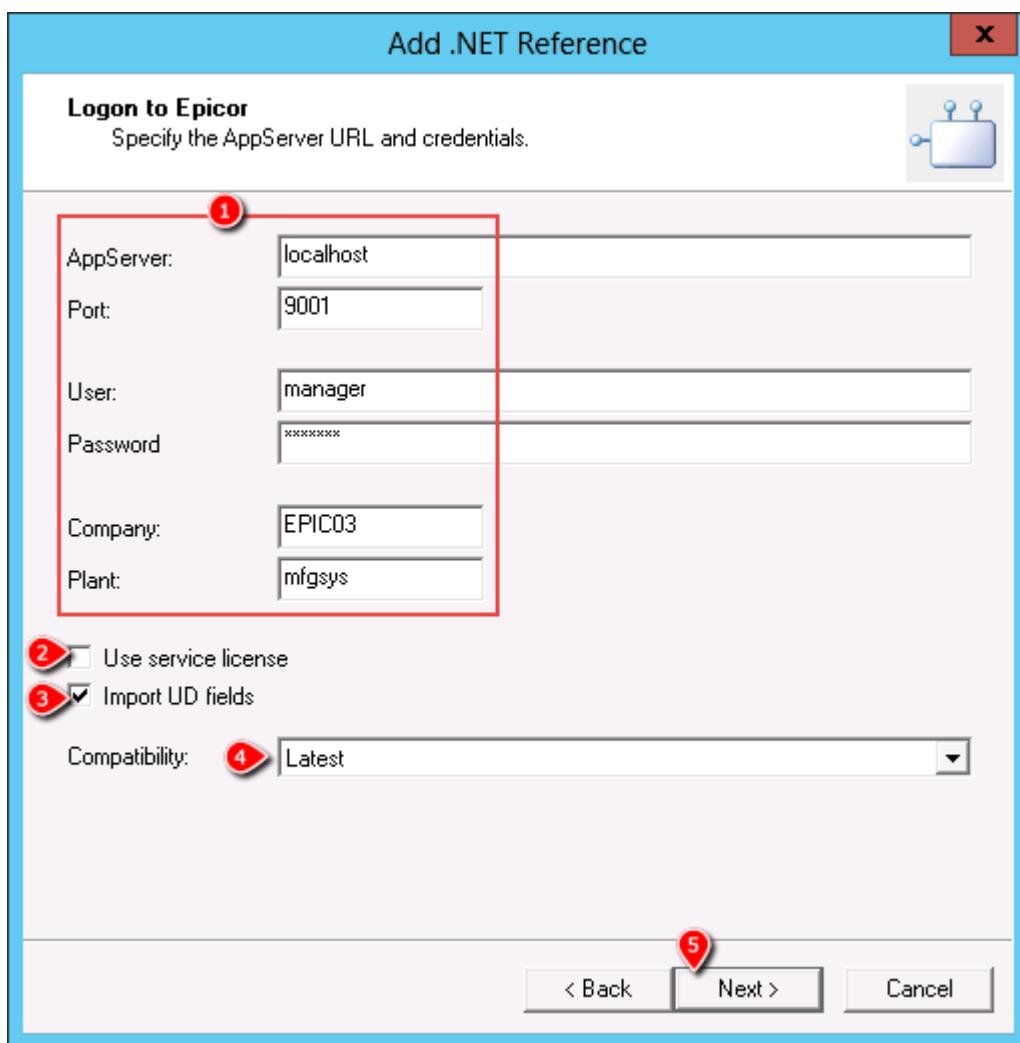
The next wizard page displays the options specific to the assembly type you selected.

Epicor 9 .NET Reference Setup

If you add an Epicor 9.05 .NET reference:

1. On the **Logon to Epicor** screen, enter the following Epicor logon settings:

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



2. 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.
3. 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.

When you add a .NET reference with the **Import UD fields** option enabled, Service Connect opens a new Epicor session during import and a license is consumed. The license type is specified in the .NET reference settings. So, before you import such .NET references, verify the following conditions are met:

- The Epicor server is available.
- At least one Epicor license is available.

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

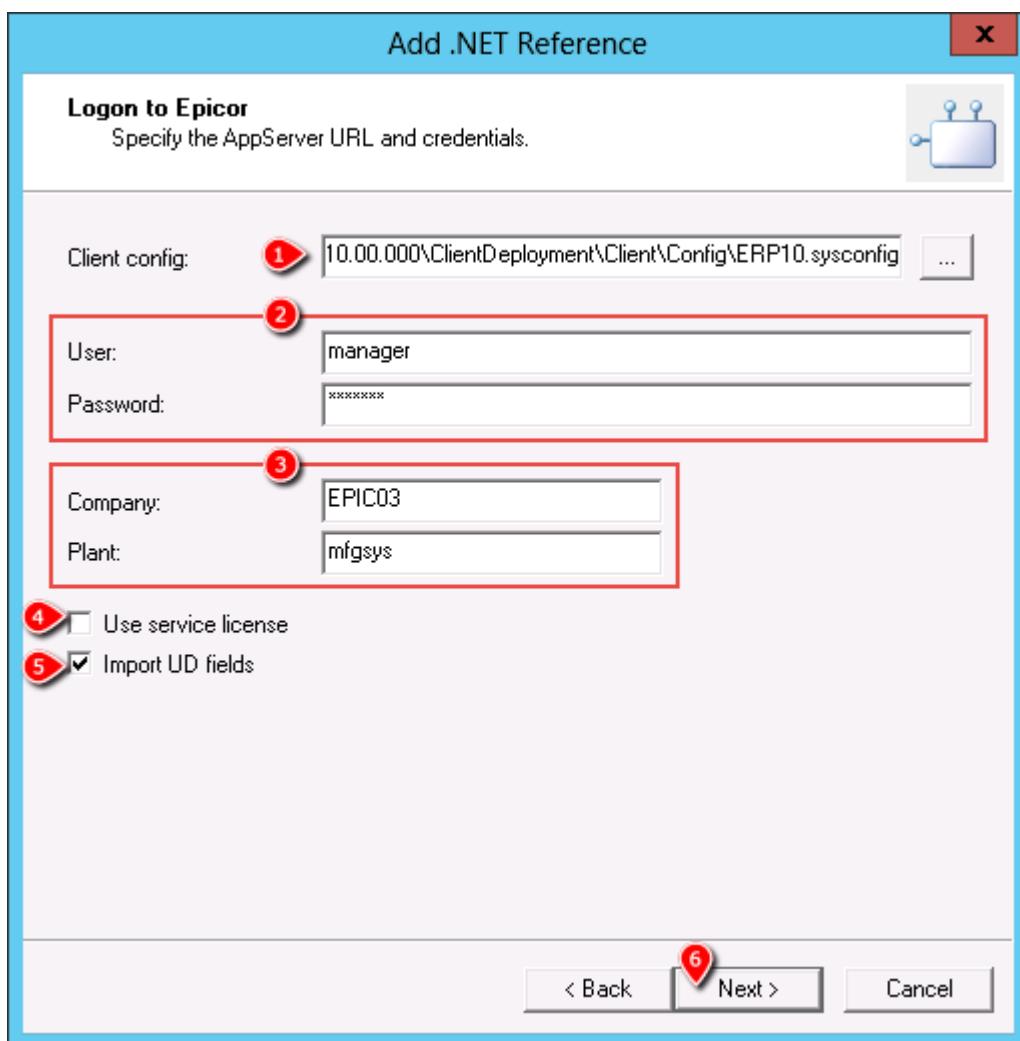
- **Latest** – Use this option if you are going to use Service Connect with the Epicor version 9.05.603 and later.
- **Epicor 9.05.602** – Use this option if you are going to use Service Connect with the Epicor versions 9.05.600 - 9.05.602.

5. Click **Next**.

Epicor 10 .NET Reference Setup

If you add an Epicor 10 .NET reference:

- In the **Client Config** field, specify Epicor client configuration file.



- In the **User** and **Password** fields, provide the Epicor logon credentials.
- In the **Company** and **Plant** fields, specify company and plant (site).
- 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.
- 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.

When you add a .NET reference with the **Import UD fields** option enabled, Service Connect opens a new Epicor session during import and a license is consumed. The license type is specified in the .NET reference settings. So, before you import such .NET references, verify the following conditions are met:

- The Epicor server is available.

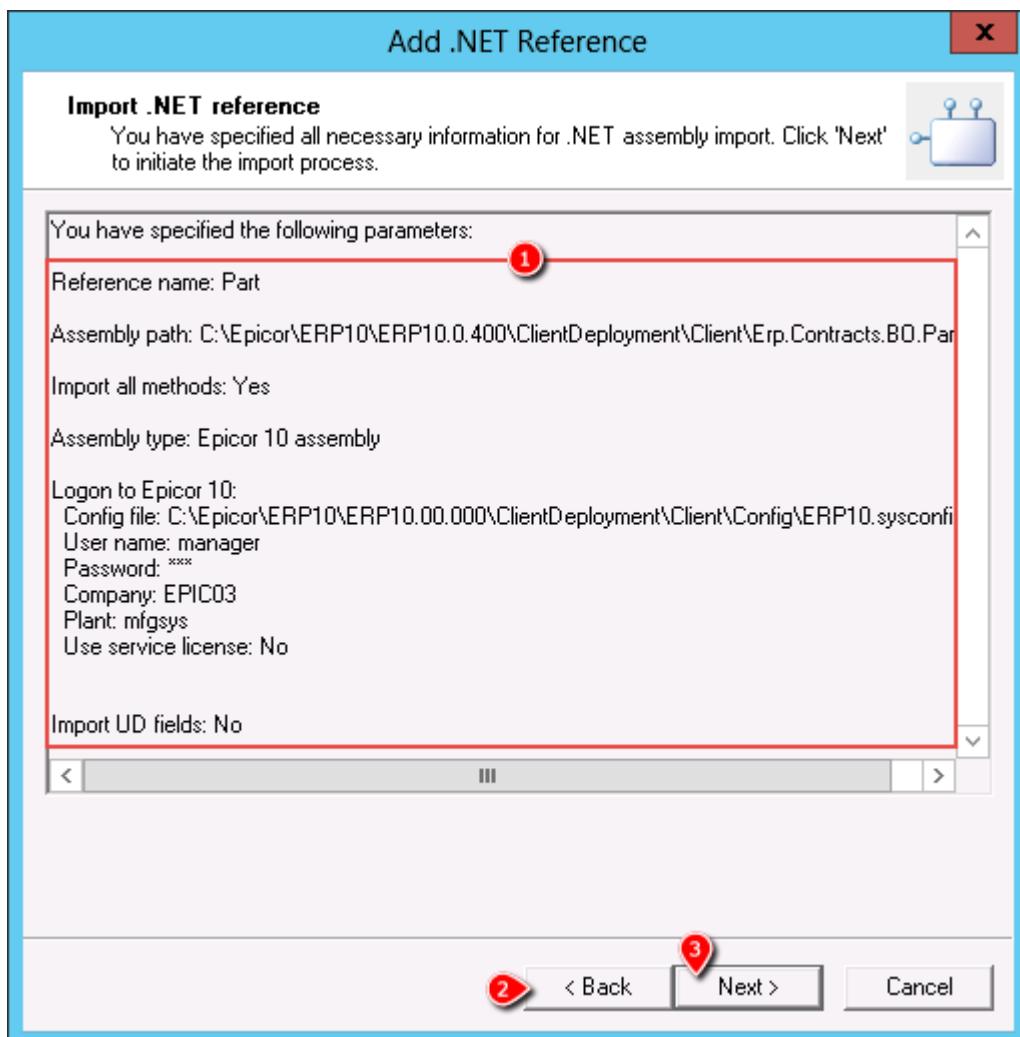
- At least one Epicor license is available.

6. Click **Next.**

.NET Import Finalization

Take the following steps to finalize .NET reference import.

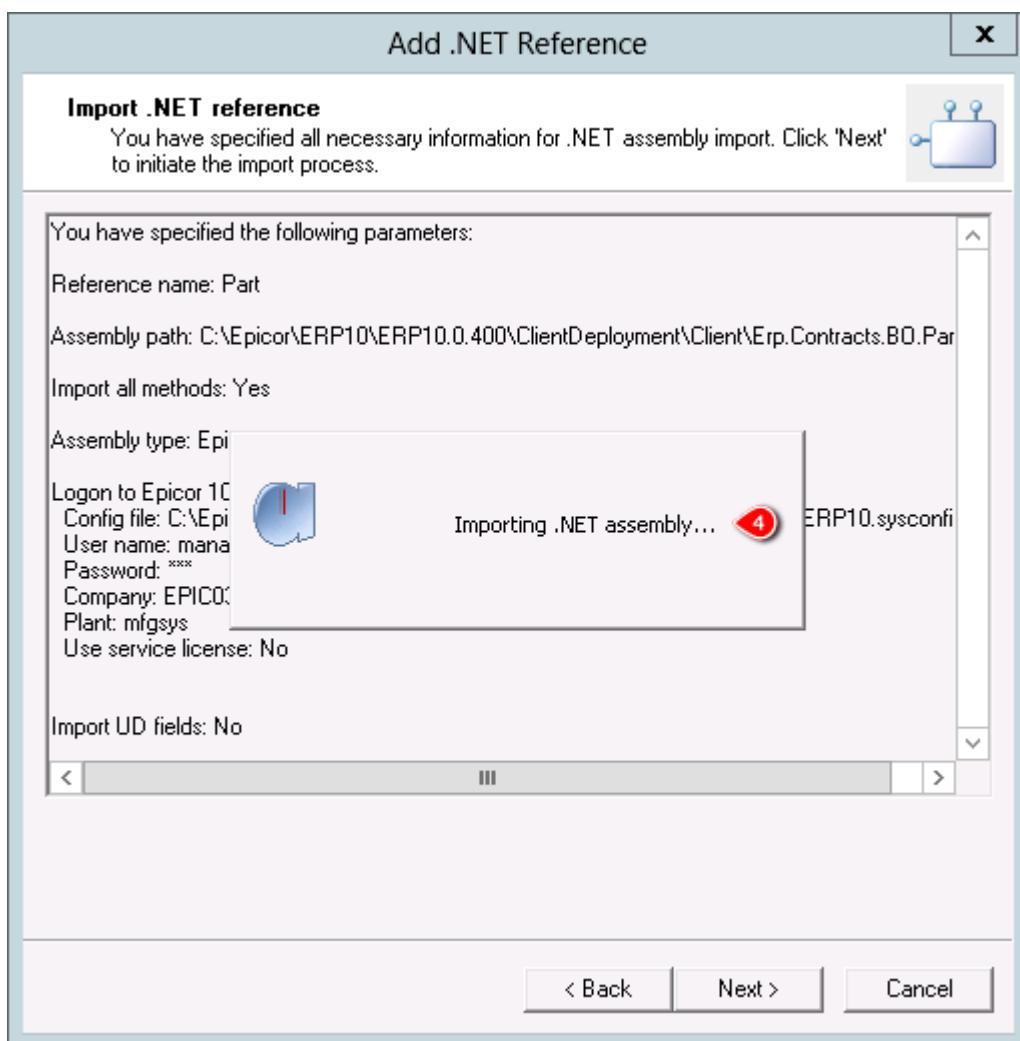
1. On the **Import .NET reference** screen, review all the information you provided about the .NET reference.



2. To change any parameter, click **Back**.

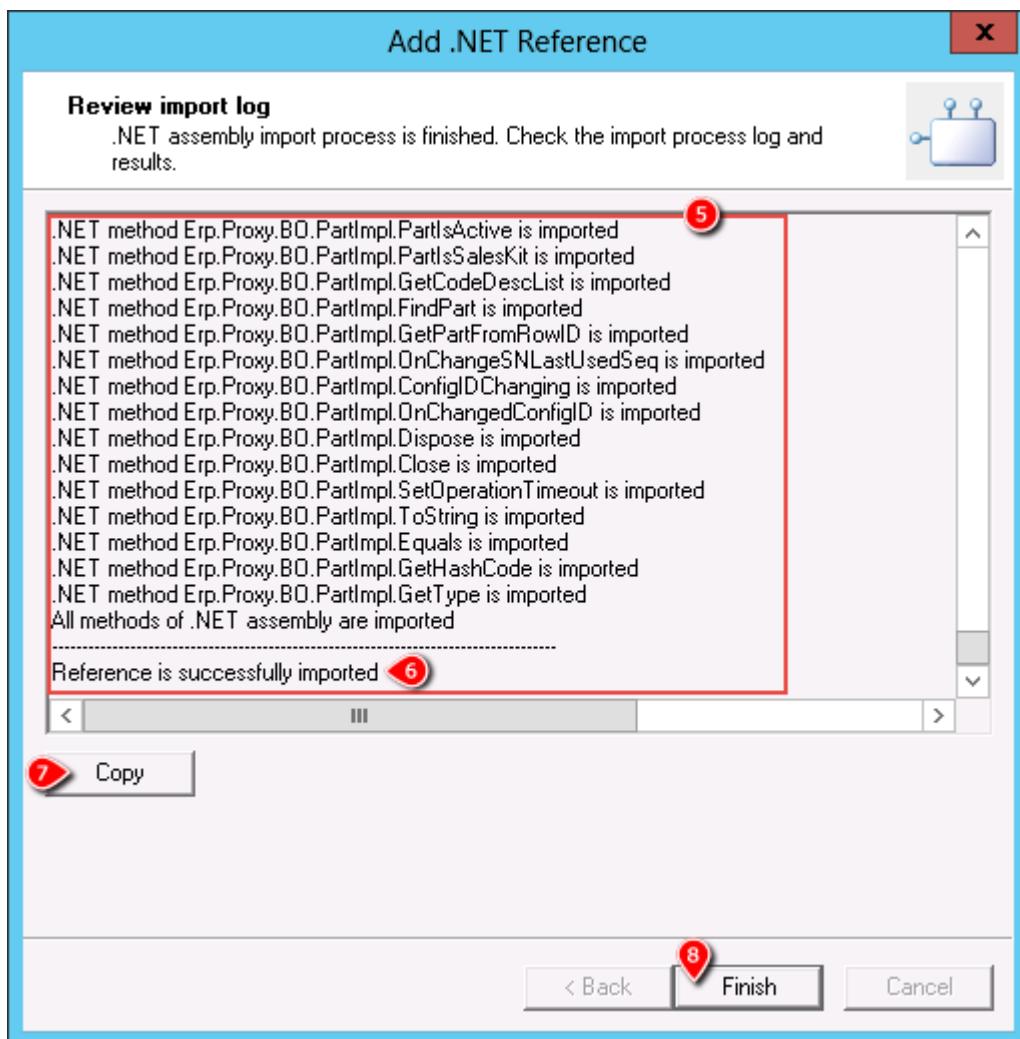
3. Click **Next**.

4. The import process starts.



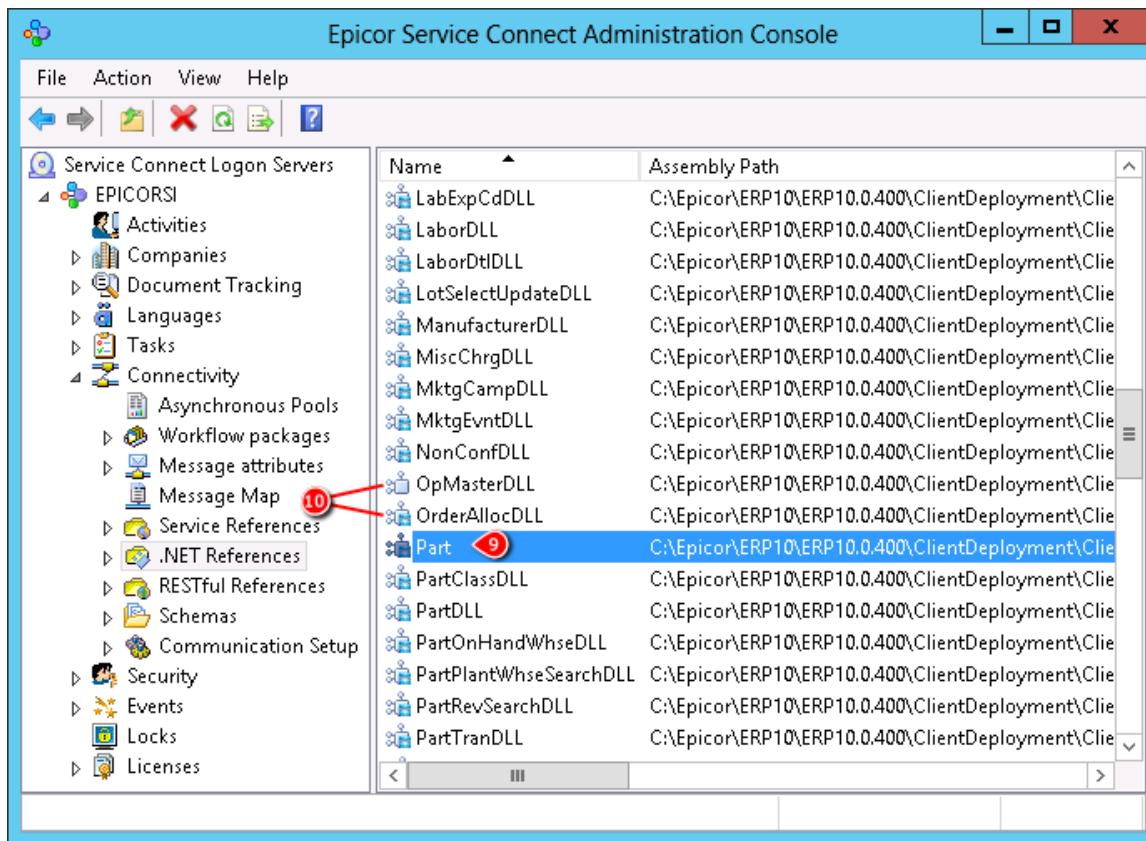
5. When the import is complete, the **Review import log** screen displays detailed log.

If the .NET Reference import failed, you can click the **Back** button to go back and adjust the reference settings, or, you can click **Cancel** to abort the import operation.



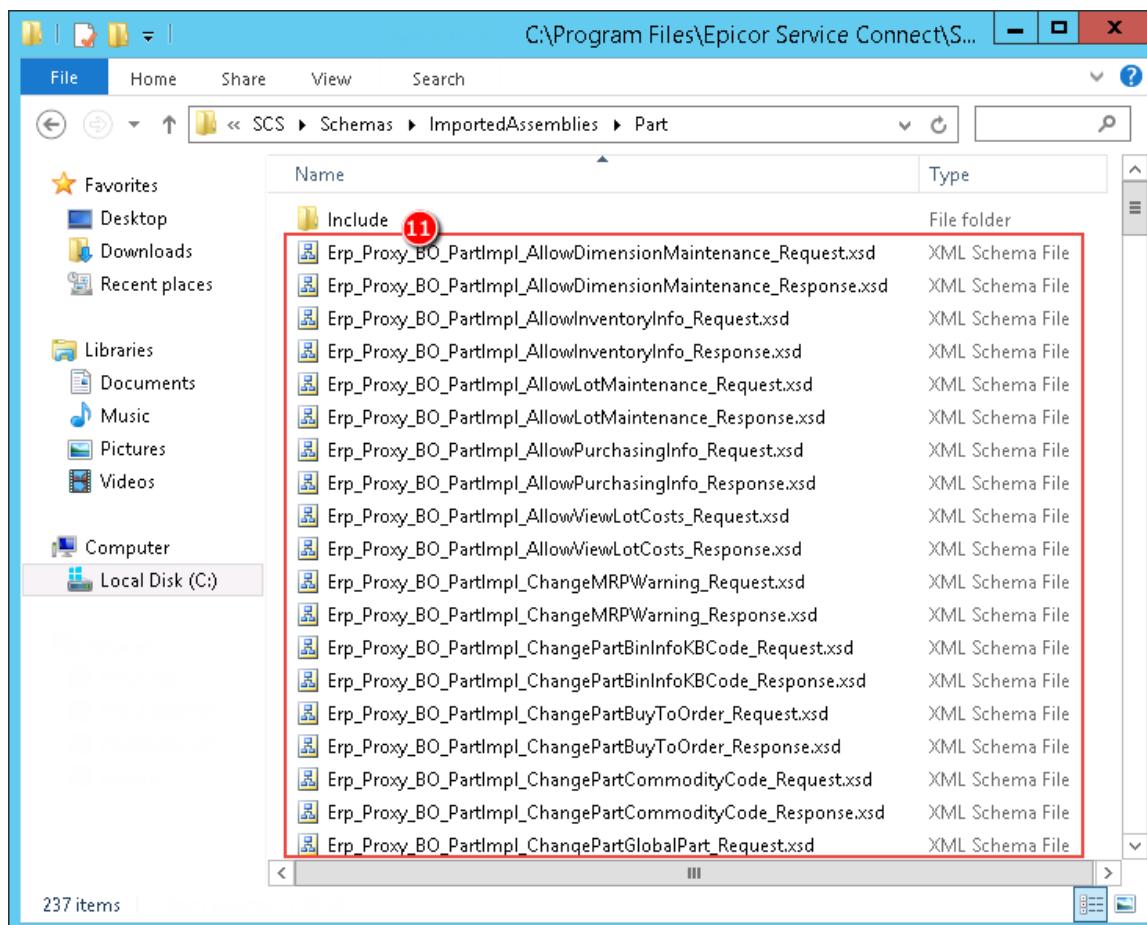
6. Notice the **Reference is successfully imported** message at the bottom of the import log.
7. Click the **Copy** button to save the import log to the clipboard.
8. Review the log information and click **Finish**.

9. Verify the new reference is now displayed in the list of .NET references.



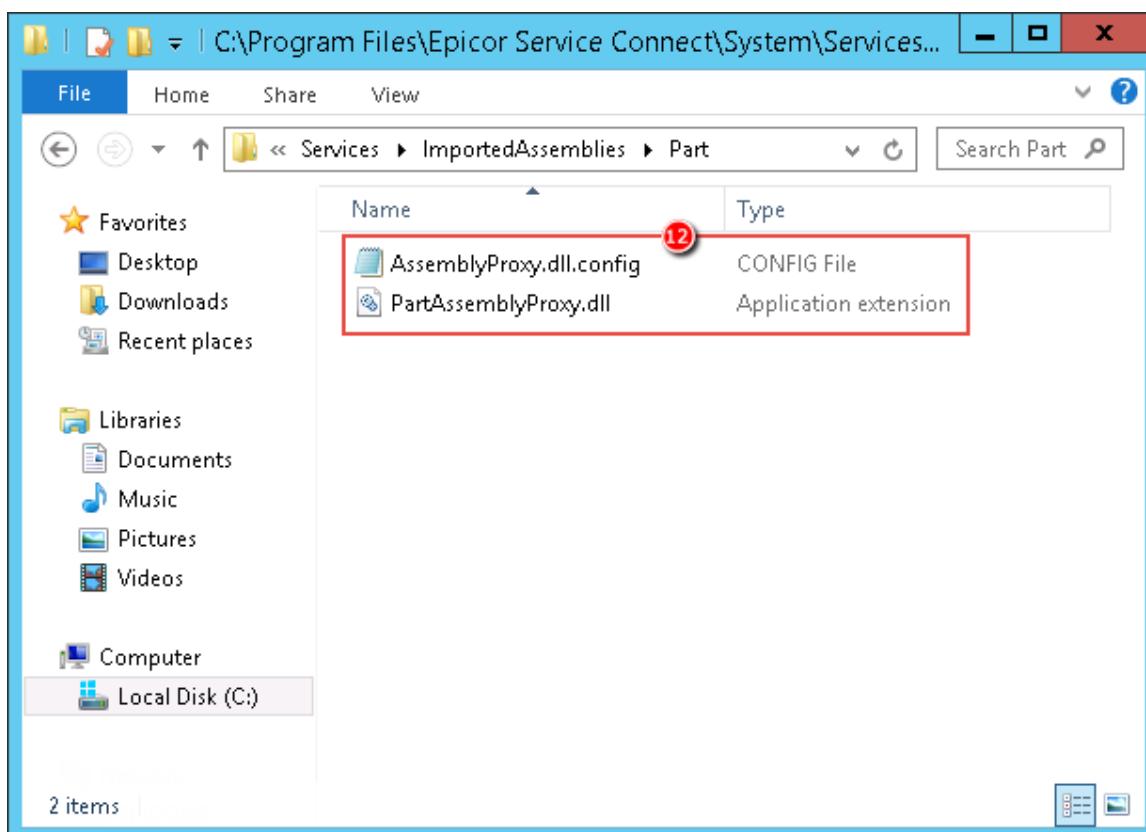
10. Notice that references to Epicor 9.05 .NET business objects have a simple icon, and references to Epicor 10 .NET business objects have an icon with a blue box on it.

11. Navigate to <SC installation folder>\SCS\schemas\ImportedAssemblies and view the schemas for imported .NET references.



You can also view schemas in Epicor Service Connect Administration Console under **Connectivity > Schemas > ImportedAssemblies** node.

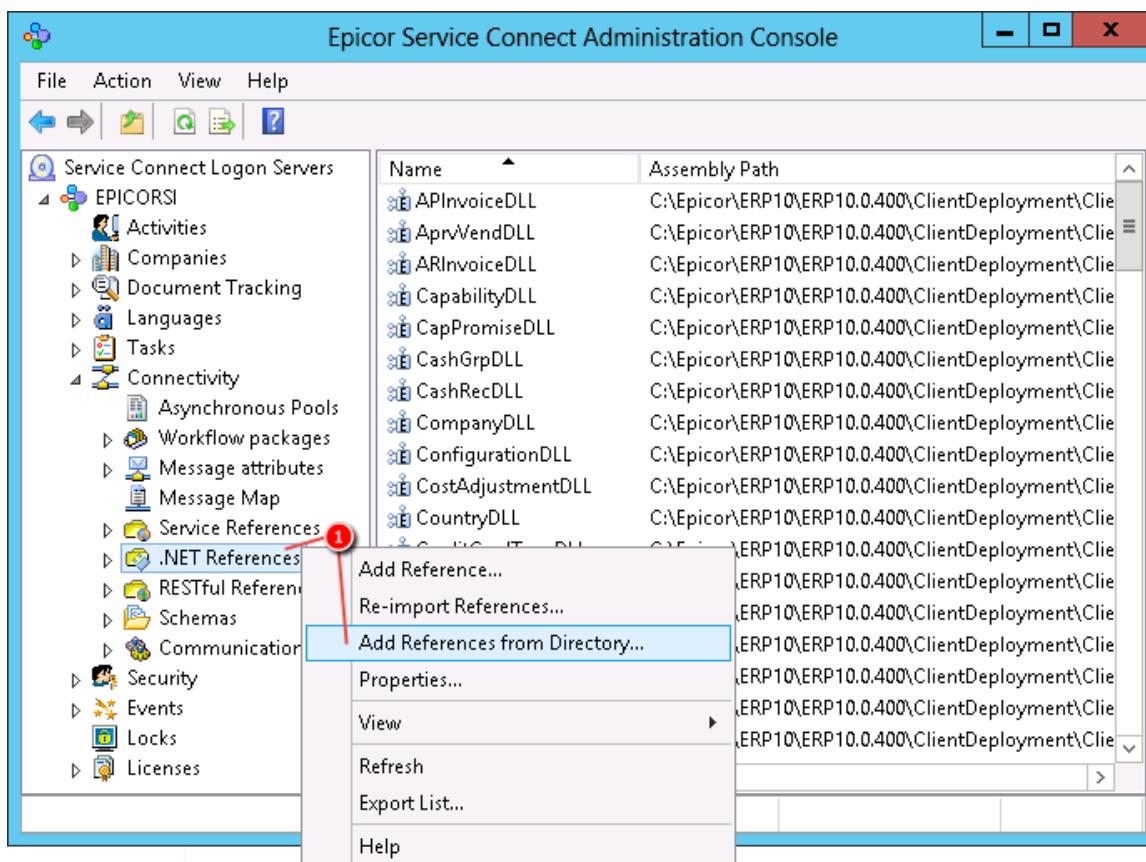
12. Navigate to <SC installation folder>\System\Services\ImportedAssemblies and view the proxy-files for imported .NET references.



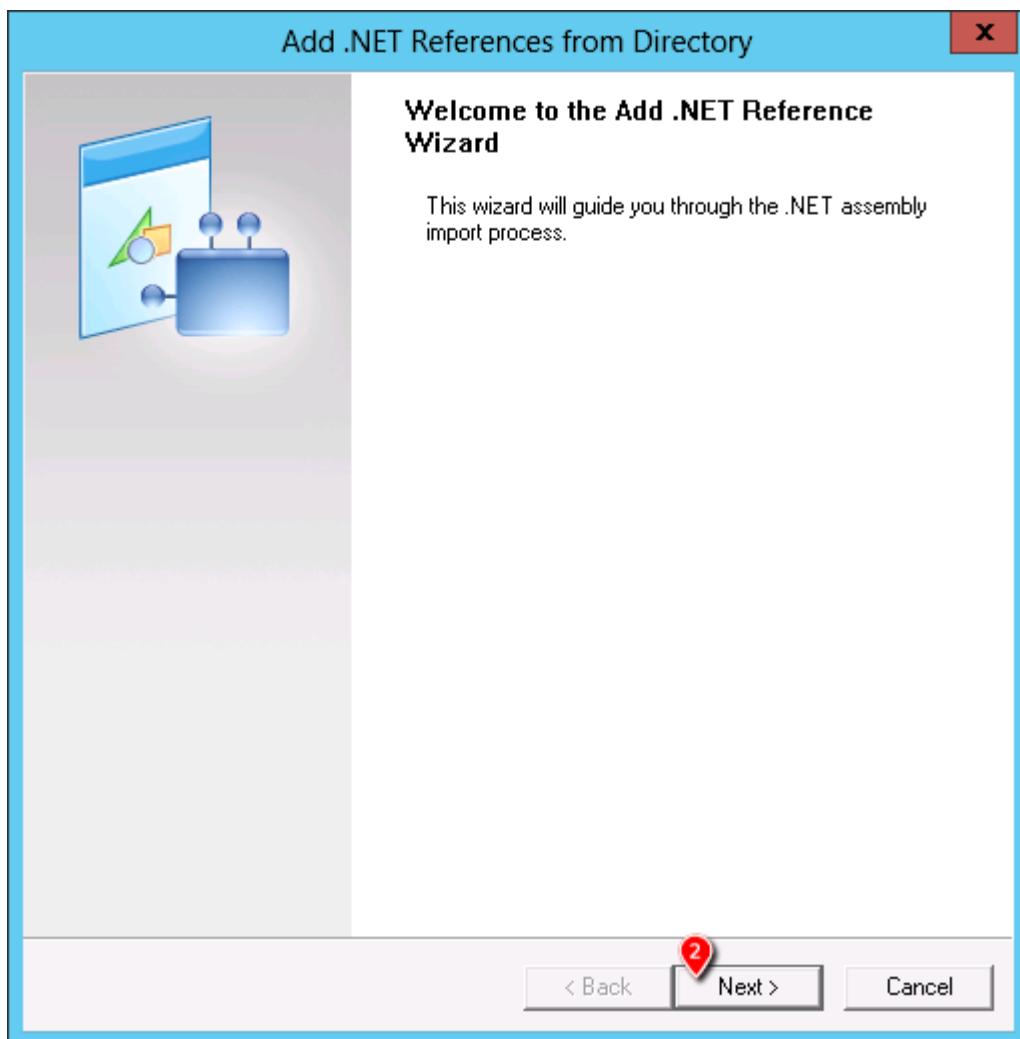
Add Multiple .NET References

To add multiple .NET references:

1. In the **tree view**, under the **Connectivity** node, right-click **.NET References** and select **Add References from Directory**.



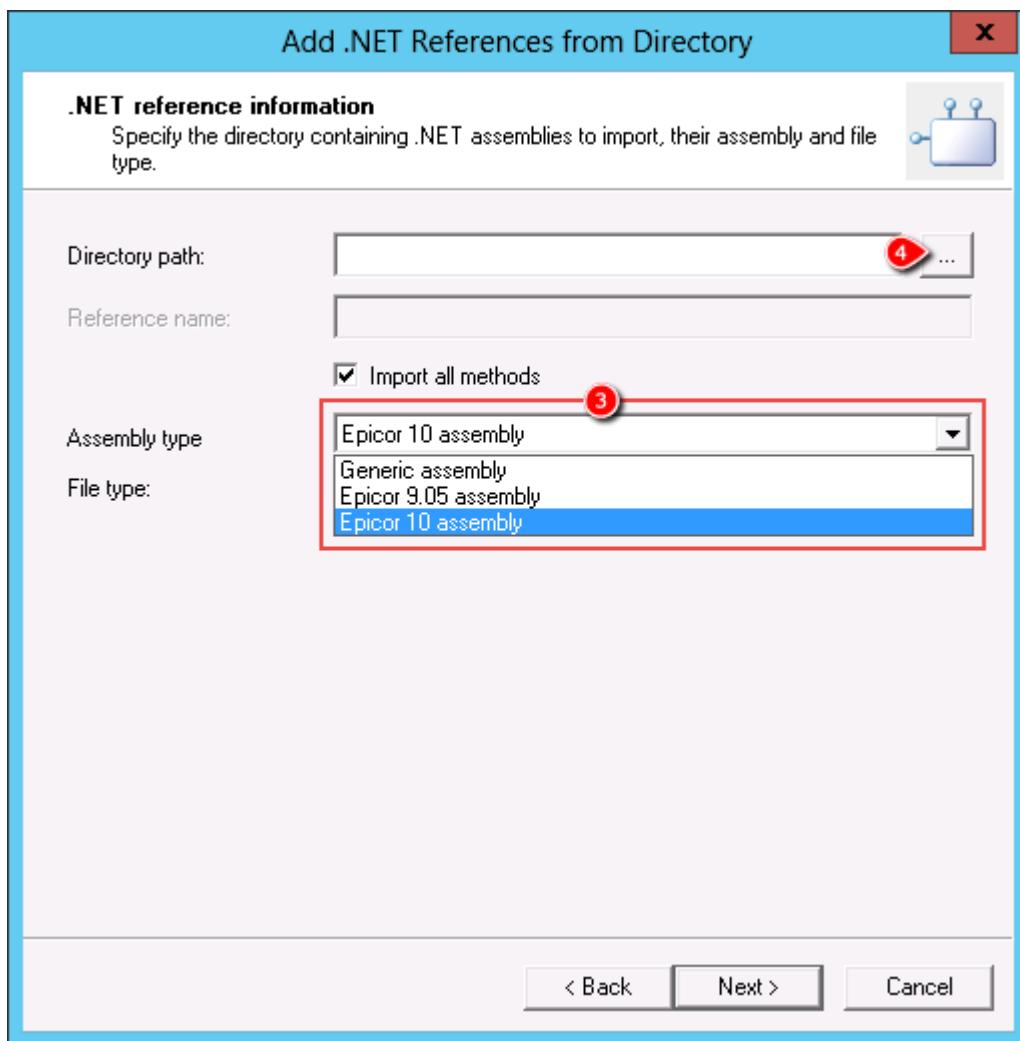
2. On the **Add .NET References from Directory** welcome page, click **Next**.



3. Click the **Assembly type** field, and select one of the .NET reference types:

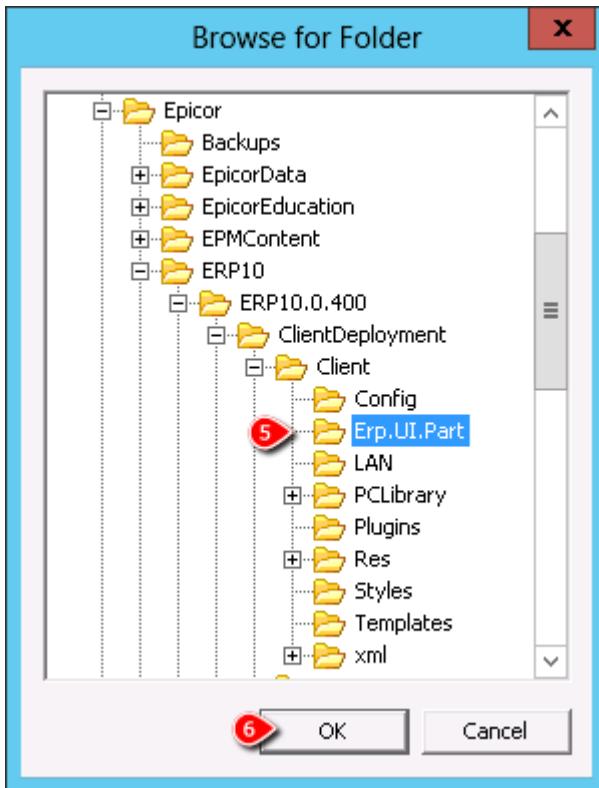
- Generic assembly
- Epicor 9.05 assembly

- Epicor 10 assembly



4. Next to the **Directory path** field, click the ... (ellipsis) button.

5. In the **Browse for Folder** window, browse to and select the directory.



6. Click **OK**.

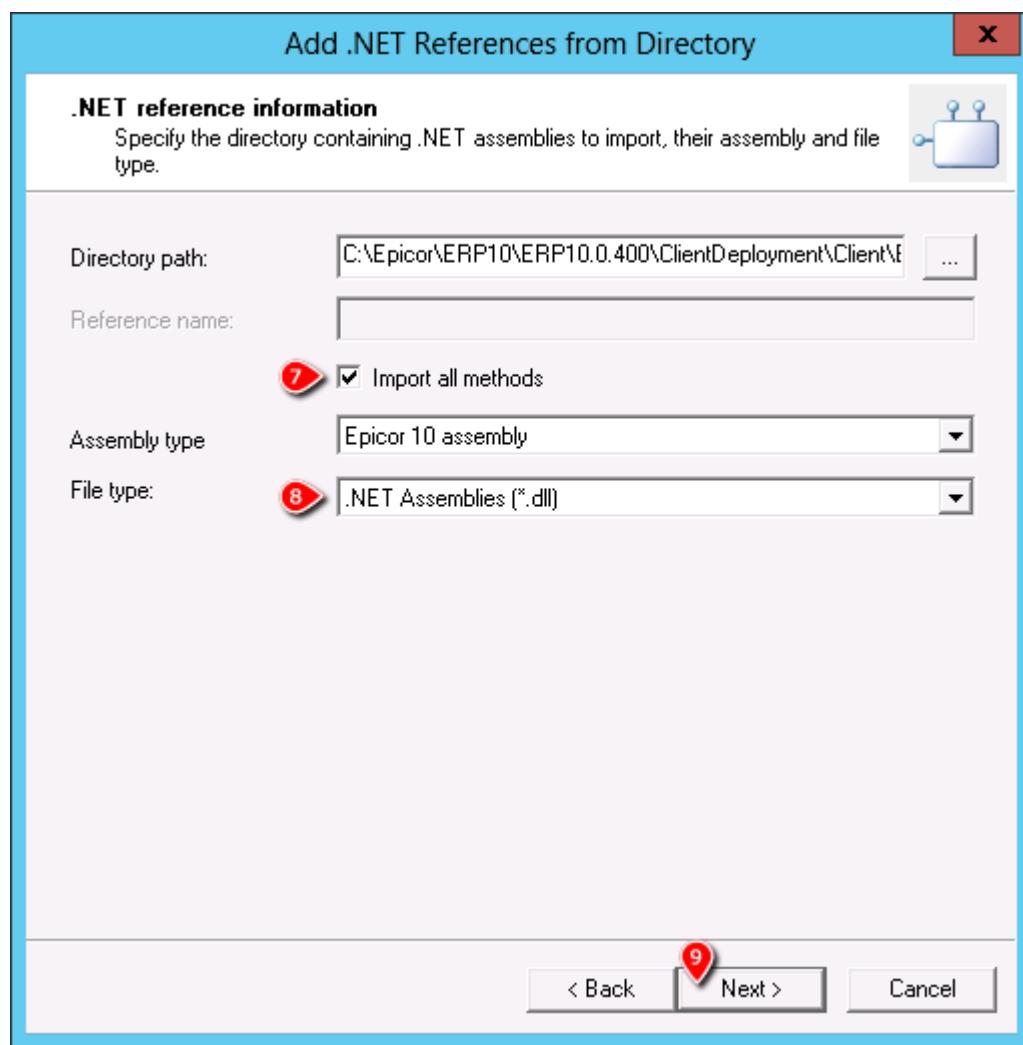
The **Directory path** field now displays the path to the directory.

This is a path on the Service Connect server.

You can alternatively enter the path to the directory directly to this field.

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

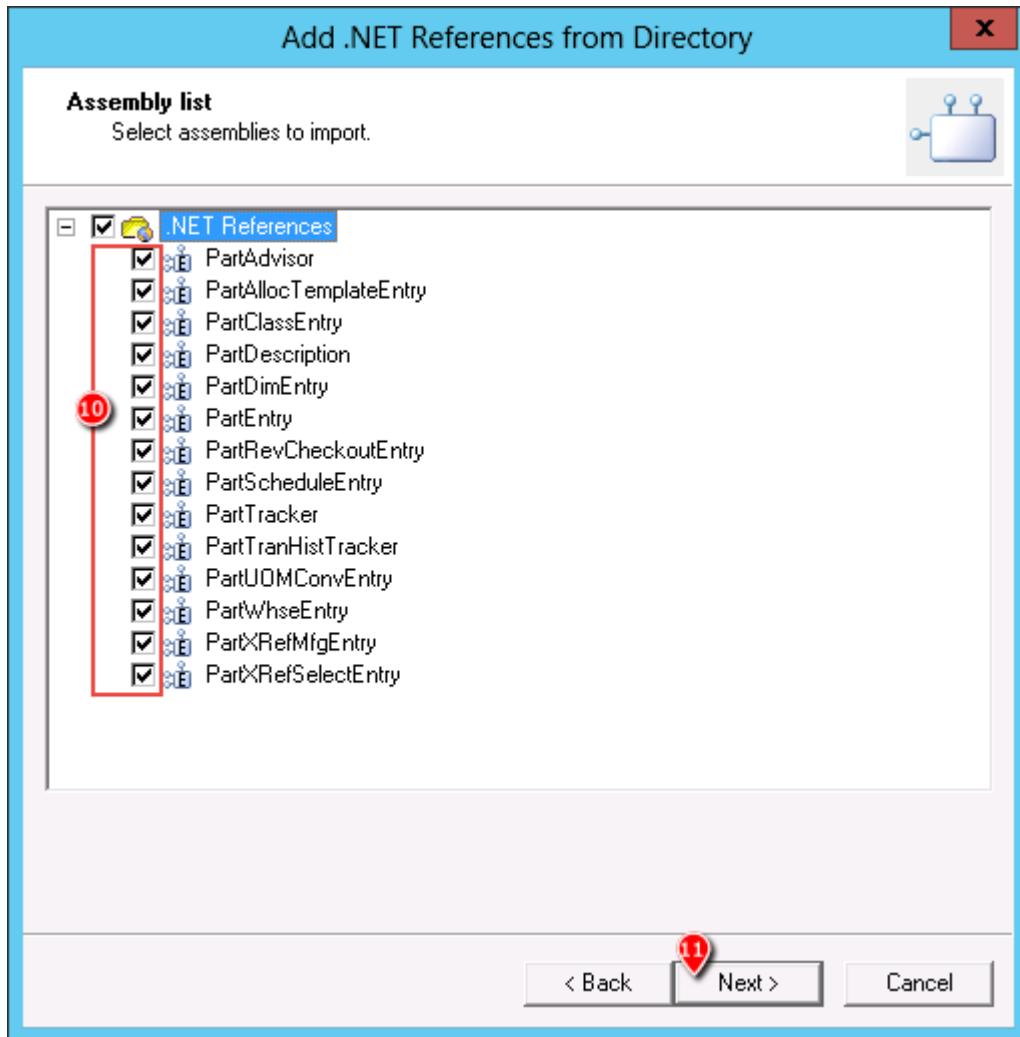
If you do not select the **Import all methods** check box, you can add the methods one by one later and configure each method individually.



8. In the **File type** field, select the file type.

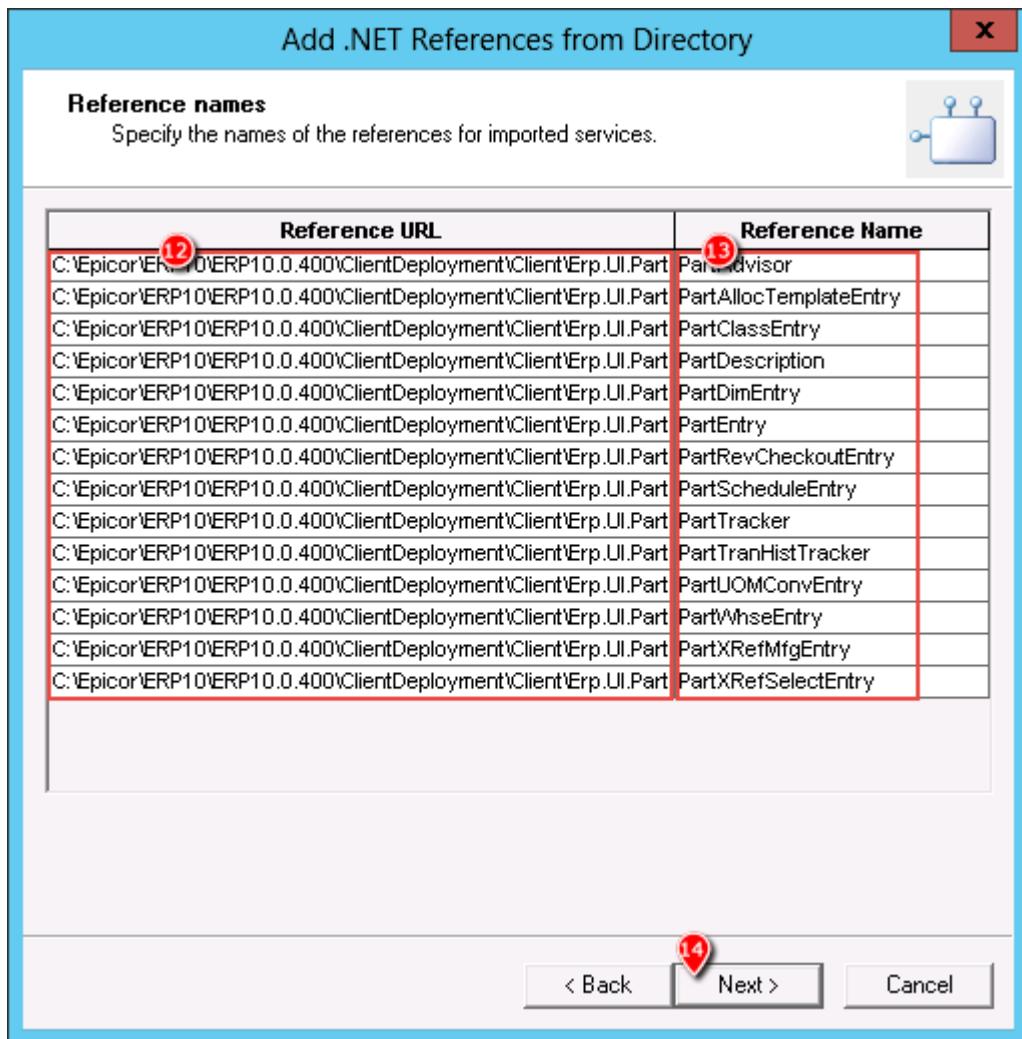
9. Click **Next**.

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



11. Click **Next**.

12. Review the **Reference URL**s.



13. Optionally modify the **Reference names**.

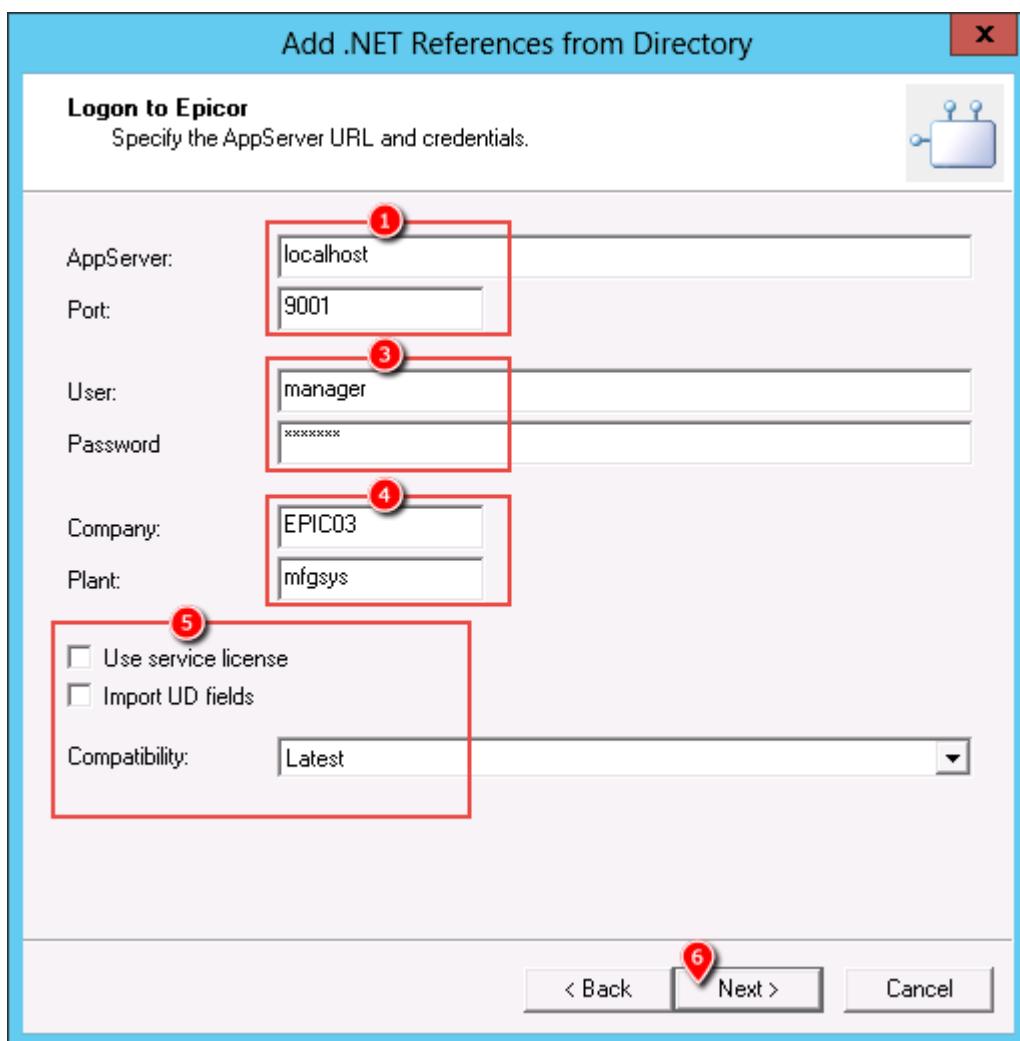
14. Click **Next**.

The next wizard page displays the options specific to the assembly type you selected.

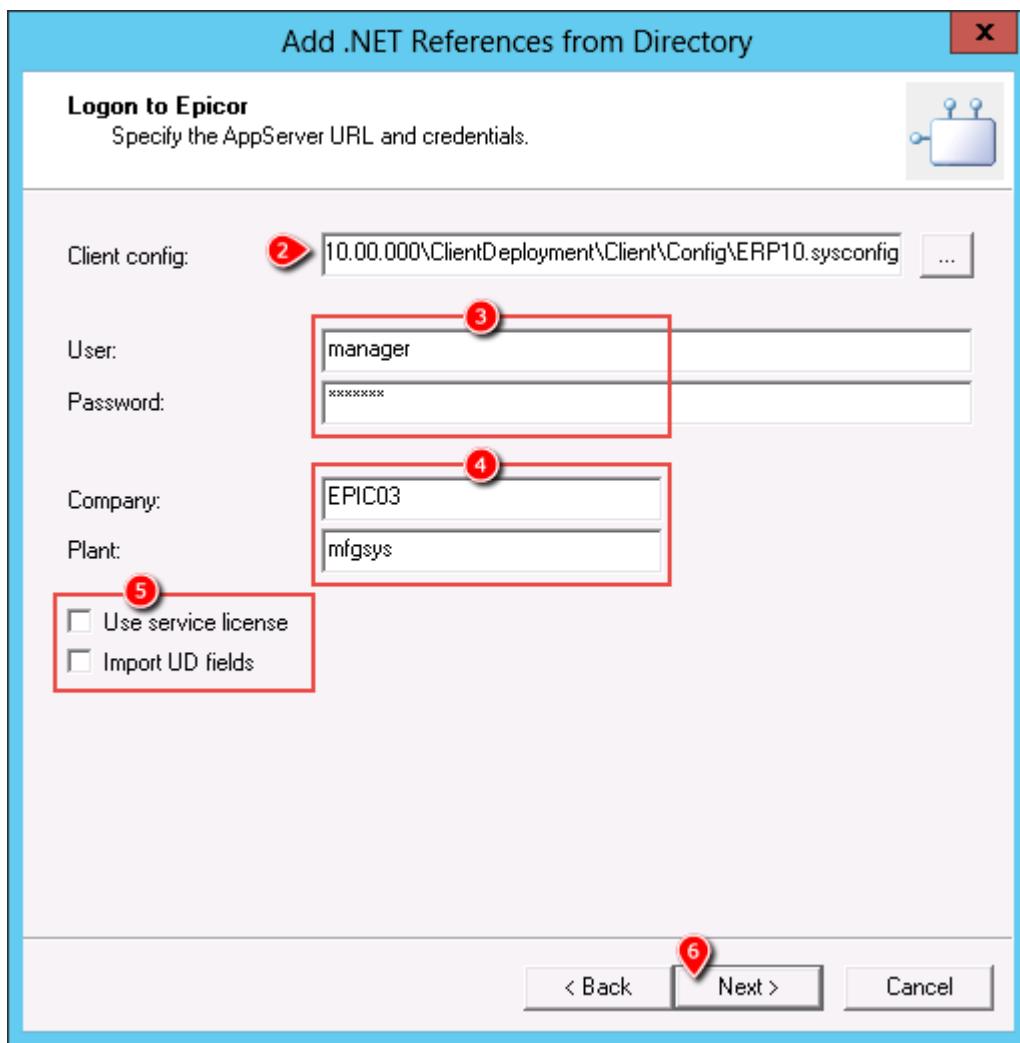
Multiple .NET References Import Finalization

Take the following steps to finalize multiple .NET references import:

1. If you selected the Epicor 9.05 assembly, on the **Logon to Epicor** screen, specify **AppServer** and **Port**.

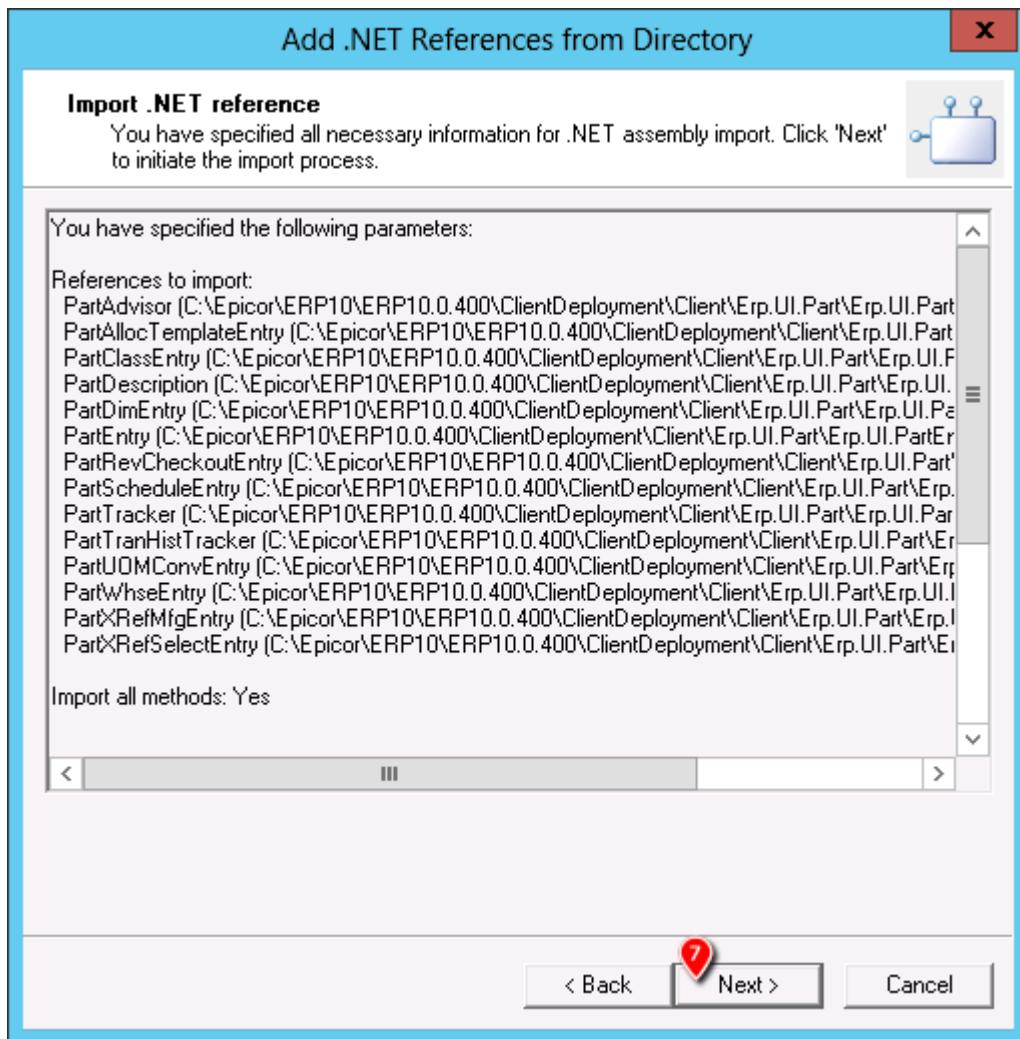


2. If you selected the Epicor 10 assembly, on the **Logon to Epicor** screen, in the **Client Config** field, specify Epicor client configuration file.

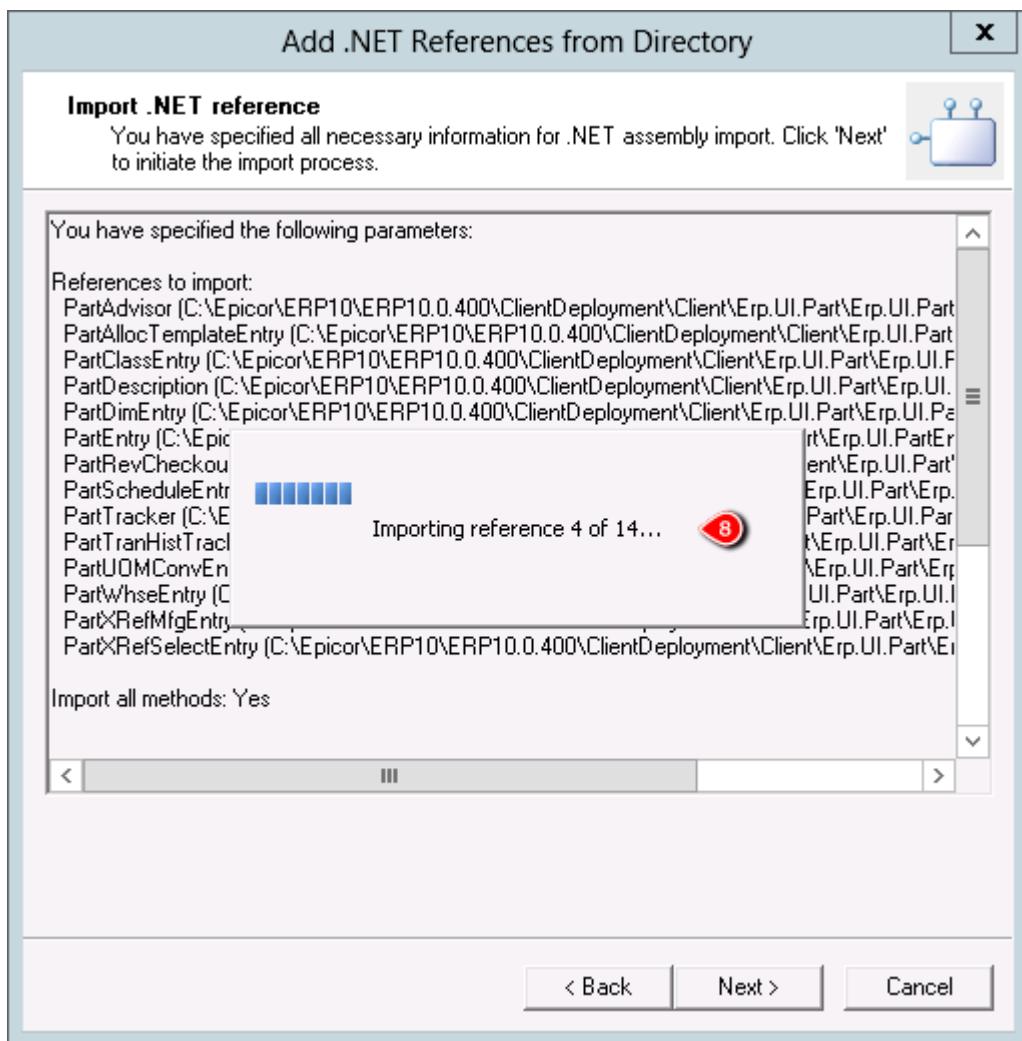


3. In the **User** and **Password** fields, provide the Epicor logon credentials.
4. In the **Company** and **Plant** fields, specify company and site.
5. Optionally, use Import UD fields, Use service license and Compatibility options as described before.
6. Click **Next**.

7. Review the .NET reference import information and click **Next**.



8. The import process starts.



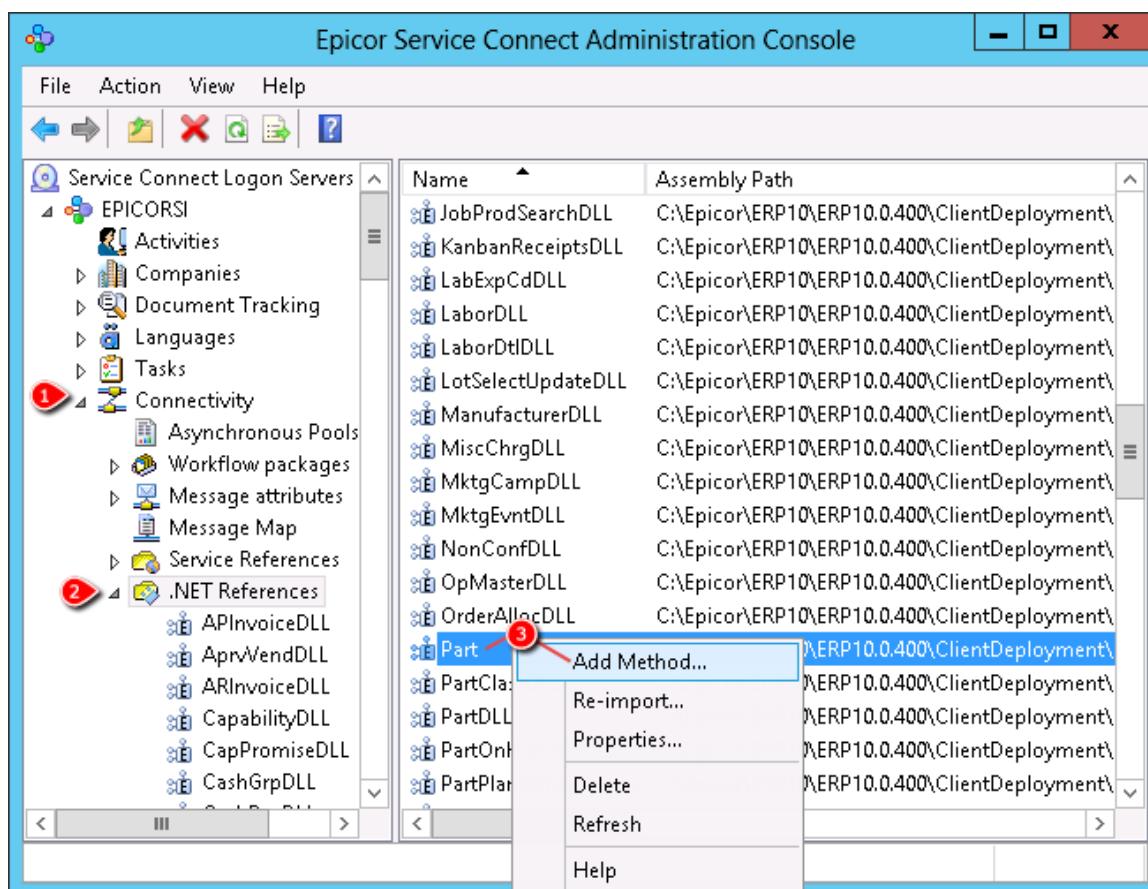
9. On the **Review import log** screen, you can use the **Copy** button to save the import log to the clipboard.

10. 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:

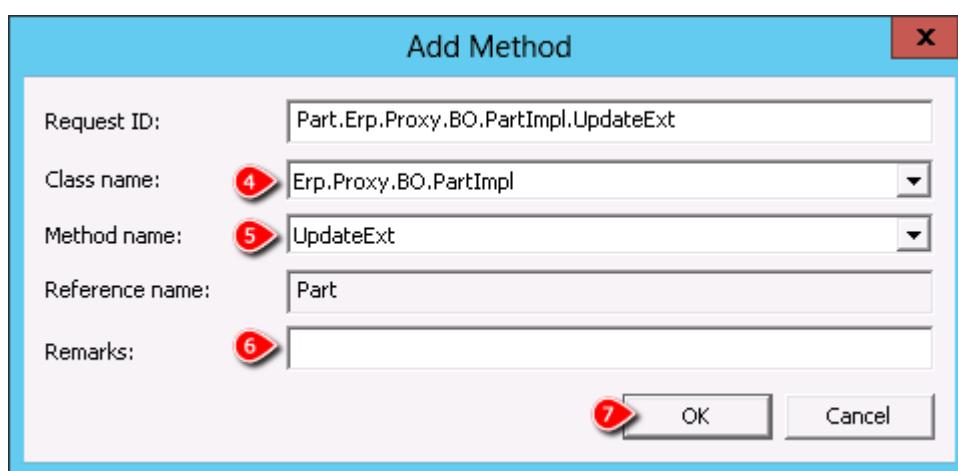
1. In the **tree view**, expand the **Connectivity** node.



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. In the **Add Method** window, select a **Class Name**.



5. Select a **Method Name**.

6. In the **Remarks** field, you can enter a description of the method.

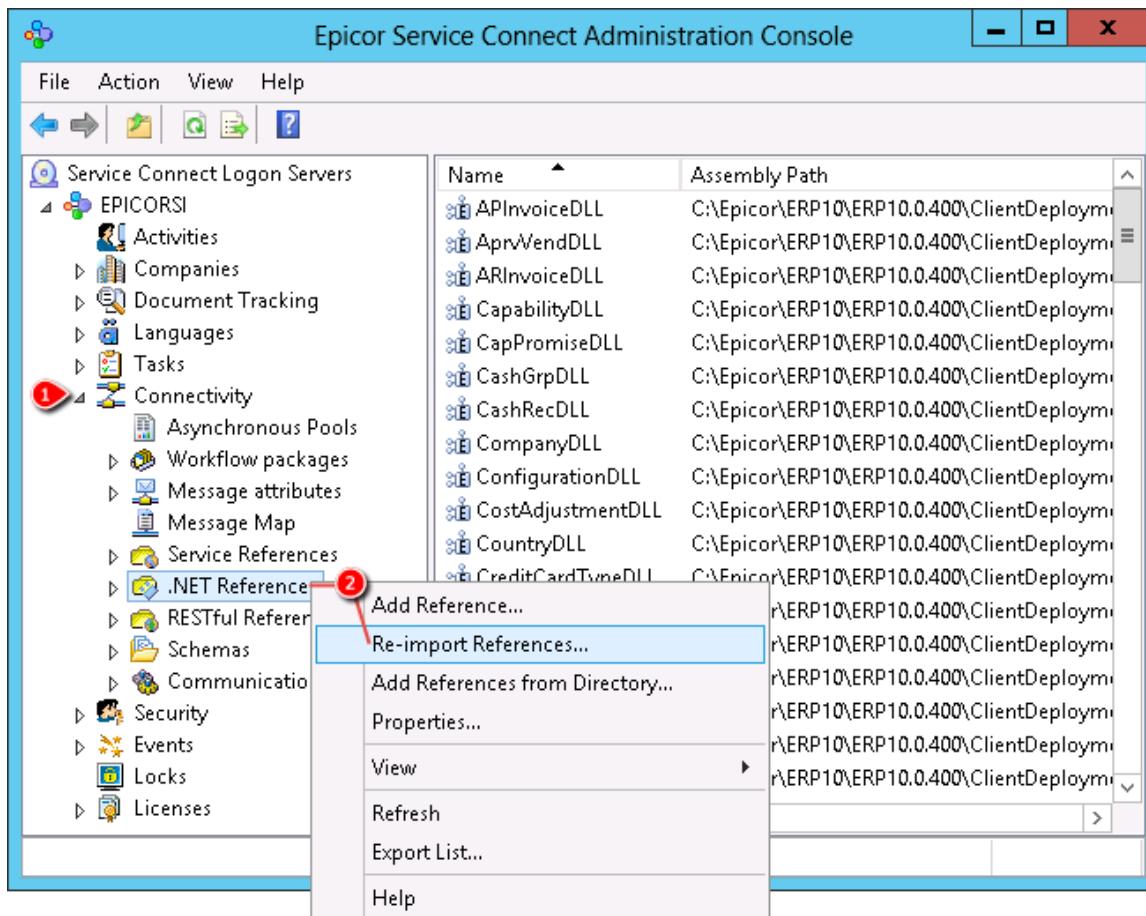
7. Click OK.

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

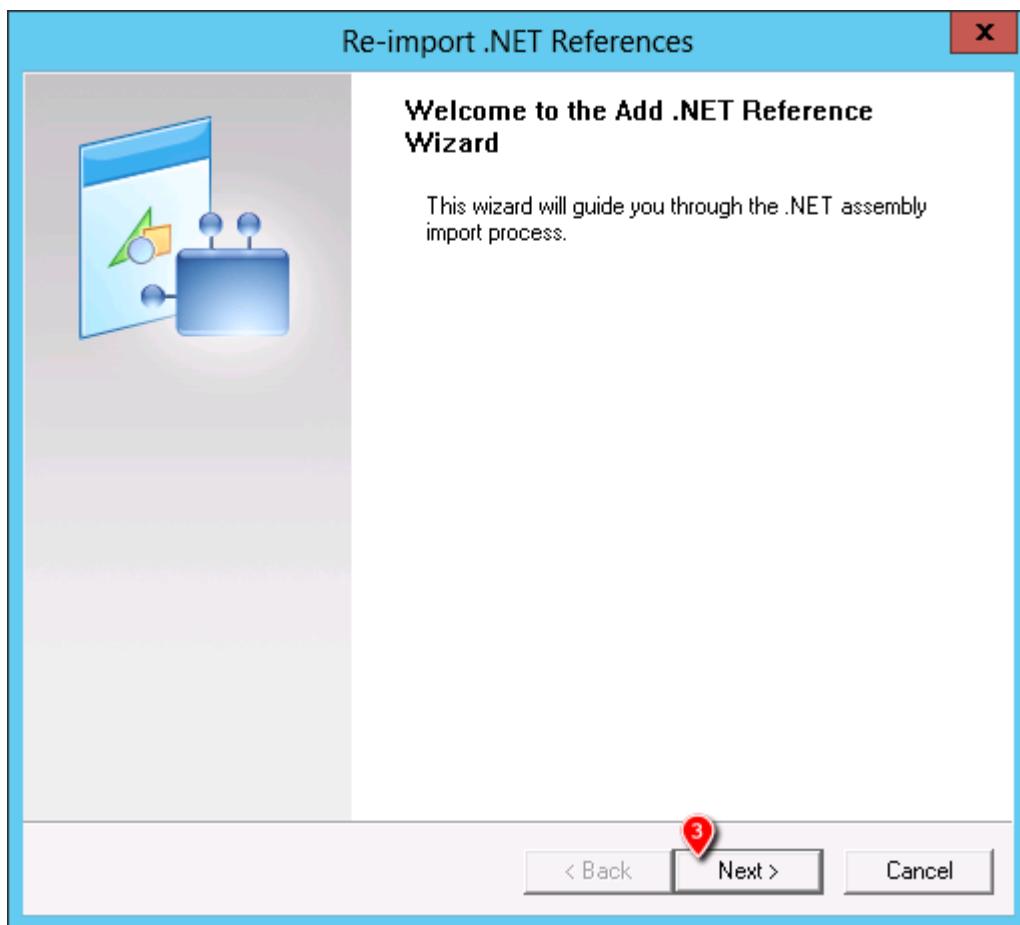
If you update the .NET assemblies, or if you install a newer version of Service Connect, follow these instructions to re-import the .NET references into the ESC Administration Console.

1. In the tree view, expand the **Connectivity** node.



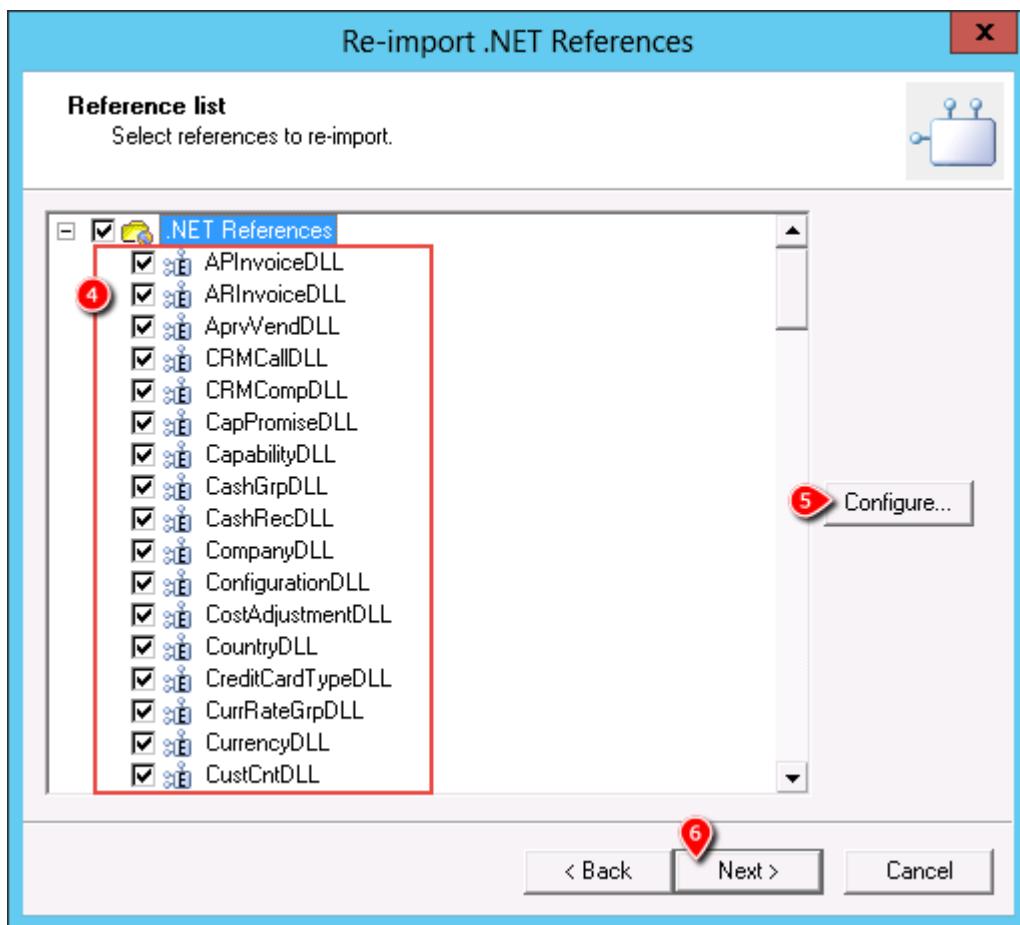
2. Right-click **.NET References** and select **Re-import References**.

3. On the **Re-import .NET References** welcome window, click **Next**.



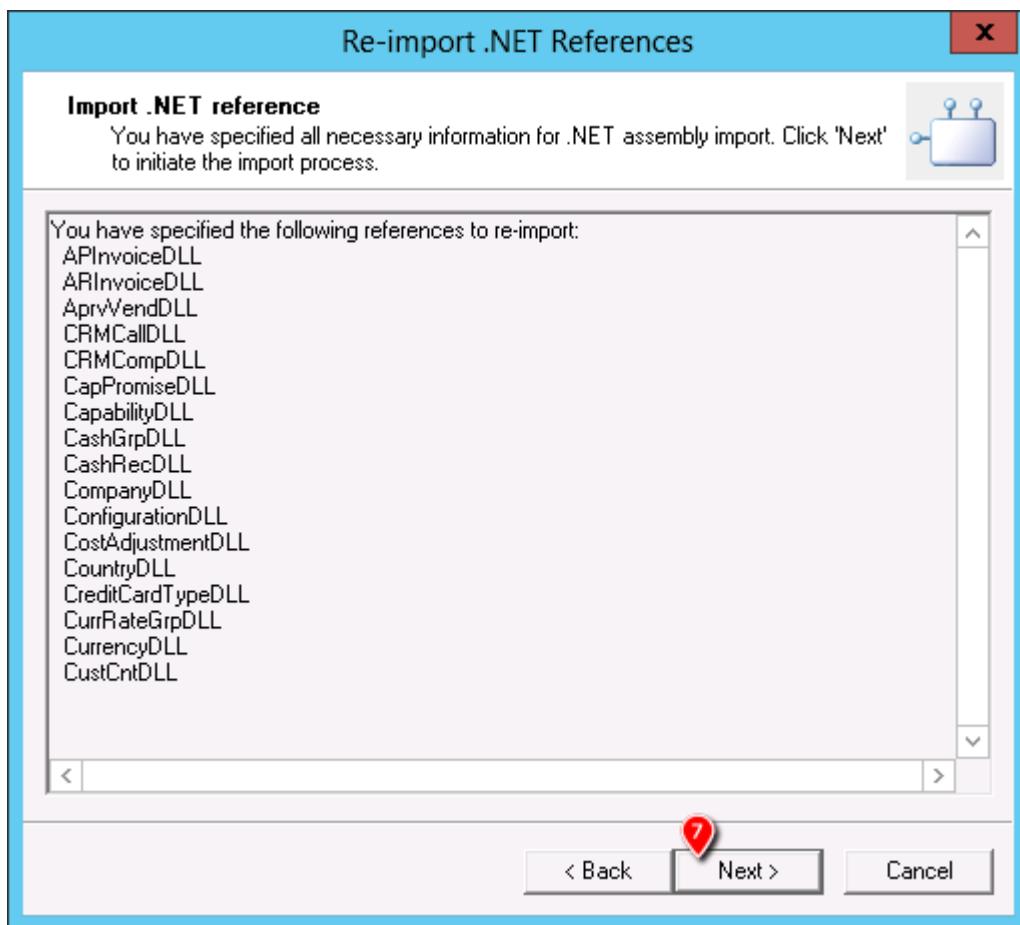
4. From the **Reference** list, select the references to re-import.

All references are selected by default.



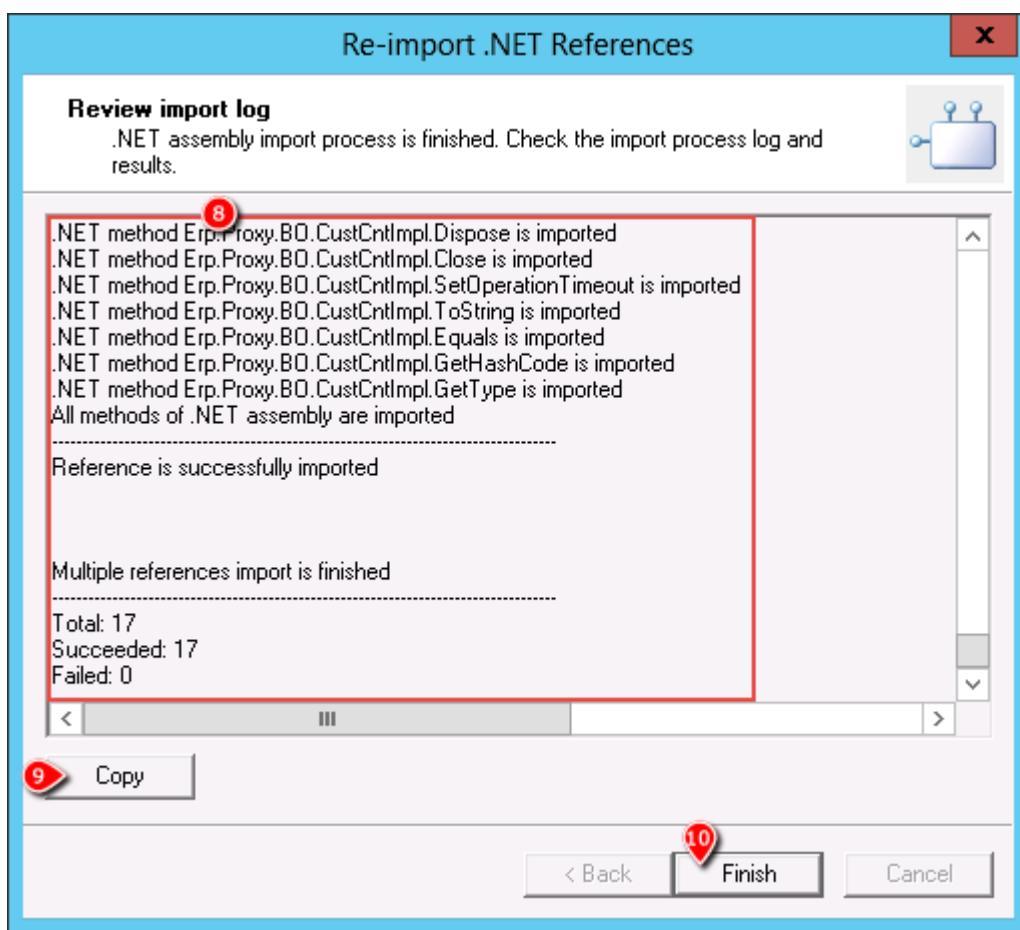
5. During reference re-import, Service Connect automatically uses the settings originally used to import the .NET reference. If you need to adjust the re-import settings, select the .NET reference, click **Configure** and change the settings.
6. Click **Next**.

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.



9. Click the **Copy** button to copy the re-import log to the clipboard.

10. Click **Finish**.

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

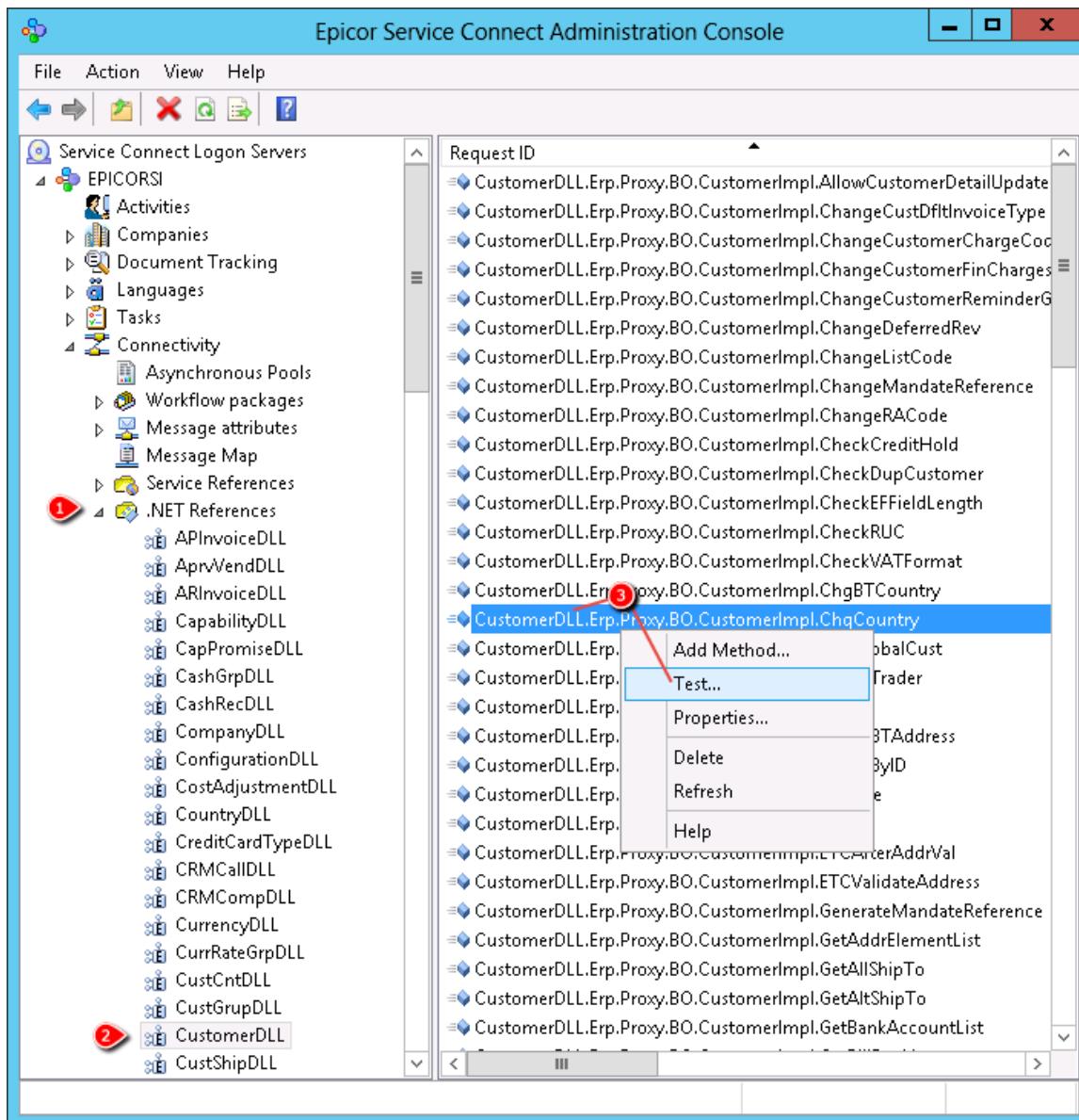
You must have security permissions to add or re-import a .NET reference. The .NET assembly must be located on the Service Connect server. If the .NET reference fails to import, verify the Assembly path and the Assembly type were correctly selected. If the .NET reference uses the Epicor assembly type,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

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.

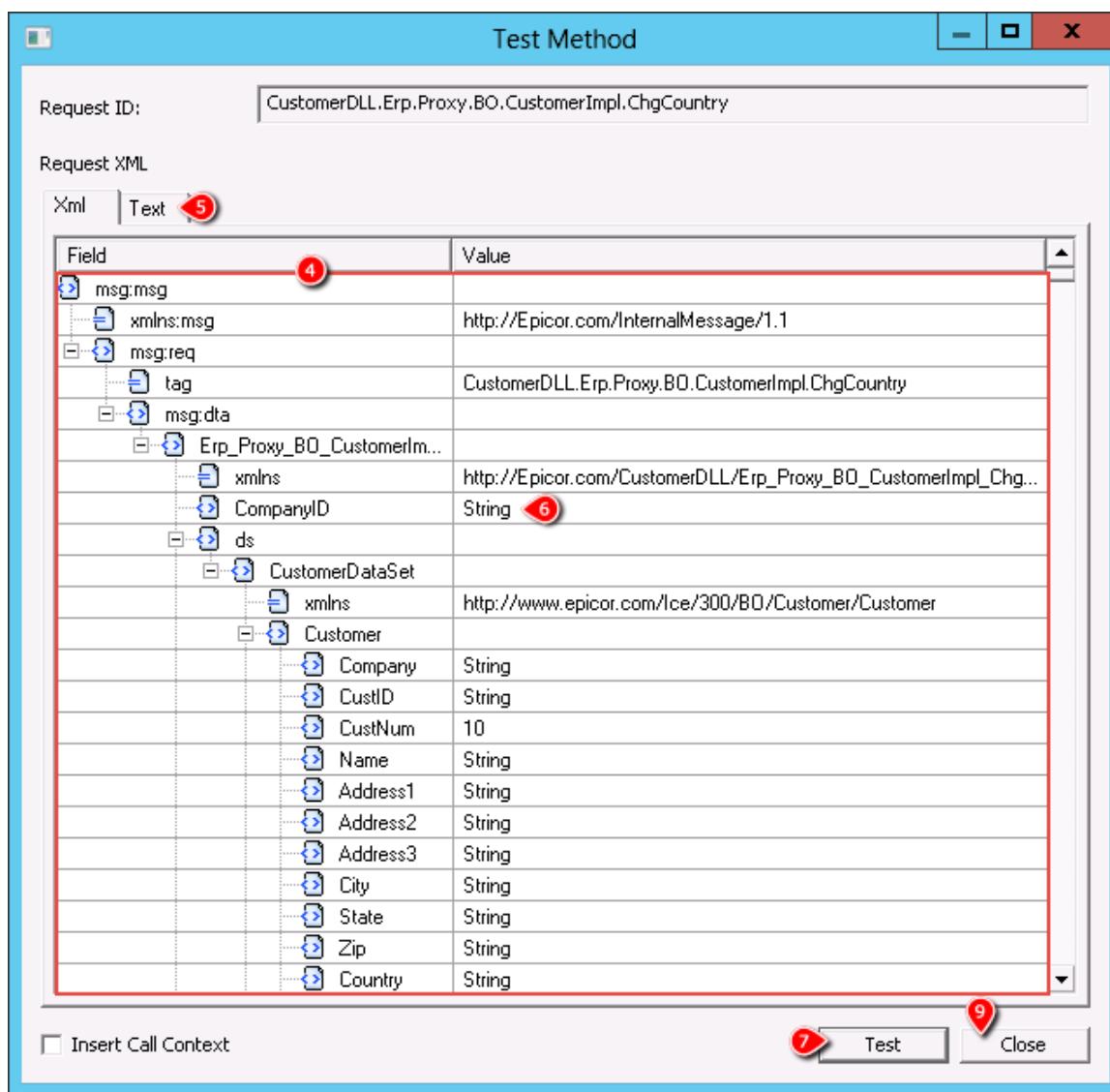


2. Select a .NET Reference.

In this example, CustomerDLL is selected.

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

4. The **Test Method** dialog opens and displays the request XML structure on the XML tab.



5. Open the **Text** tab to view xml text.

6. On the XML tab, enter the request criteria.

The request XML is a message template. You must replace the placeholders with meaningful data before you send the request. For example, in the CompanyID node, you must remove **String**, or replace it with a real company ID.

7. Click **Test**.

8. 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.

9. 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 references 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.

To learn how workflows interact and how to develop workflows refer to Workflow Designer chapter.

RESTful References

You can import RESTful web services and call them from inside Service Connect workflows.

RESTful web services do not have a standard functionality to expose metadata describing their API. Epicor recommends that you use documentation provided by the service developer in order to discover an API and supported data formats. A part of this documentation is sometimes exposed as a help page, usually available by /help relative URI. The API documentation is the only way to discover exposed resources, their URI templates, HTTP methods available on them, and their representation formats unless metadata is exposed. You can only import RESTful web service automatically if its metadata is exposed or if a API-type specific contact is reached (for example, if RESTful services in Epicor API expose metadata later). Otherwise, you must manually define all resources and operations available on them.

The Administration Console displays different icons for REST references imported automatically and manually.

Add a RESTful Reference Manually

When you import a RESTful Reference manually, perform the following steps:

1. Add Reference in the RESTful References node in Administration Console

Service Connect creates a new reference to the RESTful Web Service.

References are configured links to the published application services, such as RESTful web services, which are outside of Service Connect. After you add RESTful references in the ESC Administration Console you can pass and receive datasets to and from the RESTful web service resources from inside workflows. A wizard helps you import RESTful references, configure security, and handle other miscellaneous details regarding the exchange of datasets.

2. Add Resources Group in the reference node or existing resource group node

Service Connect creates a new Resource Group to put imported resources into.

Resources Group is similar to a folder. Use the Groups to keep the links to the resources logically grouped. A RESTful Reference should contain at least one Resource Group to add individual resources into. Any Resource Group can contain both child Resource Groups and individual resources.

3. Add Resource in the existing resource group node

Service Connect creates a new representation of the resource that belongs to the imported API in the selected resource group.

A resource is a source of information. In RESTful systems, servers expose resources. You use standard URI to address resources.

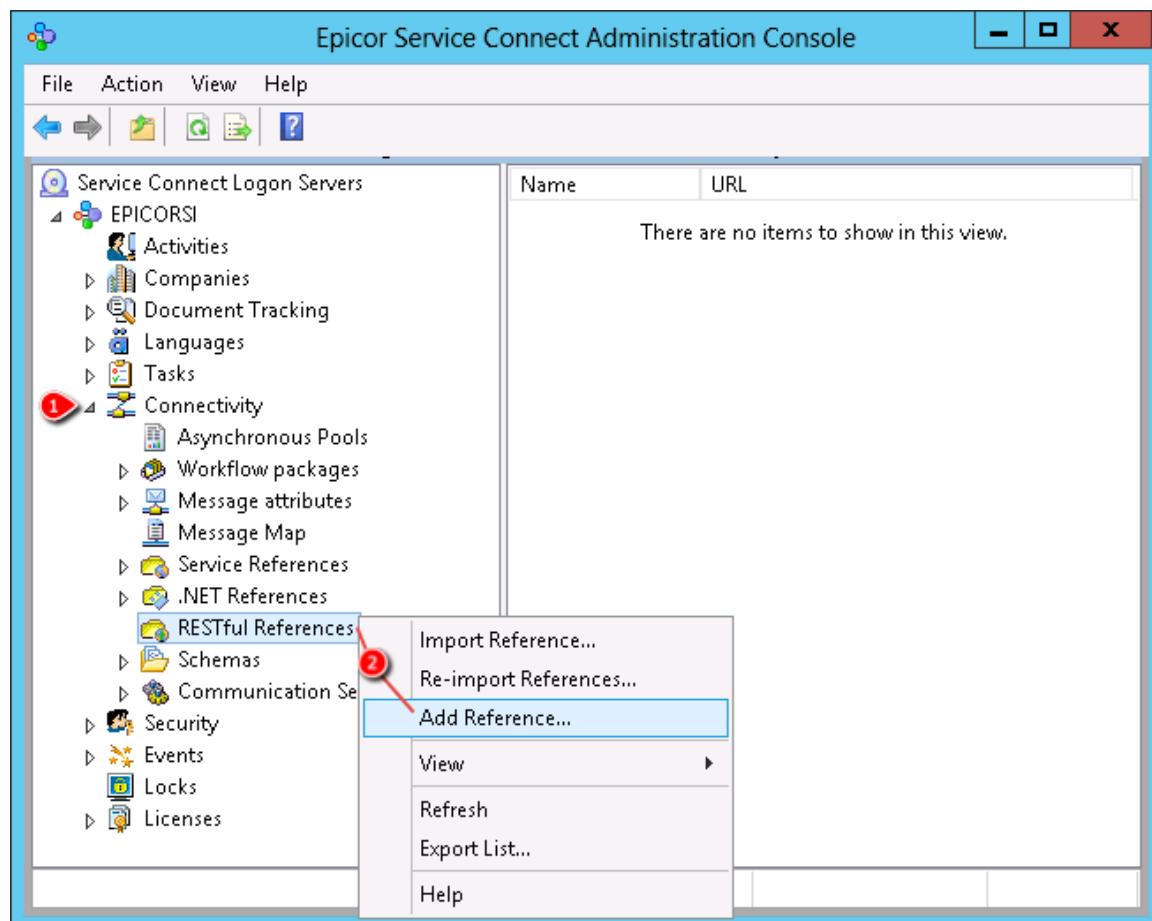
Resources can either be stored in a plain list or separated into resource groups.

4. Add Operation in the existing resource node

Service Connect creates a new operation on the selected resource. RESTful Web Service Operations are similar to Web Service methods. Each operation is defined by its name, URI template and HTTP method.

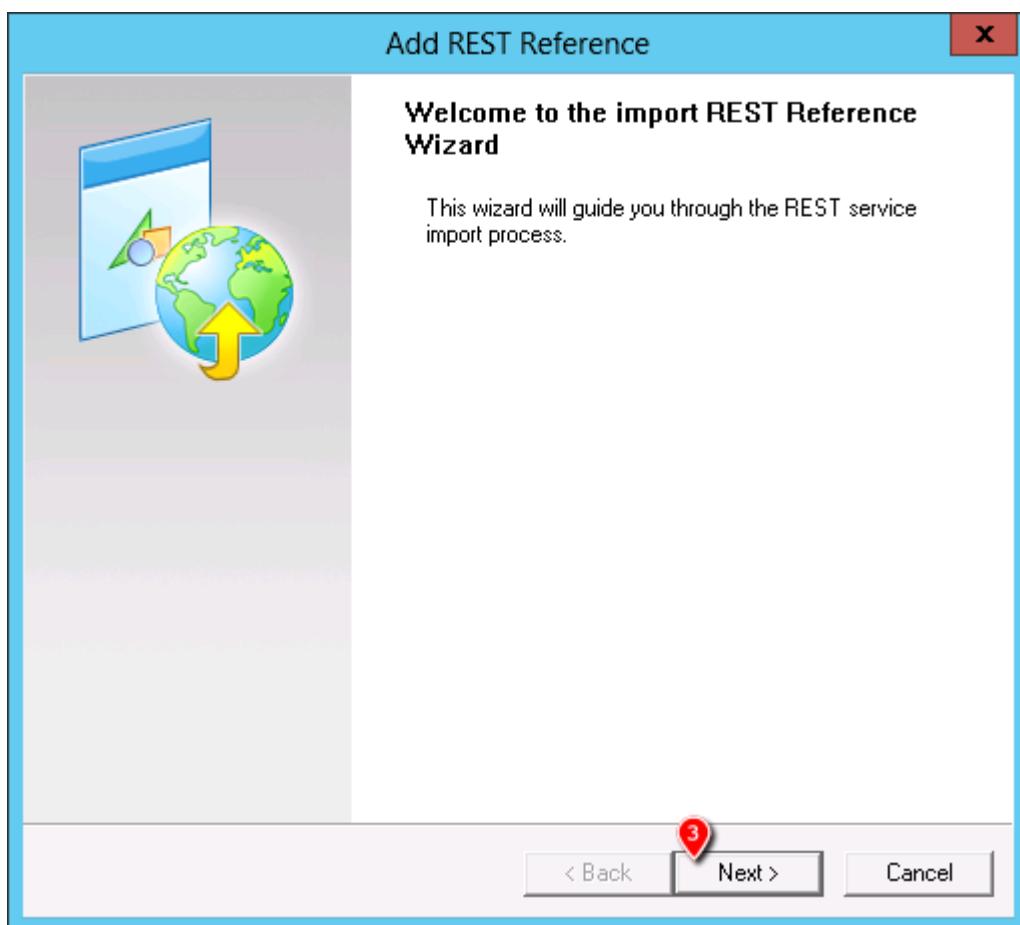
To add a RESTful reference:

1. In the **tree view**, expand the **Connectivity** node.

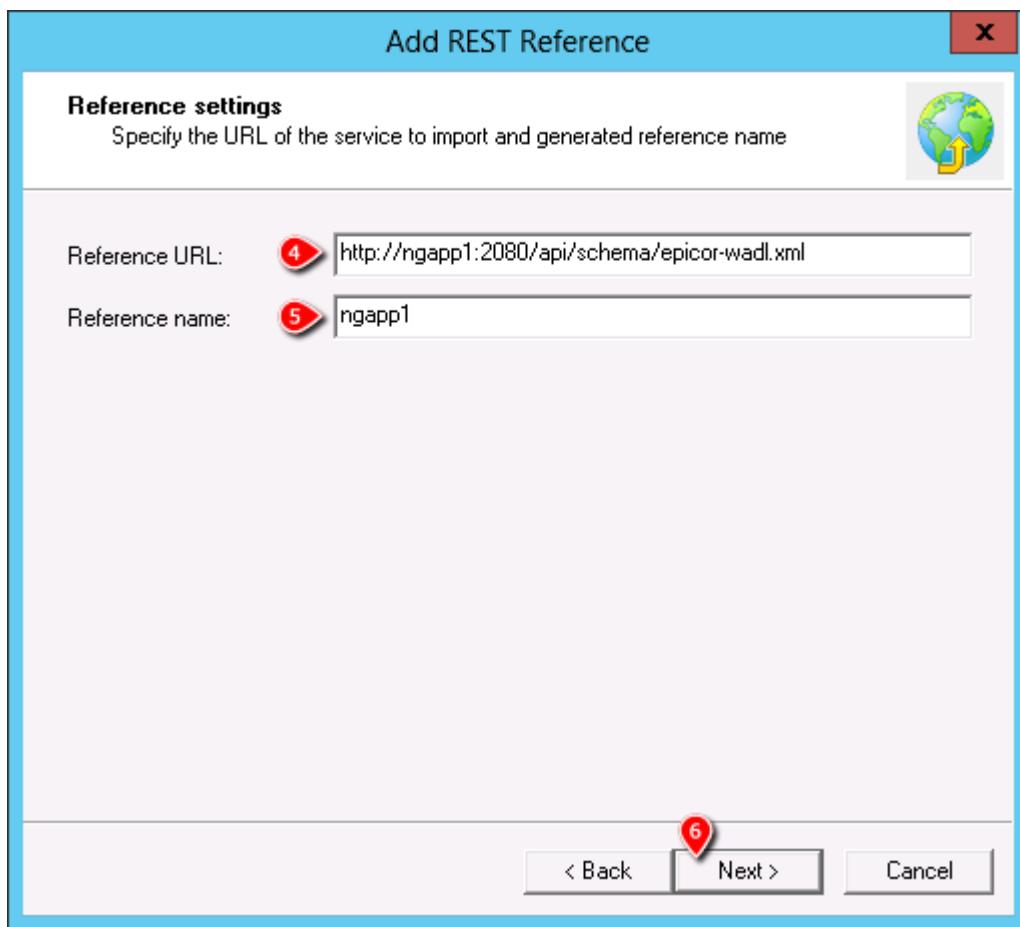


2. Right-click **RESTful References** and select **Add Reference...**.

3. On the **Add REST Reference** wizard welcome page, click **Next**.



4. On the **Reference settings** screen, in the **Reference URL** field, specify base URL of the imported RESTful service.



5. Verify the **Reference name** field displays the default RESTful reference name.

This name will display in the RESTful references list in the **RESTful References** node under the **Connectivity** node in the Administration Console.

RESTful reference name must start with a Latin letter, include only Latin letters, digits and spaces, and be no longer than 40 symbols. It must be unique for an installation.

6. Click **Next**.

7. On the **Authentication settings** screen, in the **Authentication** field, select the authentication mode. The following options are available:

- **None**
- **Basic**

Specify **User** and **Password**.

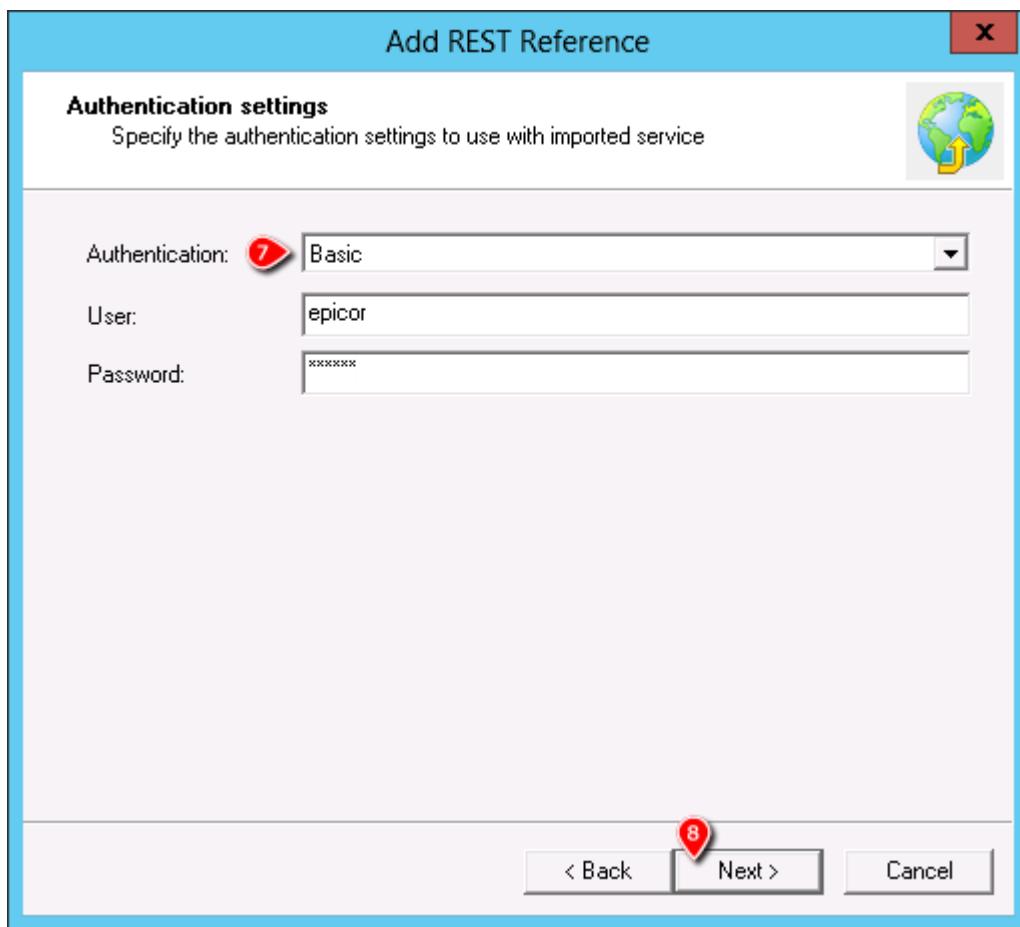
- **Windows**

Specify **Domain**, **User** and **Password**.

Alternatively, select the **Use DES service account** check box.

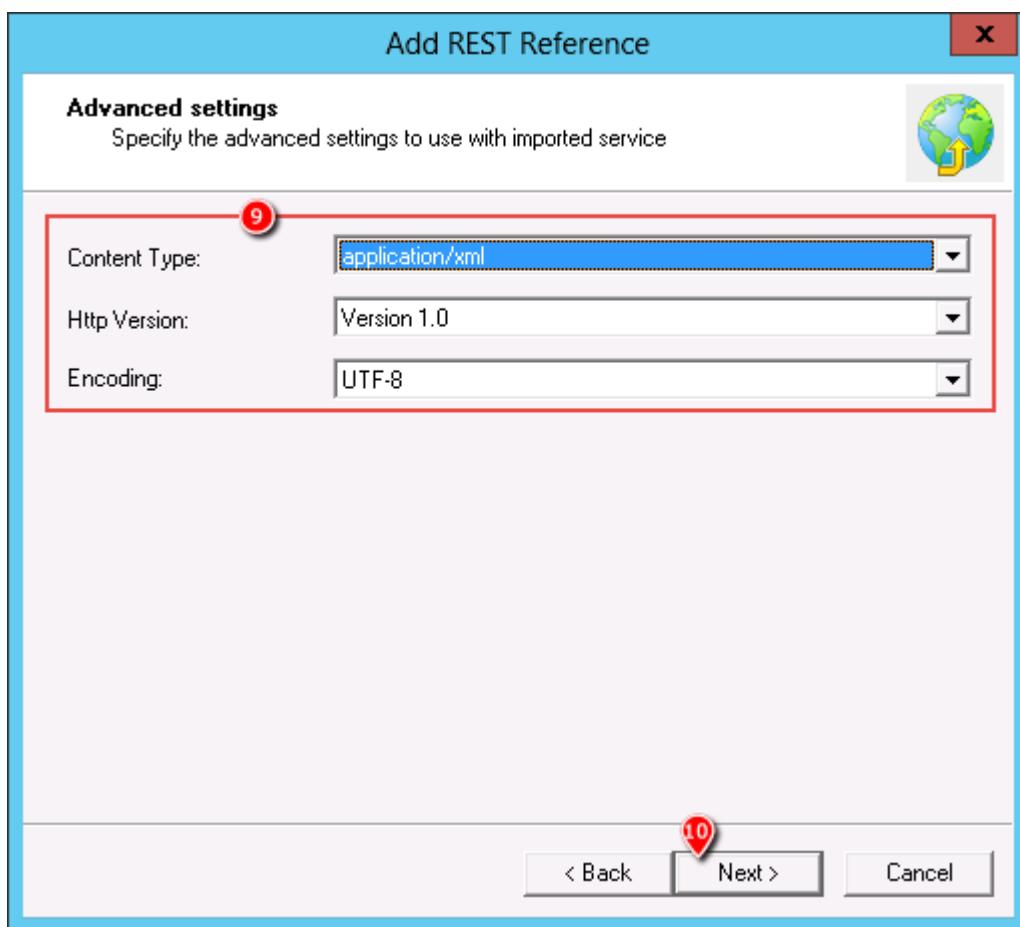
- **OAuth**

Specify **Token server URL**, **User** and **Password**.



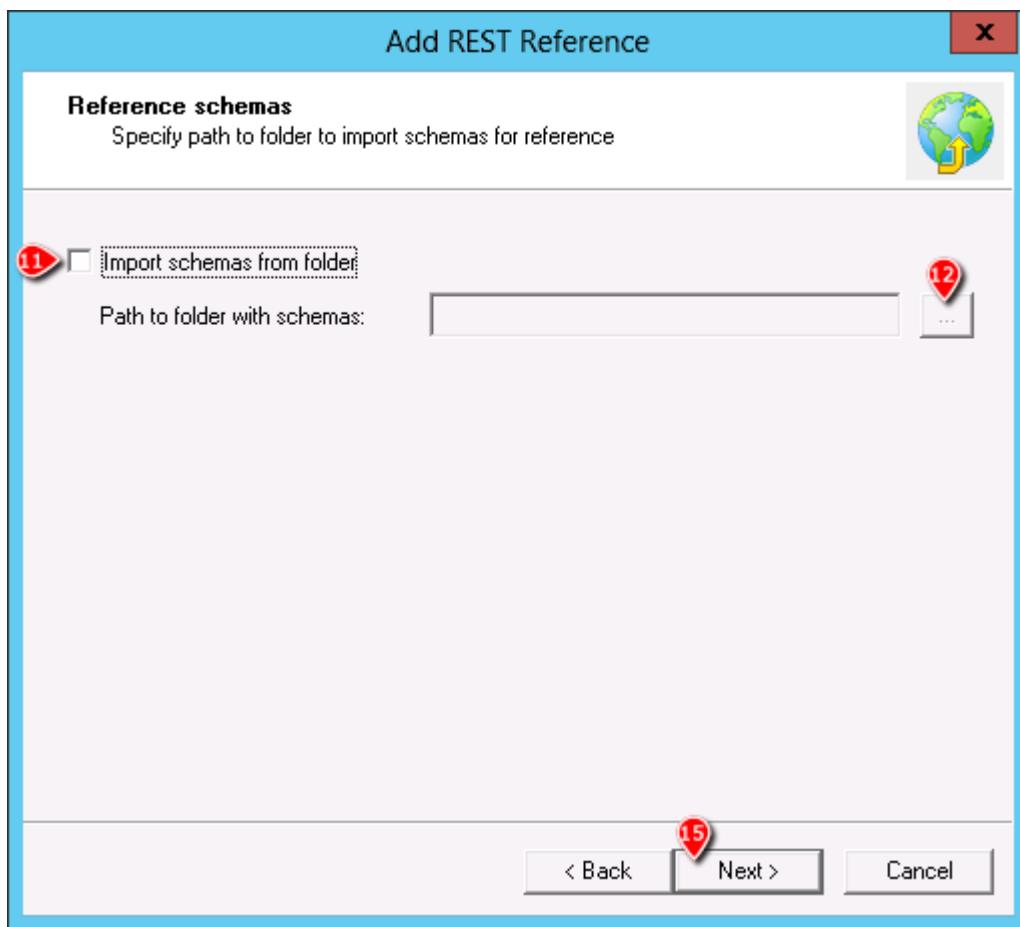
8. Click **Next**.

9. On the **Advanced settings** screen, specify content type, http version and encoding.



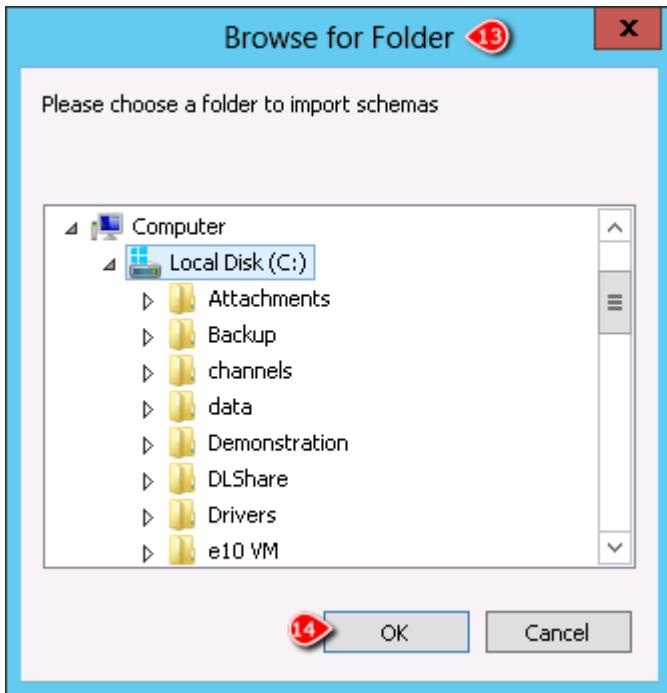
10. Click **Next**.

11. If REST schemas are available, on the **Reference schemas** screen, select the **Import schemas from folder** check box.



12. Next to the **Path to folder with schemas** field, click the ellipsis button.

13. In the **Browse for Folder** window, browse to and select the folder.



14. Click **OK**.

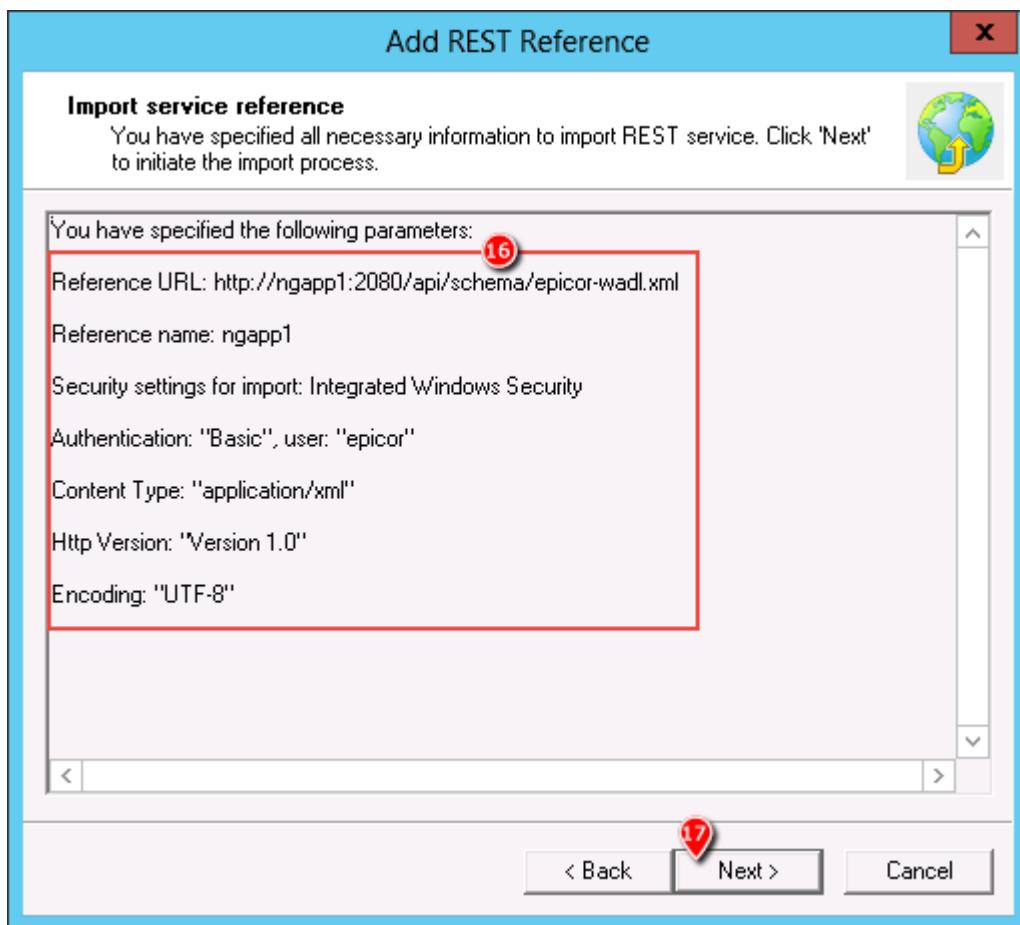
The path to the schemas folder displays in the **Path to folder with schemas** field.

This is a path on the Service Connect server.

You can also use REST reference context menu to specify or change the schemas folder for importing.

15. Click **Next**

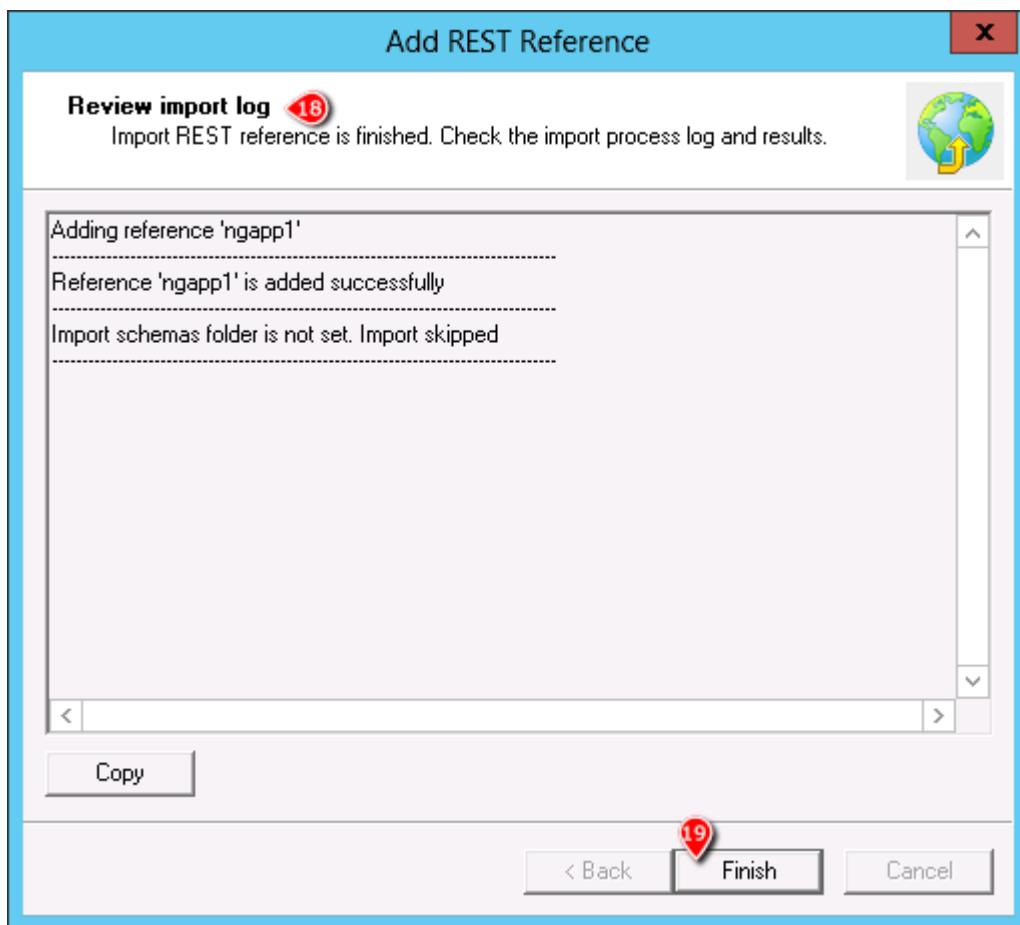
16. Review the import information you provided.



17. Click **Next** to start import process.

Alternatively click **Back** to change the import parameters.

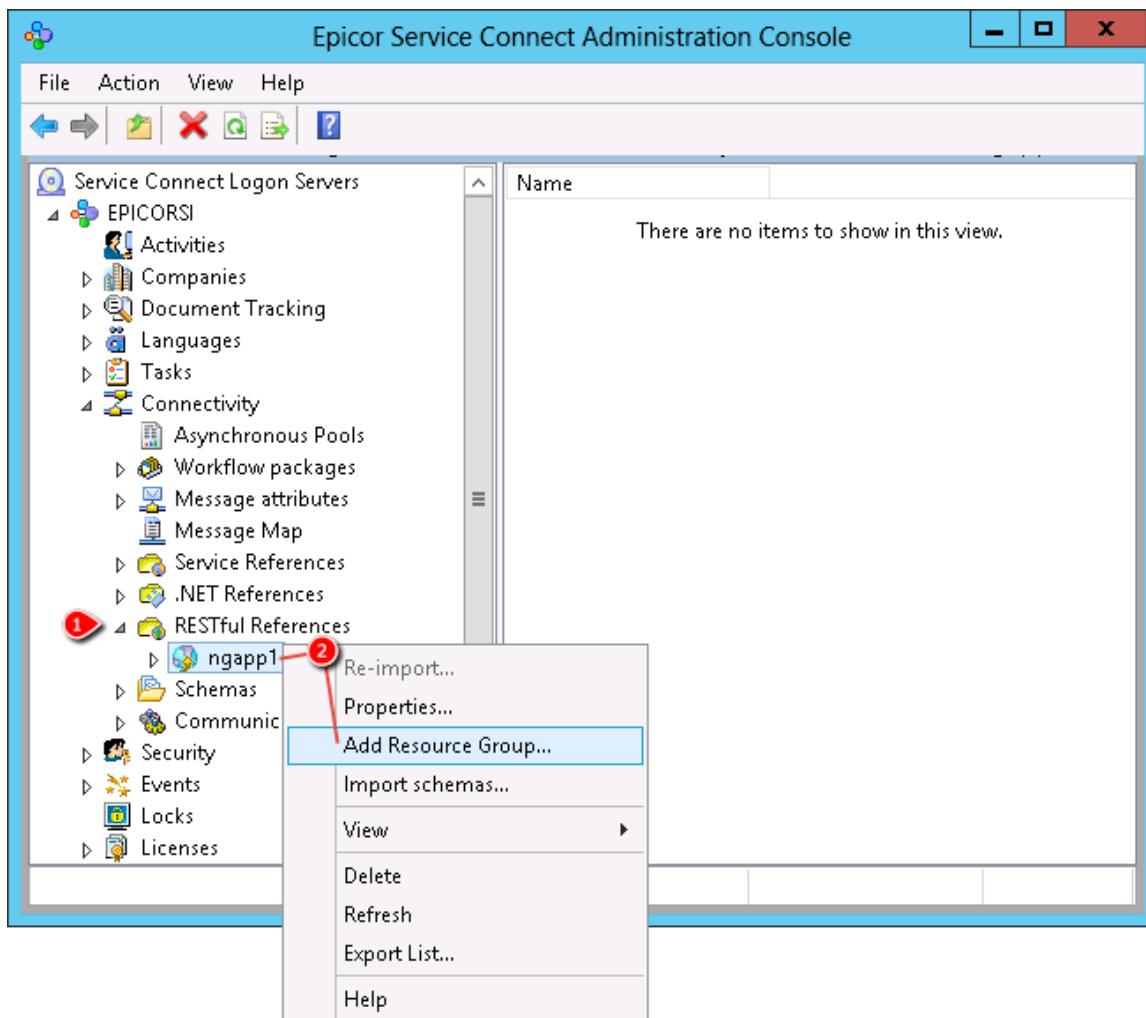
18. When the import is complete, the **Review import log** screen is displayed.



19. Click **Finish**.

Add Resource Group

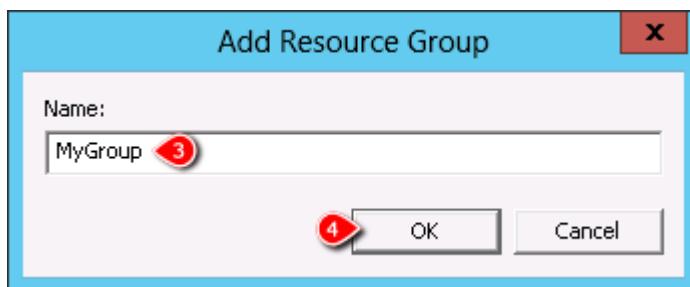
1. In the **tree view**, expand the **Connectivity > RESTful References** node.



2. Right-click a RESTful reference node and select **Add Resource Group**.

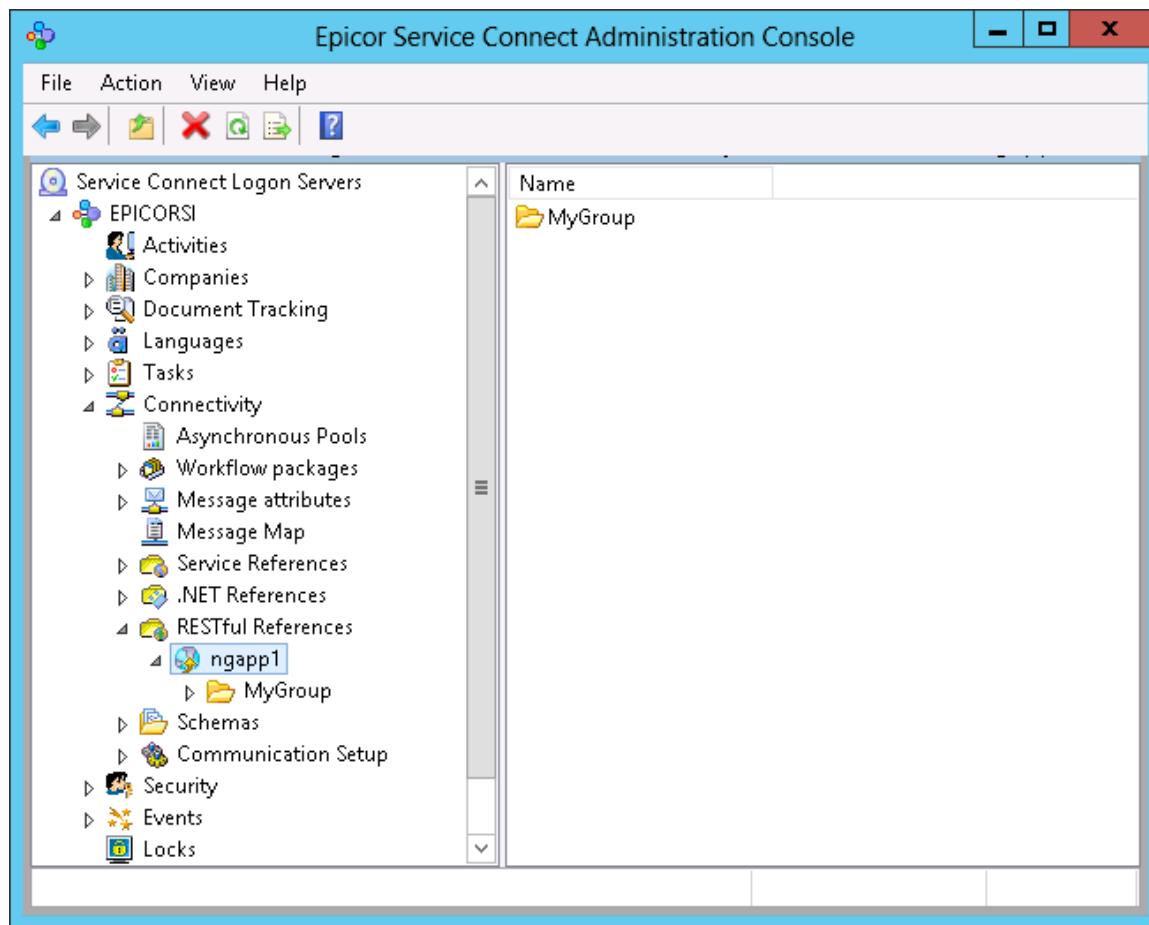
You can also right-click an existing resource group node and select **Add Resource Group** to add a child node to it.

3. In the **Add Resource Group** dialog, enter the name for the new group.



4. Click **OK**.

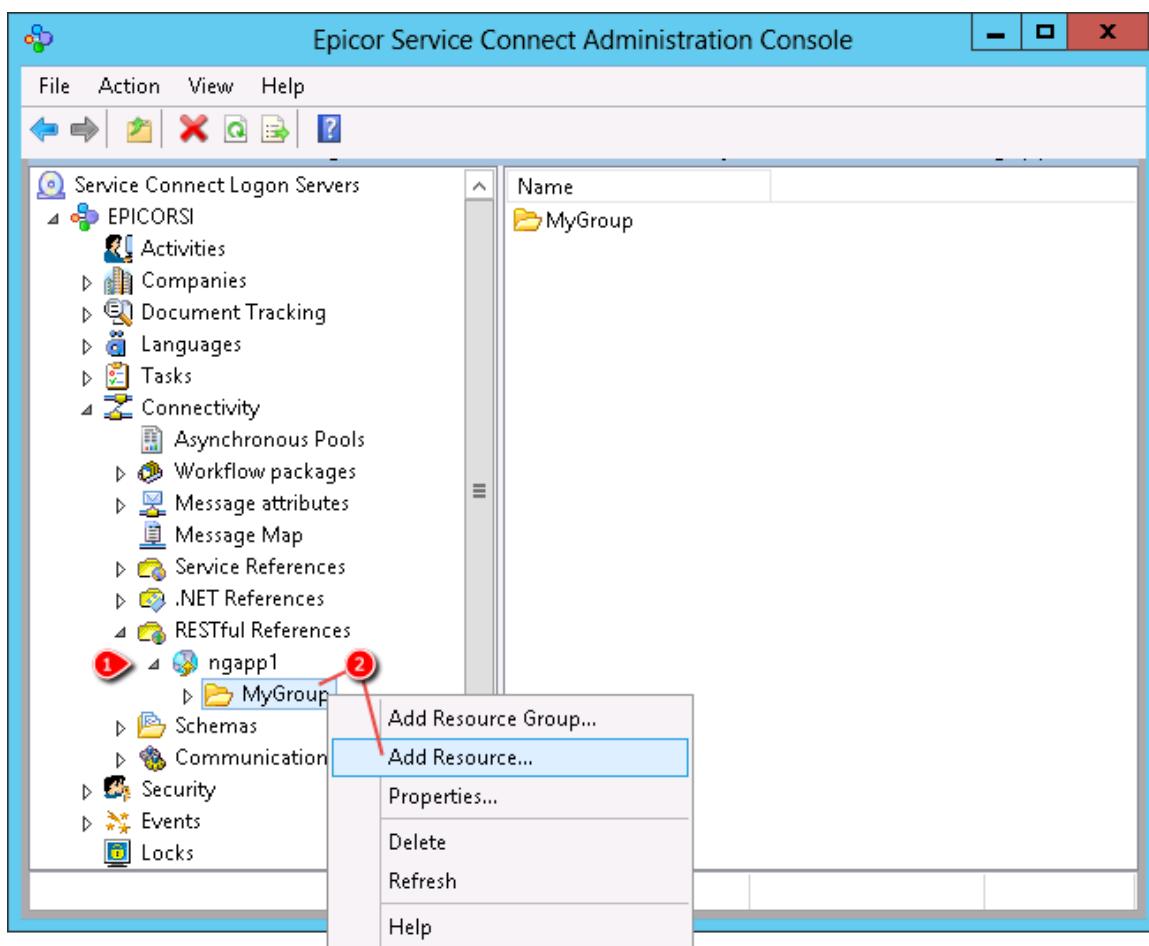
The new group displays in the Console tree in the left pane and in the right pane.



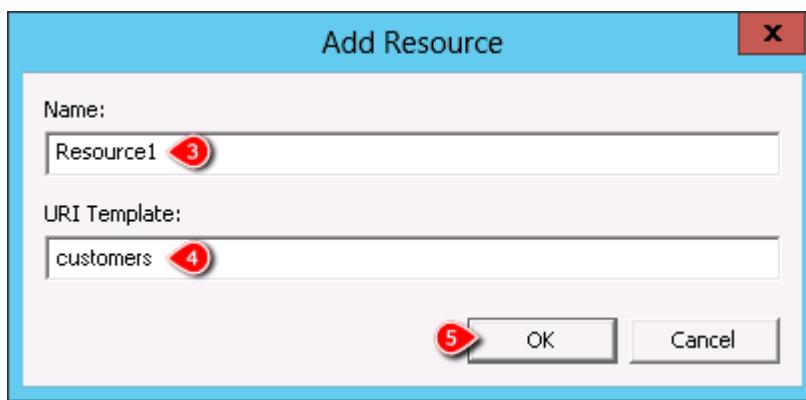
Add Resource

To add a Resource:

1. In the tree view, expand the **Connectivity > RESTful References > <RESTful reference>** node.



2. Right-click a Resource Group node and select **Add Resource**.
3. In the **Add Resource** dialog, enter the name for the new resource.

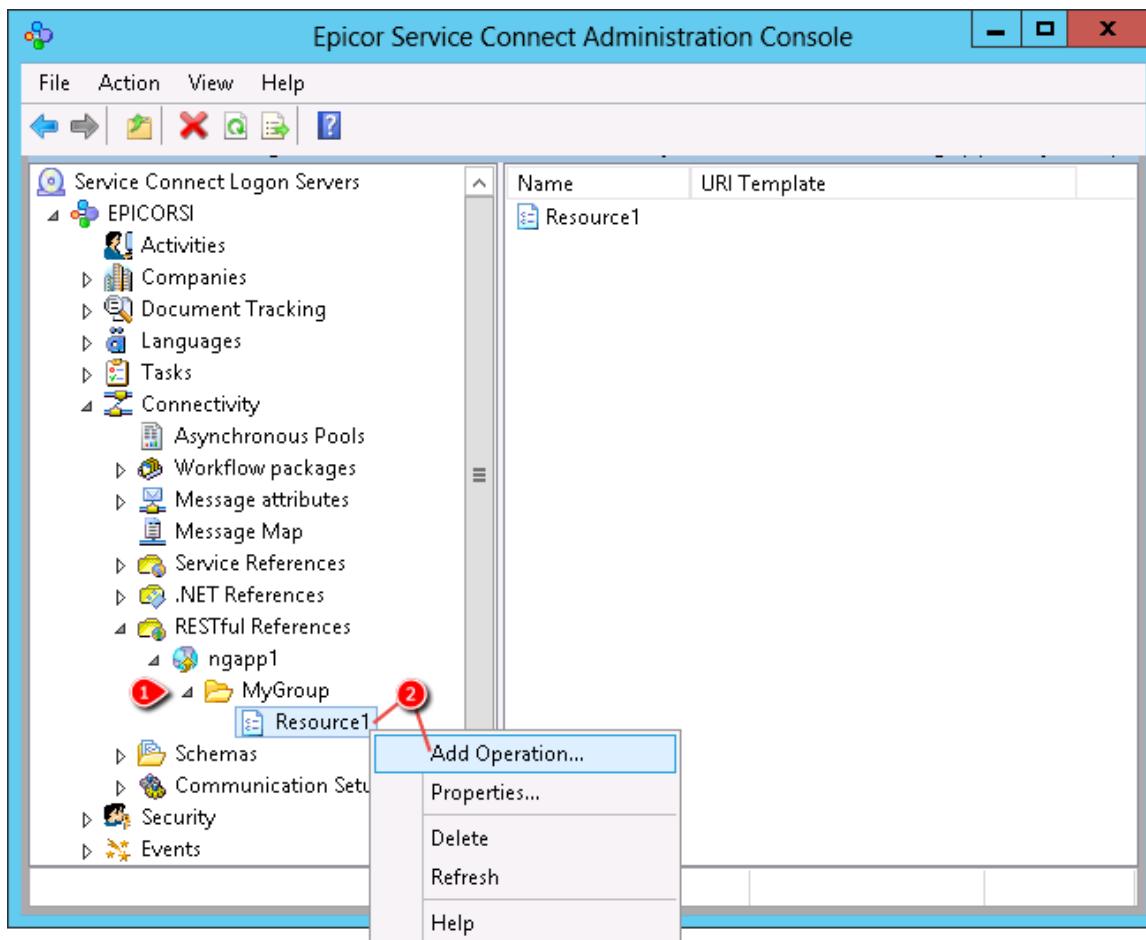


4. Optionally, enter the new resource URI template.
URI template is an optional field. The value of the URI template assigned to the resource can be overridden on any of its operations.
5. Click **OK**.

Add Operation

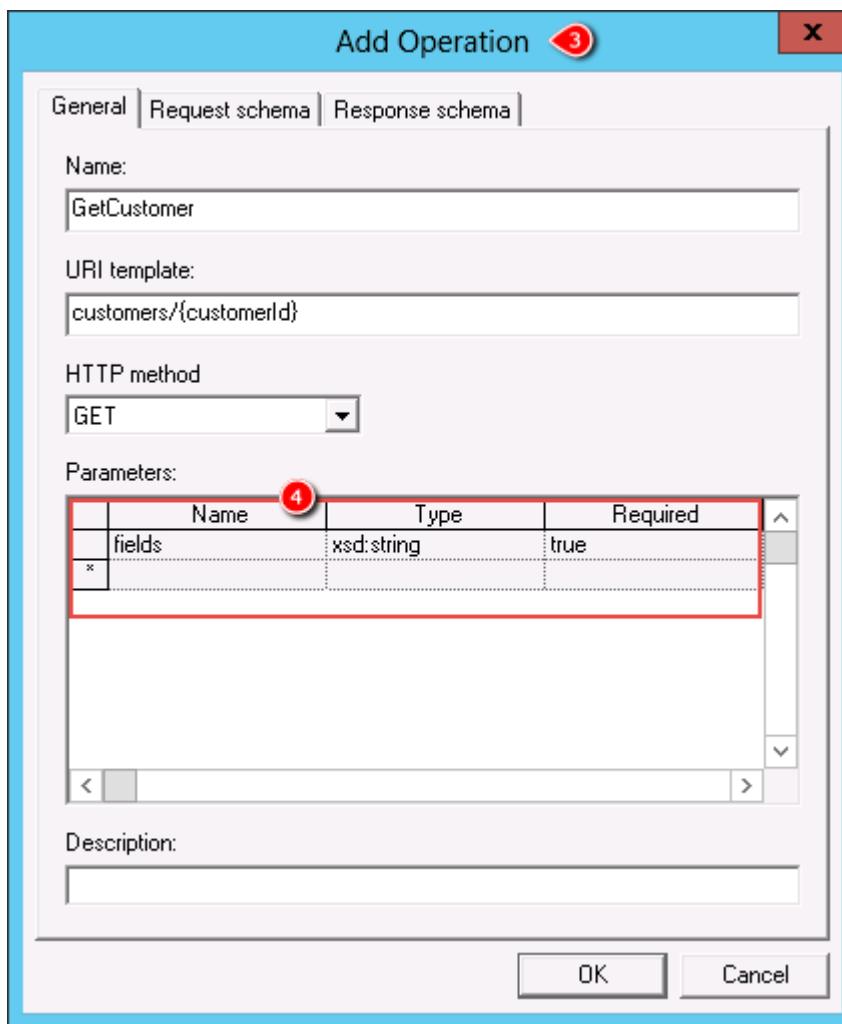
To add an Operation:

1. In the **tree view**, expand the **Connectivity > RESTful References > <RESTful reference> > <Resource Group>** node.



2. Right-click a Resource node and select **Add Operation**.

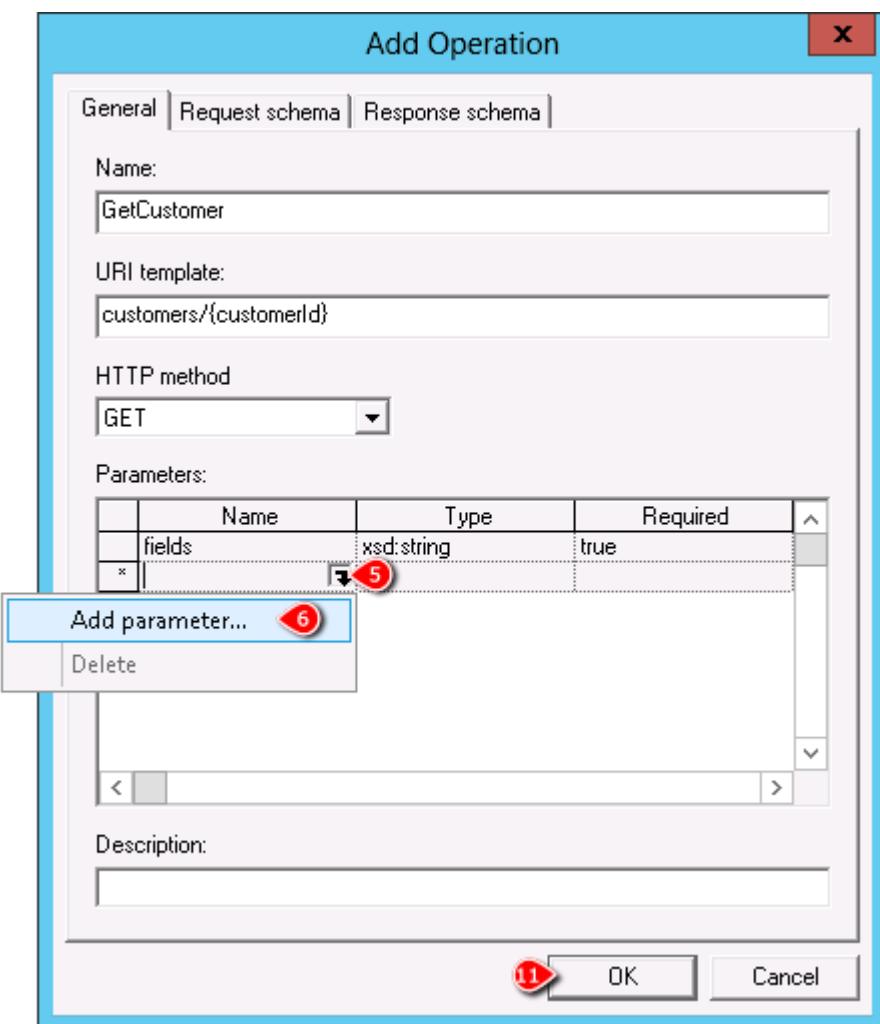
3. In the **Add Operation** dialog, enter the name for the new operation, its URI template, HTTP method used, and operation description.



URI template is an optional field. In case you leave it blank for the particular operation, the value is derived from the resource configuration.

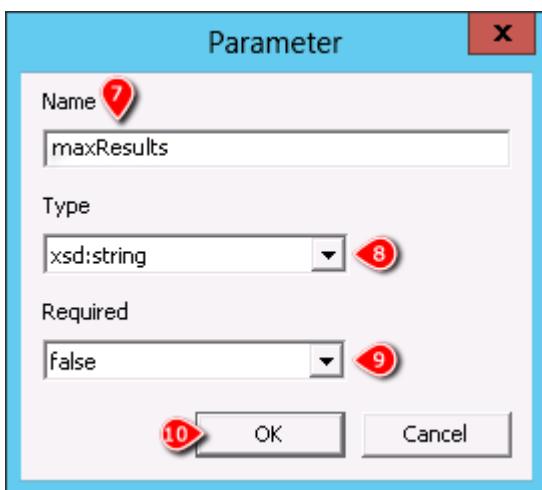
4. Use the **Parameters** pane to set up additional parameters of URI template.

5. To add a new parameter, click in an empty row in the **Name** column.



6. Select **Add parameter**.

7. Specify a new parameter **Name**.



8. In the **Type** field, select the parameter type.

9. In the **Required** field, specify if the parameter must be present or not.

The generated schema for Rest element will contain a **Parameters** node with all the parameters as child nodes.

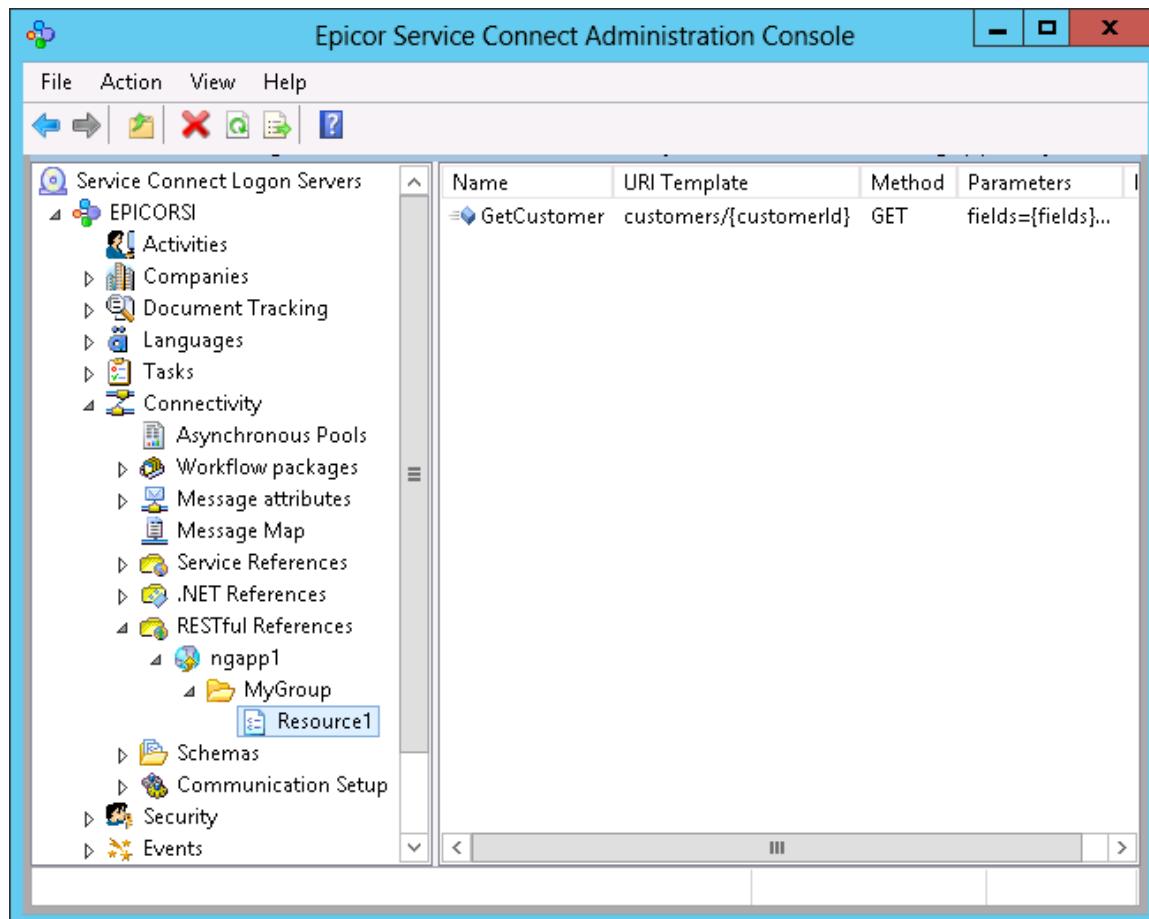
The parameters you create are available for mapping.

10. In the **Parameter** dialog, click **OK**.

The new Operation displays in the Parameters list.

11. In the **Add Operation** dialog, click **OK**.

The new Operation displays in the Console right pane.

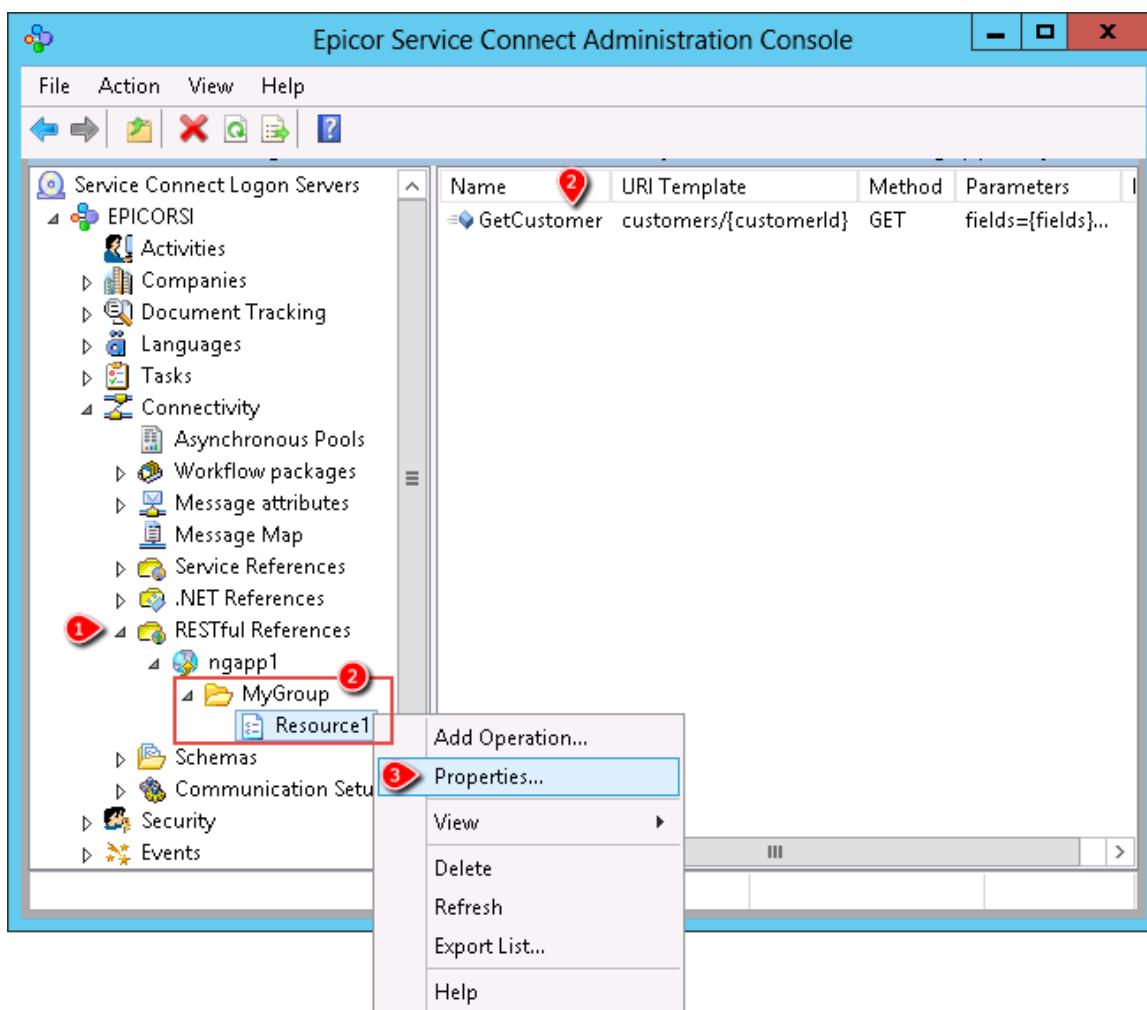


After you created references, resource groups, resources and operations, you can edit them. You can edit any of the initially supplied properties. Modification of some of them, though, can render other Epicor Service Connect objects invalid. For example, changing the name of the resource group or individual resource causes re-generation of one or more RESTful Reference operation Request IDs.

Edit Resource Group, Resource, or Operation

To edit Resource Group, Resource, or Operation:

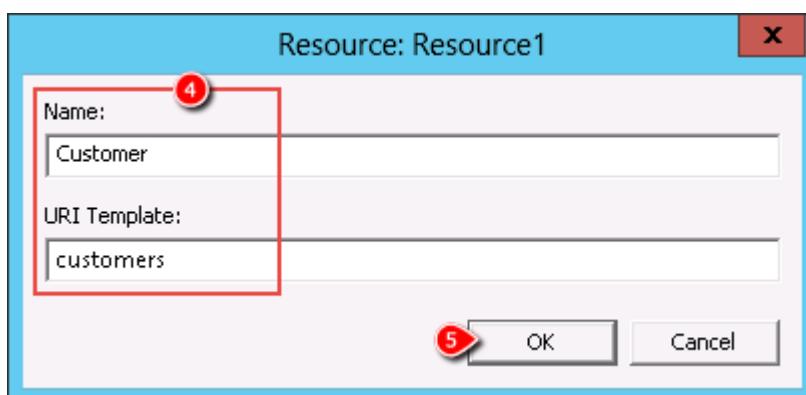
1. In the tree view, expand the **Connectivity > RESTful References** node.



2. Right-click a Resource Group, Resource, or Operation to edit.

3. Select **Properties**.

4. Edit the properties.



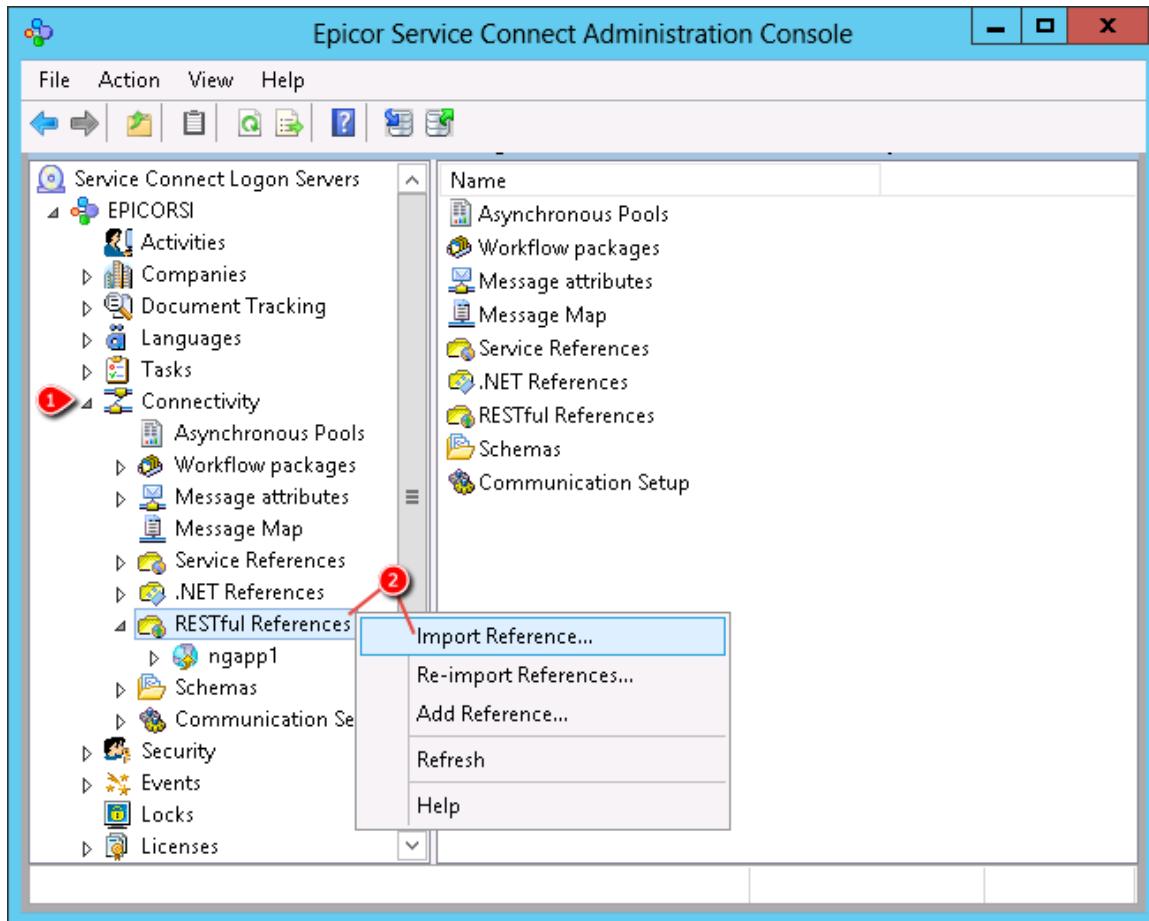
5. Click **OK**.

Import a RESTful Reference Automatically

Some RESTful web services expose metadata which can be a simple xml file which describes the whole API structure: schemas, parameters, and so on.

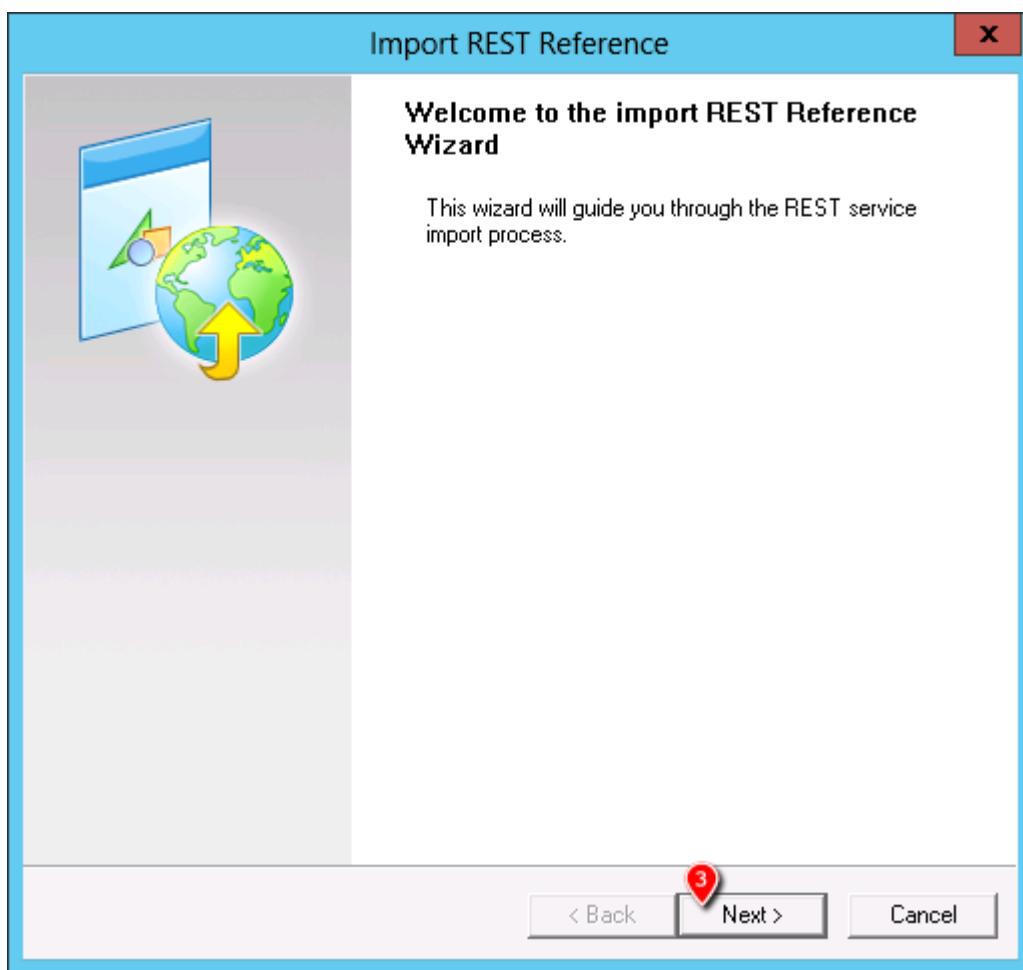
To import a RESTful reference automatically:

1. In the **tree view**, expand the **Connectivity** node.

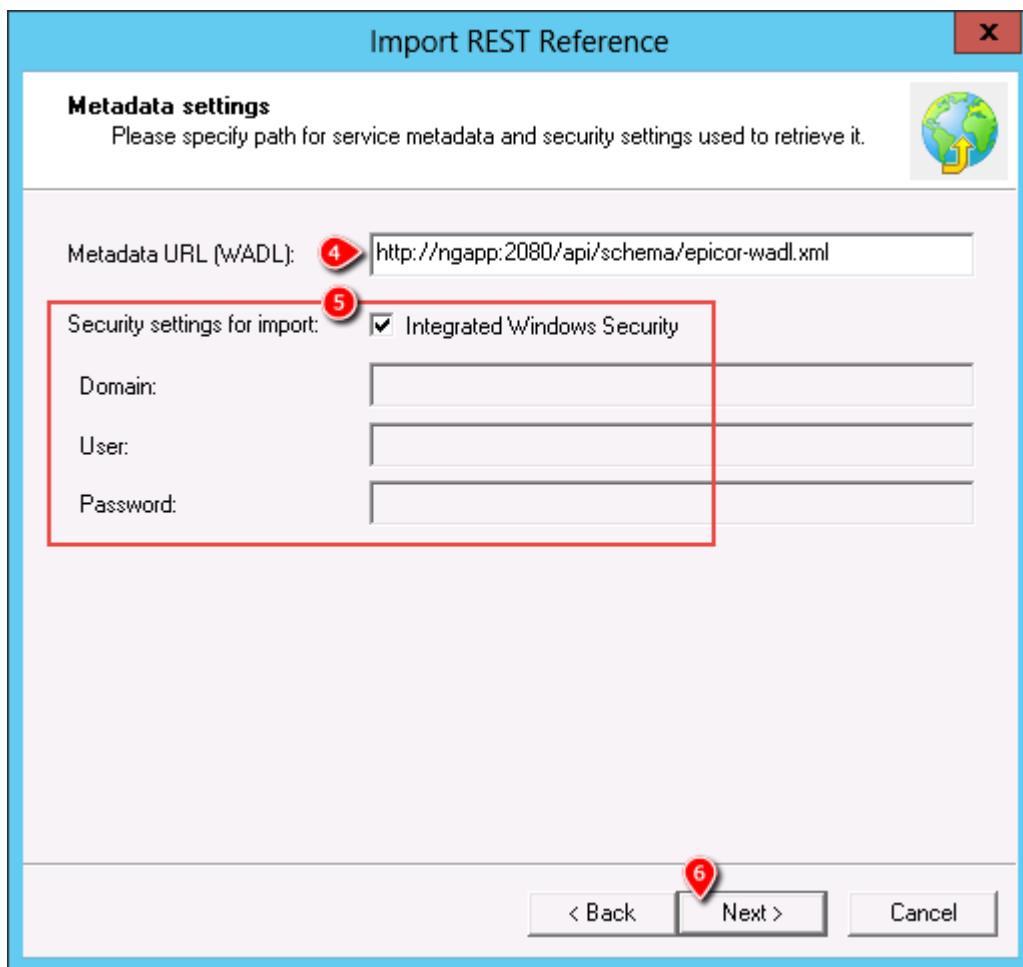


2. Right-click **RESTful References** and select **Import Reference**.

3. On the **Import REST Reference** wizard welcome page, click **Next**.

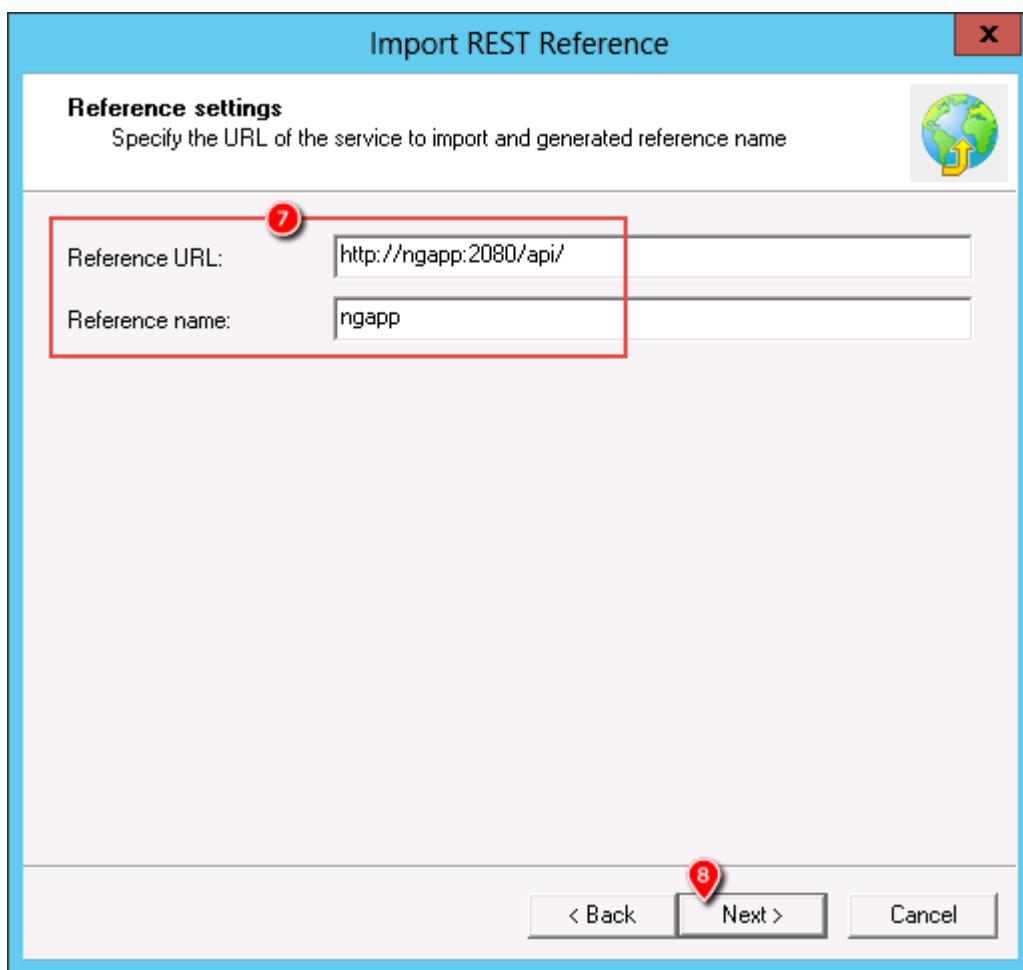


4. On the **Metadata settings** screen, in the **Metadata URL (WADL)** field, specify Metadata URL of the imported web service.



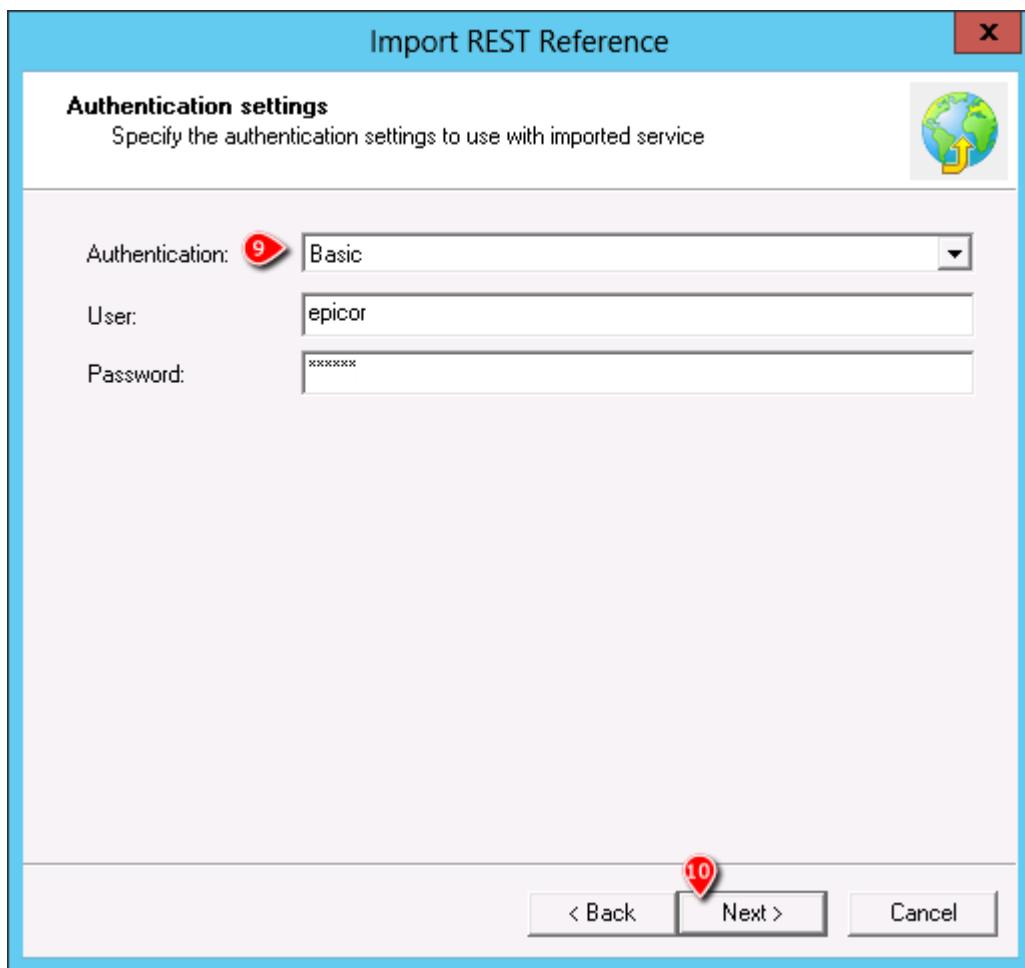
5. Specify the security settings.
6. Click **Next**.
7. On the **Reference settings** screen, view the automatically generated Reference URL and name.

You can edit these properties.



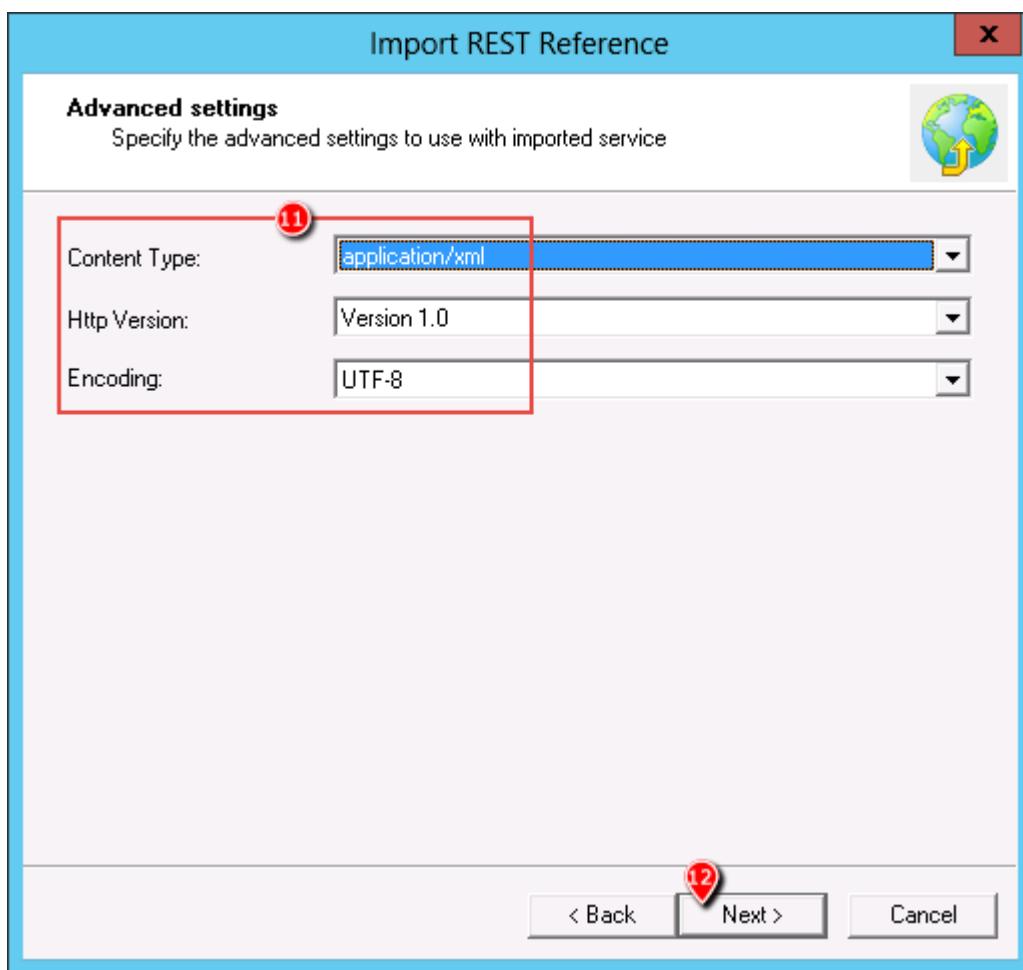
8. Click **Next**.
9. On the **Authentication settings** screen, in the **Authentication** field, select the authentication mode. The following options are available:
 - **None**
 - **Basic**
Specify **User** and **Password**.
 - **Windows**
Specify **Domain**, **User** and **Password**.
Alternatively, select the **Use DES service account** check box.
 - **OAuth**

Specify **Token server URL**, **User** and **Password**.



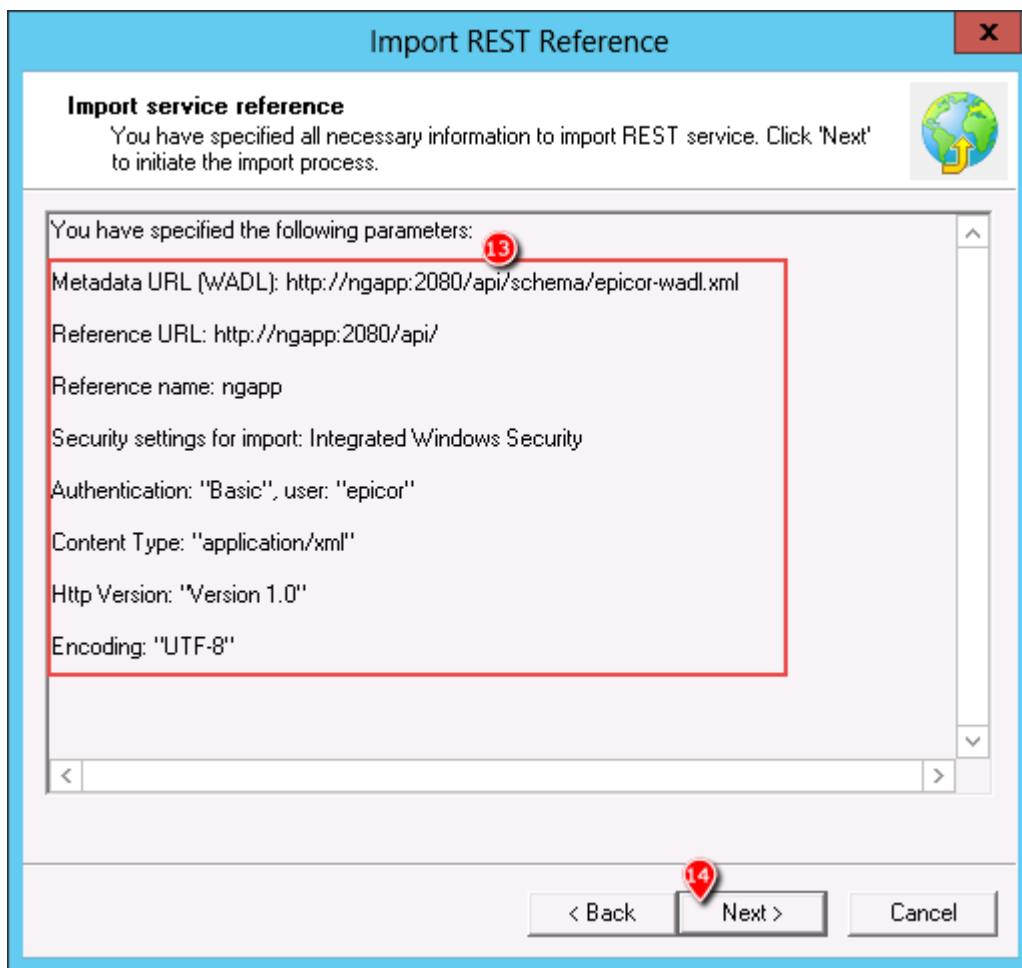
10. Click **Next**.

11. On the **Advanced settings** screen, specify content type, http version and encoding.



12. Click **Next**.

13. Review the import information you provided.



14. Click **Next** to start import process.

Alternatively click **Back** to change the import parameters.

15. When the import is complete, the **Review import log** screen is displayed.

16. If the RESTful Reference import failed, you can click the **Back** button to go back and adjust the reference settings. Or click **Cancel** to cancel the import operation.

17. Click **Finish**.

Schemas

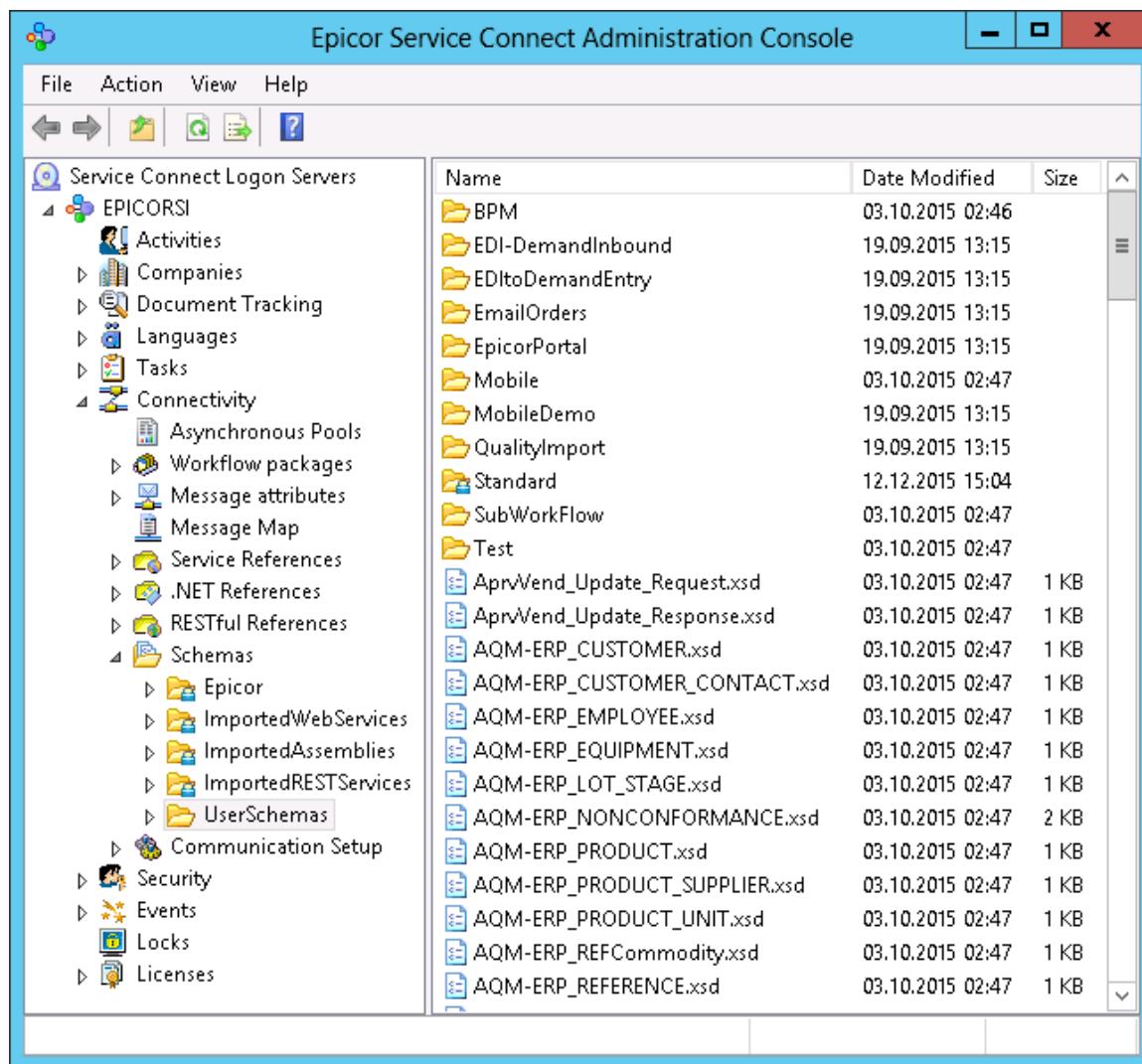
Use the Schemas node in the Epicor Service Connect Administrator Console to perform schema-related tasks.

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.

Read-only folders are marked with a lock sign on their icons.
Operations that create, change or delete schemas are not available in such folders.



For more information on schemas, review Workflow Designer chapter and Managing Schemas topic in the Service Connect application Help.

Schema snap-in supports multi-site ESC installations.

Asynchronous Pools

Use Asynchronous Pools to handle incoming asynchronous calls for message processing. The main property of a Pool is the maximum count of threads that limits the number of processes executing at the same time that receive messages from asynchronous channels and from asynchronous workflows.

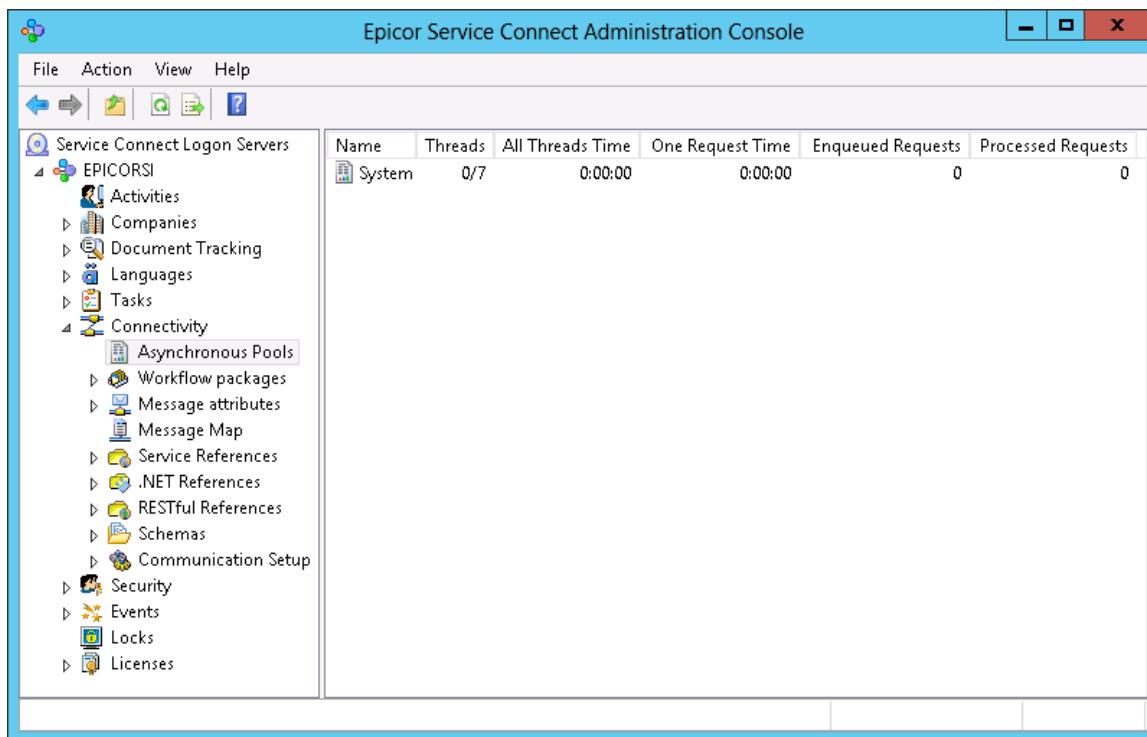
For example, if you set the thread number to 1, messages in the incoming queue will be processed one by one. If the thread number is 20, processing of up to 20 incoming messages will start simultaneously. Asynchronous processing strongly preserves the order of messages incoming into the Pool and starts processing messages according to this order.

For synchronous workflow processing the Pool setting is ignored.

By default the Pool named **System** is always available in the Service Connect installation. It is used if no other pool is set for the workflow.

The following fields display for each Asynchronous Pool in the Connectivity > Asynchronous Pools node in Epicor Service Connect Administration Console:

- **Name** – This is the Pool name.
- **Threads** – The maximum number of processes that receive messages from asynchronous channels and from asynchronous workflows. Thread number can take on a value from 1 to 99.
- **All Threads Time**– This is the summarized time of all processed message requests.
- **One Request Time** – This field displays the average time of message request processing.
- **Enqueued Requests** – This is the number of messages to be processed.
- **Processed Requests** – This is the number of successfully processed messages.
- **State** – This field displays if the Asynchronous Pool is Active or Paused.



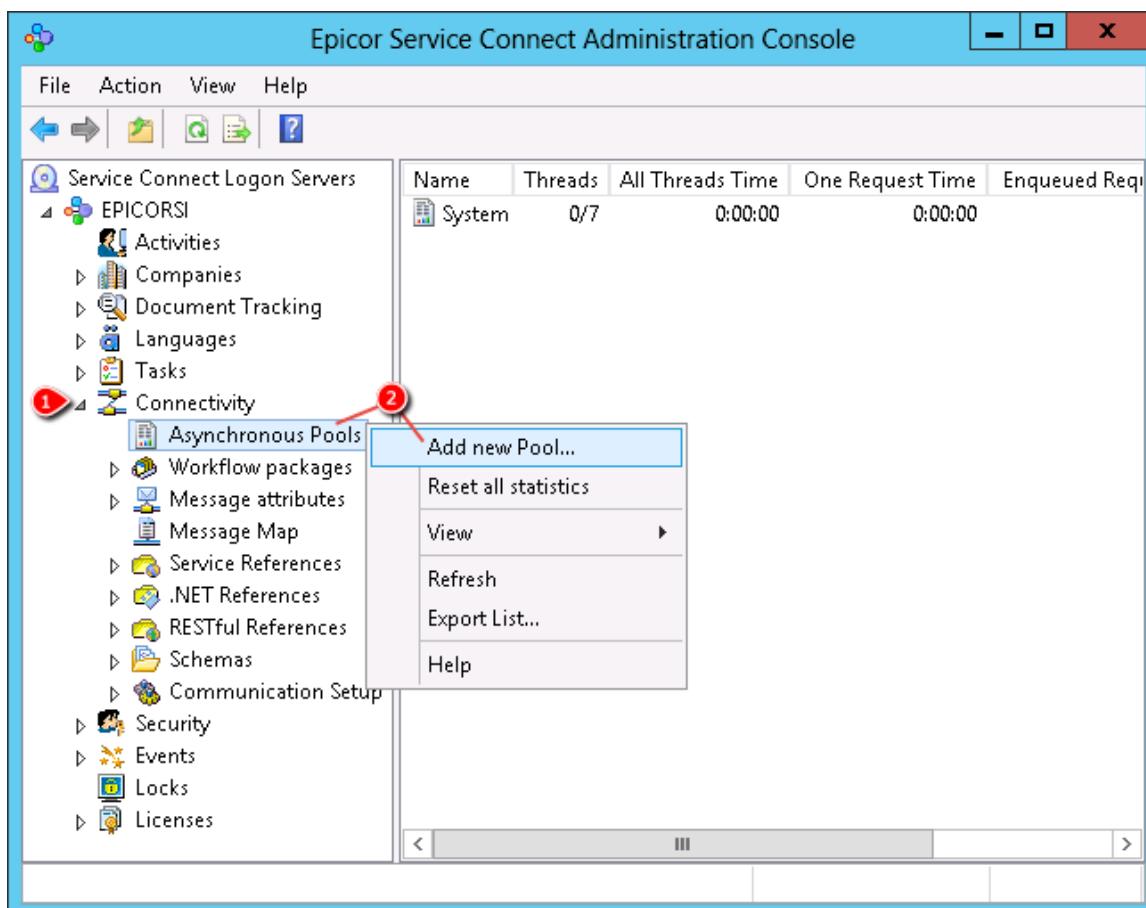
You can assign an Asynchronous Pool by name to a workflow or to a sub-workflow element. Refer to Workflow Designer chapter for more information on how to assign an Asynchronous Pool.

Manage Asynchronous Pools

You can perform the following tasks with Asynchronous Pools in ESC Administration Console.

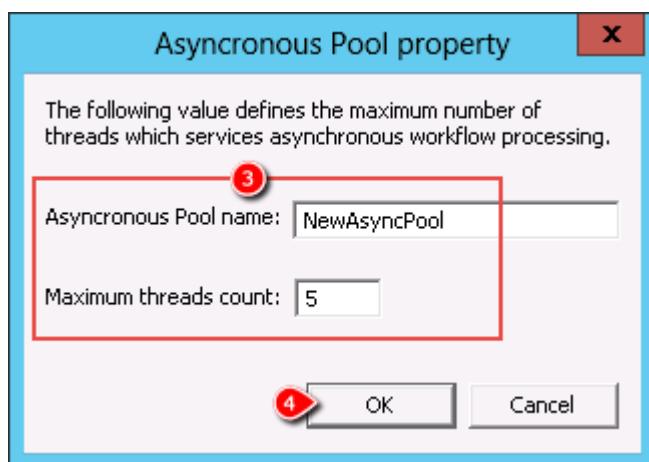
Create an Asynchronous Pool

1. In the **tree view**, expand the **Connectivity** node.



2. Right-click the **Asynchronous Pools** node, and select **Add new Pool**.

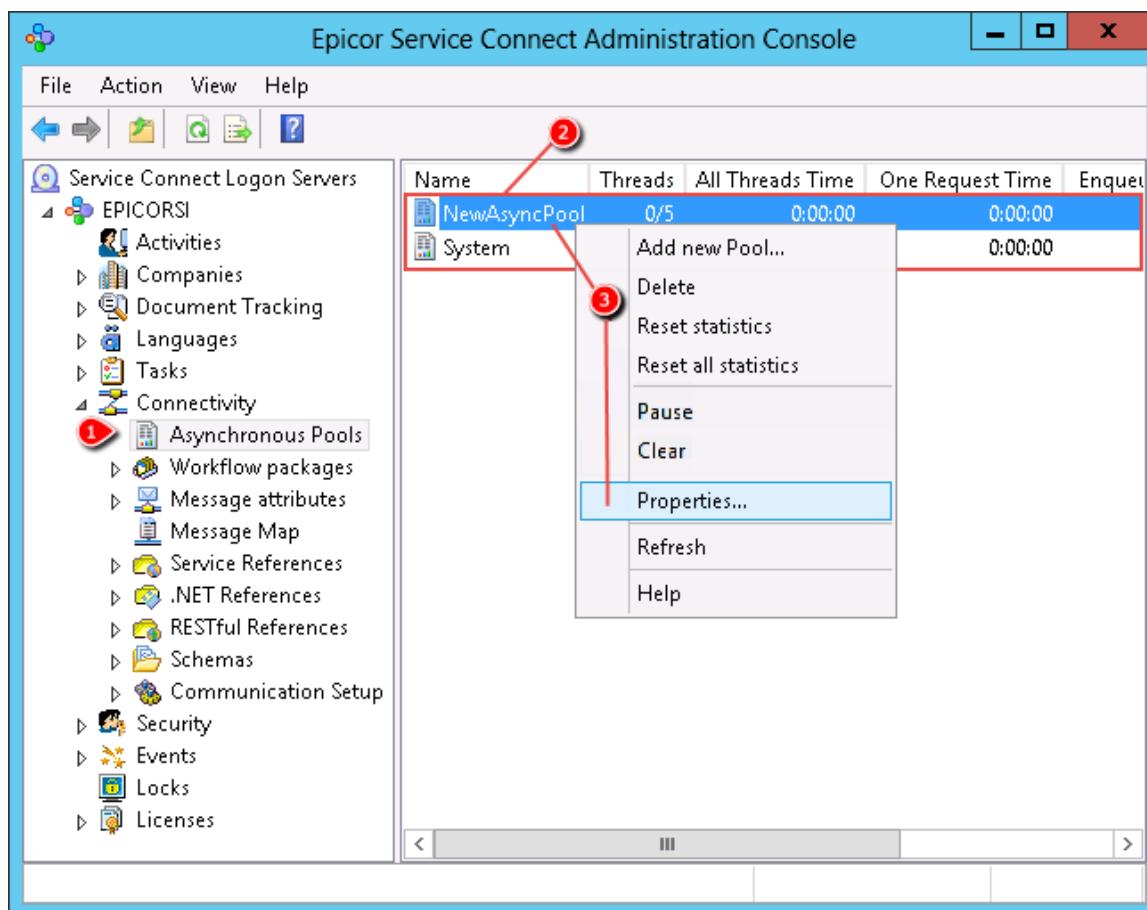
3. In the **Asynchronous Pool property** dialog, define the **Pool name** and the maximum **Threads** count in the corresponding fields.



4. Click **OK**.

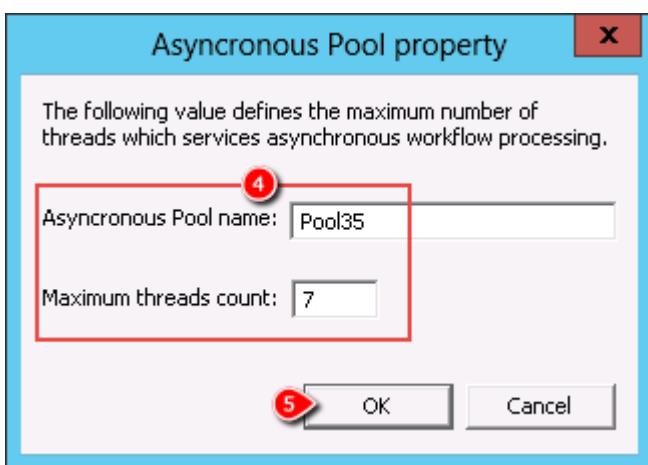
Edit an Asynchronous Pool

1. In the **tree view**, expand the **Connectivity > Asynchronous Pools** node.



2. In the right pane, view the list of all the Asynchronous Pools registered in the system.

3. Right-click the Asynchronous Pool you want to edit and select **Properties**.
4. In the **Asynchronous Pool property** dialog, update the **Pool name** and the **Maximum threads count** in the corresponding fields.

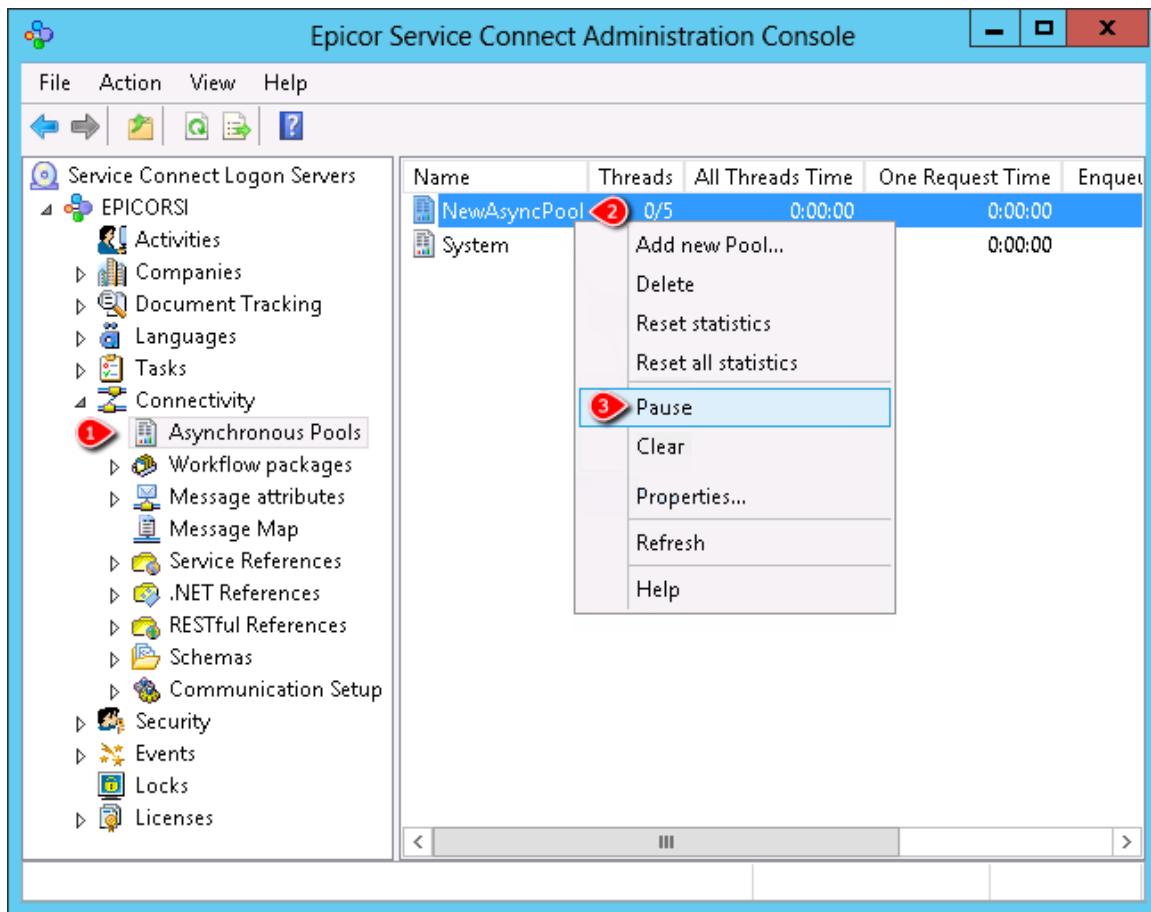


5. Click **OK**.

Pause an Asynchronous Pool

1. In the **tree view**, expand the **Connectivity > Asynchronous Pools** node.

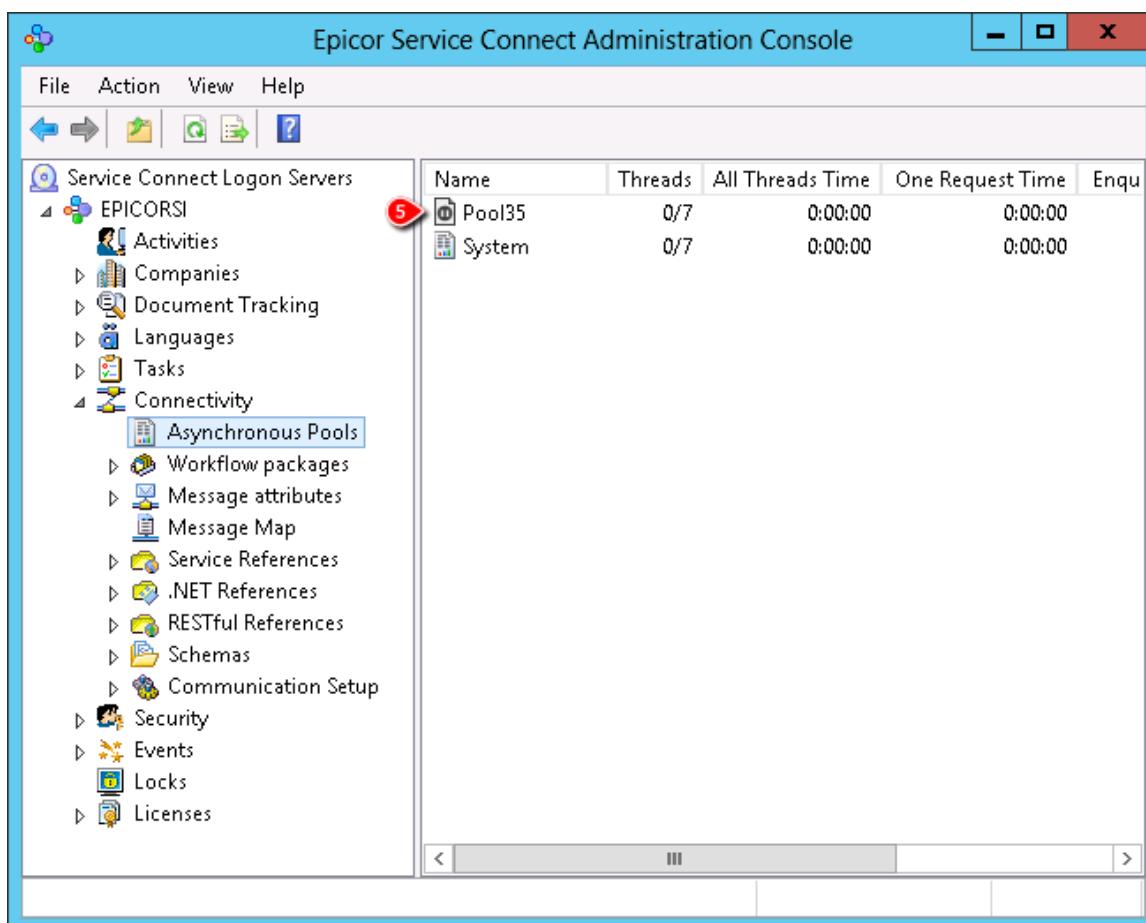
2. In the right pane, from the list of all the Asynchronous Pools registered in the system, right-click the Asynchronous Pool you want to pause.



3. Select **Pause**.

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

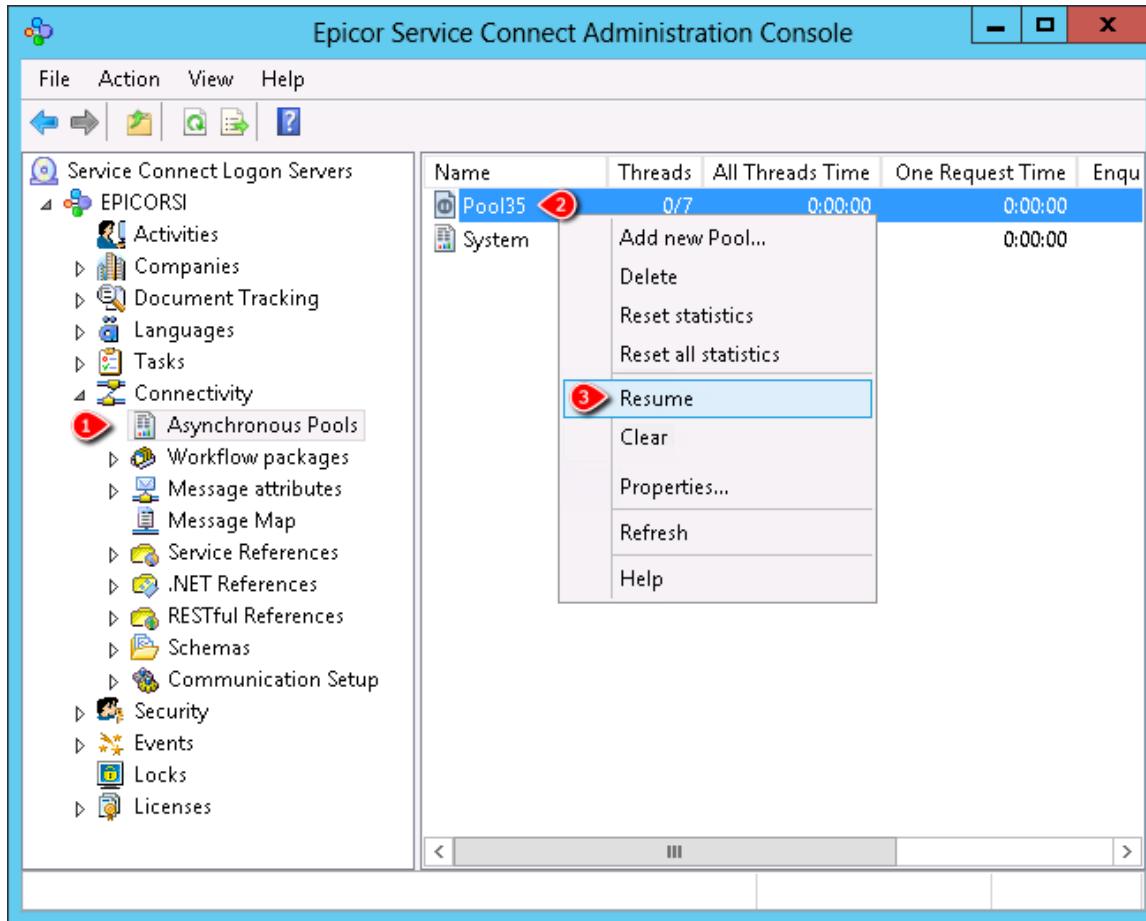
5. Notice, the Pool now has an icon with the pause sign on it and the **State** column for the Pool displays **Paused**.



When the pool is paused, messages enqueued to it remain in its queue and are not processed until the pool resumes the work.

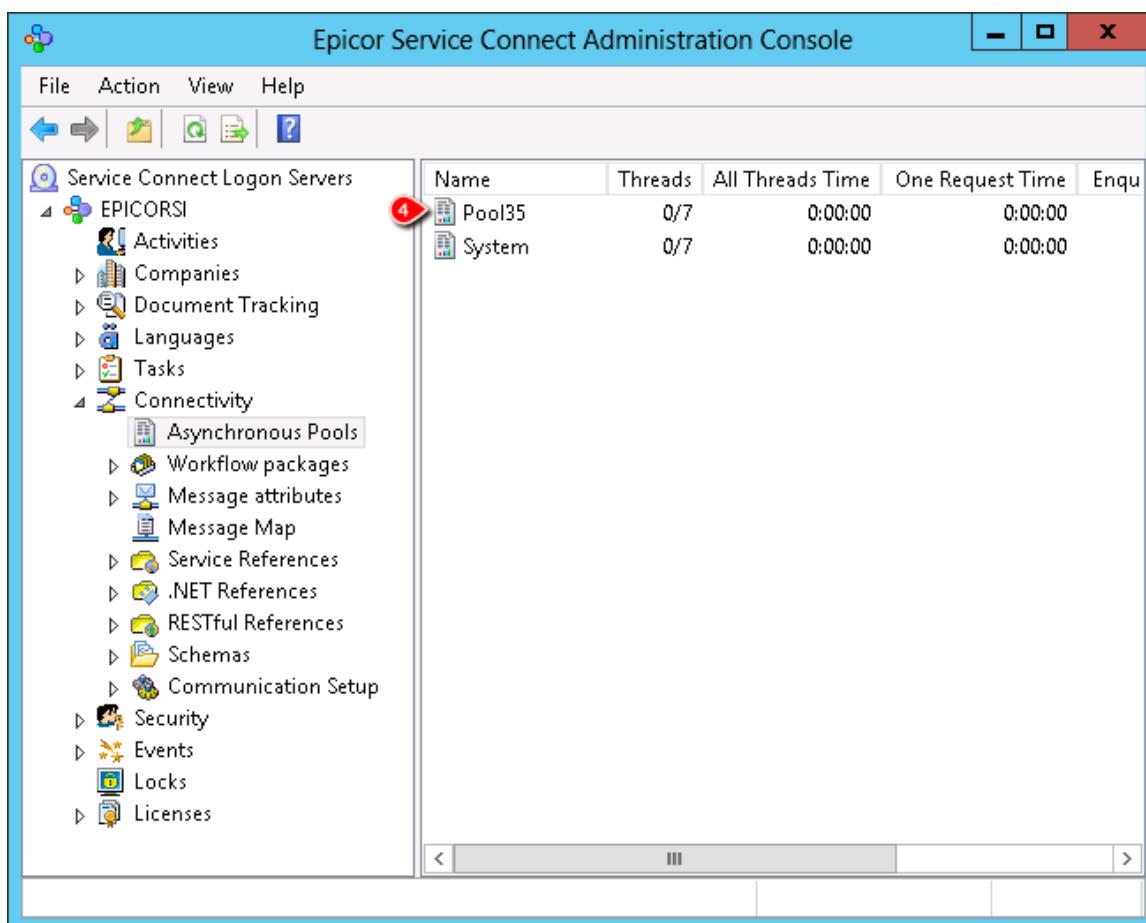
Resume a paused Asynchronous Pool

1. In the **tree view**, expand the **Connectivity > Asynchronous Pools** node.



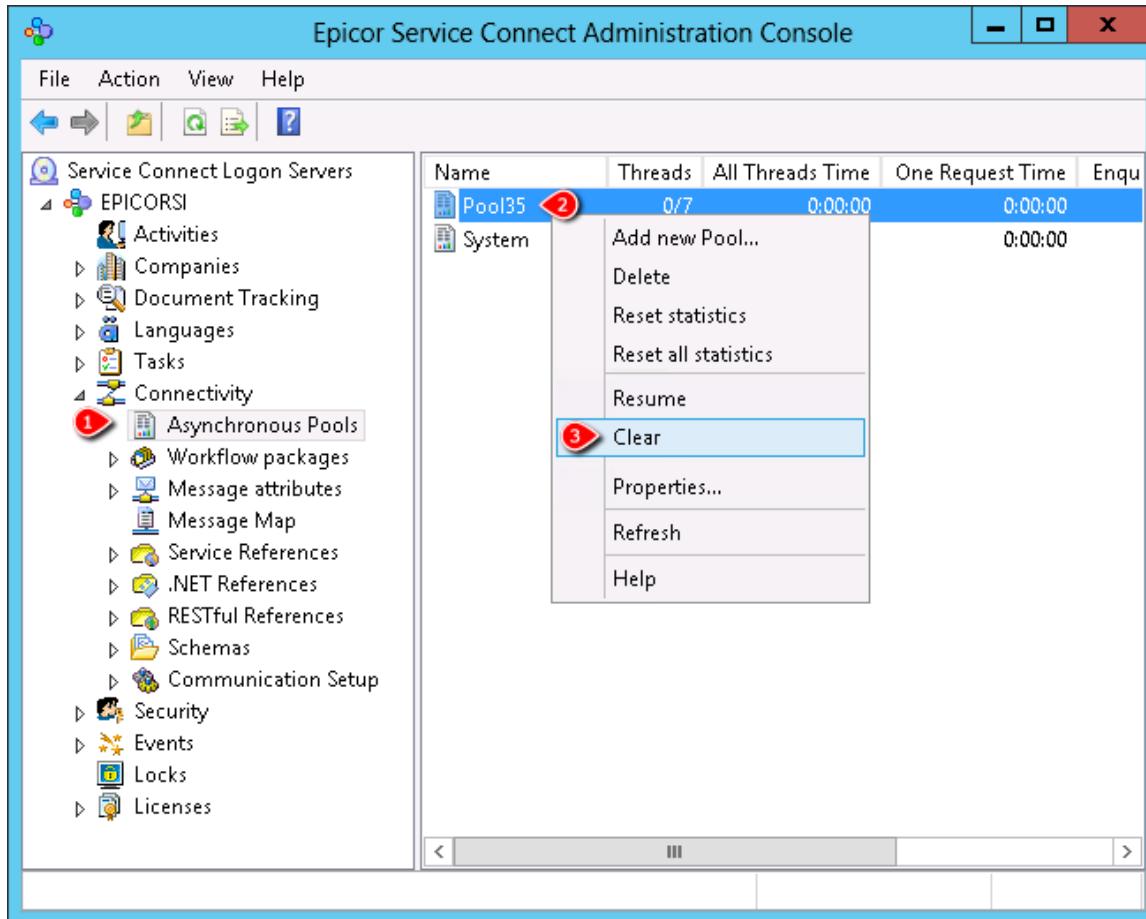
2. In the right pane, from the list of all the Asynchronous Pools registered in the system, right-click the paused Asynchronous Pool.
3. Select **Resume**.

4. Notice, the Pool now has a standard icon and the **State** column for the Pool displays **Active**.



Clear an Asynchronous Pool

1. In the **tree view**, expand the **Connectivity > Asynchronous Pools** node.

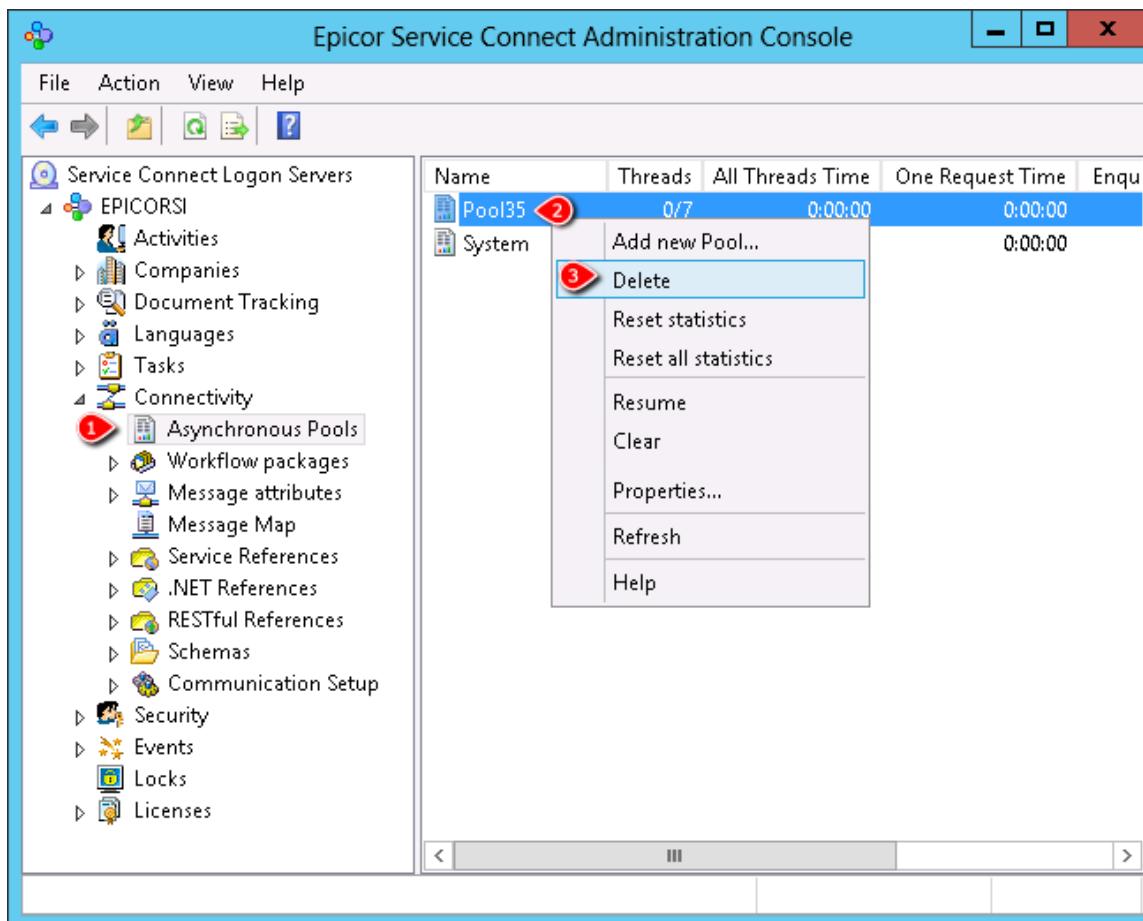


2. In the right pane, from the list of all the Asynchronous Pools registered in the system, right-click the Asynchronous Pool you want to clear.
3. Select **Clear**.
4. To the confirmation message, click **OK**.

All the messages are removed from this Pool. But the Pool still exists. To move all the messages from the Pool to the system asynchronous pool, and delete the pool, use the Delete operation.

Remove an Asynchronous Pool

1. In the **tree view**, expand the **Connectivity > Asynchronous Pools** node.



2. In the right pane, from the list of all the Asynchronous Pools registered in the system, right-click the Asynchronous Pool you want to delete.
3. Select **Delete**.
4. To the confirmation message, click **OK**.

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 in the ESC Administration Console:

- **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 (listener or speaker) 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.
- **State** – This field displays if the channel is Enabled, Disabled, or Failed. A channel has the Failed state in case of an error that happened during input or output channel work. To sort the channels by their state, click the column header.

The screenshot shows the Epicor Service Connect Administration Console window. The left pane is a tree view of the configuration structure, starting with 'Service Connect Logon Servers' and branching into 'EPICORSI' (Activities, Companies, Document Tracking, Languages, Tasks, Connectivity, Communication Setup), 'EPICORSI' (Asynchronous Pools, Workflow packages, Message attributes, Message Map, Service References, .NET References, RESTful References, Schemas), and 'EPICORSI' (Channels, Input Channels, Output Channels, Conversions). The right pane displays a table of channels with the following columns: Name, Thread, Busy, and Processed messages. The table lists 22 channels, all currently inactive (Busy = 0). The channels are: APAE-Mail, AQM-Errors, AQMCommodityCodeExport, AQMCountryExport, AQMCustomerContactExport, AQMCustomerExport, AQMCustomerGroupExport, AQMDMRUpdate, AQMDMRUpdateError, AQMERP_CUSTOMEROutput, AQMERP_CUSTOMEROutput_Save, AQMERP_CUSTOMER_CONTACTOutput, AQMERP_EMPLOYEEOutput, AQMERP_EQUIPMENTOutput, AQMERP_NONCONFORMANCE, AQMERP_PRODUCTOutput, AQMERP_PRODUCT_SUPPLIEROutput, AQMERP_PRODUCT_SUPPLIEROutput_test, and AQMERP_PRODUCT_UnitOutput.

Name	Thread	Busy	Processed messages
APAE-Mail	1	0	
AQM-Errors	1	0	
AQMCommodityCodeExport	1	0	
AQMCountryExport	1	0	
AQMCustomerContactExport	1	0	
AQMCustomerExport	1	0	
AQMCustomerGroupExport	1	0	
AQMDMRUpdate	1	0	
AQMDMRUpdateError	1	0	
AQMERP_CUSTOMEROutput	1	0	
AQMERP_CUSTOMEROutput_Save	1	0	
AQMERP_CUSTOMER_CONTACTOutput	1	0	
AQMERP_EMPLOYEEOutput	1	0	
AQMERP_EQUIPMENTOutput	1	0	
AQMERP_NONCONFORMANCE	1	0	
AQMERP_PRODUCTOutput	1	0	
AQMERP_PRODUCT_SUPPLIEROutput	1	0	
AQMERP_PRODUCT_SUPPLIEROutput_test	1	0	
AQMERP_PRODUCT_UnitOutput	1	0	

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, SFTP, FTP (.NET), IBMMQ, IBMMQ (.NET), POP3, IMAP, Schedule, MSMQ, Windows Service Bus, and SonicMQ
- **Speaker types** – Include COM, File, FTP, SFTP, FTP (.NET), HTTP, IBMMQ, IBMMQ (.NET), SMTP, MSMQ, Windows Service Bus, and SonicMQ

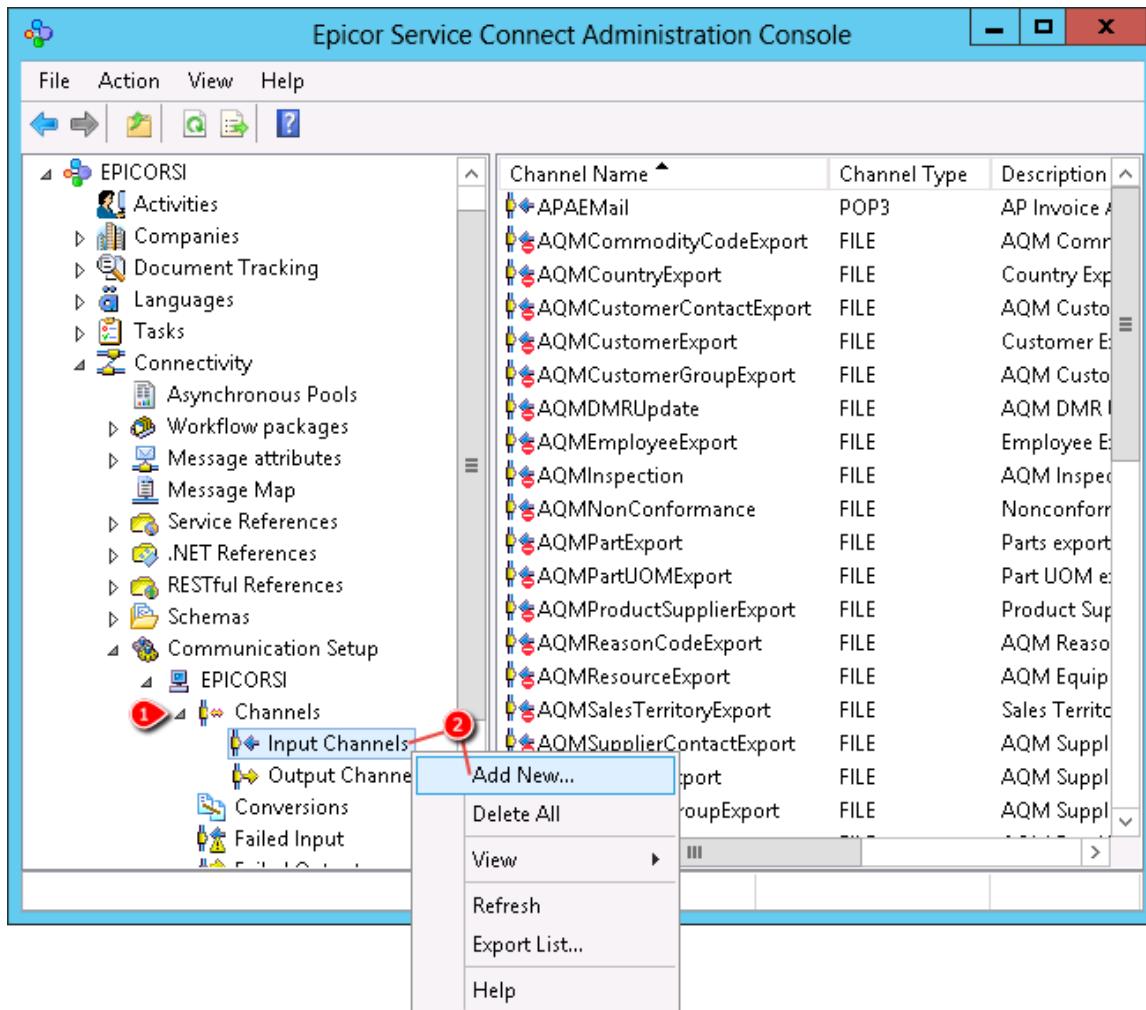
If the data inside a workflow should be posted to a specified destination, the message can be routed to a Poster activity. In the Poster properties sheet, one or more output channels can be specified.

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.

Set Up an Input Channel

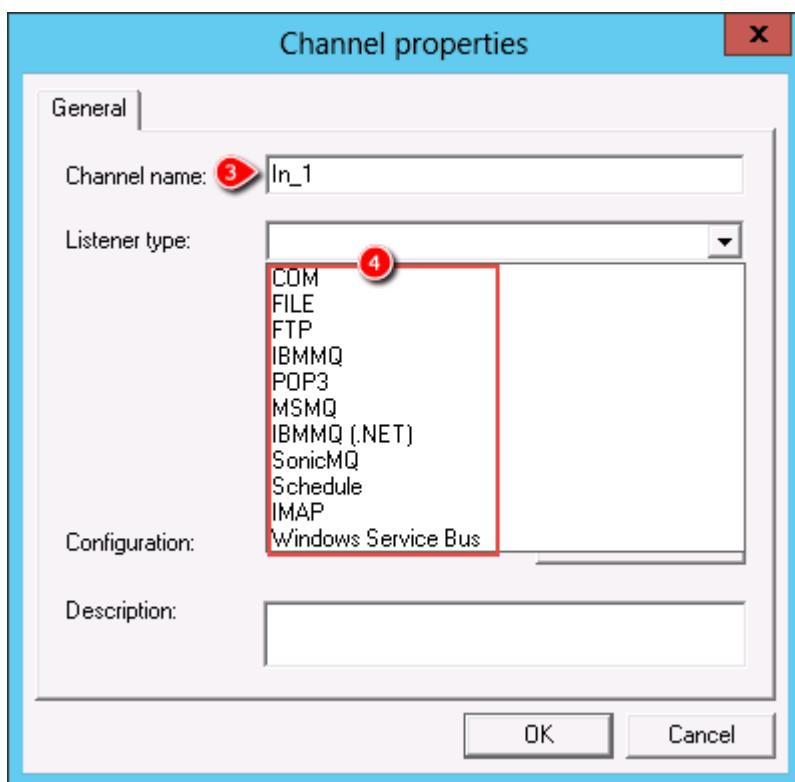
To set up an input channel:

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



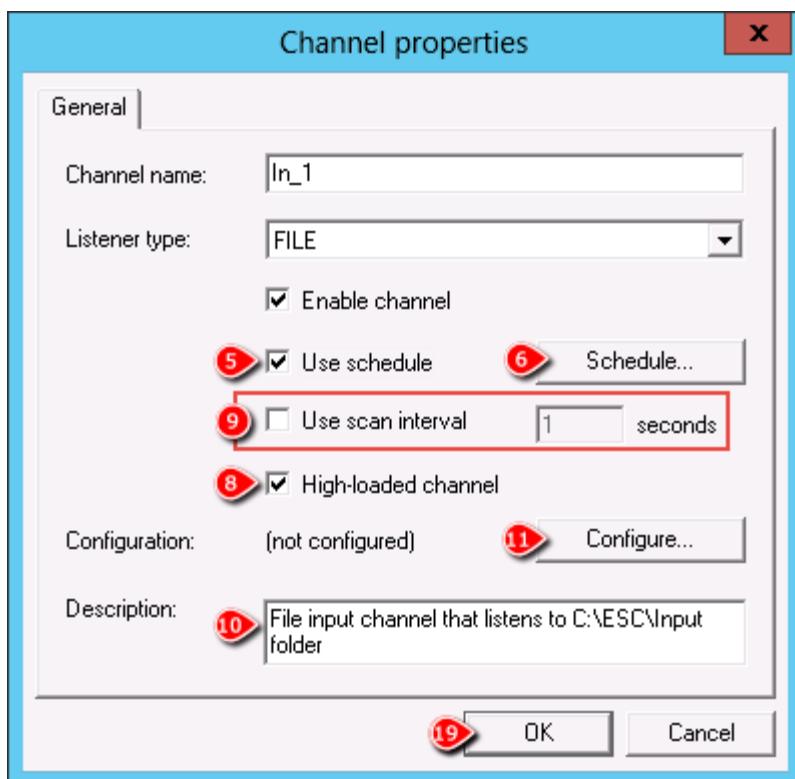
2. Right-click **Input Channels** and select **Add New**.

3. In the **Channel properties** window, enter a **Channel name**.

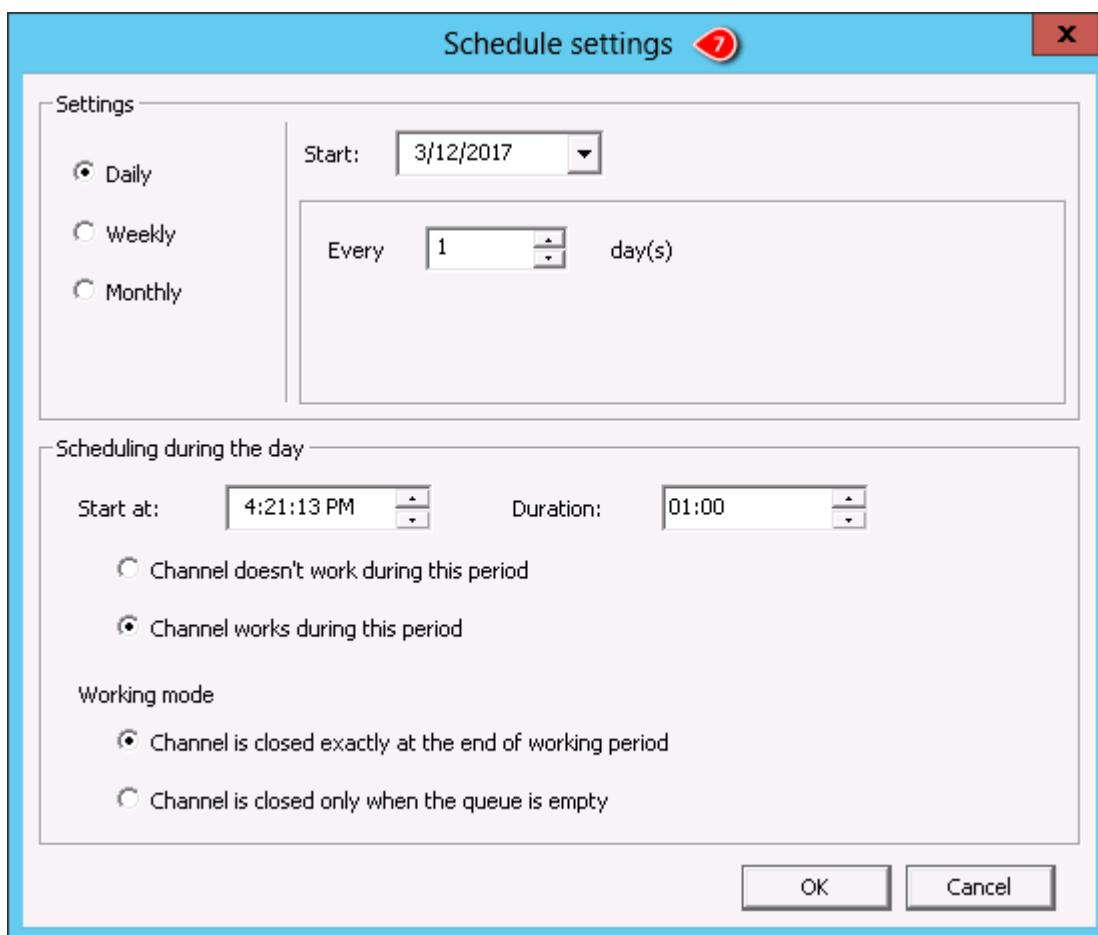


4. Select a **Listener type**.

5. Select **Use schedule**.



6. If you select Use schedule, click **Schedule** to set up a schedule.
7. On the **Schedule settings** screen, define how often the listener is checked for documents, and click **OK**. For more details on how to set channel schedule refer to the Set Up Channel Schedule section.



8. 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). In this case, Epicor recommends you adjust the settings in Properties of **Connectivity > Communication Setup > <Server Name> > Channels** so there are more threads available for the channels or decrease the number of channels marked as high-loaded.

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 settings; 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 message to be processed (for synchronous channels), the other channel waits for the first one to finish the work. 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 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.

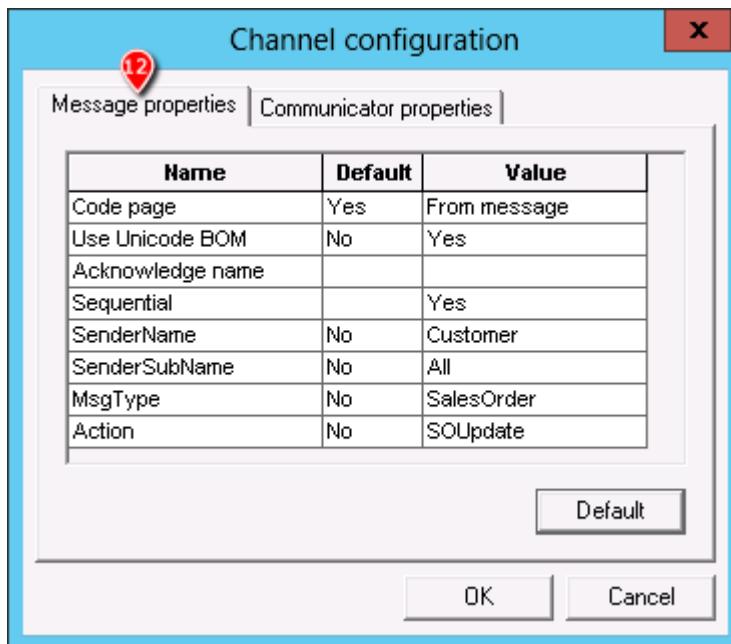
9. To check the listener for documents periodically, select the **Use scan interval** check box and specify the time interval in seconds.

10. Enter a **Description**.

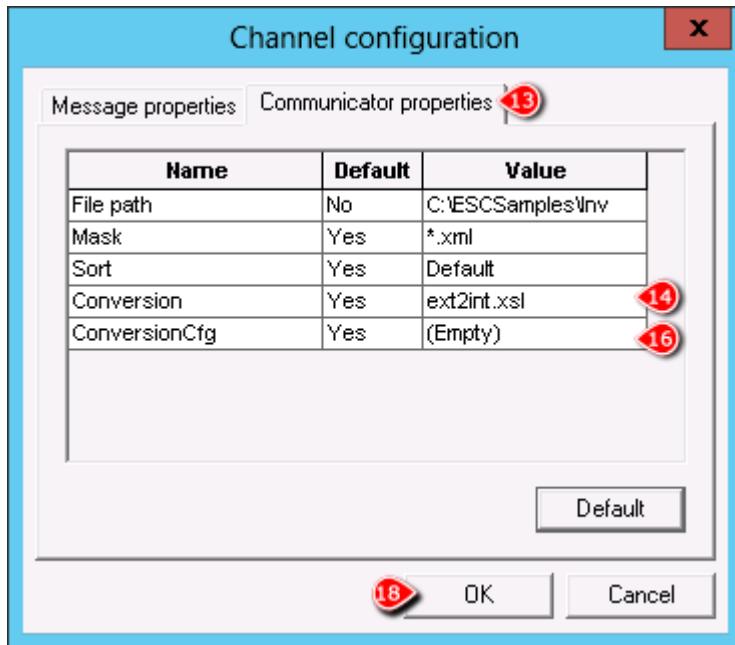
11. Click **Configure**.

12. In the **Channel configuration** window, open 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.

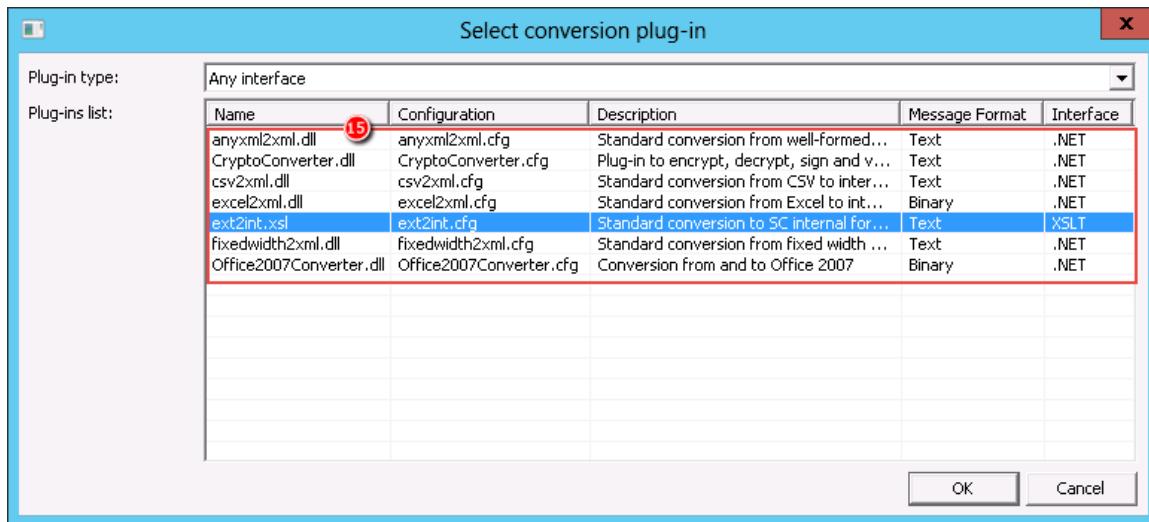


13. Click the **Communicator properties** tab.



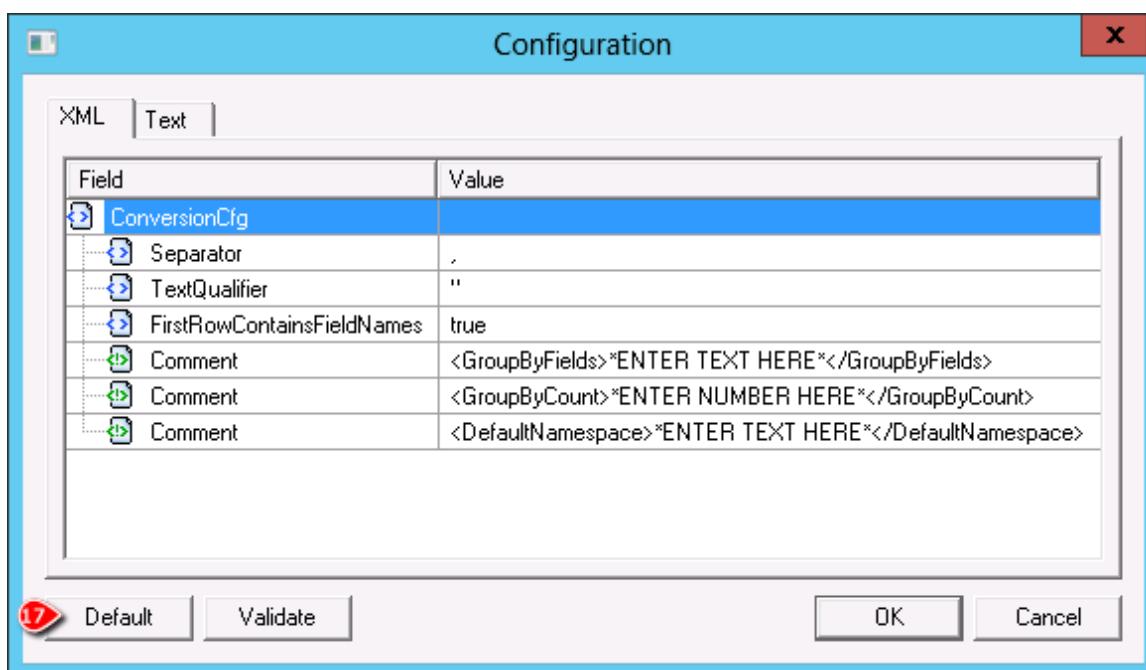
The properties listed first are based on listener type.

14. 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.
15. When you click this field, the **Select Conversion Plug-in** dialog with the list of registered conversions displays. For each conversion, its configuration file or files name, description, message format and interface type display. Select the conversion and click **OK**.



16. In the **Channel configuration** window, use the **ConversionCfg** property to enter a custom plug-in configuration. Click the down arrow button next to the ConversionCfg field. The **Configuration** window is displayed.
- If you selected a standard .NET conversion, in the **Configuration** dialog, 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.

17. You may also click the **Default** button to add the default configuration automatically for some conversions. This button is available if you selected **anyxml2xml**, **CryptoConverter**, **csv2xml**, **excel2xml**, or **fixedwidth2xml** conversion in the **Channel configuration** window.



When you click the Default button, the converter provides configuration sample with commented nodes. To be able to use these nodes, open the Text tab and remove !-- and -- from the lines with the fields that you want to use.

18. On the **Channel configuration** screen, click **OK**.

19. On the **Channel properties** screen, click **OK**.

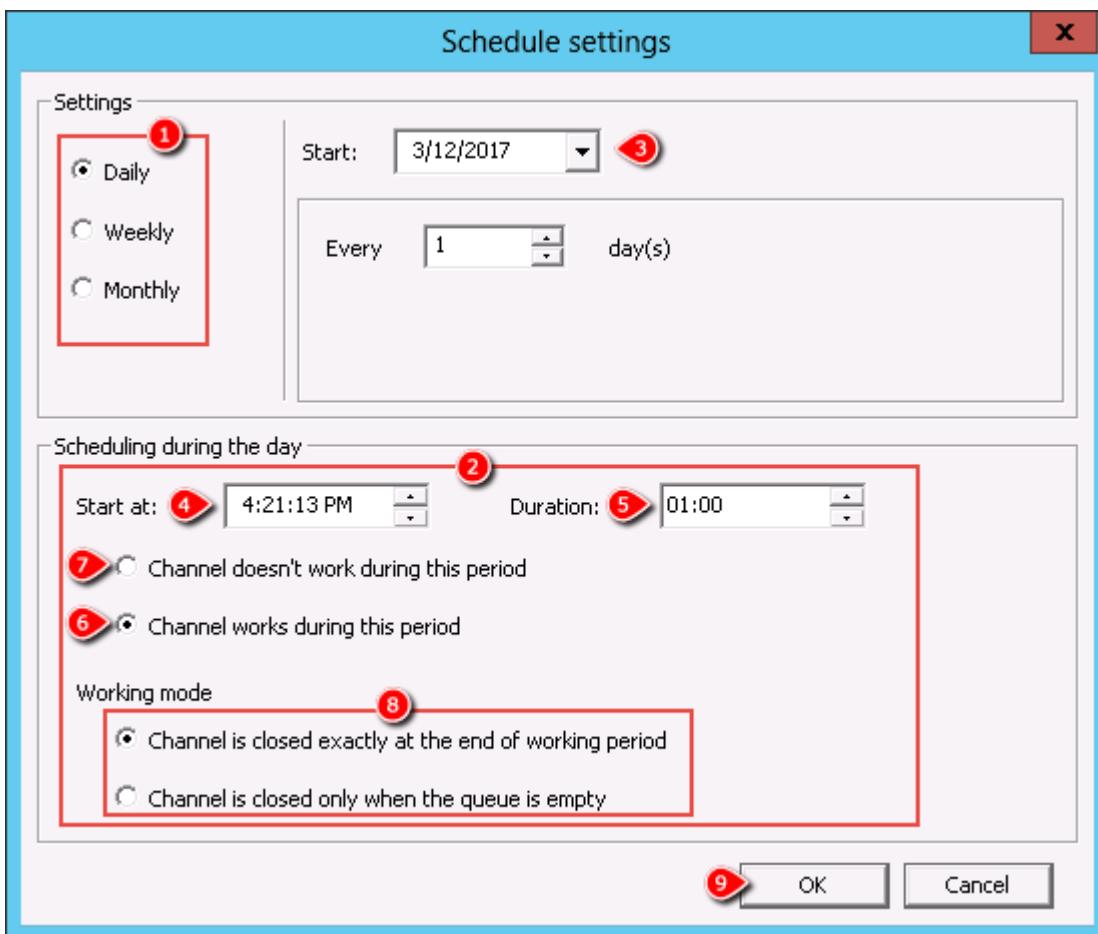
Set Up Channel Schedule

In the **Channel properties** window, if you select the **Use schedule** check box and click the **Schedule** button, the **Schedule settings** dialog is displayed. Use it to specify the time periods when the channel is active.

The following dialog is displayed for channels with type other than **Schedule** selected in the **Listener type** field in the **Channel properties** window.

To set up channel schedule:

- In the **Schedule settings** dialog, in the **Settings** pane, select one of the following options.



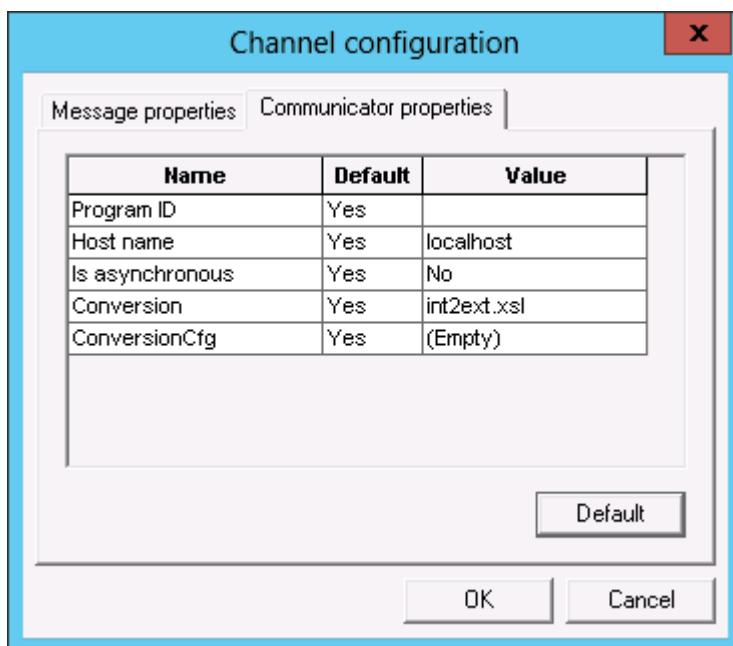
Option name	Value
Daily	The channel checks for the new messages with the daily interval you select in the Every _ day(s) field.
Weekly	The channel checks for the new messages on the days of week you specify with a weekly interval you select in the Every _ week(s) field.
Monthly	The channel checks for the new messages on the specified day with a monthly interval you select in the Every _ month field.

- The channel is activated at time intervals you specify in the **Scheduling during the day** pane.
- In the **Start** field, set the date when the channel should start to receive messages.
- In the **Scheduling during the day** group, in the **Start at** field, specify when the channel working period starts.
- In the **Duration** field, specify how long the channel working period is.
Do not set the time interval that goes past midnight. If you set such a time interval, warning displays.
- Select the **Channel works during this period** option if you want to enable the channel inside the specified period and pause it all the rest of the time outside of the period. This is the default option.

7. Alternatively, select the **Channel doesn't work during this period** option if you want to enable the channel all the time outside of the specified period, and pause the channel inside of the period.
8. In the **Working mode** group, select if the channel should be closed exactly when the specified period expires, or only if the queue is empty.
9. In the **Schedule settings** dialog, click **OK**.

Configure COM Channel Options

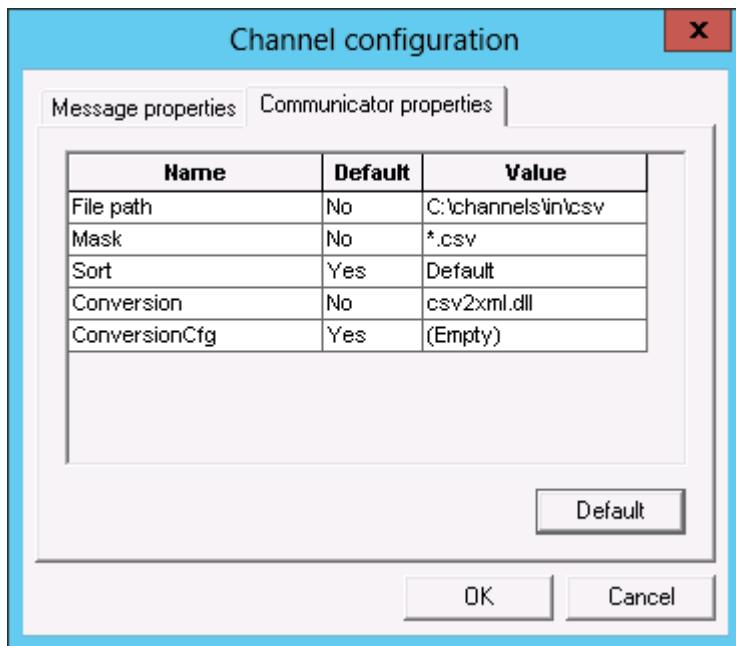
A COM communication channel can be both a listener and a speaker. As a listener, it is used exclusively to receive messages from a SharePoint workflow when Windows SharePoint Services is integrated with Service Connect. As a speaker, it is used to send messages to reusable Windows-based software components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



Property	Description
Program ID	The ID of the program sending the message. For example, ScaMessengerSrv.Gate
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 Option 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	<p>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.</p> <p>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.</p> <p>To use an environment variable, insert the variable name, surrounded by percent signs; for example, %sc_input_folder%.</p> <p>If you use a folder on a remote server, the account used to run ScaMessengerSrv service must have read-write access to that folder. You can adjust the account for ScaMessengerSrv service, use Windows' Administrative Tools->Services snap-in.</p>
Mask	<p>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.</p> <p>For output channels, this indicates the format and name for 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.</p> <p>The following macros are available:</p> <ul style="list-style-type: none"> • %GUID% • %DATE(FORMAT)%

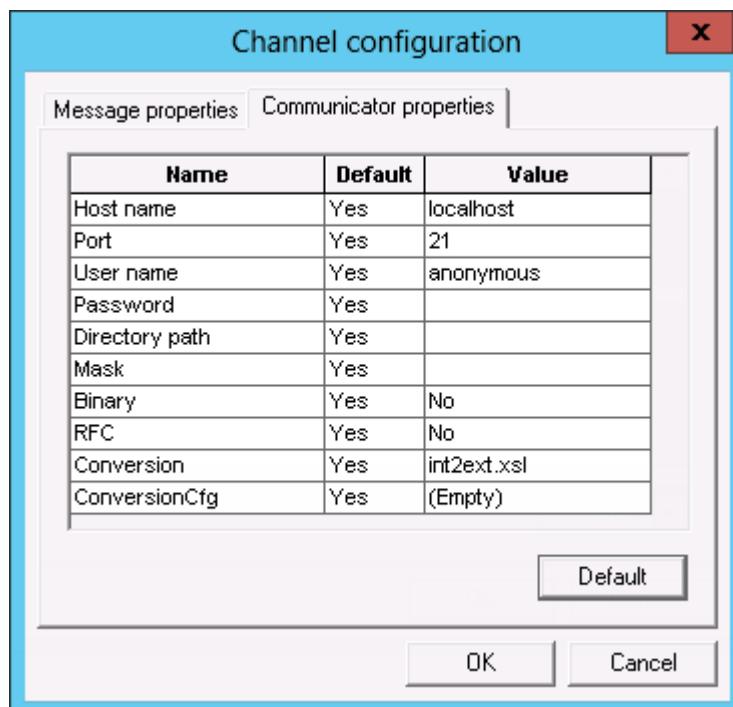
Property	Description
	<ul style="list-style-type: none"> • %TIME% • %DATETIME(FORMAT)% • %COUNTER% <p>If you set this field to *.csv, the filename will be a globally unique identifier (GUID).</p>
Sort	<p>Files from the folder specified in the Input channel configuration can be processed sequentially in accordance with the following file parameters:</p> <ul style="list-style-type: none"> • Name (Asc.) • Name (Desc.) • Creation date (Asc.) • Creation date (Desc.) • Modification date (Asc.) • Modification date (Desc.) <p>This option is only available for Input Channels and only takes effect if Sequential property on the Message Properties tab is set to Yes.</p>
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.

If you are going to use an FTP input channel that monitors an FTP folder, you must grant the FTP user the permission to modify and delete files in this folder.

Service Connect only supports passive FTP connection mode.



Property	Description
Host name	The name of the FTP server.
Port	The port number to use with the FTP host specified in the Host name field. The specified port number is validated on the FTP host.
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.

Property	Description
Binary	<p>The option used to transfer the files, either binary or ASCII. If the property is set to No, the data is transferred in ASCII.</p> <p>This option is related to the Binary conversion property of the Conversion defined for this channel. You can only select a Binary conversion for a channel if the channel has the Binary option set to Yes. If a Binary conversion is selected for a channel, you cannot set the Binary option to No.</p>
RFC	<p>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.</p> <p>RFC option should be set to Yes if file share is created on the AS400 system.</p>
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 SFTP Channel Options

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

If you are going to use an SFTP input channel that monitors an SFTP folder, you must grant the SFTP user the permission to modify and delete files in this folder.

Service Connect only supports passive FTP connection mode.

Only password-based authentication is supported.



Property	Description
Host name	The name of the SFTP server.
Port	The port number to use with the SFTP host specified in the Host name field. The specified port number is validated on the SFTP host.
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 SFTP 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.

Property	Description
Binary	<p>The option used to transfer the files, either binary or ASCII. If the property is set to No, the data is transferred in ASCII.</p> <p>The default value for the SFTP input channel Binary option is Yes.</p> <p>This option is related to the Binary conversion property of the Conversion defined for this channel. You can only select a Binary conversion for a channel if the channel has the Binary option set to Yes. If a Binary conversion is selected for a channel, you cannot set the Binary option to No.</p>
Compression	The default value for the SFTP input channel Compression is No.
Proxy host	<p>The following values are available:</p> <ul style="list-style-type: none"> • None • Socks4 • Socks4a • Socks5 • HttpConnect
Conversion	<p>The conversion plug-in used when consuming or posting the file.</p> <p>When you click this field, the Select Conversion Plug-in dialog with the list of registered conversions displays.</p>
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.

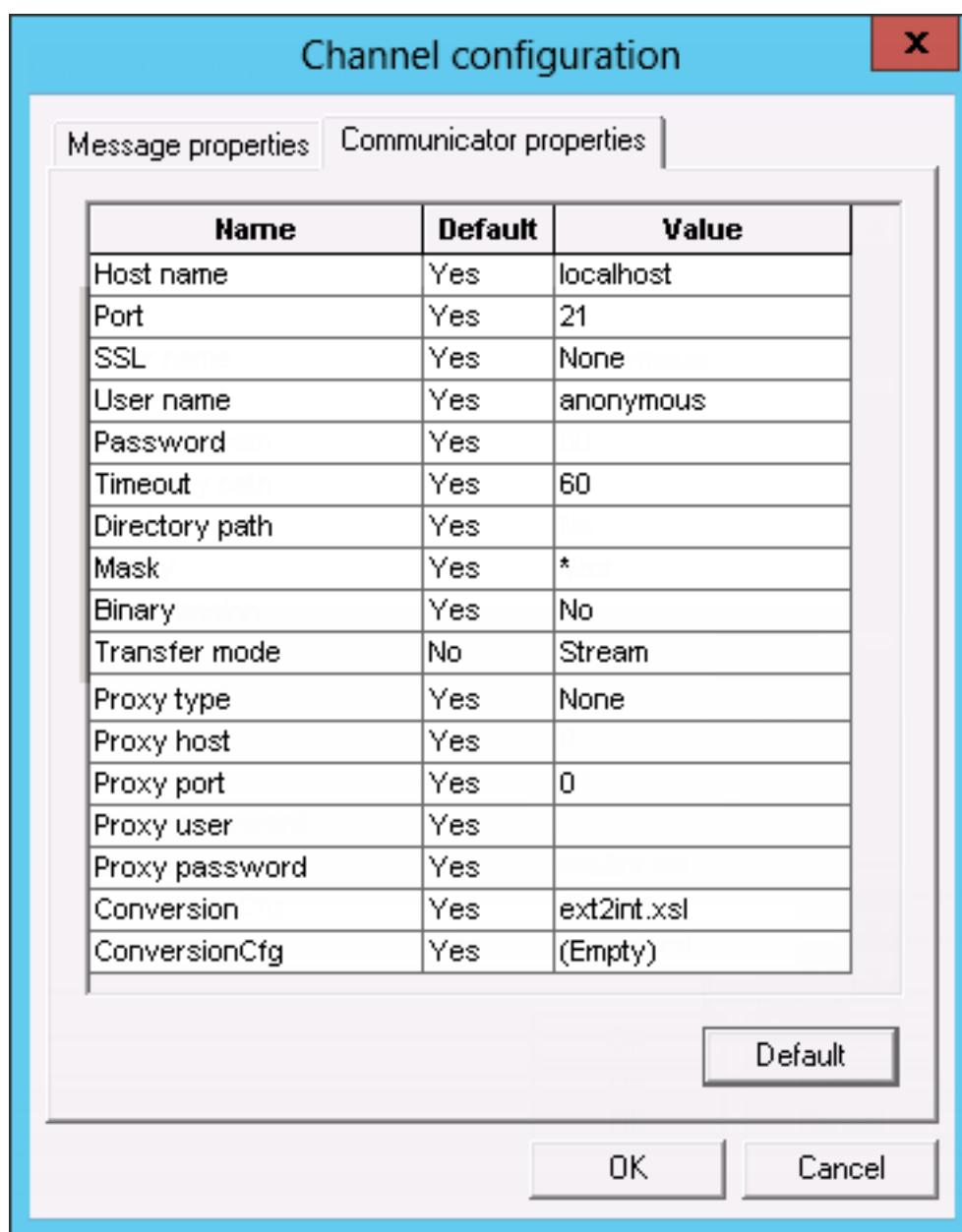
Configure FTP (.NET) Channel Options

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.

If you are going to use an FTP input channel that monitors an FTP folder, you must grant the FTP user the permission to modify and delete files in this folder.

Service Connect only supports passive FTP connection mode.

Only password-based authentication is supported.



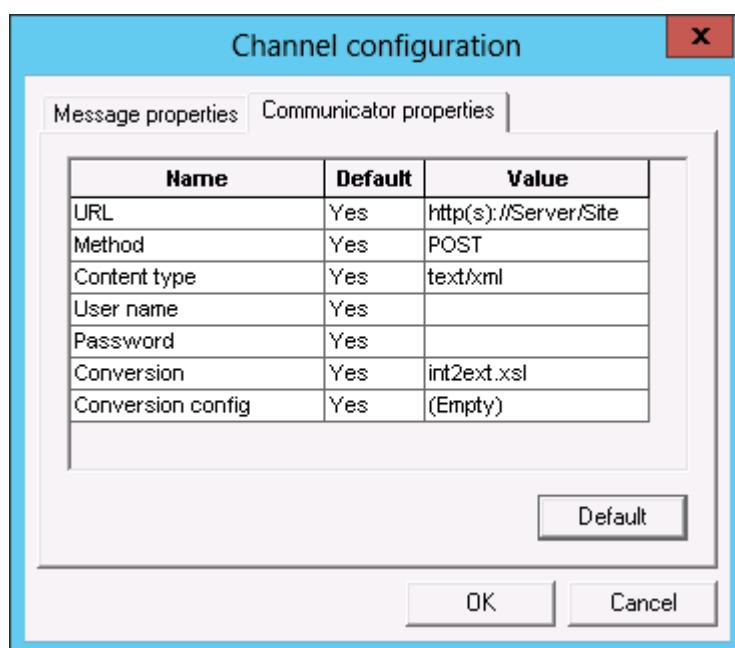
Property	Description
Host name	The name of the FTP server.
Port	The port number to use with the FTP host specified in the Host name field. The specified port number is validated on the FTP host.
SSL	The property defines if the secure socket layer (SSL) is enabled and which mode of SSL connection to the server is used. The available options are: <ul style="list-style-type: none">• Explicit

Property	Description
	<ul style="list-style-type: none"> • Implicit • None <p>The default value for the FTP (.NET) channel SSL option is None.</p>
User name	The user account name that will log onto the server.
Password	The user account password.
Timeout	<p>The default value for the FTP (.NET) input channel Timeout is 60 seconds. 0 indicates that the request does not time out.</p>
Directory path	The path to the folder on the FTP server where the file will be posted; for example, pub/sc/inchannel.
Mask	<p>The mask that determines the file types that can be transferred via the channel. Click the arrow in the Mask field to activate the File name template editor window. The template can be a combination of exact file name characters, macro definitions and wildcards For example, to allow only CSV files, set this field to *.csv.</p>
Binary	<p>The option used to define the file transfer mode, either binary or ASCII. If the property is set to No, the data is transferred in ASCII. The default value for the (.NET) channel Binary option is No. This option is related to the Binary conversion property of the Conversion defined for this channel. You can only select a Binary conversion for a channel if the channel has the Binary option set to Yes. If a Binary conversion is selected for a channel, you cannot set the Binary option to No.</p>
Transfer mode	<p>This property determines how client and server transfer data. The following values are available:</p> <ul style="list-style-type: none"> • Stream • Zlib • Block <p>The default value for the FTP .NET channel Transfer mode is Stream.</p>
Proxy type	<p>The following values are available:</p> <ul style="list-style-type: none"> • None • Socks4 • Socks4a • Socks5 • HttpConnect • FtpSite • FtpUser • FtpOpen • FtpDoubleLogin

Property	Description
Conversion	The conversion plug-in used when consuming or posting the file. When you click this field, the Select Conversion Plug-in dialog with the list of registered conversions displays.
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	<p>The URL of the ASP page to which the file should post.</p> <p>The following is a sample ASP page.</p> <pre><%@ 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);</pre>

Property	Description
	<pre>Response.OutputStream.Write(content, 0, content.Length) ; } </script></pre>
Method	http verb PUT or POST
Content type	The ContentType property contains the media type of the request. It gets or sets the value of the Content-type HTTP header.
Conversion	The conversion plug-in used when posting a file
Conversion config	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 Schedule Channel Options

Schedule is a special type of input channel you use to schedule workflow execution or any other type of plug-ins including output channels, Web references, and .NET references. A Schedule communication channel can be only a listener.

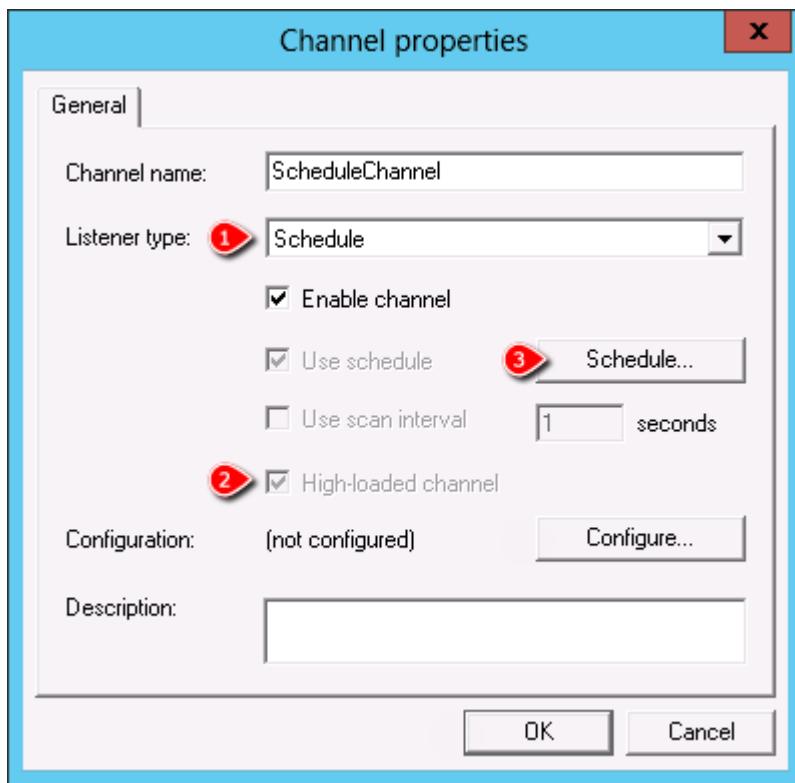
Schedule channels are always created as non-sequential and high-loaded to guarantee the most accurate schedule execution.

You can create multiple instances of Schedule channels.

Schedule channels always use processing schedule. By default, it is configured so that the required plug-in is executed daily, once a day, at the time of channel creation.

To set up a Schedule channel:

1. In the **Channel properties** window, in the **Listener type** field, select **Schedule**.

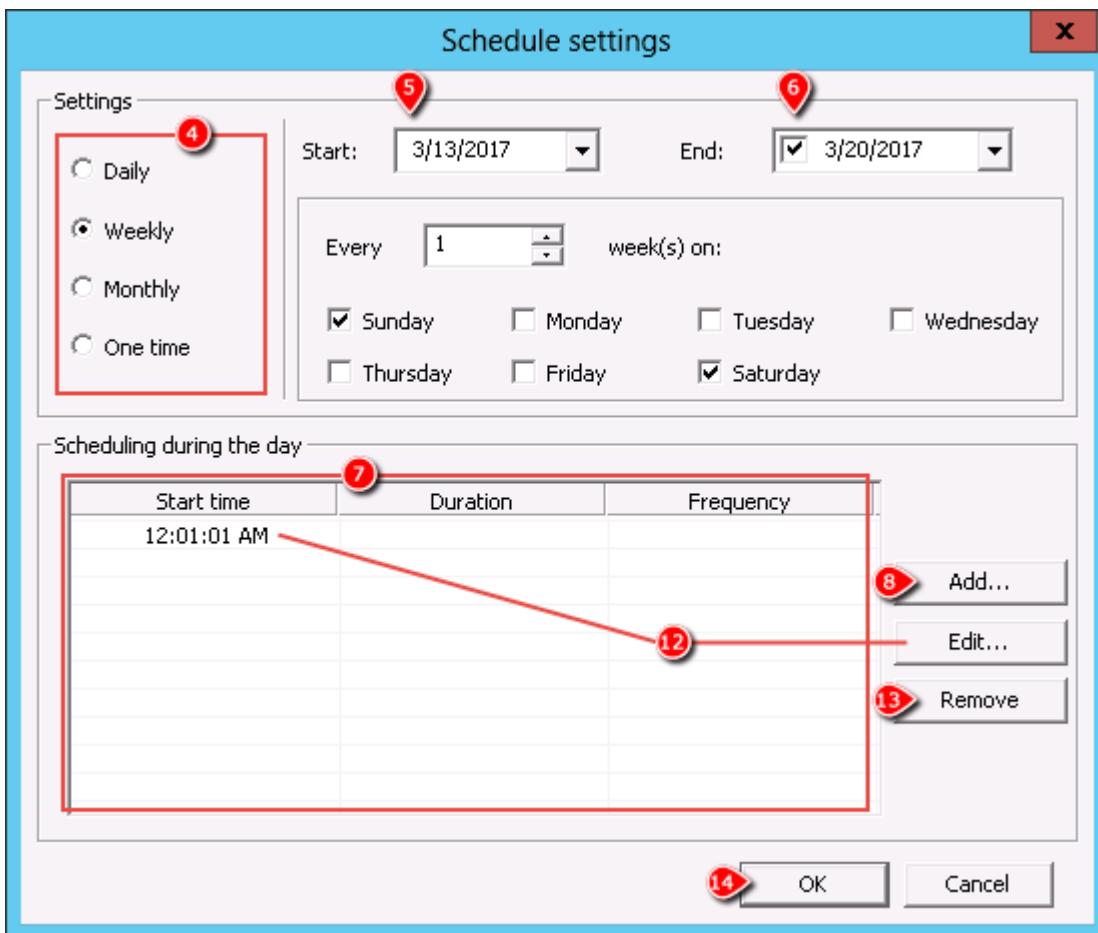


2. Notice, the **High-loaded channel** check box is selected and grayed.

Schedule channel is a high-loaded channel by default and it always uses a dedicated thread.

3. Click the **Schedule** button.

4. In the **Schedule settings** dialog, in the **Settings** pane, select one of the following options.



Option name	Value
Daily	The task is executed with a specified daily interval.
Weekly	The task is executed on specified days of week with a specified weekly interval.
Monthly	The task is executed on the specified day of the specified month or months.
One time	The task is executed on the date specified in the Start field.

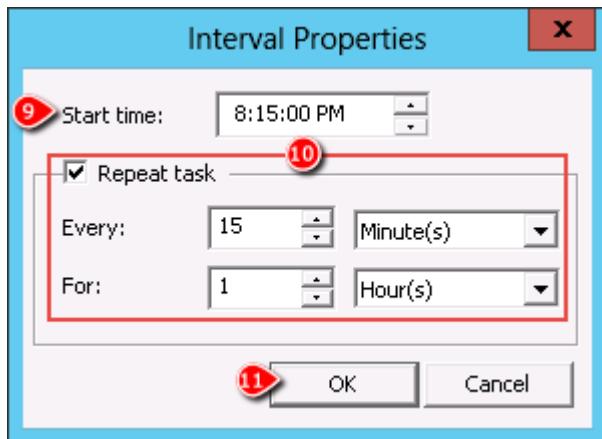
The channel is activated at time intervals you specify in the **Scheduling during the day** pane.

- In the **Start** field, set the date of the schedule activation.
- In the **End** field, select the check box and set the date of the schedule de-activation.
This property is optional.
- In the **Scheduling during the day** grid, define the channel activity interval or intervals during a day.

The intervals cannot intersect one another. An interval cannot cross a day.

- To define a new interval, click the **Add** button.

9. In the **Interval Properties** dialog, define an interval **Start time**.



10. Optionally, select the **Repeat task** check box and define interval duration and frequency with which the channel generates a message during activity interval.

If you only define the **Start time**, the channel generates a message only once at the defined time.

11. In the **Interval Properties** window, click **OK**.

12. To correct an existing interval, select it in the grid and click **Edit**.

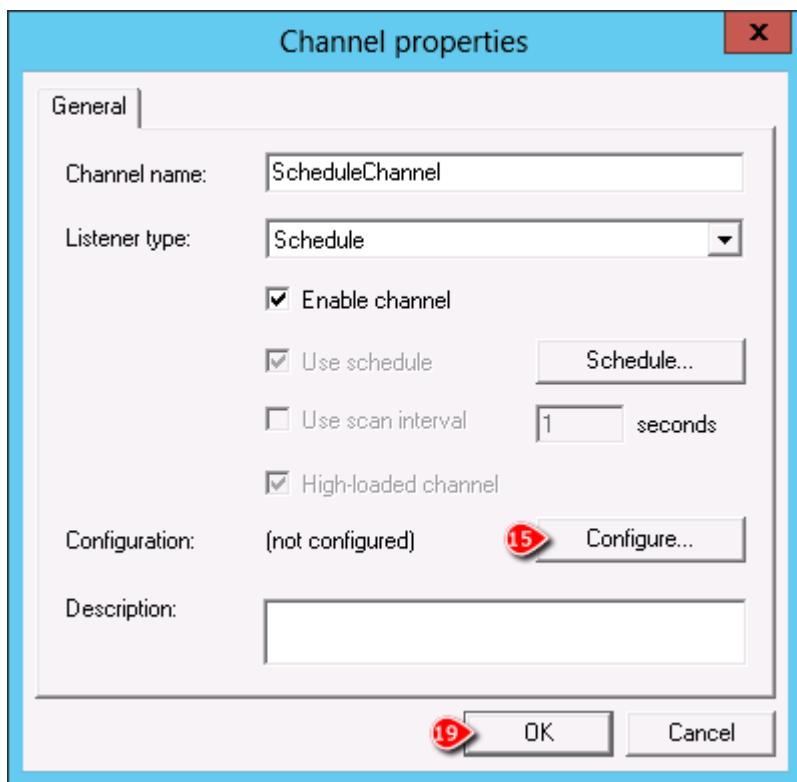
You can alternatively double-click the row.

In the **Interval Properties** window, perform the required updates and click **OK**.

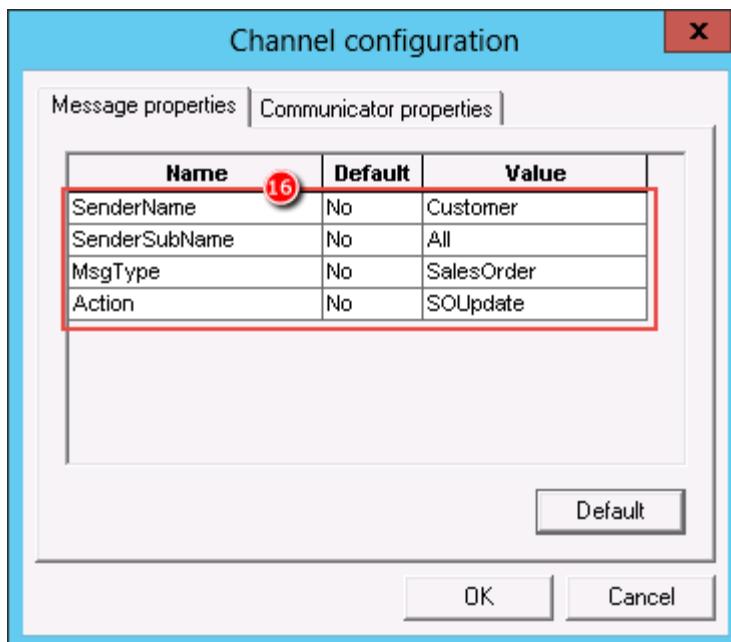
13. To remove an existing interval, select it in the grid and click **Remove**.

14. In the **Schedule settings** window, click **OK**.

15. In the **Channel properties** window, click **Configure**.

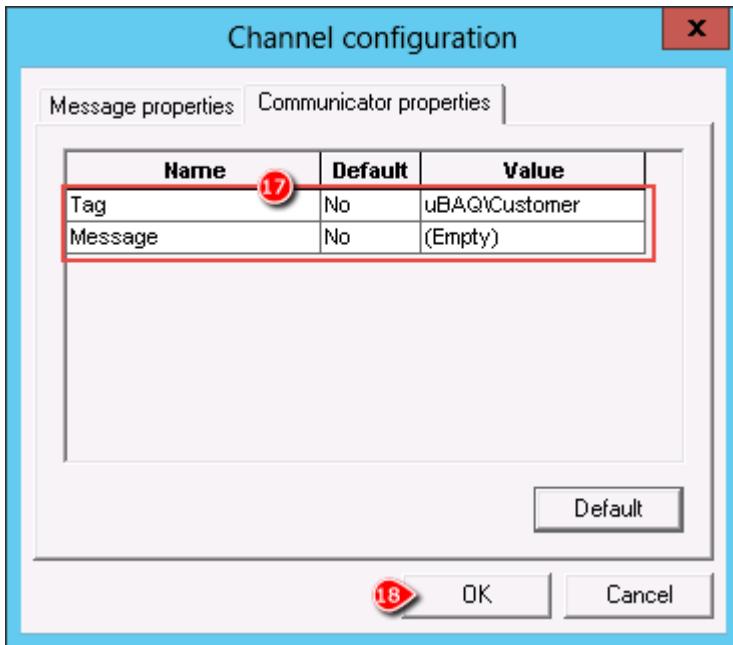


16. You can set up the following message properties to provide message mapping that defines which plug-in should be executed according to the specified schedule:



- SenderName
- SenderSubName
- MsgType
- Action

17. On the **Communicator properties** tab, the following communicator properties are available:



Property name	Description
Tag	Use it to explicitly define which plug-in should be executed according to the specified schedule, without using message mapping.
Message	Use it to specify the contents of the <dta> section of the message that should be passed to the plug-in and executed according to the specified schedule. If this property is not defined, an empty message is passed to the plug-in.

18. On the **Channel configuration** screen, click **OK**.

19. On the **Channel properties** screen, click **OK**.

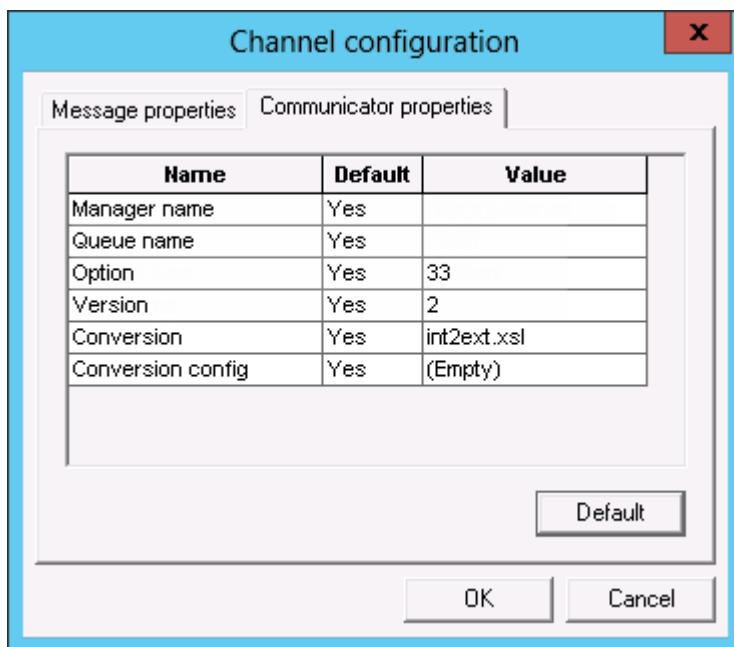
Configure IBM MQ and IBM MQ (.NET) Channel Options

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.

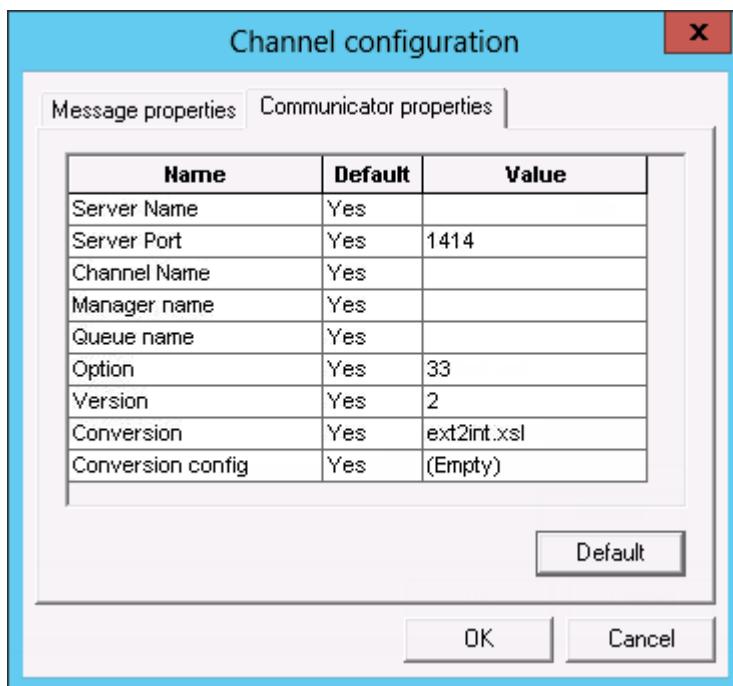
IBM MQ Channel Options

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.
Conversion config	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 Options



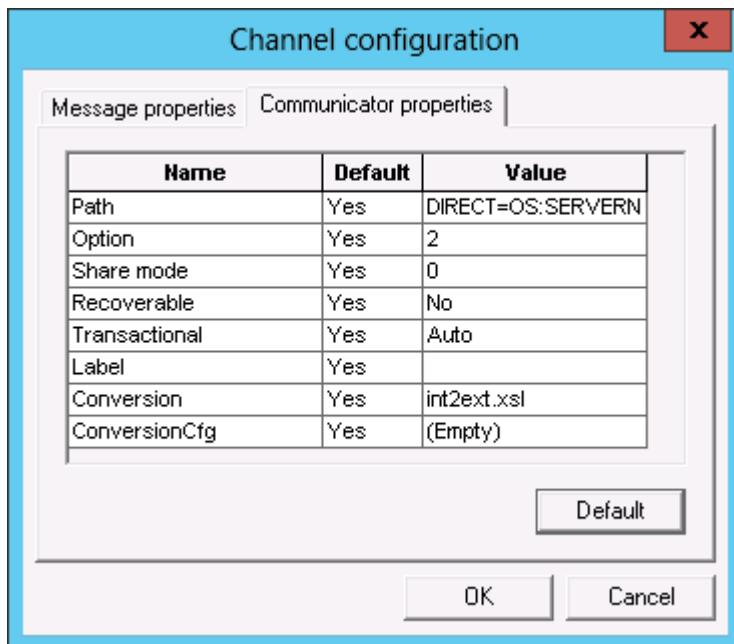
IBM MQ (.NET) channel has the following properties:

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 if 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.
Conversion config	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.

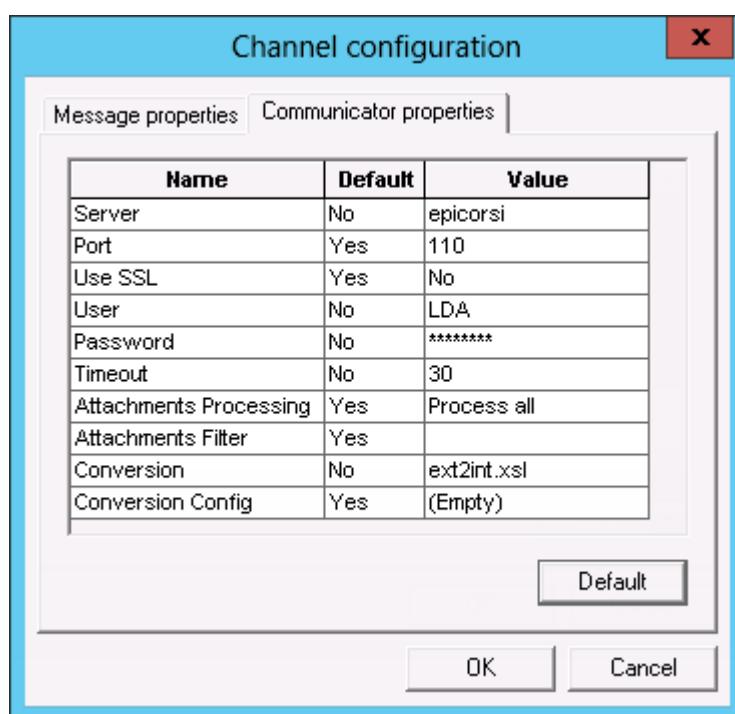


Property	Description
Path	The path to the message queue.
Option	<p>One of the following:</p> <ul style="list-style-type: none"> • 32 - You can view messages but cannot remove them from the queue. • 2 - You can send messages to the queue. • 1 - You can view and remove messages from the queue. <p>Epicor strongly recommends you keep the default value.</p>
Share mode	<p>One of the following:</p> <ul style="list-style-type: none"> • 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 are 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.

Property	Description
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.

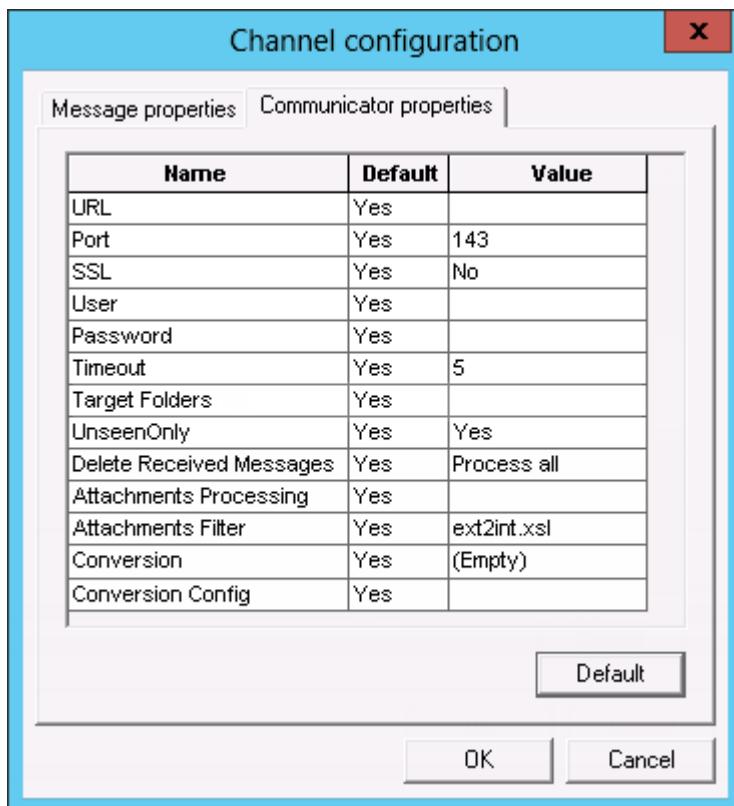


Property	Description
Server	The POP3 server name. If the server is within a trusted domain, you can enter just the server name. If the server is outside a trusted domain, enter the Fully Qualified Domain Name (FQDN). For example, if mycompany.pop3server.net is within a trusted domain, you can enter pop3server.
Port	The server port number.
Use SSL	Select the mode to use the Secure Sockets Layer (SSL) protocol to manage message transmission security on the Internet. The available options are: <ul style="list-style-type: none">• Explicit• Implicit

Property	Description
	<ul style="list-style-type: none"> • None <p>Explicit mode is also referred to as TLS.</p> <p>The Implicit setting is most commonly used for POP3 channels.</p>
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 the inactivity. The receipt of any command during the timeout intervals resets the timer.
Attachments Processing	<p>The mode to process attachments. There are three modes:</p> <ul style="list-style-type: none"> • Process all • Process by mask • Skip by mask <p>In Process all mode, all incoming email message attachments are treated as data and put into ESC through conversion layer. This is the default channel behavior, compatible with previous versions.</p> <p>If you select Process by mask or Skip by mask mode, specify Attachments Filter. Process by mask means that only attachments that match given mask or masks will be processed. Skip by mask means that attachments that match given mask or masks will not be processed.</p>
Attachments Filter	<p>Enabled only when Attachments Processing is set to Process by mask or Skip by mask mode. You can set the filter in format: *.pdf, *.doc, *.gif, *.*</p> <p>Use , and ; as delimiters.</p>
Conversion	The conversion plug-in used when consuming or posting a file.
Conversion Config	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 IMAP Channel Options

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

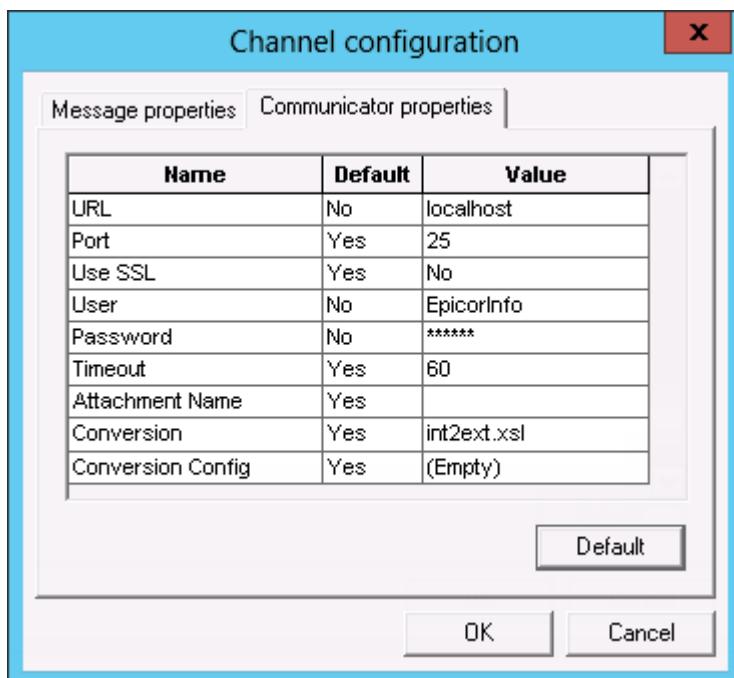


Property	Description
SSL	Select the mode to use the Secure Sockets Layer (SSL) protocol to manage message transmission security on the Internet. The available options are: <ul style="list-style-type: none"> • Explicit • Implicit • None Explicit mode is also referred to as TLS. The Implicit setting is most commonly used for IMAP channels.
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 DeleteReceivedMessages set to No with UnseenOnly set to Yes, or set DeleteReceivedMessages to Yes.

Configure SMTP Channel Options

An SMTP communication channel is a speaker that can send an e-mail message to 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 Workflow Designer chapter for an example of how to use a Poster activity to send an e-mail using an SMTP channel.



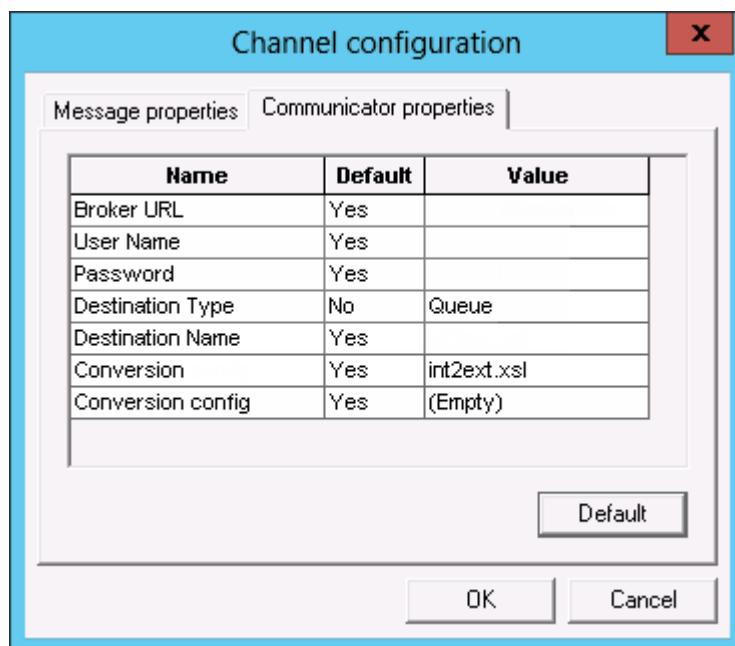
Property	Description
URL	The URL of the SMTP server. If the server is within a trusted domain, you can enter just the server name. If the server is outside a trusted domain, enter the Fully Qualified Domain Name (FQDN). For example, if smtpserver.mycompany.net is within a trusted domain, you can enter smtpserver.
Port	The server port number is 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 the mode to use the Secure Sockets Layer (SSL) protocol to manage message transmission security on the Internet. The available options are: <ul style="list-style-type: none"> • Explicit • Implicit • None Explicit mode is also referred to as TLS. The Explicit setting is most commonly used for SMTP channels.

Property	Description
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.
Attachment Name	Name of the attachment file. Click the field to activate the File name template editor. The template can be a combination of exact file name characters, macro definitions and wildcards. Default value is empty, which will result in GUID file name with extension, corresponding to the channel conversion.
Conversion	The conversion plug-in used when consuming or posting the file.
Conversion Config	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 connection include: <ul style="list-style-type: none"> • 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 sever. 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: <ul style="list-style-type: none"> • 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 if the destination on the Java Messaging Server.
Conversion	The conversion plug-in used when consuming or posting the file.
Conversion config	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 Windows Service Bus Channel Options

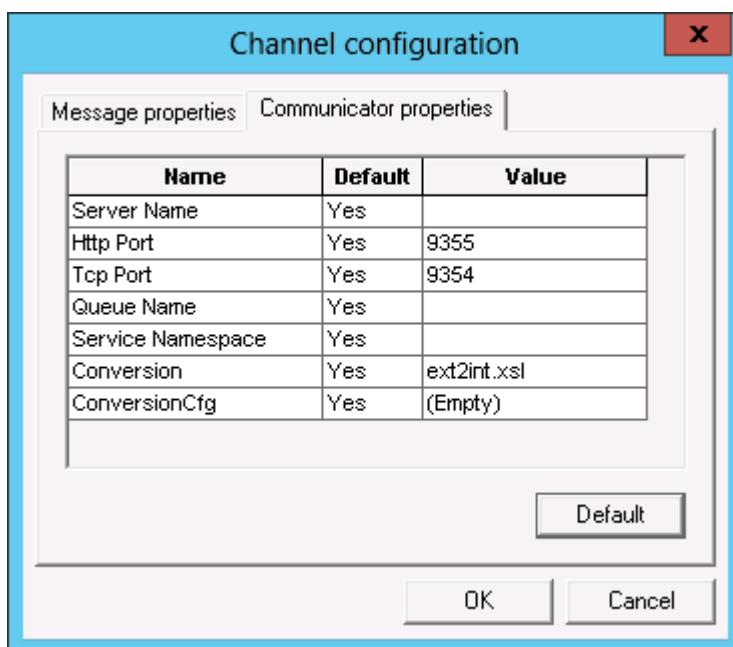
Use Windows Service Bus input and output communication channels to interact with Windows Service Bus.

To be able to use Windows Service Bus communication channels perform the following steps:

1. Install Windows Service Bus 1.0.
2. Locate the Microsoft.ServiceBus.dll library. The default location is C:\Program Files\Service Bus\1.0.
3. Copy Microsoft.ServiceBus.dll to the same location where ScaMessengerSrv.exe resides: <ESC installation folder>\System\Services\DES.

Windows Service Bus 1.0 library is required to work with Windows Service Bus 1.1.

To be able to use the Windows Service Bus channel, the account under which ScaMessengerSrv.exe Service Connect service is running must have the permission to access Service Bus.



Property	Description
Server Name	The name of the server that hosts the Service Bus for Windows Server
Http Port	HTTPS Port for the Service Bus
Tcp Port	Standard TCP port to use Service Bus
Queue Name	Service Bus queue
Service Namespace	A service namespace that provides a container for addressing Service Bus resources within Service Connect.
Conversion	The conversion plug-in used when consuming or posting a file. When you click this field, the Select Conversion Plug-in dialog with the list of registered conversions displays.
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.

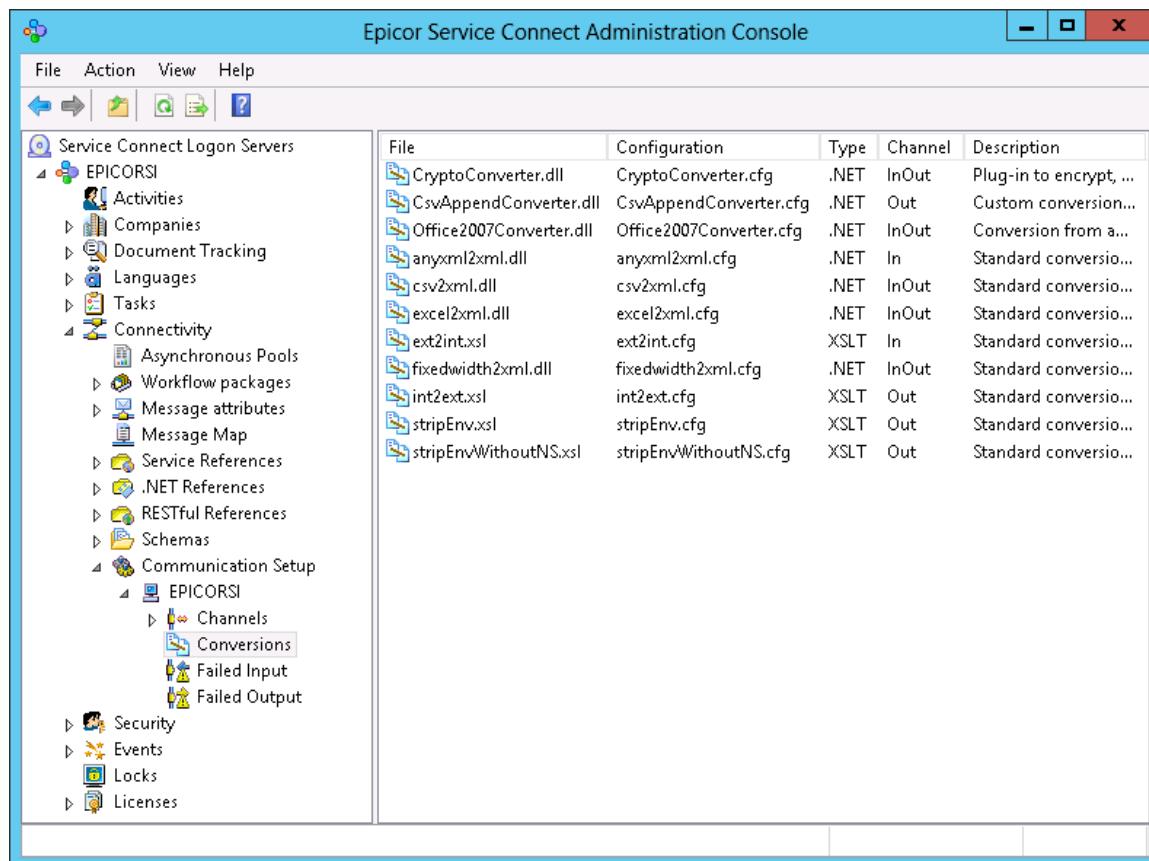
Conversions

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 .Net BO or 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 Workflow Designer chapter 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 necessarily 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 **Connectivity > Communication Setup > server name > Conversions** node 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.



Configuration files that were parsed with errors are displayed with error configuration icon



instead of the standard one.

The following information for each conversion displays in the columns in the right pane:

Column	Description
File	Conversion file name. For example, "ext2int.xls"
Configuration	Configuration file name for the conversion. One conversion can have more than one configuration. In this case, there are multiple entries of this conversion file name for each configuration file name. For example, if

Column	Description
	a conversion has 3 configurations, this conversion will be represented with 3 lines for each configuration but in all 3 lines the conversion file name will be the same.
Type	For example, "XSLT", "COM", ".NET"
Channel	"In", "Out", or "InOut"
Description	Designates the accepted file format, and the format it is converted to

If a mandatory conversion parameter was not read correctly, Administration Console displays a dash sign ('-') in the corresponding column.

The following conversions are provided with Service Connect:

Conversion	Description
anyxml2xml	Converts well-formed xml to the internal envelope format.
ext2int	Converts message to the internal envelope format.
int2ext	Converts from the internal envelope format to the external envelope format.
csv2xml	Converts from the comma-separated value format to the internal envelope format, and from the internal envelope format to comma-separated value format.
CsvAppendConverter	<p>Plug-in similar to csv2xml converter but it is output-only and appends data to the end of specified comma-separated value file instead of creating a new file for every message.</p> <p>CsvAppendConverter requires a filename (OutFile) and an OutFilePath on the ConfigXML. The converter appends the csv it receives to the OutFile. Then the converter returns data to the output channel. The channel also creates a new file with the name specified under mask in the Poster configuration. You can use the LastChunk.xml name in the Channel configuration to avoid mess.</p> <p>When you set the channel to use CsvAppendConverter, the channel is still pushing out one file per message. CsvAppendConversion acts as a secondary outputting entity within the channel. When the channel receives the message it first runs it through the conversion. CsvAppendConverter performs its own output to the file configured in its settings. Then the output channel creates a new file with a single message according to the output channel settings.</p> <p>So, the channel with CsvAppendConverter converter always creates two output files. The file that gets all the data appended is the one configured within the CsvAppendConverter settings. The file that gets each message separately is the one configured within the output channel settings.</p>
CryptoConverter	Plug-in to encrypt, decrypt, sign and verify XML documents.
excel2xml	<p>Converts from the Excel format to the internal envelope format, and from the internal envelope format to the Excel format. To be able to create schemas for xlsx, xlsm, xlsb (Excel 2007 and 2010) files, you should first install the Office System Driver package.</p> <p>Service Connect can only process Excel files without protection.</p> <p>You can set up the converter to select a sheet by name from a multi-sheet spreadsheet.</p> <p>excel2xml converter can handle multiple sheets within a spreadsheet.</p>

Conversion	Description
	If you use the excel2xml converter to convert from xml to xlsx format, setup output file name with the .xlsx extension in the output channel properties. If xls or any other extension is specified, standard xls file will be created.
Office2007Converter	Converts from and to Office 2007 format.
fixedwidth2xml	Converts from the fixed width text file format to the internal envelope format, and from the internal envelope format to the fixed width text file format.
TextToXmlRegExConverter	Converts text file to XML using regular expressions.
stripEnv	Strips the internal envelope from the message.
stripEnvWithoutNS	Strips the internal envelope from the message (without namespace prefix).

These conversions are referred to as standard.

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, AlwaysAddText Qualifier, GroupByFields and GroupByCount.
- **CsvAppendConverter.dll** - This supports the Separator, FirstRowContainsFieldNames, AllowMultipleTables, TextQualifier, AlwaysAddTextQualifier, OutFileEncoding and OutFilePath settings.
- **excel2xml.dll** - This supports DefaultNamespace, GroupByFields, GroupByCount, SheetName, CanonicalFormat and XLSXOutputFormat.
- **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	<p>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.</p> <div style="background-color: #ffffcc; padding: 10px;"> Example <DefaultNamespace> http://myorgnamespace </DefaultNamespace> </div>
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

Configuration Option	Description
	<p>considered a single column. If this option is absent, a comma is used as the separator. This option affects both input and output channels.</p> <div data-bbox="621 304 931 367" style="background-color: #ffffcc; padding: 5px;"> Example <Separator>,</Separator> </div>
FirstRowContainsFieldNames	<p>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.</p> <div data-bbox="621 551 985 677" style="background-color: #ffffcc; padding: 5px;"> Example <FirstRowContainsFieldNames> false </FirstRowContainsFieldNames> </div>
TextQualifier	<p>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.</p> <div data-bbox="621 889 985 952" style="background-color: #ffffcc; padding: 5px;"> Example <TextQualifier>"</TextQualifier> </div>
AlwaysAddTextQualifier	<p>This node may be used to disable quote marks around all the data.</p> <p>Even if the AlwaysAddTextQualifier setting is set to "false", you still must add text qualifier in the following cases:</p> <ul style="list-style-type: none"> • specified non-empty separator is found • specified text qualifier is found • Line Feed (LF symbol) is found <p>The workaround is to use an empty separator and specify text qualifier symbol that does not exist in the data.</p> <div data-bbox="621 1381 985 1444" style="background-color: #ffffcc; padding: 5px;"> Example <AlwaysAddTextQualifier>false</AlwaysAddTextQualifier> </div>
GroupByFields	<p>Defines which fields are used to group nodes.</p> <div data-bbox="621 1558 985 1622" style="background-color: #ffffcc; padding: 5px;"> Example <GroupByFields> Field1,Field3 </GroupByFields> </div> <div data-bbox="621 1674 985 1738" style="background-color: #ffffcc; padding: 5px;"> Example <GroupByFields> 1,3 </GroupByFields> </div> <div data-bbox="621 1790 948 1854" style="background-color: #ffffcc; padding: 5px;"> Example <GroupByFields> Field1,3 </GroupByFields> </div>

Configuration Option	Description
	<p>The GroupByFields is parsed the same way as any CSV line - settings like Separator and Text</p> <p>Qualifier are applicable.</p> <p>For example, if you set Separator to ; :</p> <pre><Separator>;</Separator></pre> <p>You must then use the following format for the GroupByFields:</p> <pre><GroupByFields>2;3;25;26;27;28;29</GroupByFields></pre> <p>If you have FirstRowContainsFieldNames off, you can specify only field indexes.</p> <p>If you have FirstRowContainsFieldNames on, you can specify field names or indexes.</p> <p>Field names to group by must be valid.</p>
GroupByCount	<p>Limits the number of nodes in the group.</p> <div style="background-color: #ffffcc; padding: 10px;"> Example <GroupByCount> 30 </GroupByCount> </div> <p>limits the number of nodes in the group to 30.</p>
OutFileEncoding	<p>The encoding of the output file. The following values are accepted:</p> <ul style="list-style-type: none"> 'UTF8' 'UTF16' 'ASCII' 'Unicode'
OutFilePath	<p>Output file with full path.</p>
SheetName	<p>The name of the Excel sheet or sheets from a multi-sheet spreadsheet that should be converted. To specify multiple sheets, separate them by comma.</p> <p>Use '*' to process all Excel document sheets.</p> <p>If you omit SheetName parameter definition, the first not empty sheet is processed.</p>
CanonicalFormat	<p>Use this setting to convert date, datetime and numeric values to the corresponding values in W3C standard.</p> <p>If you do not define this setting for your conversion (the setting is absent in conversion configuration), the data in the excel file is treated as string.</p> <p>In the default conversion configuration, the CanonicalFormat setting is true, which means the excel2xml conversion transforms date, datetime and numeric values to the corresponding values in W3C standard.</p>
XLSXOutputFormat	<p>Use the XLSXOutputFormat format option with excel2xml output converter to define if output file is saved to xls or xlsx format. To save the file as xlsx, add the following line to the output channel configuration:</p> <pre><XLSXOutputFormat>true</XLSXOutputFormat></pre> <p>If XLSXOutputFormat is not specified, the file format is determined by filename extension defined in the channel Mask property where applicable.</p>

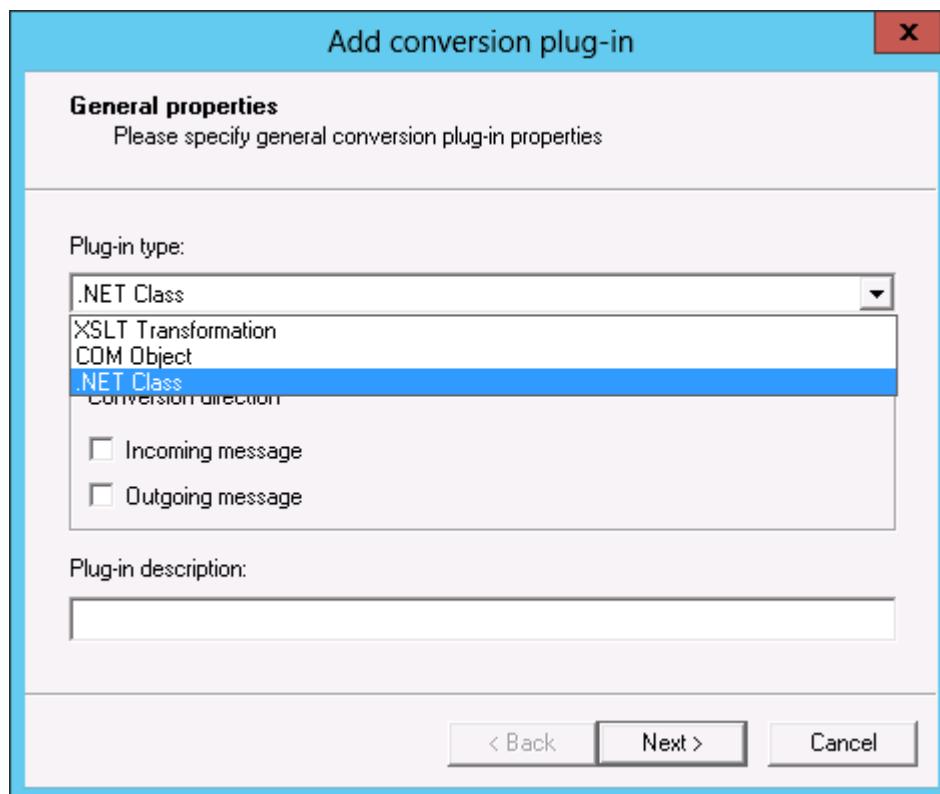
Configuration Option	Description
	<p>Limitation:</p> <p>File extension for channels where Mask is not applicable is always xls even if <XLSXOutputFormat>true</XLSXOutputFormat>, but the file structure isxlsx if <XLSXOutputFormat>true</XLSXOutputFormat>.</p> <p>In SMTP channel, xlsx extension can be specified in the Attachment Name setting.</p>
TemplateDocument	<p>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.</p>
Signing and Encryption algorithms	<p>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. ServiceConnect can digitally sign xml documents and read digital signatures in incoming messages using the W3C xml signature standard.</p> <p>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.xslt or int2ext.xslt transformation. So, all transformation changes affect CryptoConverter plug-in workability.</p>
Field width and name	<p>Field width is a mandatory parameter of an incoming file. Field name is an optional parameter of an incoming file.</p> <div style="background-color: #ffffcc; padding: 10px;"> <p>Example</p> <pre><Fields> <Field width = "number" name = "fieldname" /> </Fields></pre> </div>
CheckLength	<p>This flag enables line length checking; it is an optional incoming file parameter.</p> <div style="background-color: #ffffcc; padding: 10px;"> <p>Example <CheckLength/></p> </div>
DontTrimValues	<p>This flag disables the trimming of ending spaces in table values.</p> <div style="background-color: #ffffcc; padding: 10px;"> <p>Example <DontTrimValues/></p> </div>
DontTrimNames	<p>This flag disables the trimming of ending spaces in field names.</p> <div style="background-color: #ffffcc; padding: 10px;"> <p>Example <DontTrimNames/></p> </div>

Configuration Option	Description
Table and Row	<p>These optional parameters set custom names for tables and lines.</p> <p>Example</p> <pre><Table>table_title</Table> <Row>row_title</Row></pre>

Manage Custom Conversions

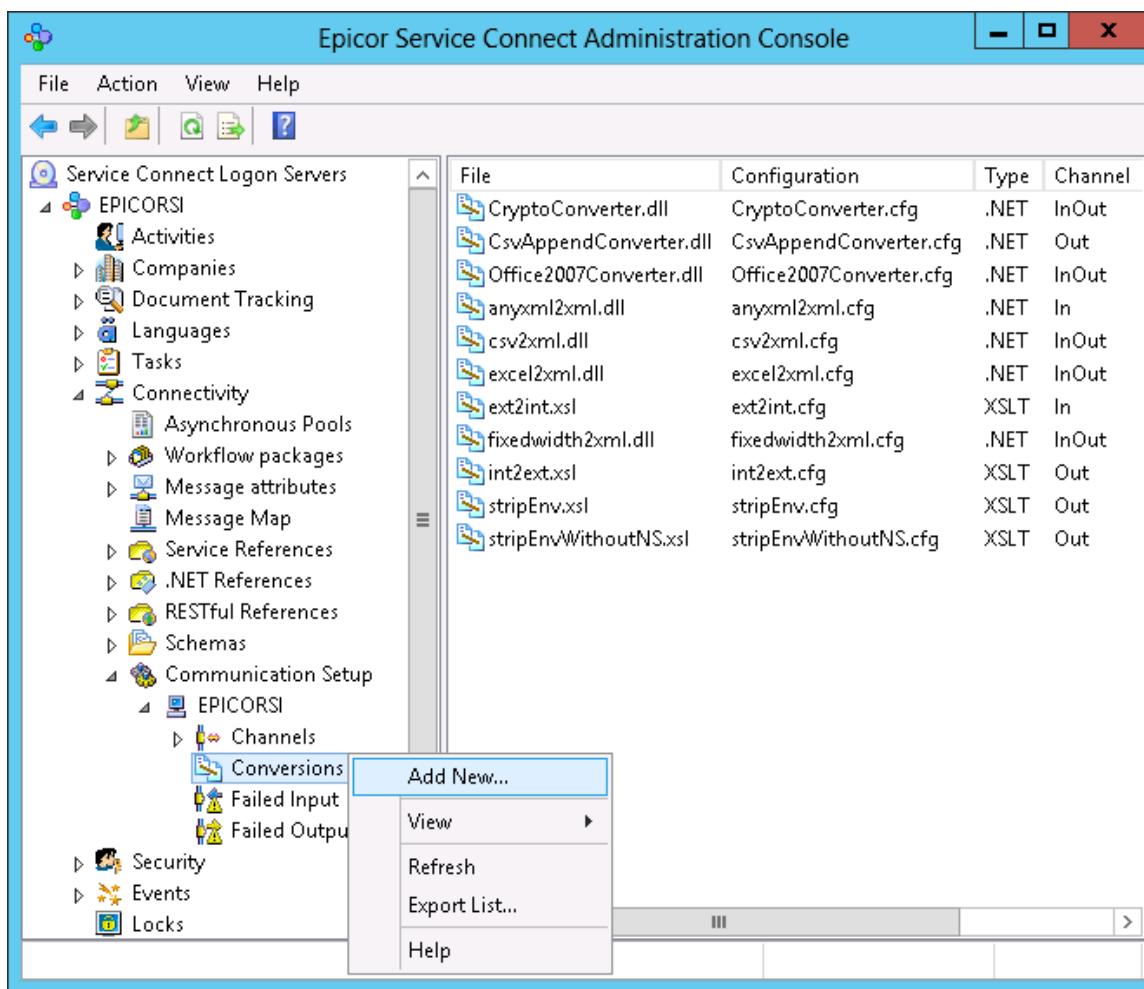
In ESC Administration Console, you can register a new conversion.

There are three conversion types available for creation: **XSLT Transformation**, **COM Object** or **.NET Class**. You can assign the custom 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>.

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



2. On the **General properties** screen, specify the following conversion parameters:

Parameter	Description
Plug-in type	XSLT Transformation, COM Object or .NET Class
Binary conversion	If you selected COM Object or .NET Class, select the Binary conversion check box to register a binary conversion. The following three channel types have a related Binary option: FTP, SFTP, FTP (.NET). So, the Binary conversion plug-in can be used with these channels only when the Binary option set to Yes.
Incoming message, Outgoing message	To specify the conversion direction, select the Incoming message check box to assign the conversion to output channel, or/and select the Outgoing message check box to assign the conversion to input channel.
Plug-in description	Provide the conversion description.

3. Click **Next**.
4. On the **<Conversion type> conversion plug-in** screen, specify type-specific parameters.

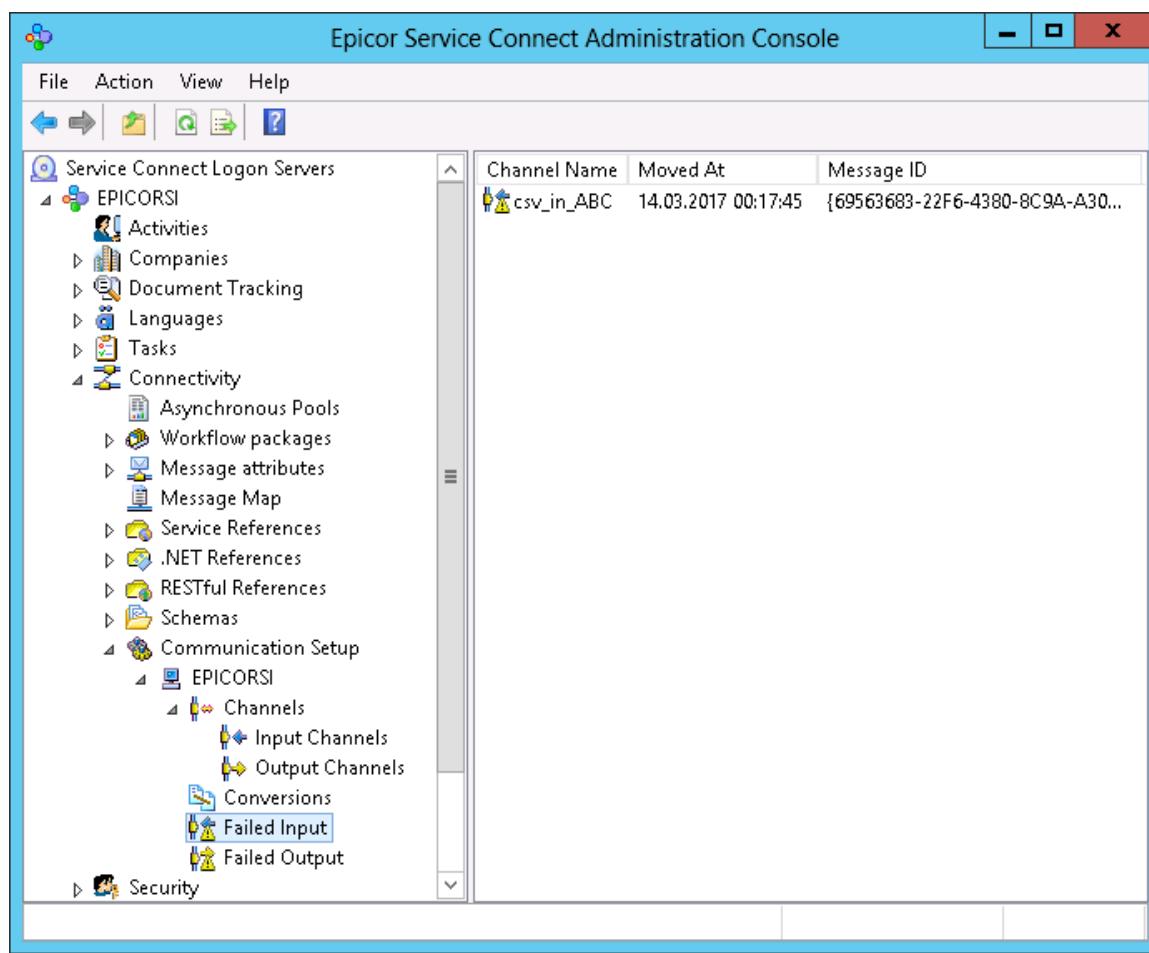
5. Click **Next**.
6. For .NET Class, on the Default input conversion configuration screen, enter the default conversion configuration for input or output channel.
7. Click **Next**.
8. For COM Object or .NET Class, on the **Execution mode** screen, in the **Launch type** field, select one of the following types:
 - **In-process**
 - **Out-of-process**If you selected Out-of-process launch type, in the **Plug-in platform** field, select one of the following platforms:
 - Default
 - Win32
 - x64
9. Click **Next** to load, validate and install the conversion plug-in.
10. If errors occur during installation, the warning message displays. Click the **Details** button to view the error details.
11. The transformation file is copied to the Converters folder of the corresponding messenger. Configuration file is created and placed in the same folder.
12. On the **Process completed** screen, click **Finish**.

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.

The screenshot shows the Epicor Service Connect Administration Console interface. The left pane is a navigation tree with the following structure:

- Service Connect Logon Servers
 - EPICORSI
 - Activities
 - Companies
 - Document Tracking
 - Languages
 - Tasks
 - Connectivity
 - Asynchronous Pools
 - Workflow packages
 - Message attributes
 - Message Map
 - Service References
 - .NET References
 - RESTful References
 - Schemas
 - Communication Setup
 - EPICORSI
 - Channels
 - Input Channels
 - Output Channels
 - Conversions
 - Failed Input
 - Failed Output**
 - Security

Channel Name	Posted At	Moved At	File Name
EmailFromHelpDesk	13.01.2017...	13.01.2017...	{8D369AF6-F72B...
EmailFromHelpDesk	13.01.2017...	13.01.2017...	{AFFEA99D-2E82...
EmailFromOrder	16.01.2017...	16.01.2017...	{1C334414-DF8D...
EmailFromOrder	16.01.2017...	16.01.2017...	{20907D14-5435...
EmailFromOrder	13.01.2017...	13.01.2017...	{20AB4CF5-A51...
EmailFromOrder	16.01.2017...	16.01.2017...	{20C11B7C-5B70...
EmailFromOrder	13.01.2017...	13.01.2017...	{66D4CDCF-B66...
EmailFromOrder	13.01.2017...	13.01.2017...	{68BA5CA2-64A...
EmailFromOrder	16.01.2017...	16.01.2017...	{9C001298-9951...
EmailFromOrder	16.01.2017...	16.01.2017...	{B8C8706D-F789...
EmailFromOrder	16.01.2017...	16.01.2017...	{C622D397-BD84...
EmailFromOrder	16.01.2017...	16.01.2017...	{D135D4F4-BA67...

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.

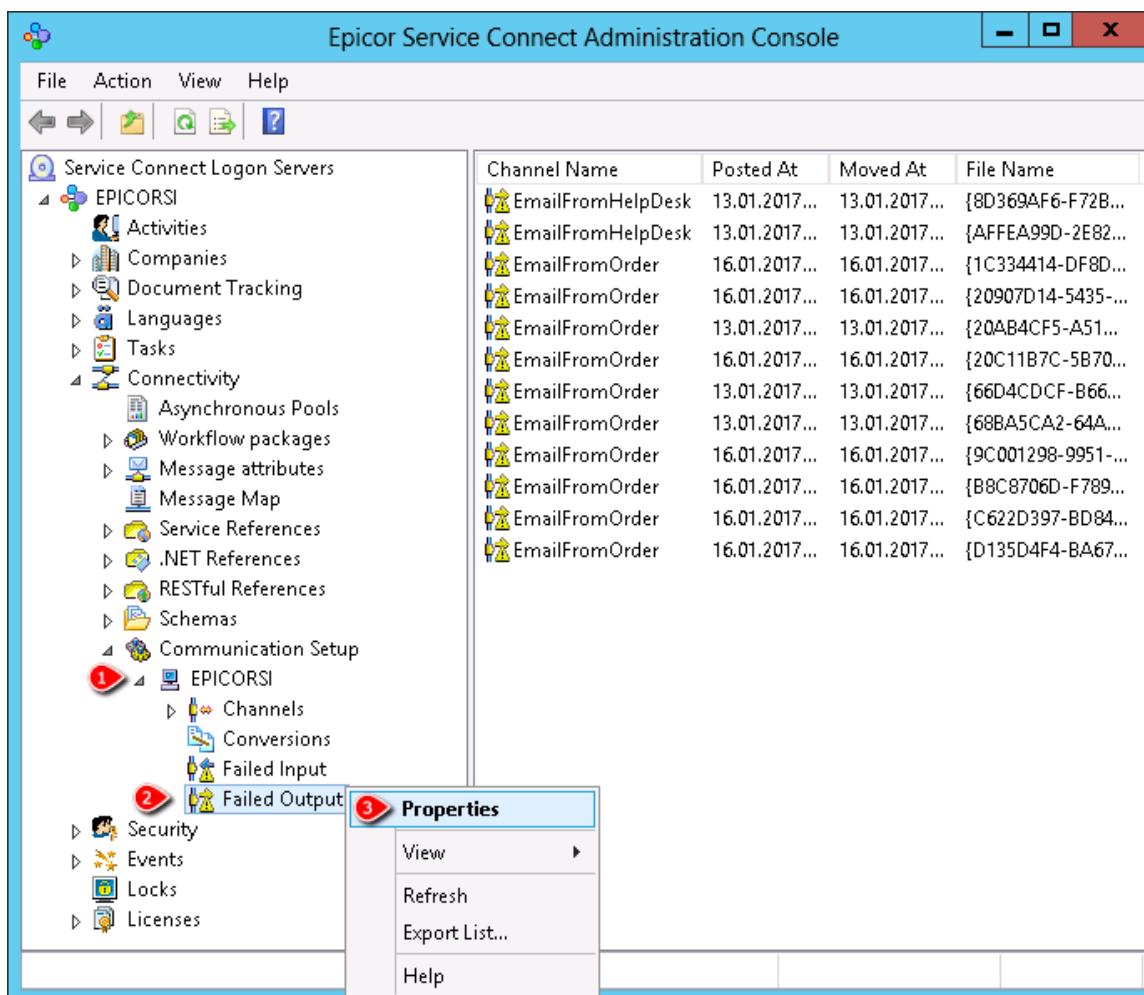
Use the Failed Output node to:

- 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.

Set Number of Attempts for Failed Output

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

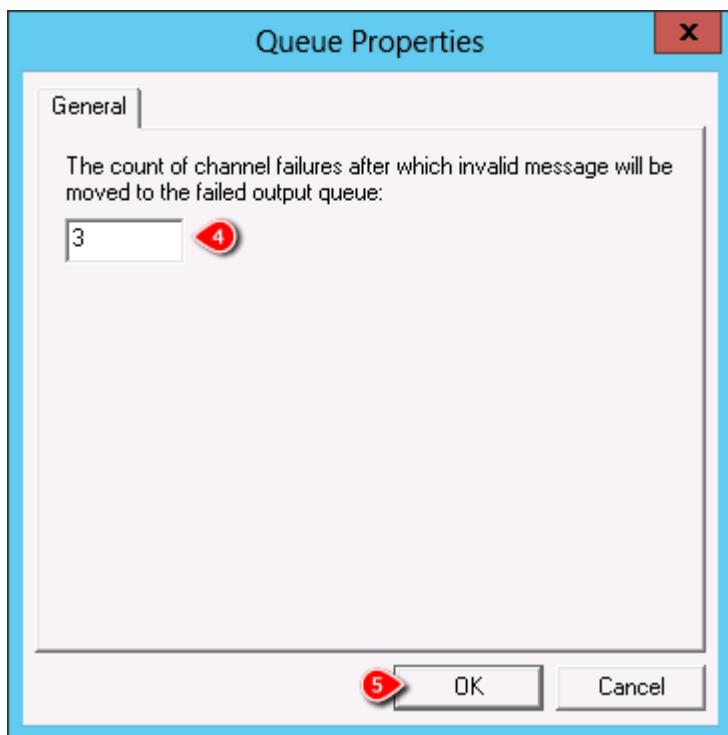
1. In the **tree view**, expand the **Connectivity > Communication Setup ><machine name>** node.



2. Right-click **Failed Output**.

3. Select **Properties**.

4. In the **Queue Properties** window, set the number of failures after which the invalid message will be moved to the failed output queue.



5. Click **OK**.

Workflow Converter

Use the Workflow Converter utility to perform the following conversions:

- Epicor 9.05 WCF web services to Epicor 10 .NET BOs
- Epicor 9.05 WSE web services to Epicor 9.05 .NET BOs or to Epicor 10 .NET BOs (Data Set based)
- Epicor 9.05 .NET BOs to Epicor 10 .NET BOs (Data Set based)
- Epicor 9.05 BPM schemas to Epicor 10 BMP schemas
- Epicor 10 WCF web services (Table Set based) to Epicor 10 .NET BOs (Data Set based)
- Epicor 10 .NET BOs (Data Set based) to Epicor 10 WCF web services (Table Set based)

Workflow Converter is a part of the standard Service Connect installation package.

The Workflow Converter receives a Service Connect workflow with Web Method elements or .NET Call elements and generates a new workflow with .NET Calls or Web Methods. The Workflow Converter also patches the Conversion, Condition, Choice, Sub-Workflow elements within the workflow and the e-mail templates which the workflow uses.

The WorkflowConverter converts all e-mail templates.

The Workflow Converter does not convert and does not copy the .xslt files if they are not used in the workflow.

The converted workflow is saved to a new package with the name of the original package plus the "_Converted" suffix. So, the Workflow Converter does not make any changes in the original workflows.

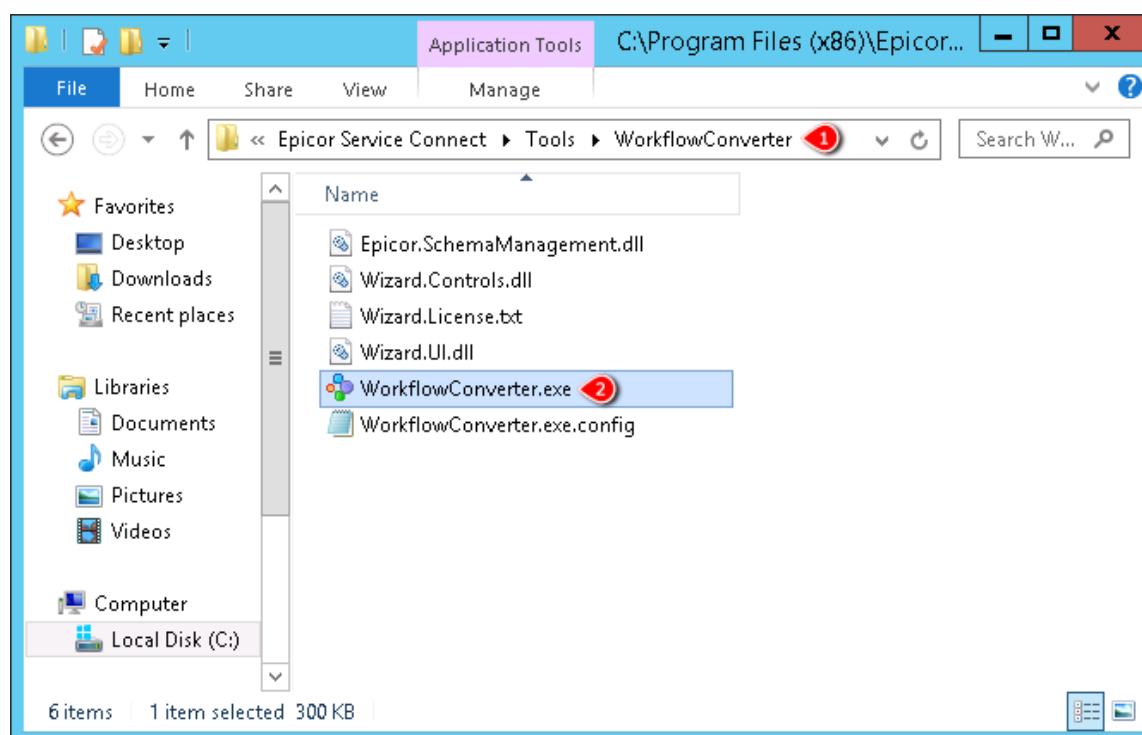
Before you run the workflow conversion process, perform the following steps:

1. Open your source workflow in the Workflow Designer.
2. Note down the names of all the Web Method and .NET call elements used in the workflow.
3. Use the Administration Console to import the .NET or Web references to replace the Epicor 9.05 .NET references and Web Methods under new names. Workflow Converter requires that old 9.05 references and new Epicor 10 references are defined within the same Service Connect installation during conversion.
4. If you convert a workflow which was created to process BPM directives, use the Administration Console to import Epicor 10 BPM schemas to replace Epicor 9 BPM schemas. The Workflow Converter analyzes available schemas to suggest corresponding replacement for Epicor 9 BPM schemas.

If you plan to convert workflows with 9.05 .NET references with UD fields, create UD fields with the same name and type in Epicor 10. Then import these .NET references from Epicor 10 with the **Import UD Fields** setting enabled.

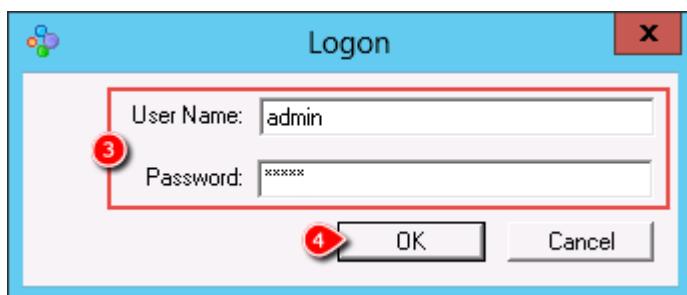
To start the Workflow Converter:

1. Navigate to the <SC Installation Folder>/Tools/WorkflowConverter folder.



2. Double-click the WorkflowConverter.exe file.

3. On the **Logon** screen, enter the credentials.

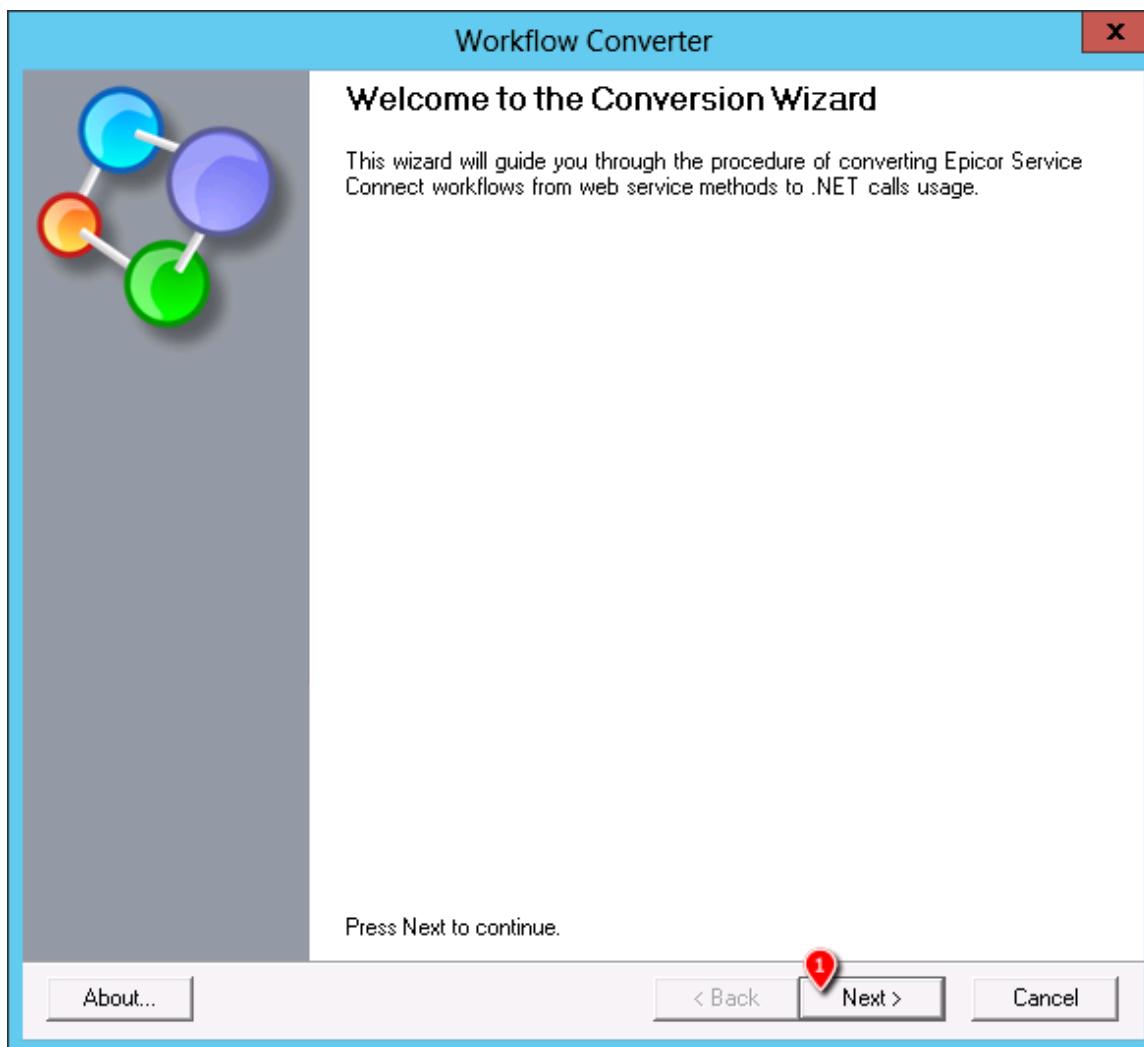


4. Click **OK**.

Convert Workflow

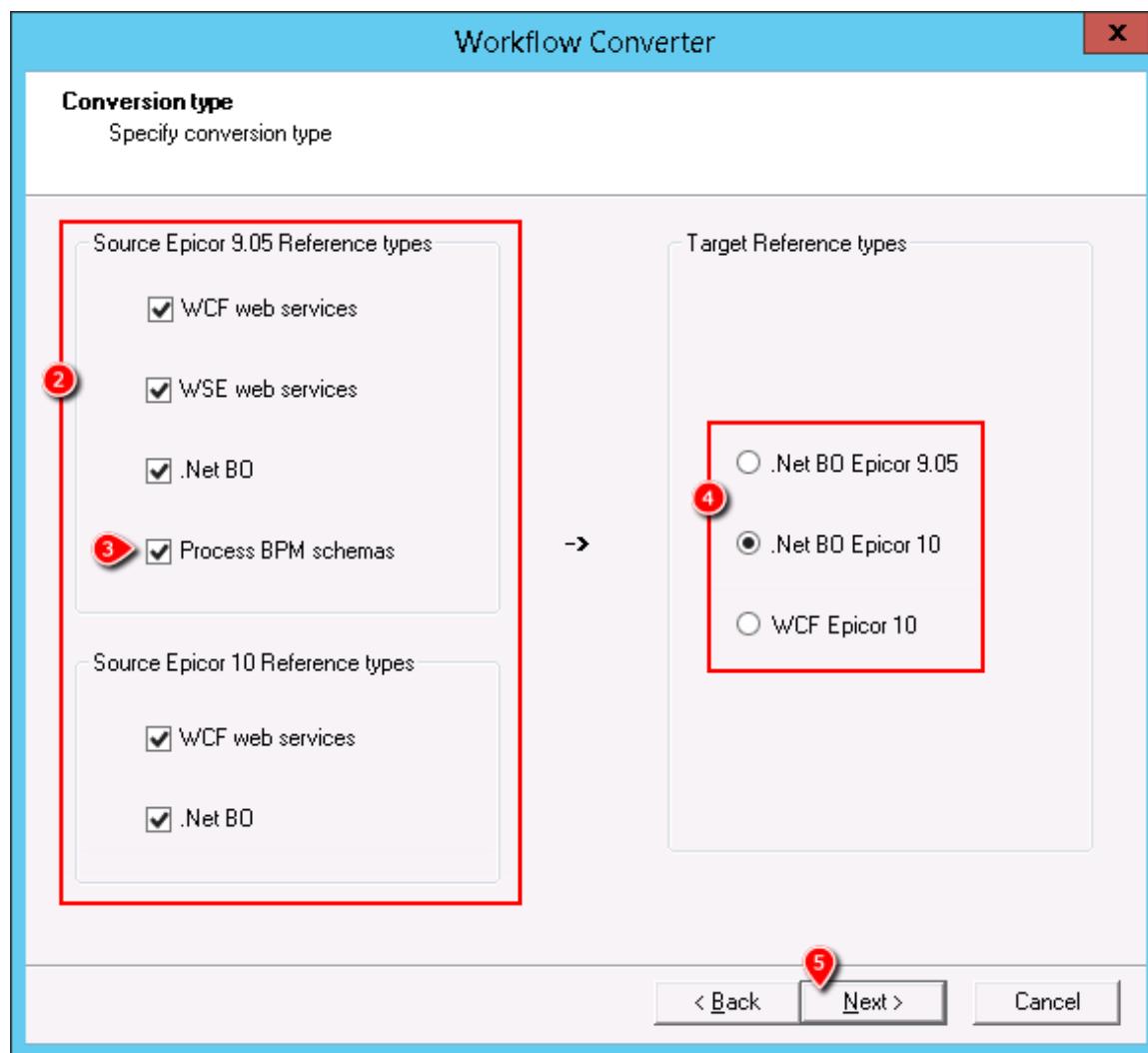
To convert a workflow:

1. On the **Welcome to the Conversion Wizard** screen, click **Next**.



The next screen lists the available conversion directions.

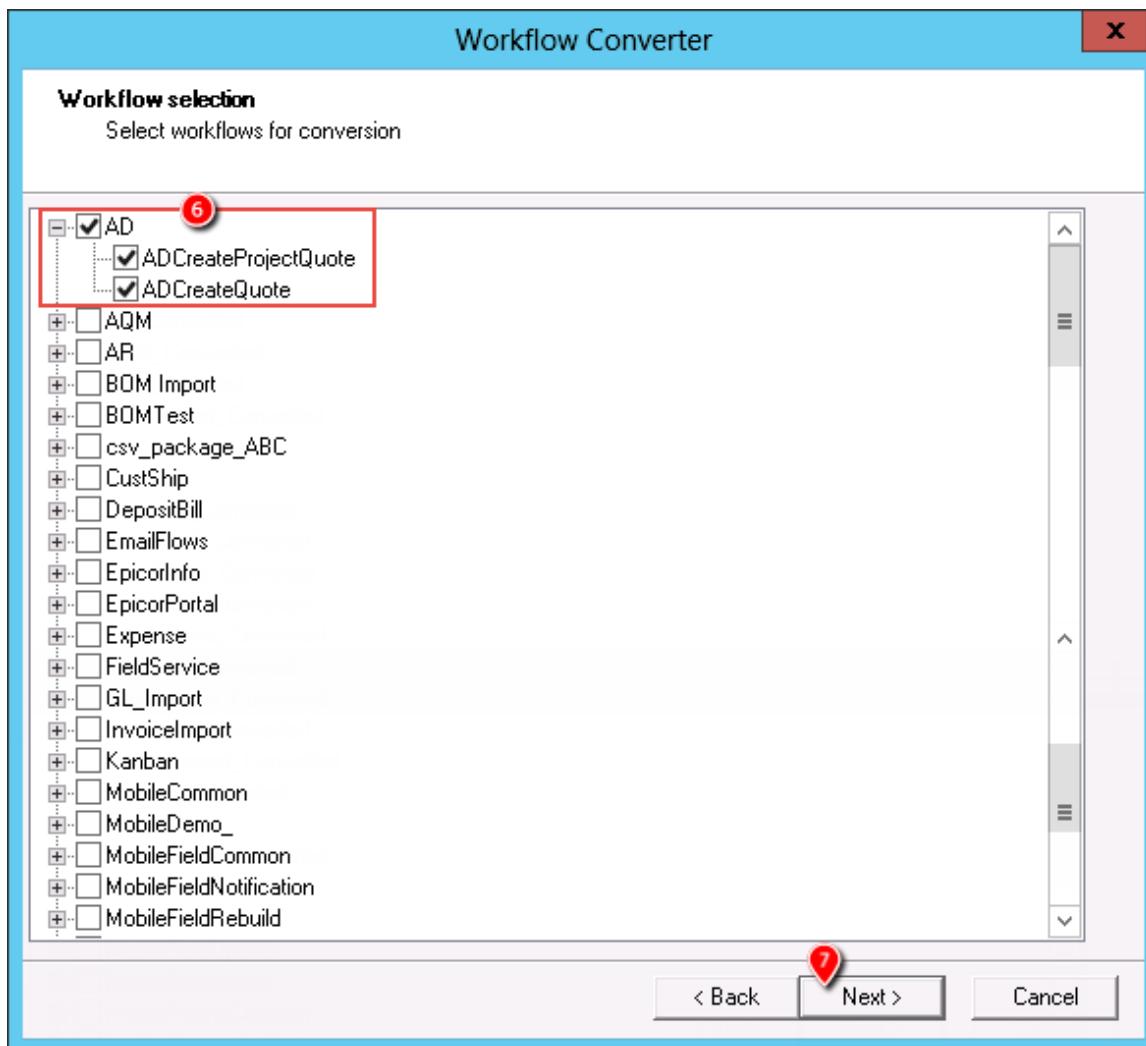
2. In the **Source Epicor 9.05 Reference types** and **Source Epicor 10 Reference types** panes, select the type or types of the references you want to convert.



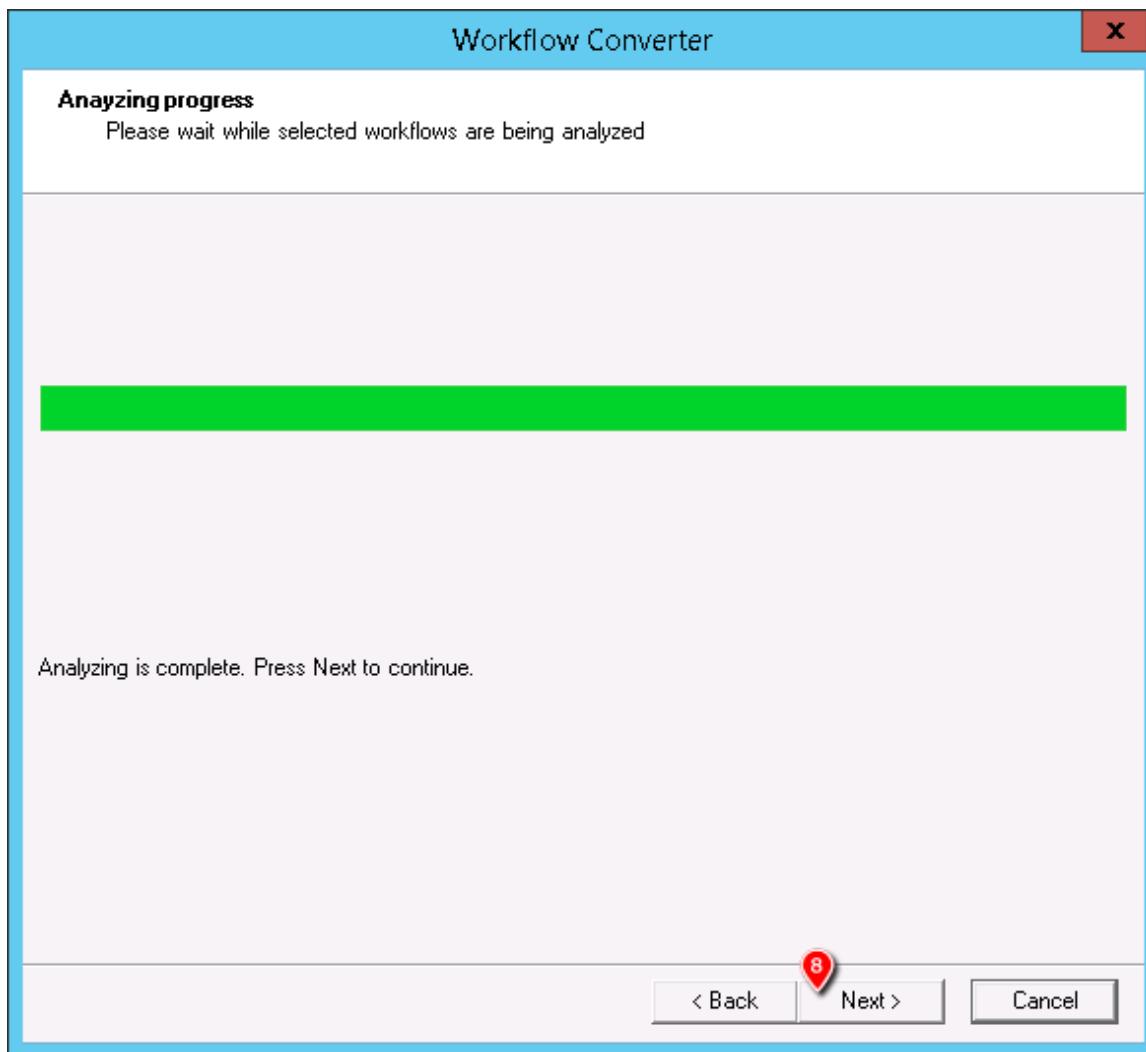
You can leave all the check-boxes selected even if you need to convert workflows with just one reference type.

3. If you convert a workflow which was created to process calls from BPM directives, select the **Process BPM schemas** check box to replace the BPM schemas used in the workflow with Epicor 10 BPM schemas.
4. In the **Target Reference types** pane, select the type of the references to which you want to convert your web services or .NET business objects.
5. Click **Next**.

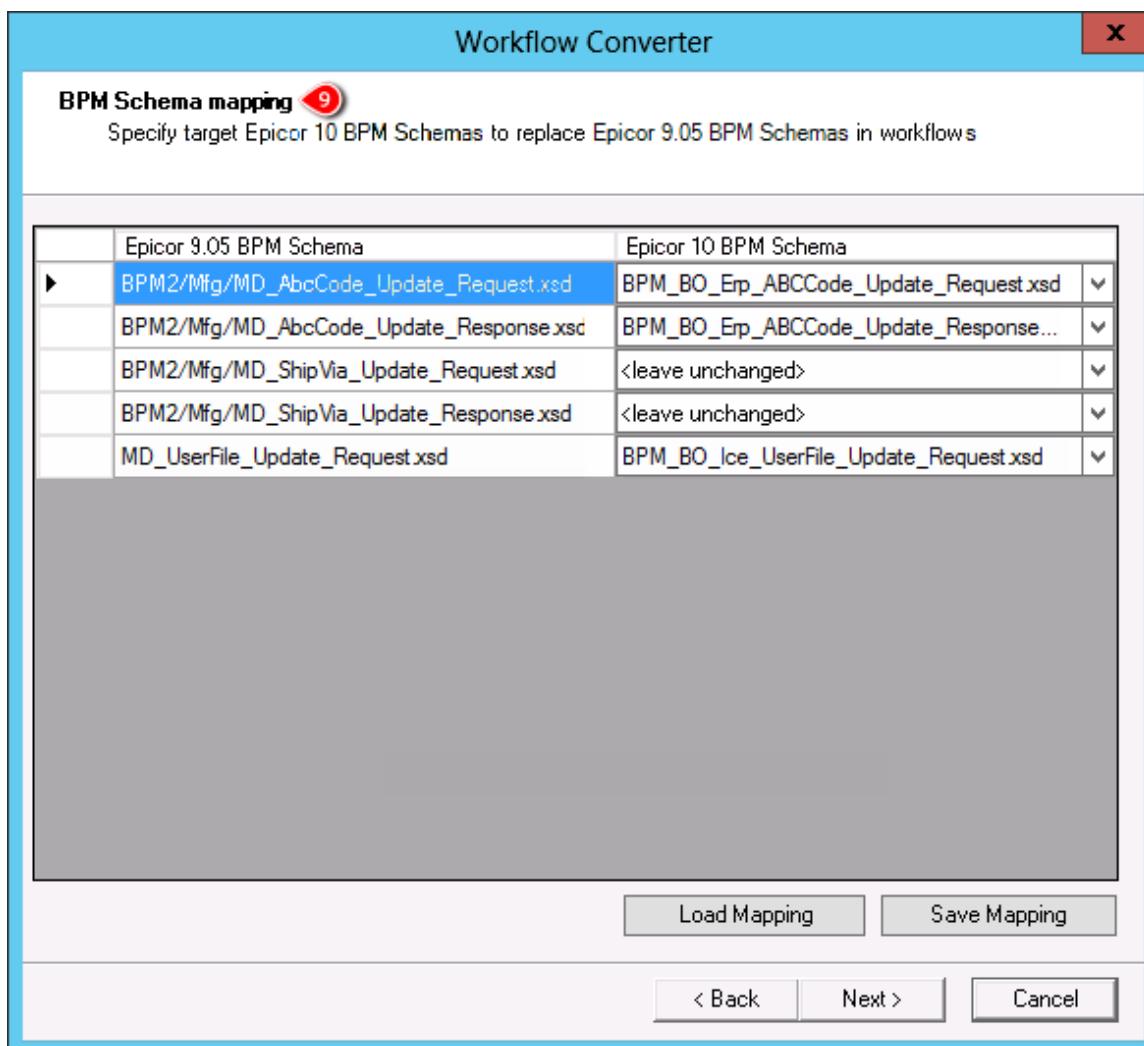
On the **Workflow selection** screen, all the workflow packages that exist in the system display.



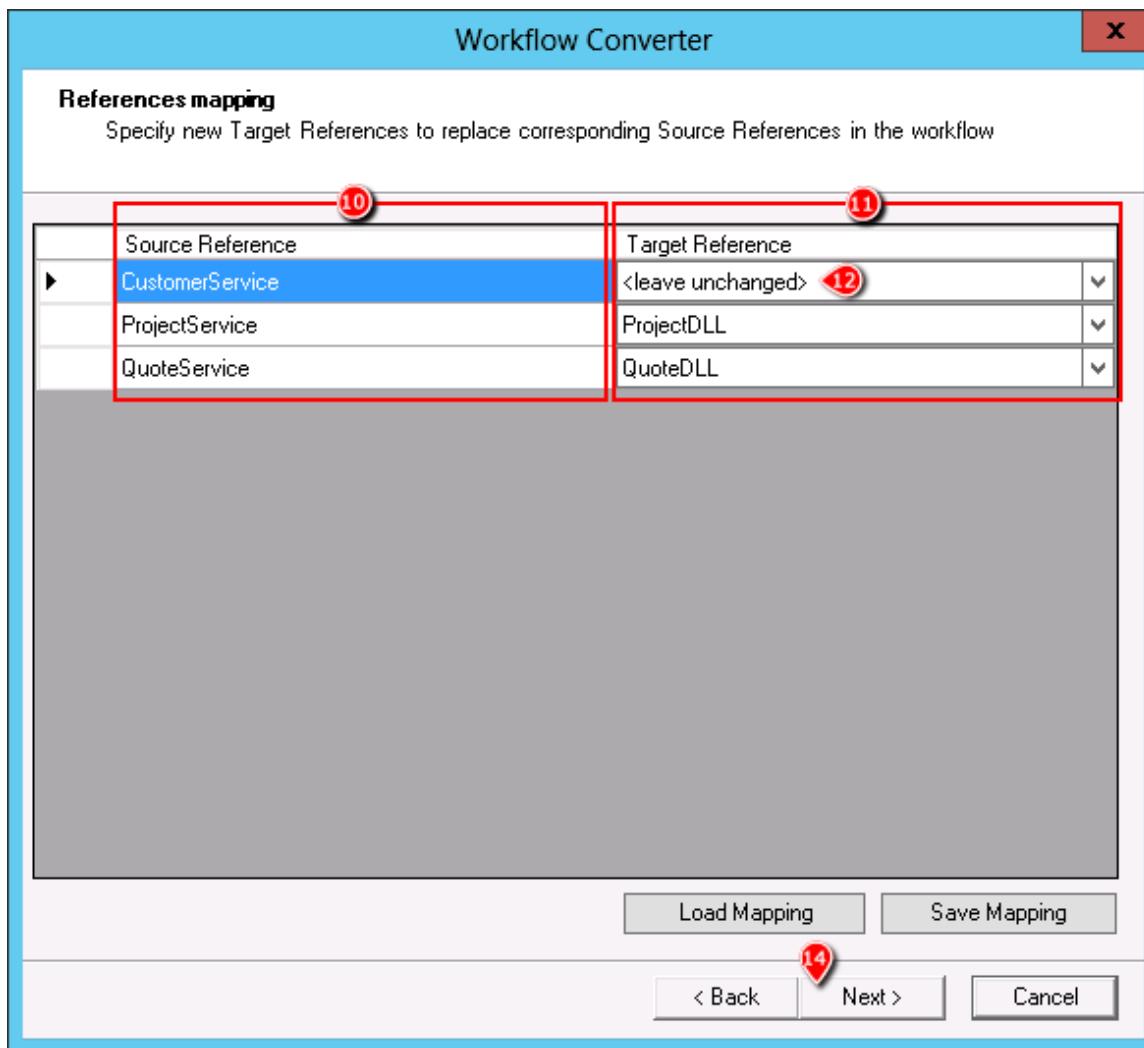
6. Select a workflow package or packages, or expand one or more of the workflow packages and select a workflow or workflows.
7. Click **Next**.
8. On the **Analyzing progress** screen, wait while the analyzing progress bar is complete, and click **Next**.



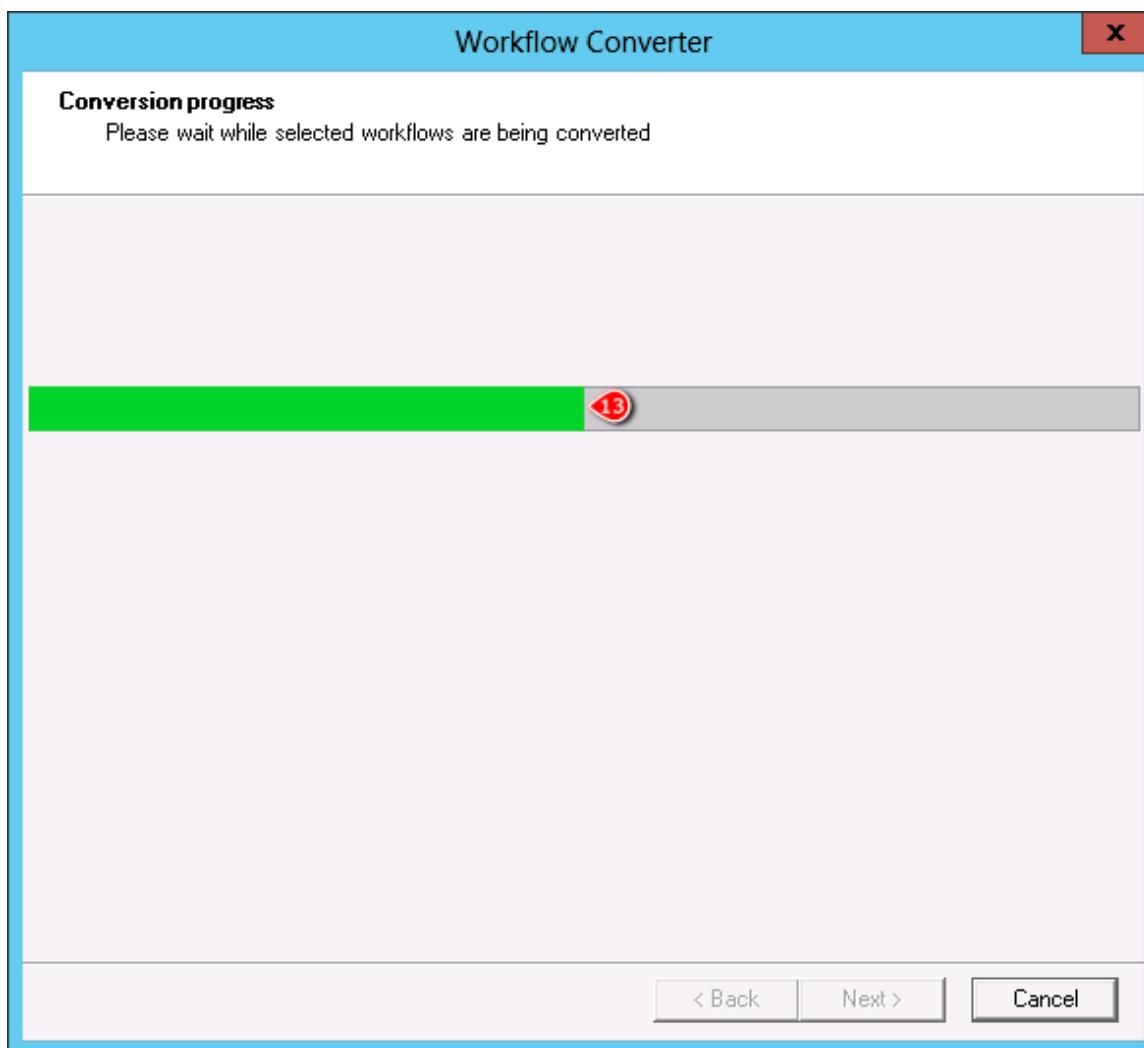
9. If you selected the Process BPM schemas option, on the **BPM Schema mapping** screen, for each BPM schema used in the source workflow specify the Epicor 10 BPM schemas that will replace them.



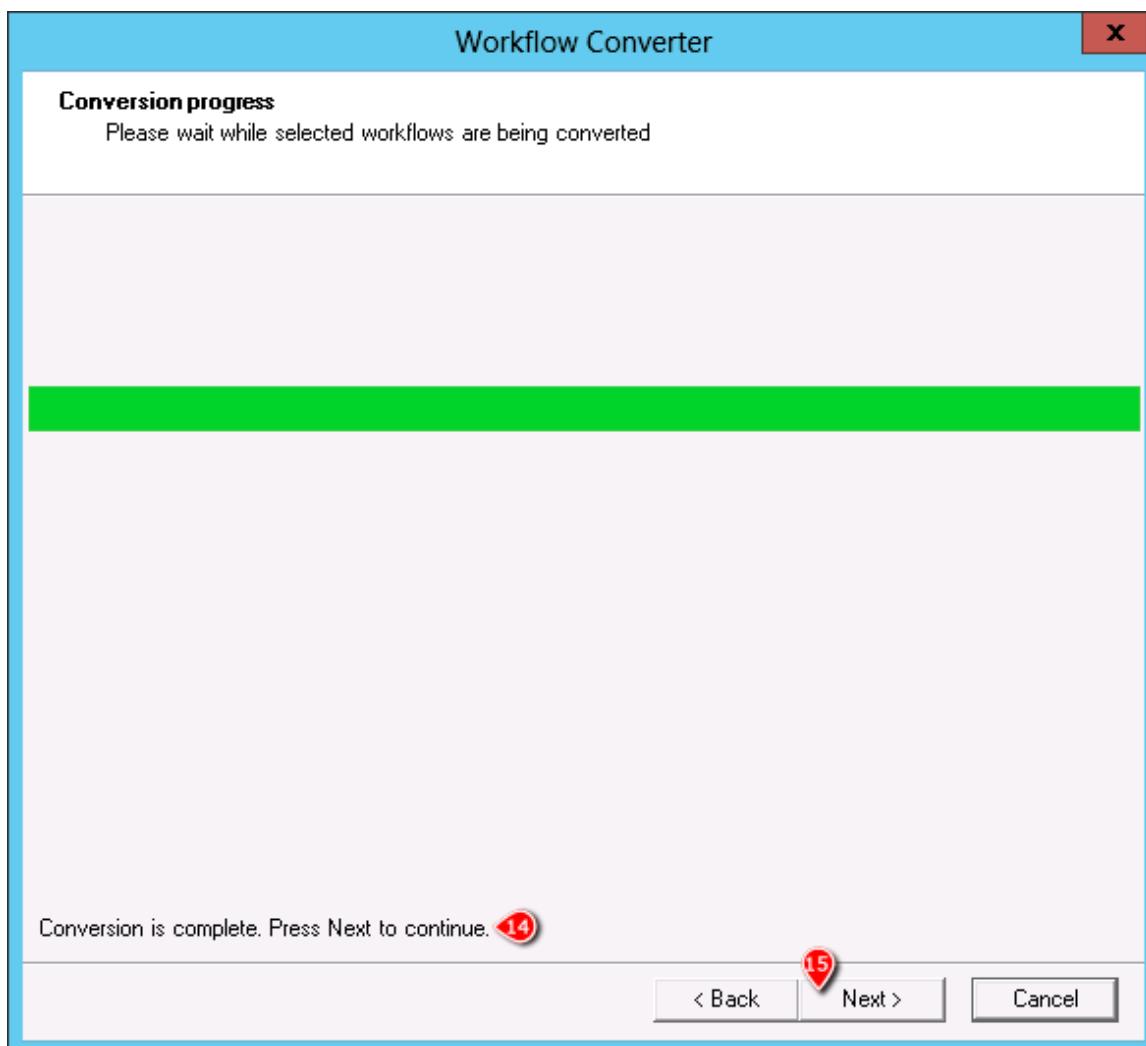
10. On the **References mapping** screen, in the **Source Reference** column, all the Service References or .Net References used in the selected workflows are displayed.



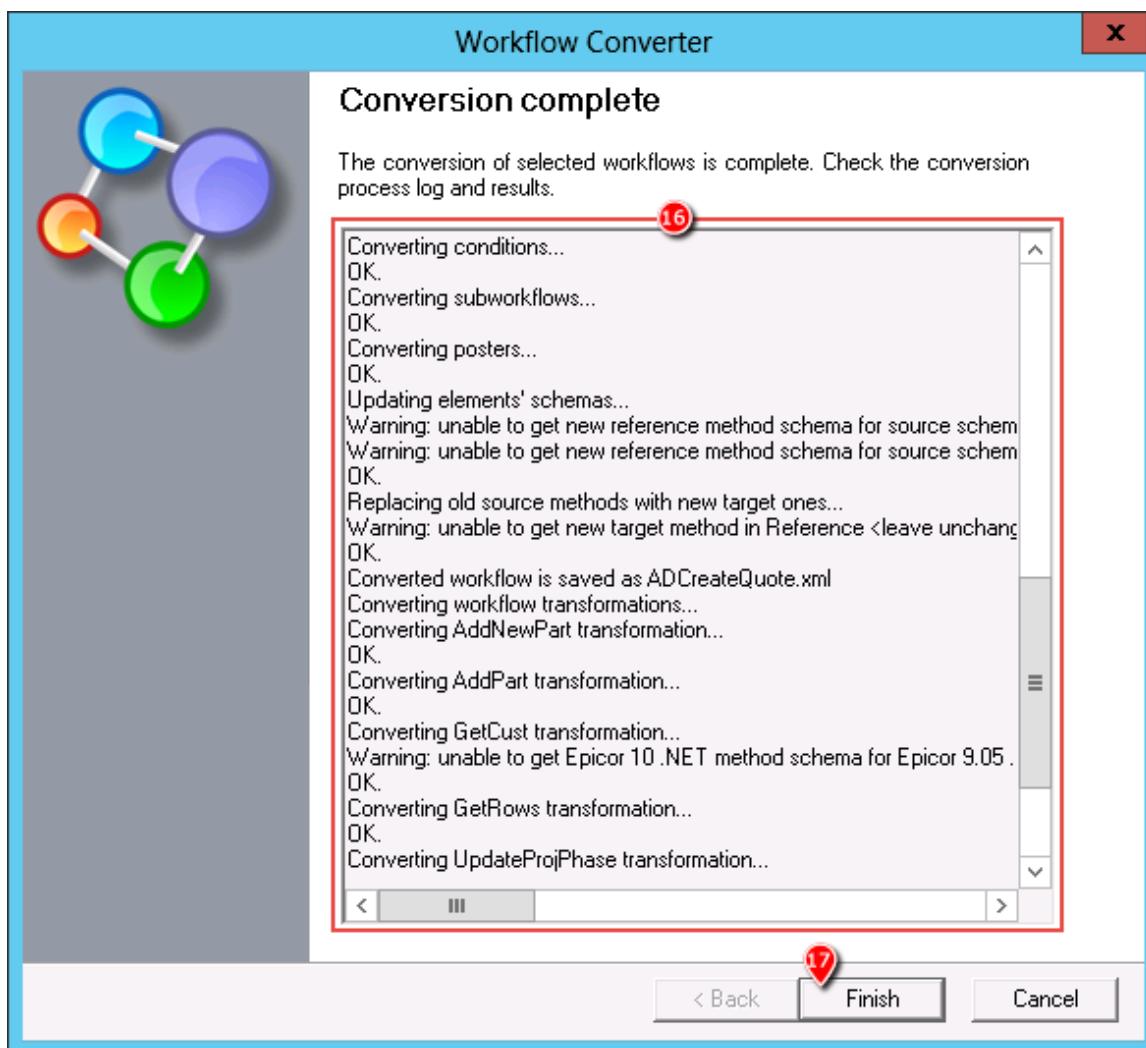
11. For each Source Reference, a replacing .NET reference displays in the **Target Reference** column. Verify the mapping is accurate.
12. If the Converter does not find the mapping, the **<leave unchanged>** option is used. When you convert from Epicor 9.05 to .NET BO Epicor 10, and the method with the same name cannot be found, the original Epicor 9.05 web service or Epicor 9.05 .NET BO is used in the resulting workflow and marked with red text.
13. If you want to perform no conversion for a specific Service Reference, in the **Target Reference** column, select **<leave unchanged>**. The elements for which you selected the **<leave unchanged>** option, are marked with green text in the resulting workflow.
14. Click **Next**. The conversion process starts.



15. Wait for the **Conversion is complete. Press Next to continue** message.



16. Click **Next**.
17. On the **Conversion complete** screen, view the conversion log.

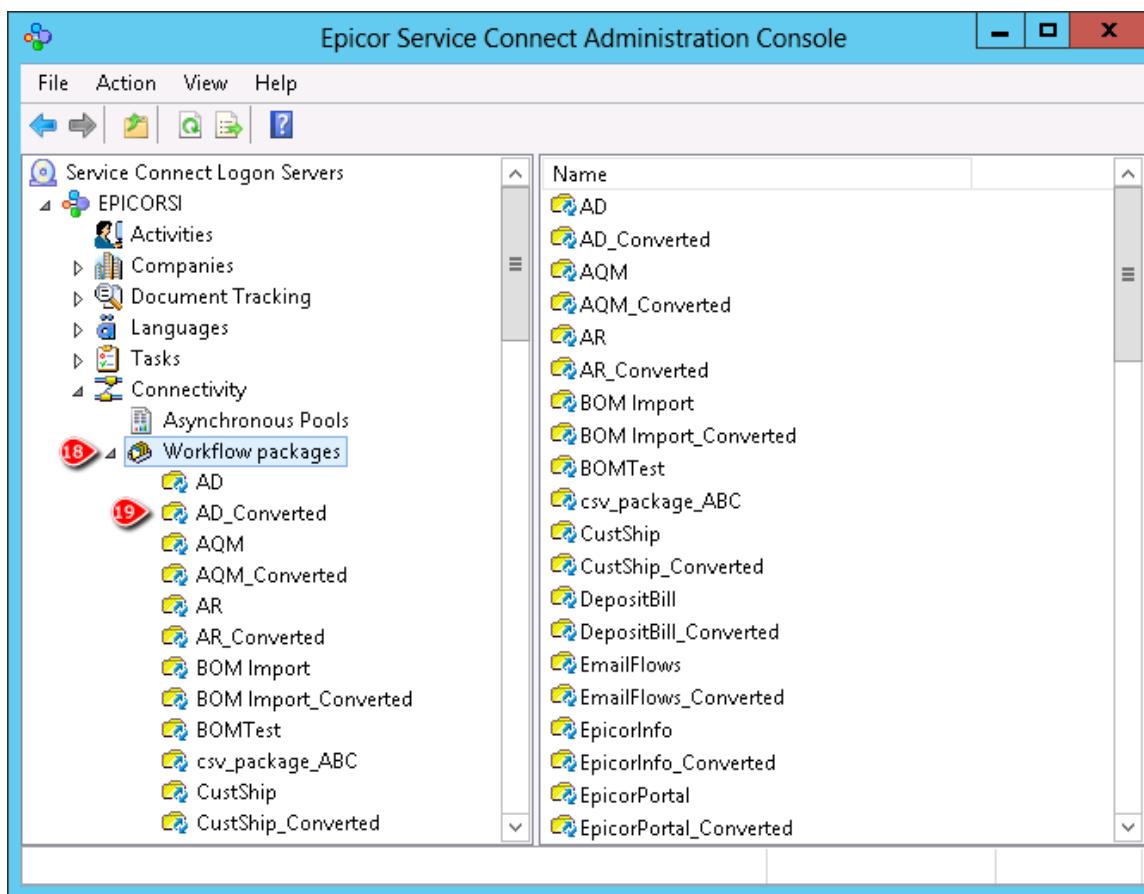


18. Click **Finish**.

The Workflow Converter creates a new workflow package named <SourceWorkflowPackage>_Converted.

19. In the Administration Console, expand the **Connectivity > Workflow packages** node.

20. Locate the workflow package named <SourceWorkflowPackage>_Converted.



The resulting converted workflow is located within this workflow package and has the same name as the source workflow.

You usually use the Workflow Converter when you upgrade to Service Connect 10 and want to transform your workflows from Epicor 9.05 WSE or WCF web services, or Epicor 9.05 .NET Business Objects (BOs) to Epicor 10 .NET BOs. For the detailed instructions on how to use Workflow Converter in live environment refer to the Best Practices: Workflow Converter page in the How To's section in ESC 10 Application Help.

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.

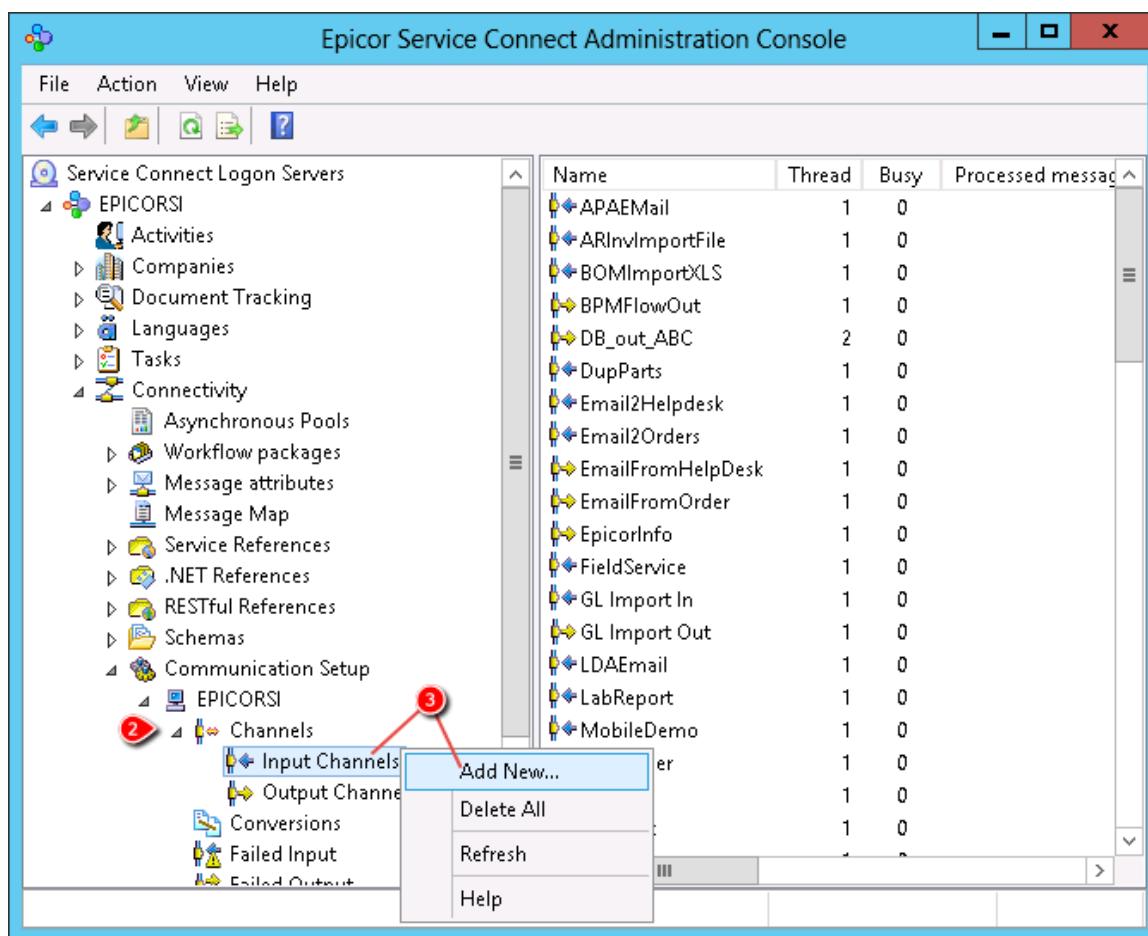
Set Up the Input Channel

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**.

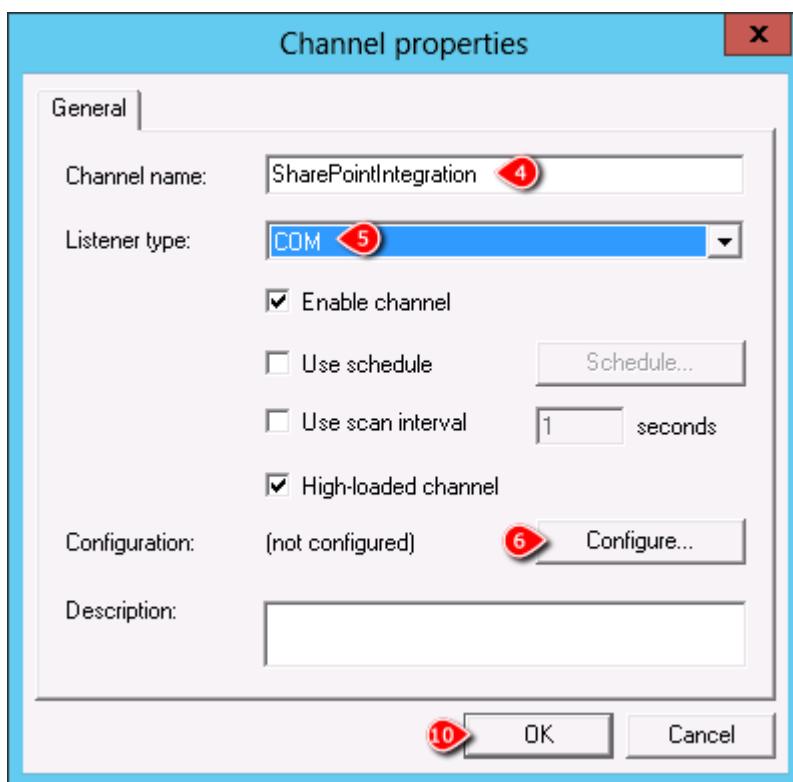
2. In the tree view, expand the **Connectivity > Communication Setup ><machine name>> Channels** node.



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

The **Channel properties** window displays.

4. Enter a **Channel name**.

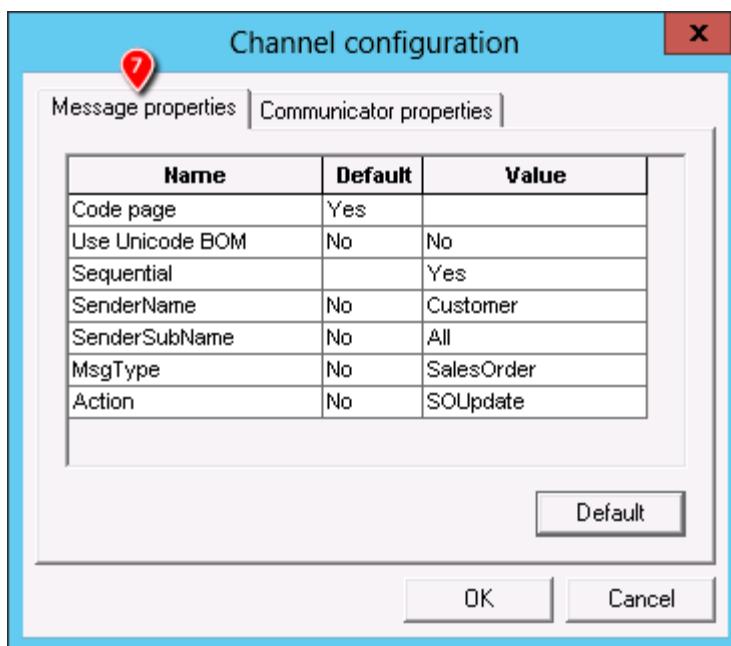


5. Click the **Listener type** drop-down list and select **COM**.

6. Click **Configure**.

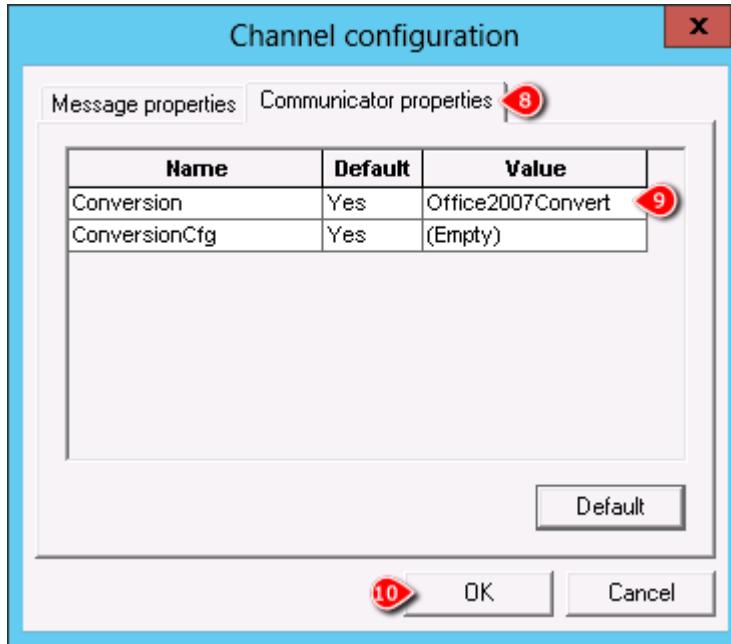
The **Channel configuration** window displays.

7. On the **Message properties** tab, enter the appropriate message attributes.



Message attributes are added to the documents when it enters the input channel. Service Connect looks for a message map with the same attribute 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.

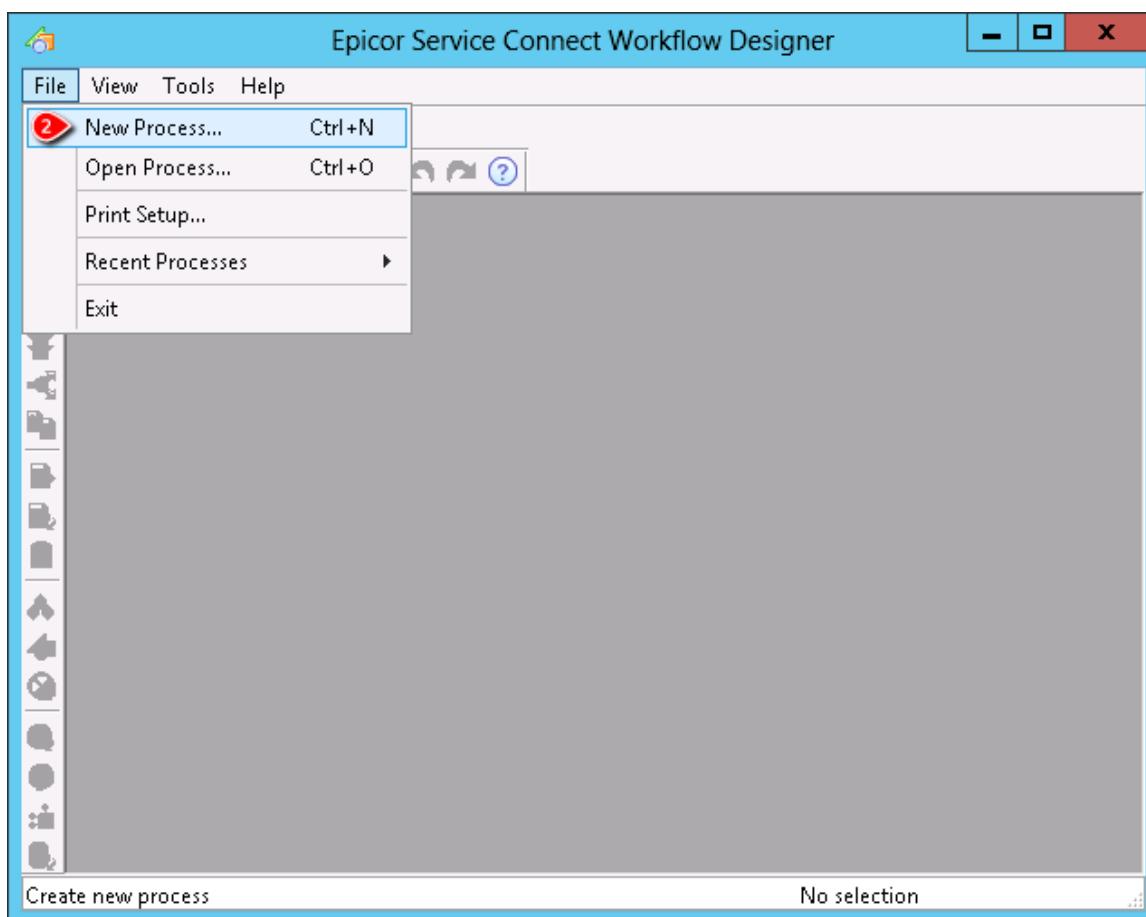
Create a Workflow

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

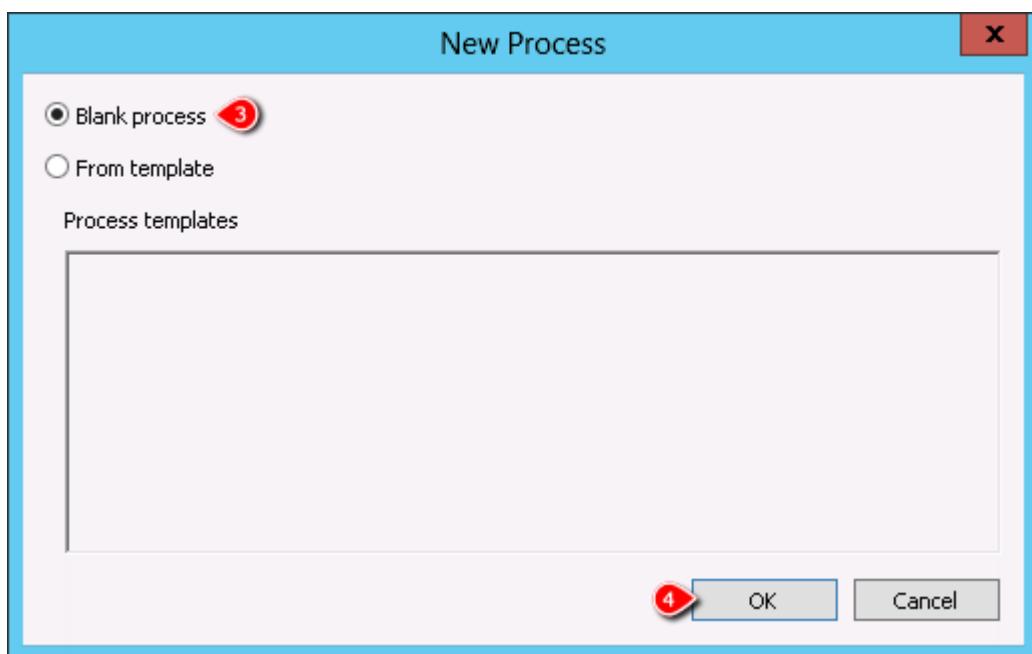
To create a new workflow:

1. Log into the Workflow Designer using the menu path **Start > All Programs > Epicor Service Connect > Workflow Designer**.

- From the **File** menu, select **New Process**.

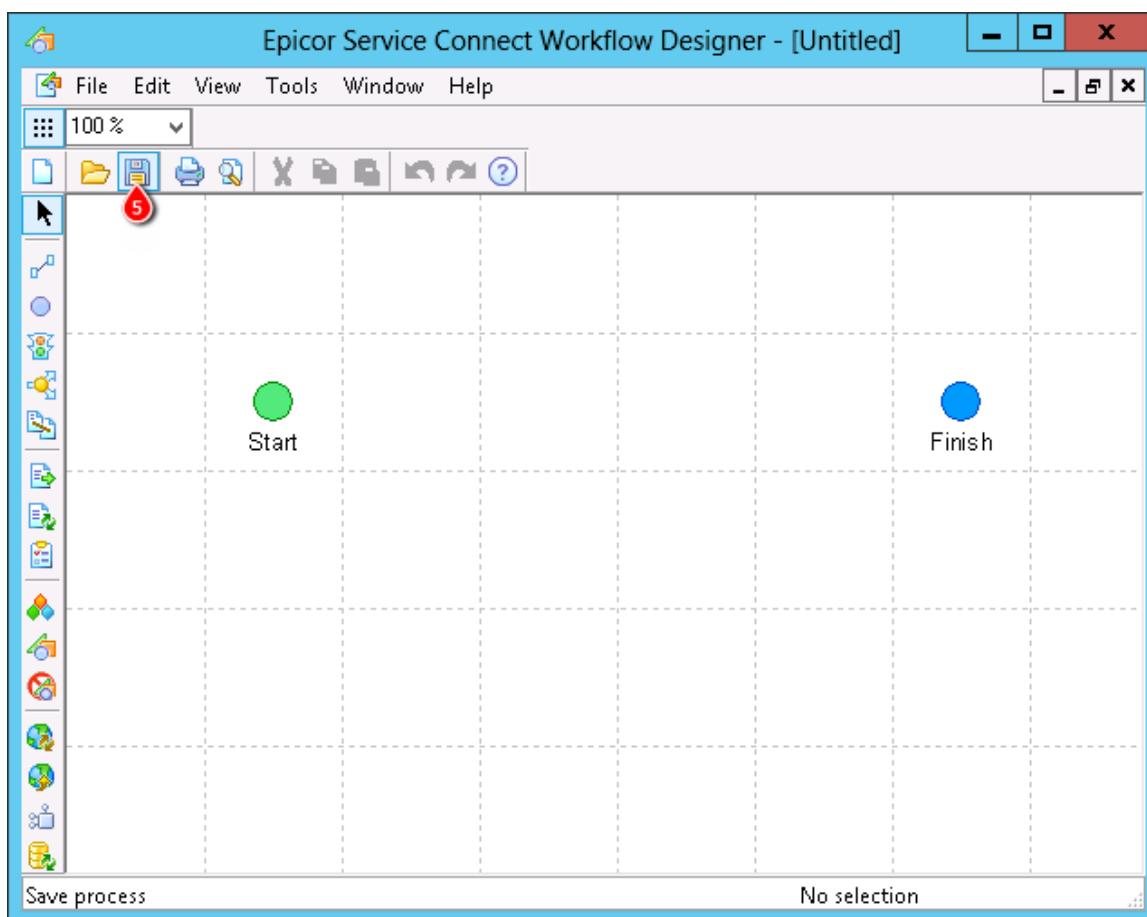


- In the **New Process** window, leave the **Blank process** option selected.

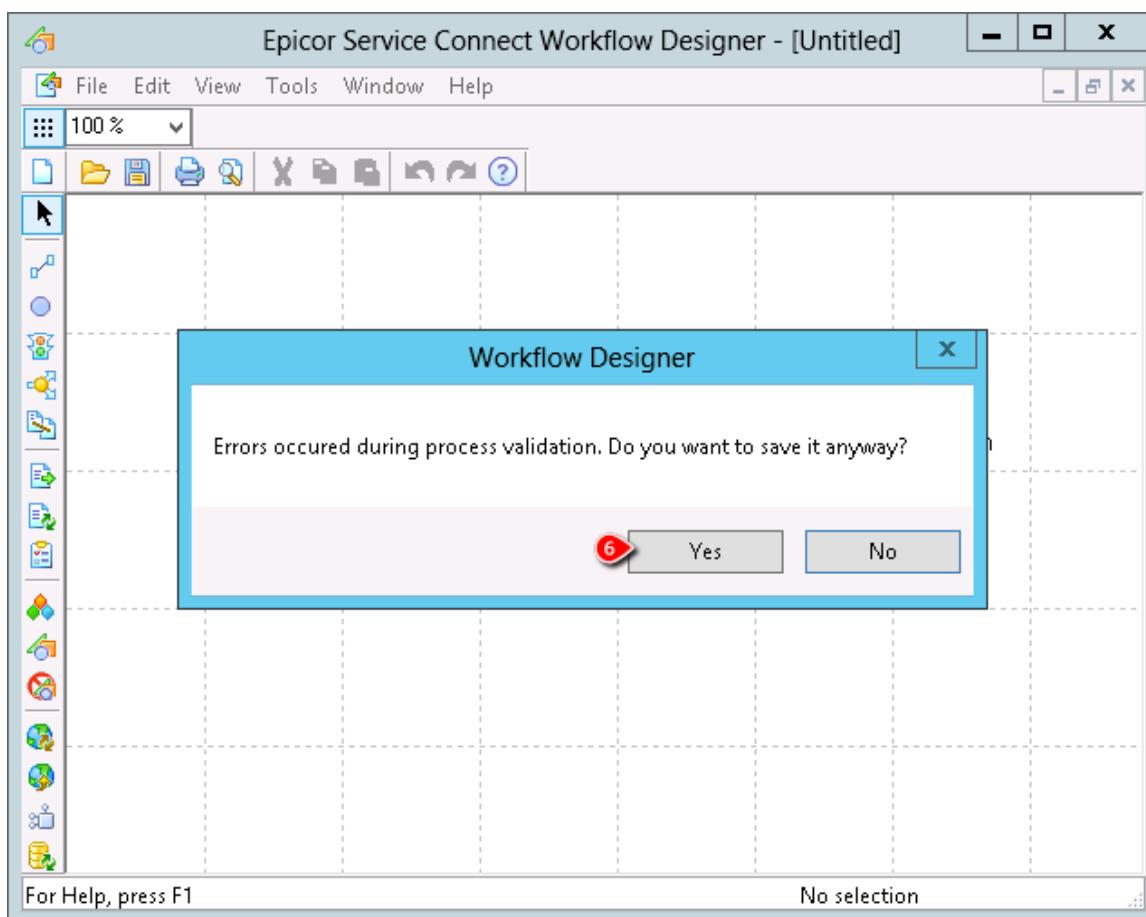


- Click **OK**.

5. Click Save.

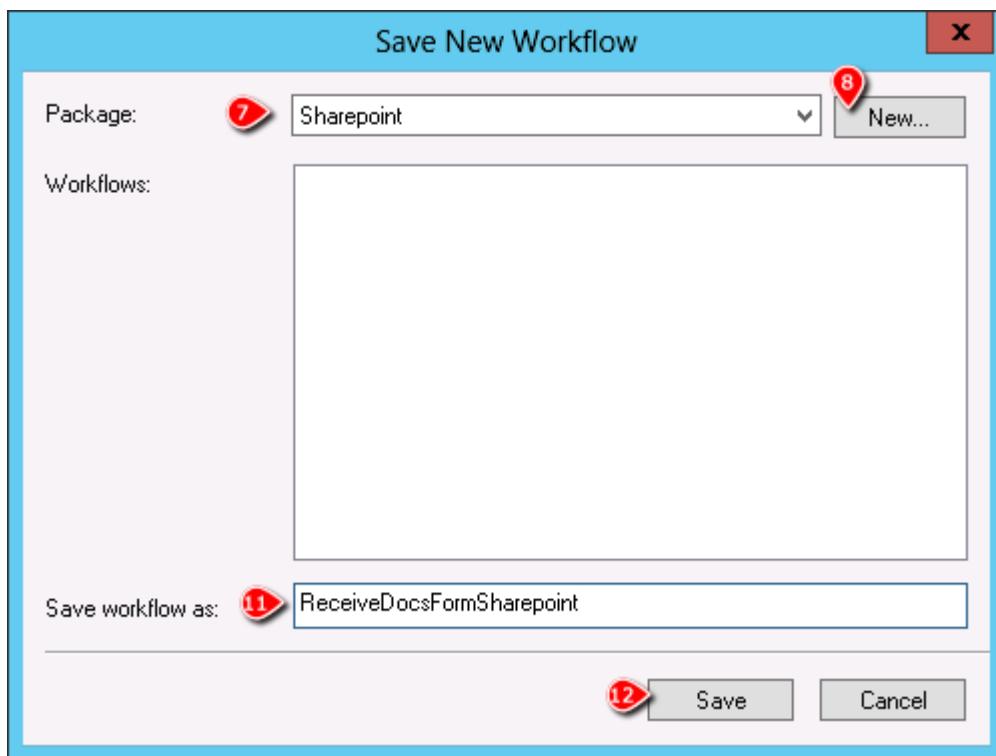


6. To the warning message, click **Yes**.

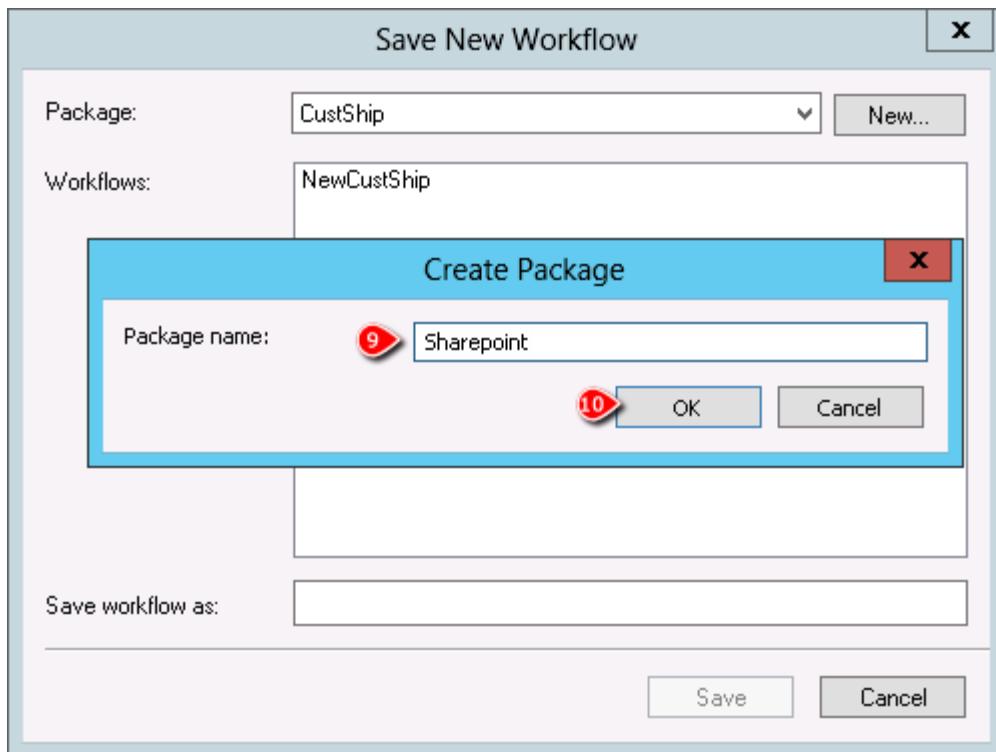


7. Click the **Package** drop-down list to select a workflow package.

8. Alternatively, click **New** to create a new workflow package.



9. In the **Create Package** window, type the **Package name**.



10. Click **OK**.

11. In the **Save workflow as** field, enter a workflow name.

12. Click **Save**.

13. Click **Yes** when prompted to save the schema.

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

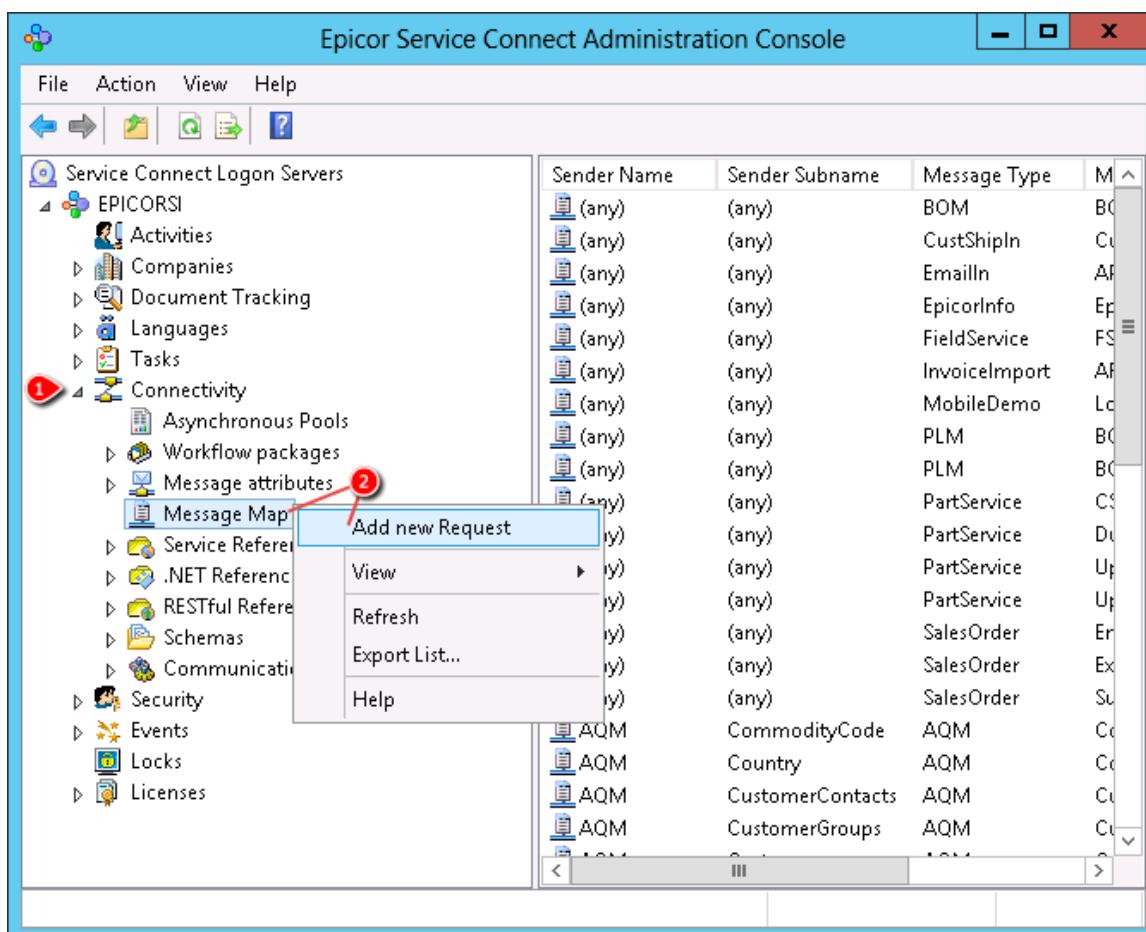
To use the incoming document in the workflow, you must create a schema. You can do it in the Administration Console in the Schemas node, or use the Generate Schema from Sample Data tool in Workflow Designer.

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.



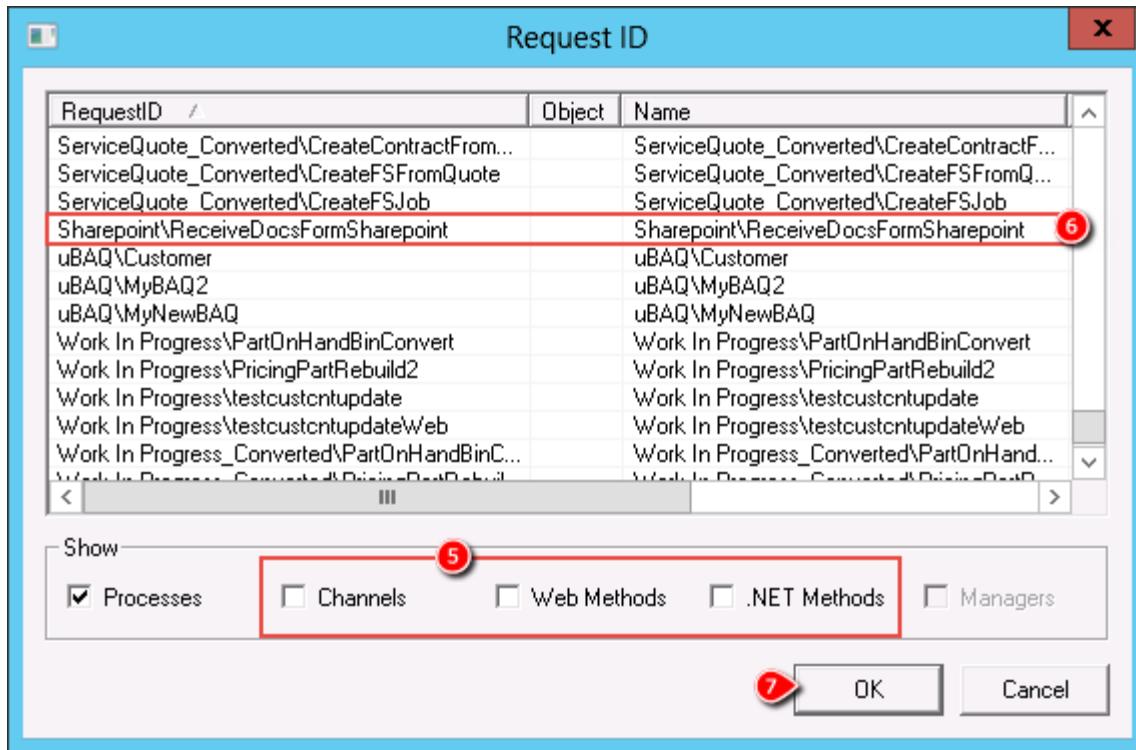
2. Right-click **Message Map** and select **Add new Request**.

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**.

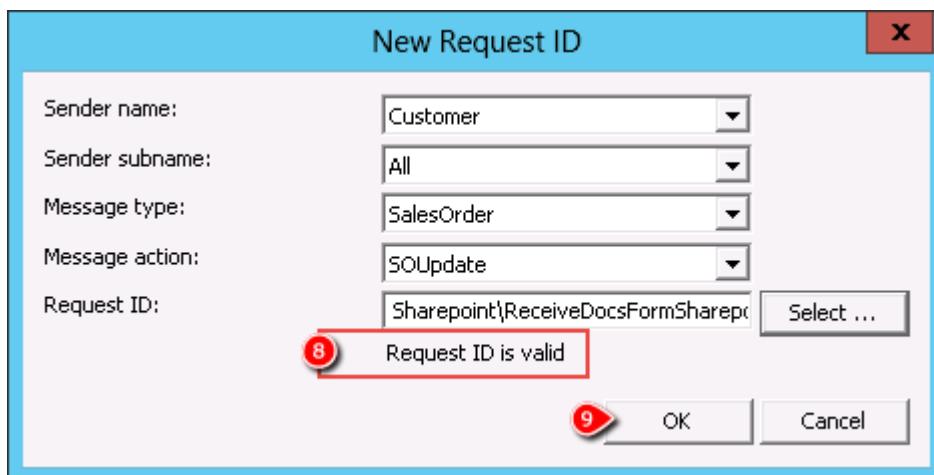
5. The **Request ID** window displays. Clear the **Channels**, **Web Methods**, and **.NET Methods** check boxes to hide these items.



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**. CO
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.

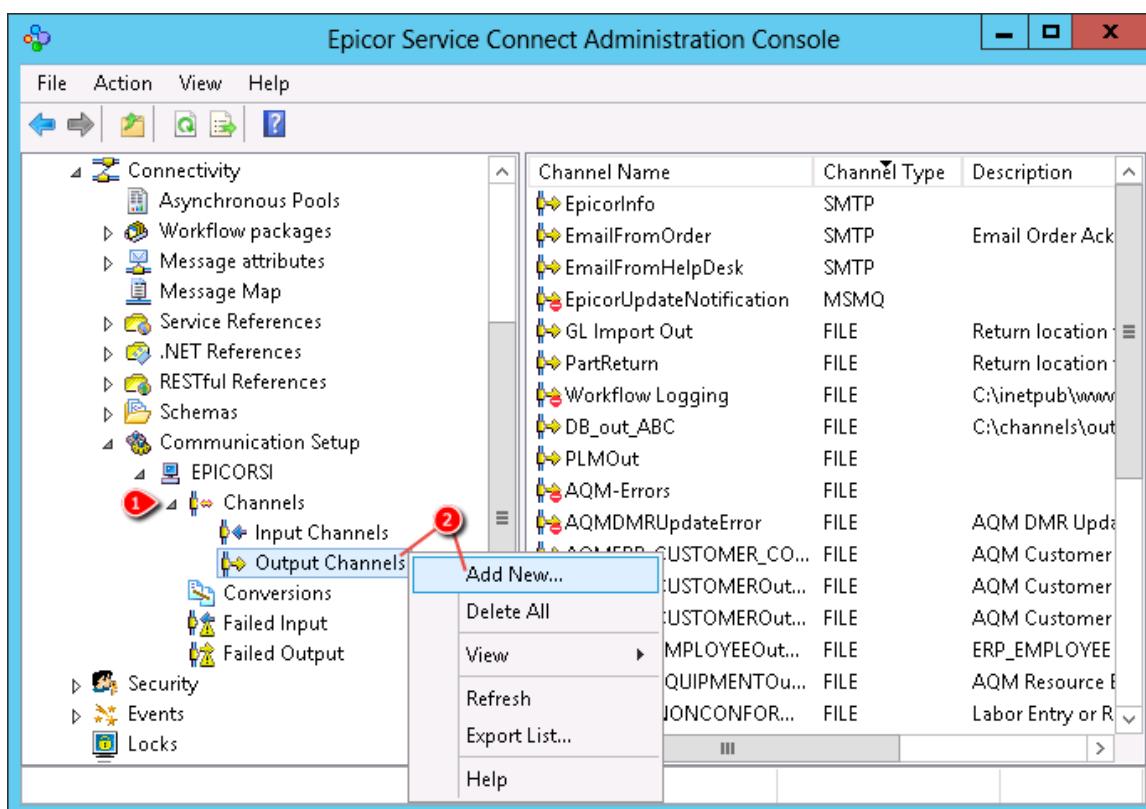
The SharePoint workflow and integration to Service Connect is now complete. When you upload a document to the SharePoint document library, it will be sent to a Service Connect workflow based on the message map set up in the Administration Console.

Add an Output Channel

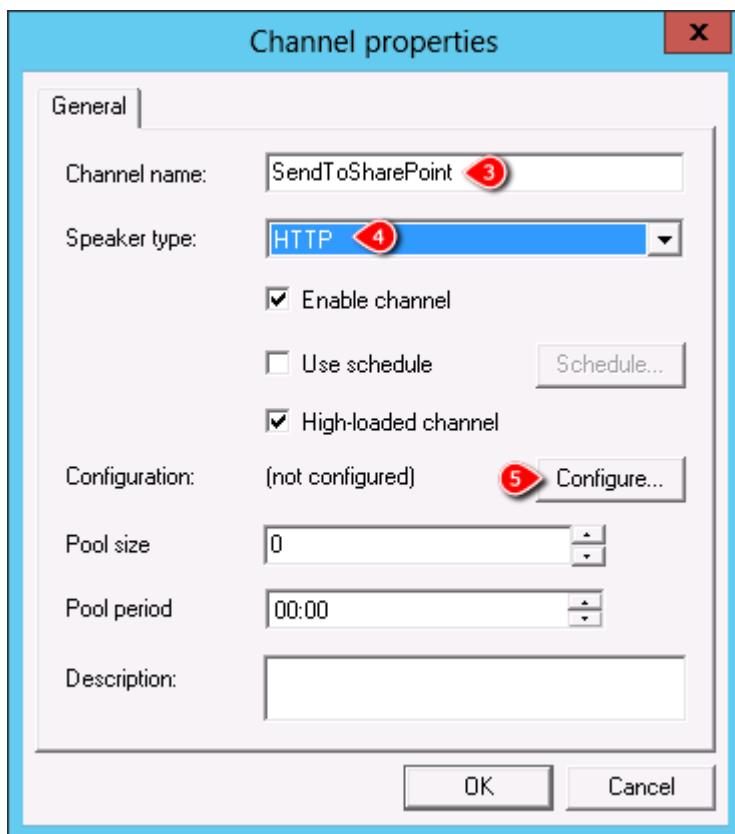
Use an HTTP output channel to send documents to a SharePoint library from a Service Connect workflow.

To add a new output channel:

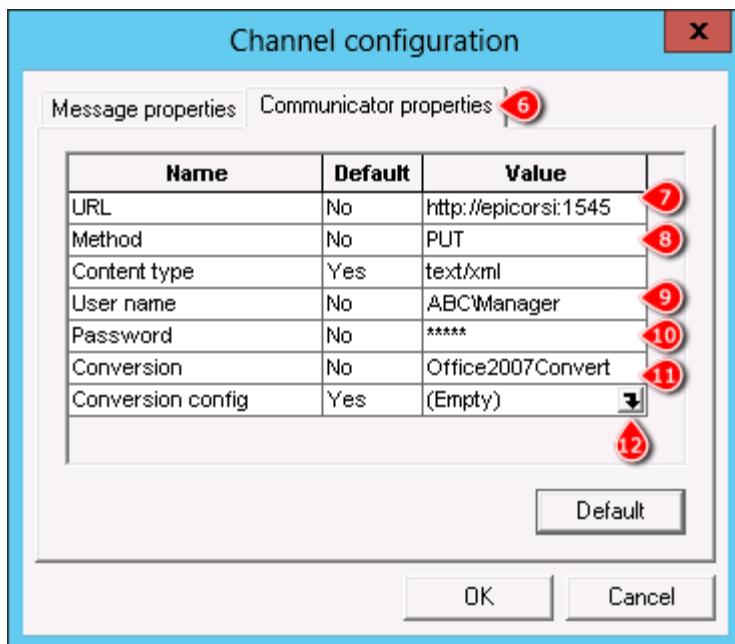
1. In the tree view of the **Epicor Service Connect Administration Console**, expand the **Connectivity > Communication Setup ><Machine Name>> Channels** node.
2. Right-click **Output Channels** and select **Add New...**.



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



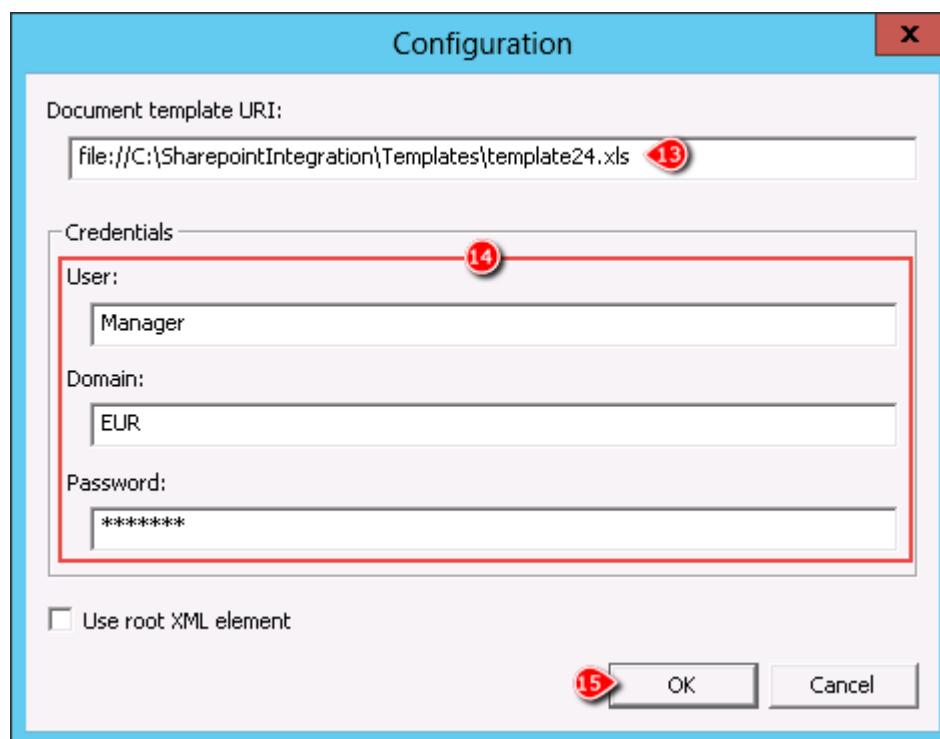
4. Click the **Speaker type** drop-down list and select **HTTP**.
5. Click **Configure**.
6. The **Channel configuration** window displays. Click the **Communicator properties** tab.



- 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.

- In the **Method** field, select **PUT**.
- 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.
- In the **Password** field, enter the account password.
- In the **Conversion** field, select the appropriate conversion for the document you want to send from Service Connect.
- Click the **Conversion config** field, and then click the arrow button that is displayed.
- 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.



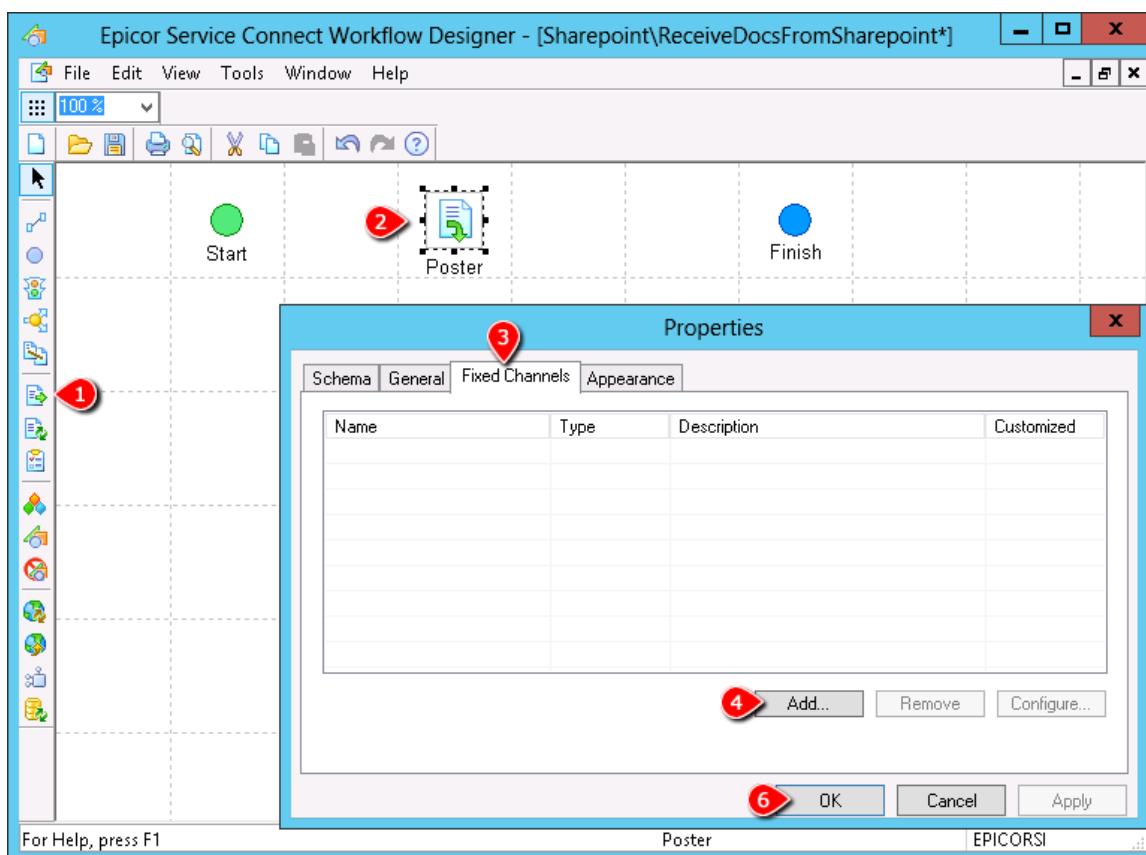
- If the document is on another server, you can enter account credentials to access the file.
- Click **OK** until you exit all dialog boxes.

Use the Output Channel from a Poster

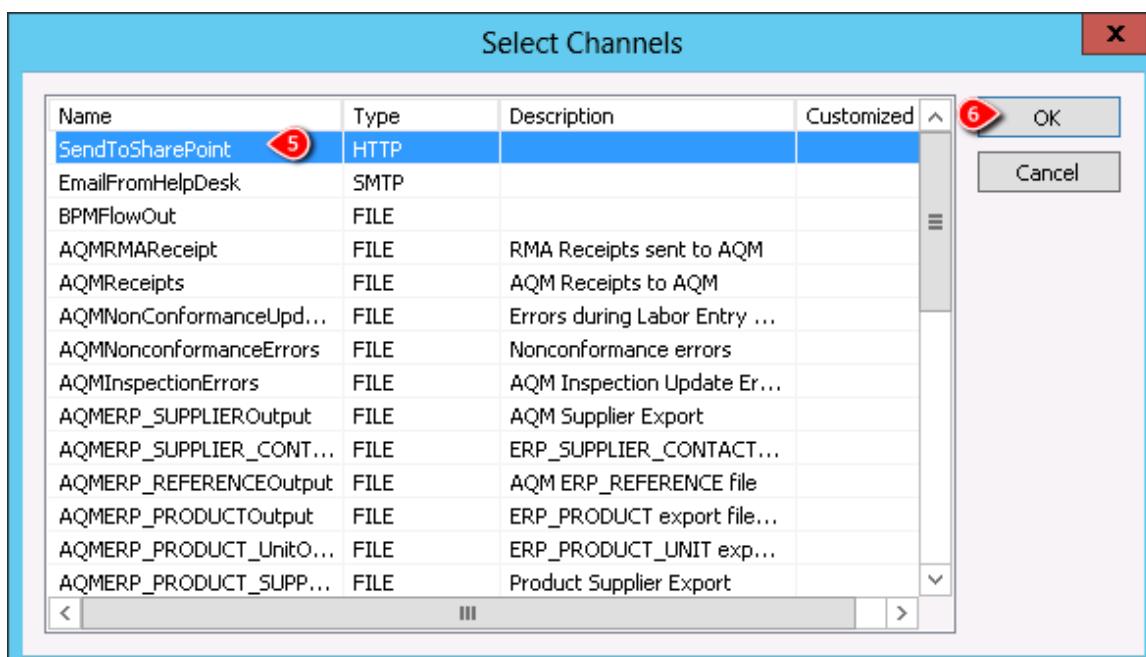
A Poster activity uses the output channel from within the workflow.

To use the output channel:

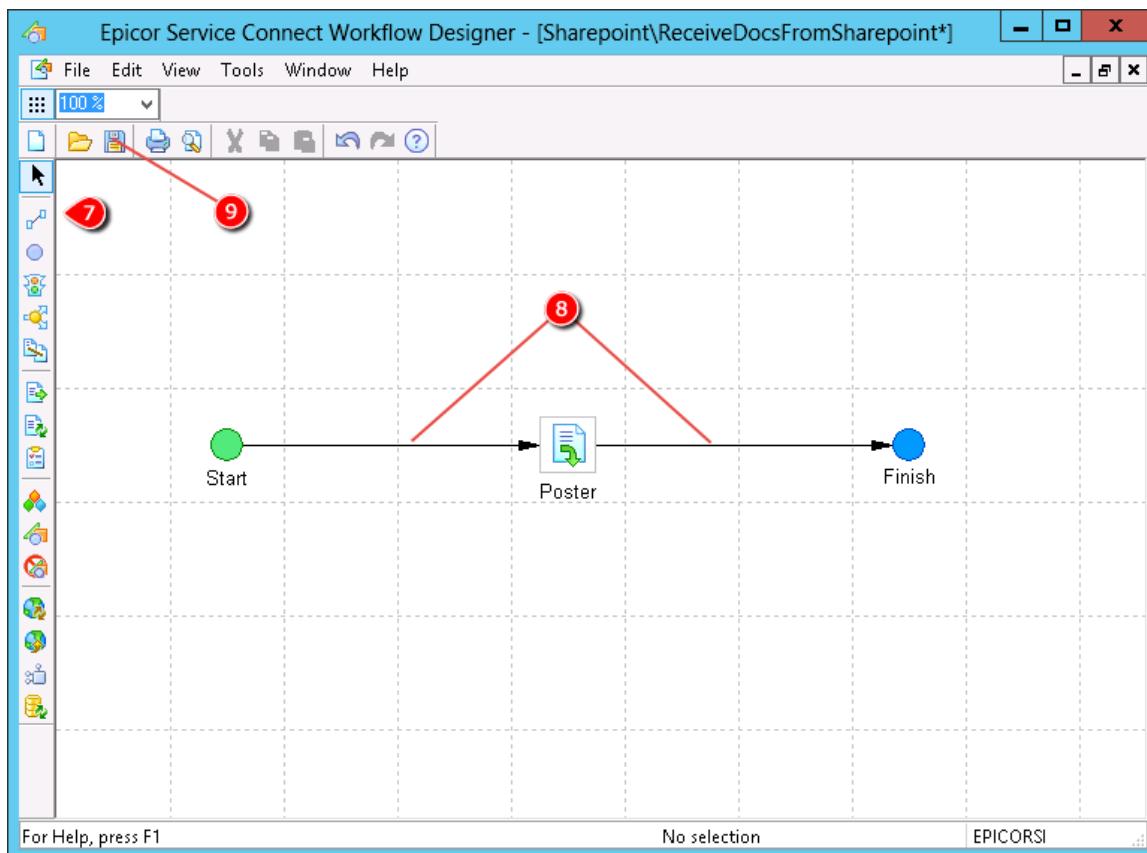
1. In the Workflow Designer, click the **Poster** button on the Items toolbar.



2. Click the design area to add a Poster activity.
3. The **Properties** window displays. Click the **Fixed Channels** tab.
4. Click **Add...**.



5. Select the output channel created for SharePoint.
6. Click **OK** until you exit all dialog boxes.
7. Click the **Connection** tool on the **Items** toolbar.



8. Connect the **Start** to the **Poster** and the **Poster** to the **Finish**.
9. Click **Save**.

Tasks

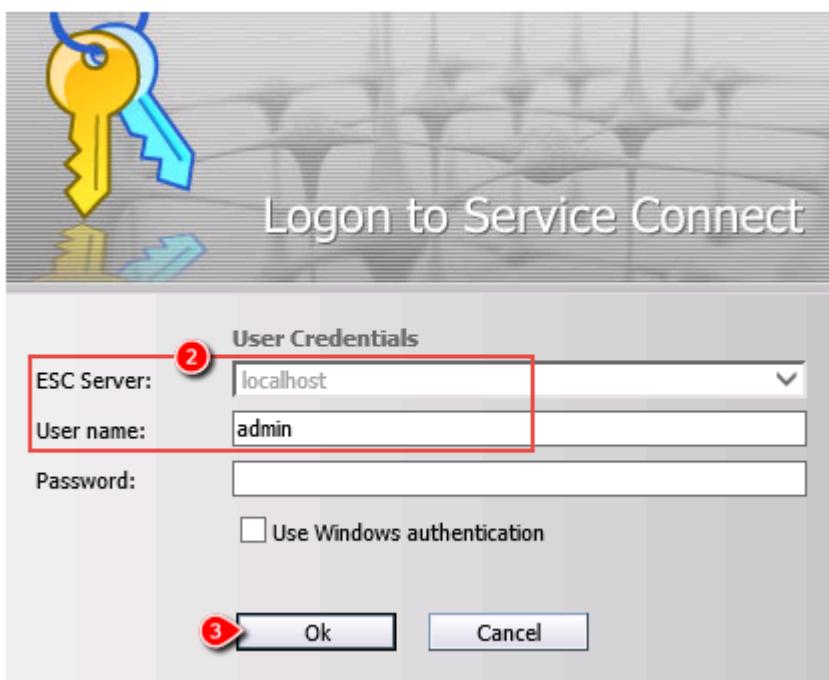
Tasks are workflow activities that can halt a workflow under certain circumstances and then send data to the Task Monitor web site where an authorized user can take action and possibly choose how to route the information. Two typical uses of the Task activity are in authorization scenarios (for example,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

Process a Task

To process a task:

1. Log into the Task Monitor using the menu path **Start > All Programs > Epicor Software > Epicor Service Connect > Task Monitor**.

2. On 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.

3. Click **OK**.

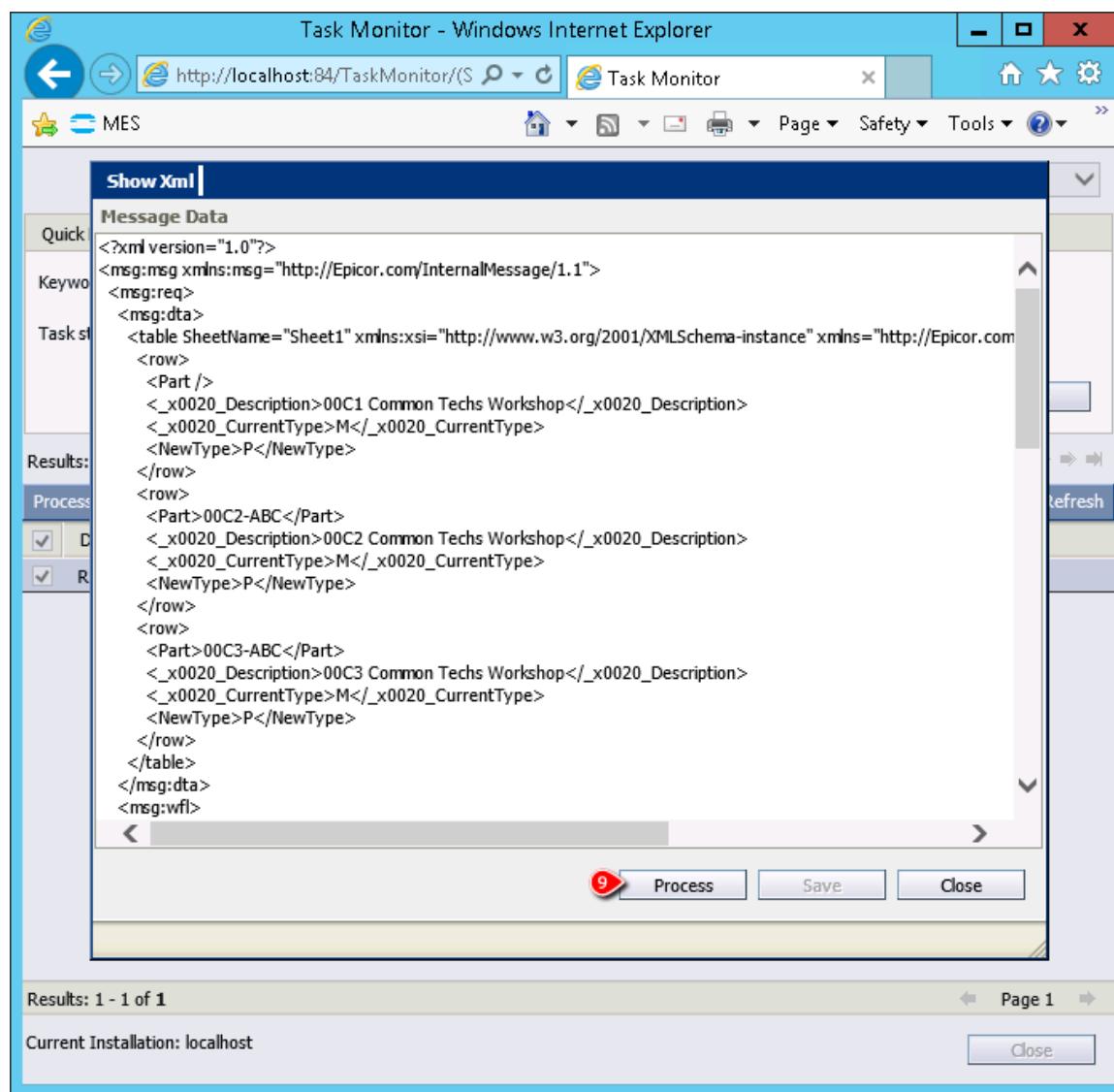
4. The **Task Monitor** displays all tasks.

The screenshot shows the 'Task Monitor - Windows Internet Explorer' window. At the top, there are navigation buttons (Back, Forward, Stop, Refresh), a search bar, and a title bar. Below the title bar, the URL is http://localhost:84/TaskMonitor/ and the page title is 'Task Monitor'. The main interface has a header with 'View: Active Tasks' and 'Schema: [All]'. A 'Quick Filter' section contains a 'Keyword:' input field (circled in red with number 5) and a checked 'All Users' checkbox. Below it is a 'Task status:' dropdown set to 'All Tasks' (circled in red with number 4). To the right of the filter section is an 'Apply' button (circled in red with number 6). The results grid shows one task: 'Resolve Errors' (checkbox selected, circled in red with number 7), assigned on 3/17/2017 at 11:52:56 AM, with an expire time of 'Never'. The grid has columns for Description, Assignment Time, Expire Time, and Short Error. Below the grid, the message 'Results: 1 - 1 of 1' is displayed. At the bottom left, it says 'Current Installation: localhost'. On the right side, there are 'Pages' navigation buttons and a 'Close' button. A red circle with number 8 points to the top-left corner of the results grid area.

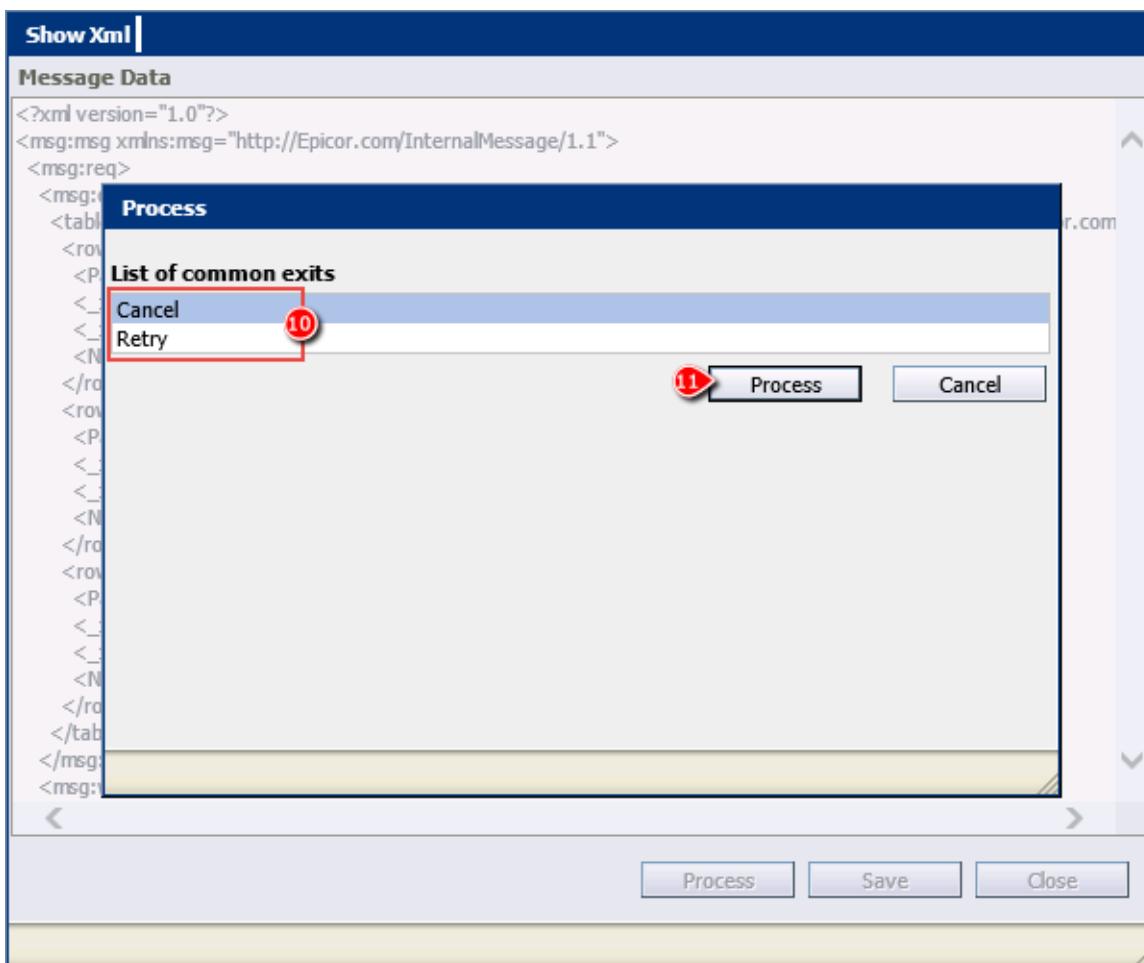
Description	Assignment Time	Expire Time	Short Error
<input checked="" type="checkbox"/> Resolve Errors	3/17/2017 11:52:56 AM	Never	

5. Use the **Quick Filter** options at the top of the form to filter the list of tasks
6. Click **Apply**.
7. In the **Results** grid, select the check box next to the tasks you to process.
8. Click **Show XML** to edit or process the XML message sent to the task.

9. 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**.



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



11. 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.

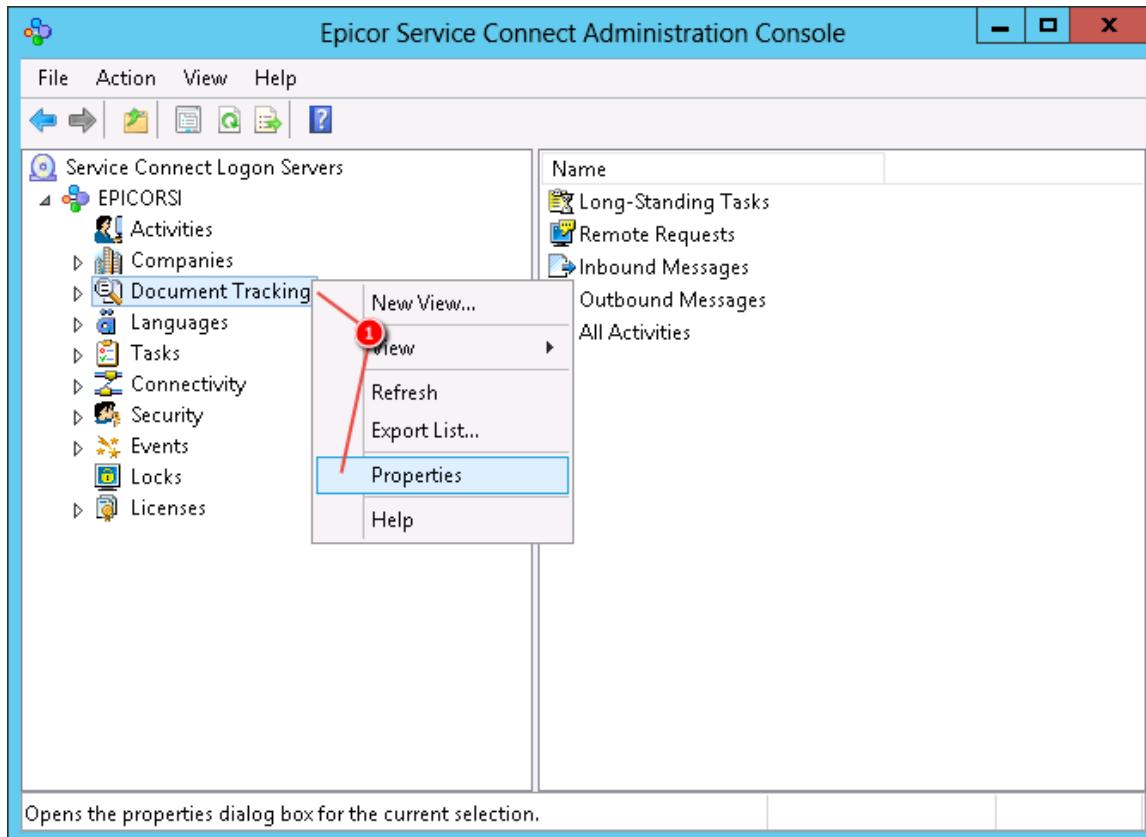
Set Up Document Tracking

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.

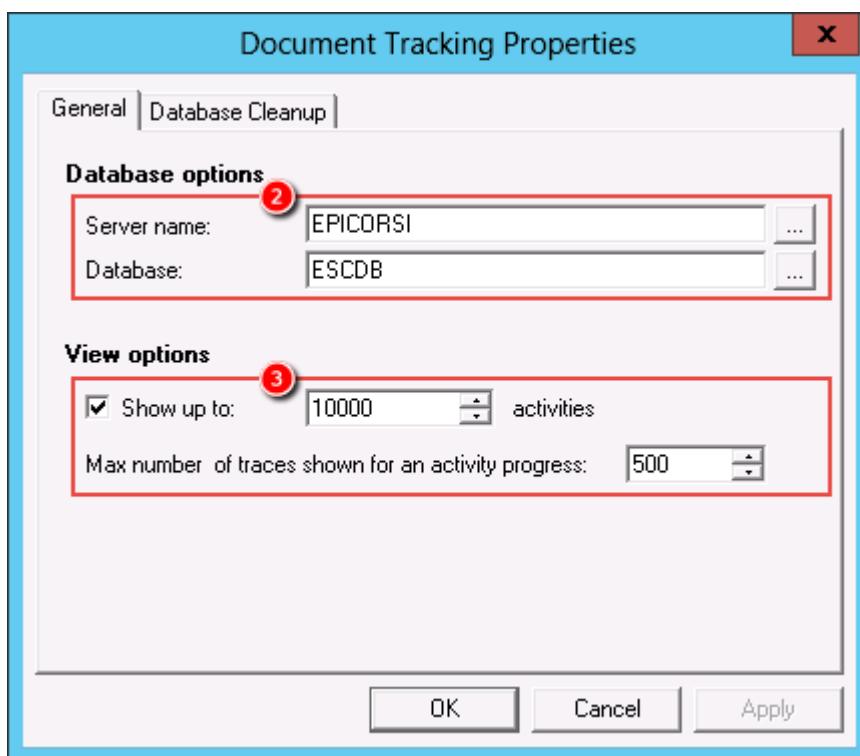
To set up the Document Tracking properties :

1. In the **Epicor Service Connect Administration Console**, right-click the **Document Tracking** node and select **Properties**.

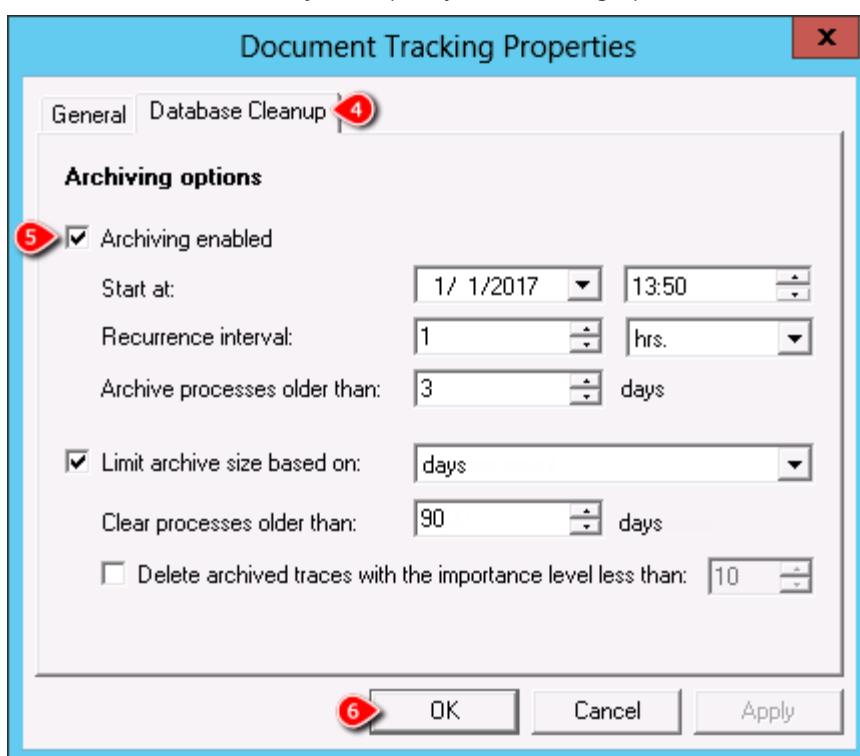


The **Document Tracking Properties** dialog box is displayed.

2. On the **General** tab, in the **Database options** group, select the server name and database.



3. In the **View options** group, specify the maximum number of activities to be displayed, and the maximum number of traces, shown for every activity progress.
4. On the **Database Cleanup** tab, specify the archiving options.



- Select the **Archiving enabled** check box, if you want to enable archiving.

Only completed process traces are moved to the archived tables.

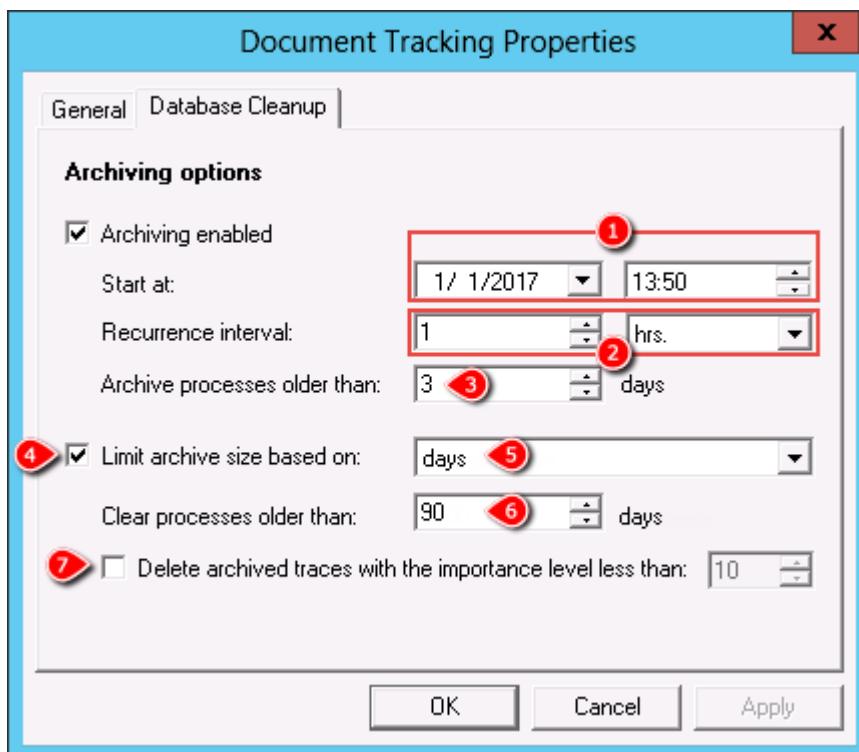
Refer to the **Archiving Options** section below for the detailed description of the options you can use to control your archive size.

- Click **OK**.

Archiving Options

On the **Database Cleanup** tab, after you enabled archiving, you can adjust a number of archiving options that can help you control the log size and contents.

- In the **Start at** fields, specify the date and time when the archiving should start.



- In the **Recurrence interval** fields, specify the time period counted from the latest archiving. When this time period elapses, subsequent archiving procedure starts automatically.

Standard views in Document Tracking like **All activities** show traces from both the main Document Tracking table and the archiving table. So if you do not want to see archived traces, use the **Limit archive size based on** setting to remove the old traces. Another option is to create a new Custom View and filter activities there to show traces only from the main Document Tracking table. To do it, set a filter, then on the Advanced tab, define an additional criteria by the **Search in** field, select the **Equal to** condition, and select **WorkingTables** condition value.

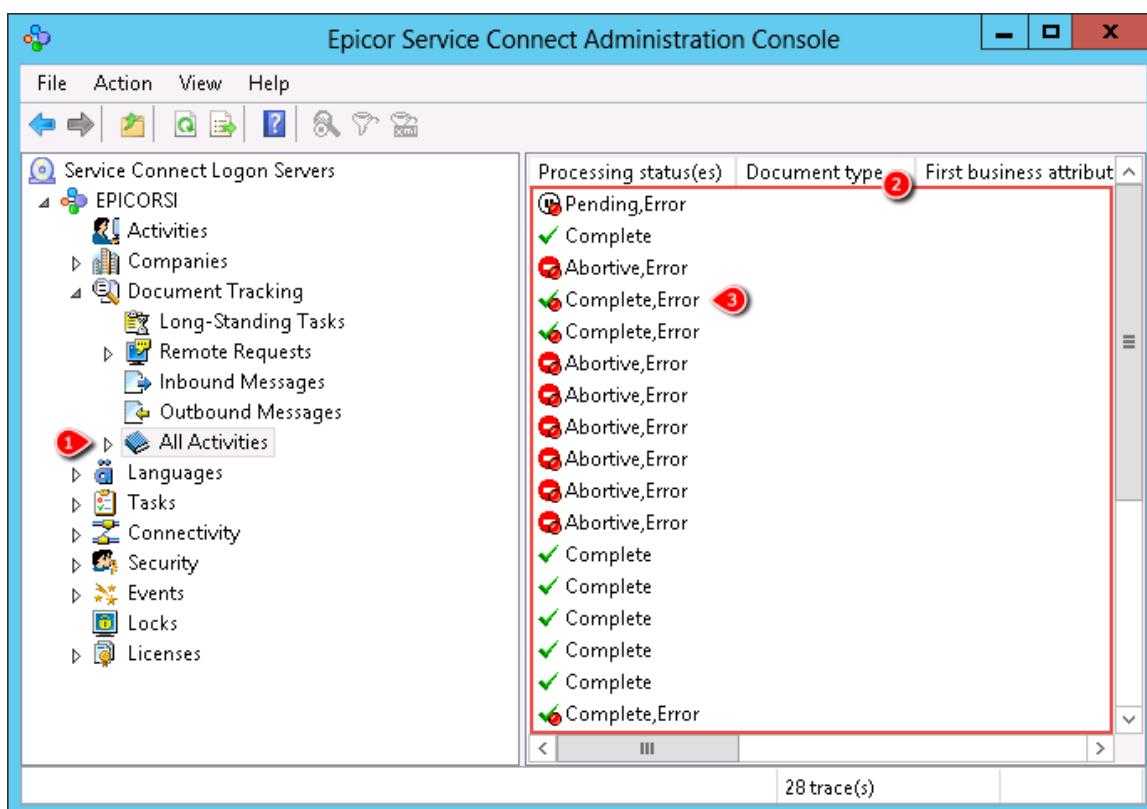
Use the next field to set the archiving condition based on the time period.

3. In the **Archive processes older than** field, enter the number of days.
All the processes older than the specified number of days will be archived.
4. Select the **Limit archive size based on** check box to be able to specify the time period the archive should be stored.
5. Select **days**.
6. In the **Clear processes older than** field, specify the time period, during which the process should be stored.
7. Select the **Delete archived traces with the importance level less than** check box to enable trace deletion, and specify the importance level threshold value.
The default value for this field is 10. The processes with the importance level less than this value are deleted. By default, these archived traces are not deleted.

View Activities

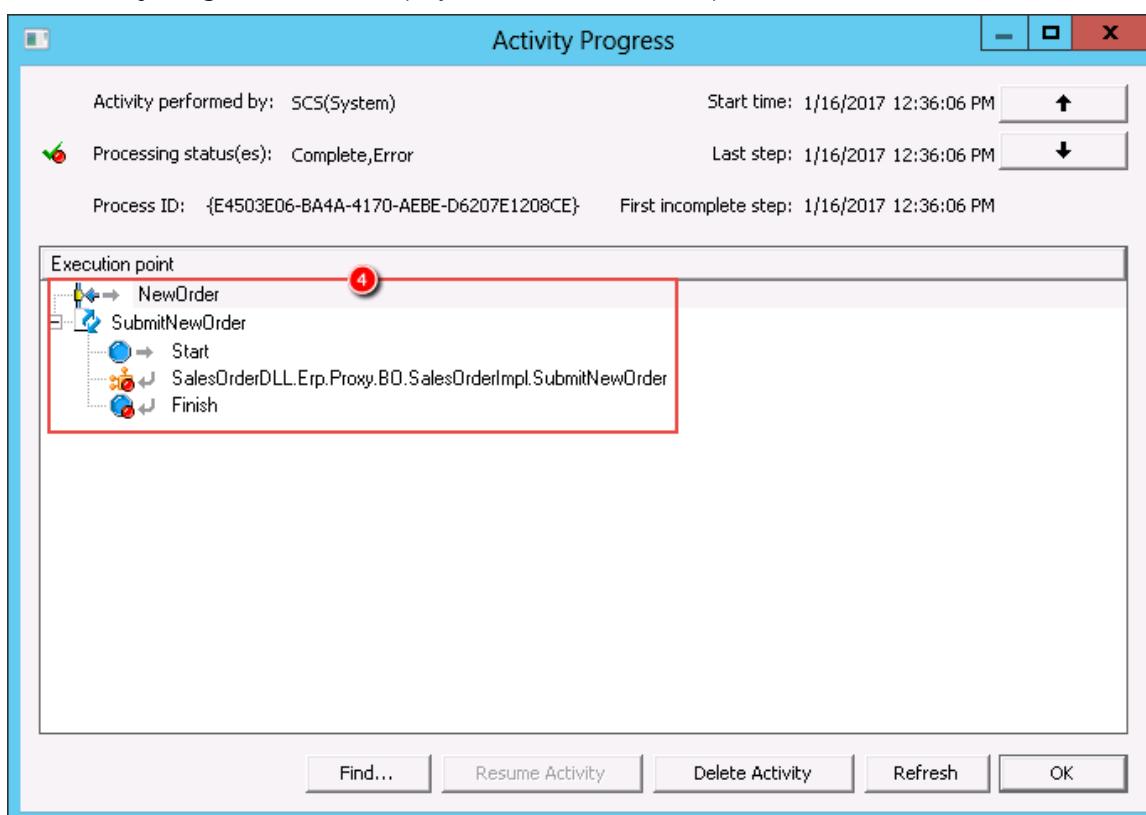
To view activities:

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



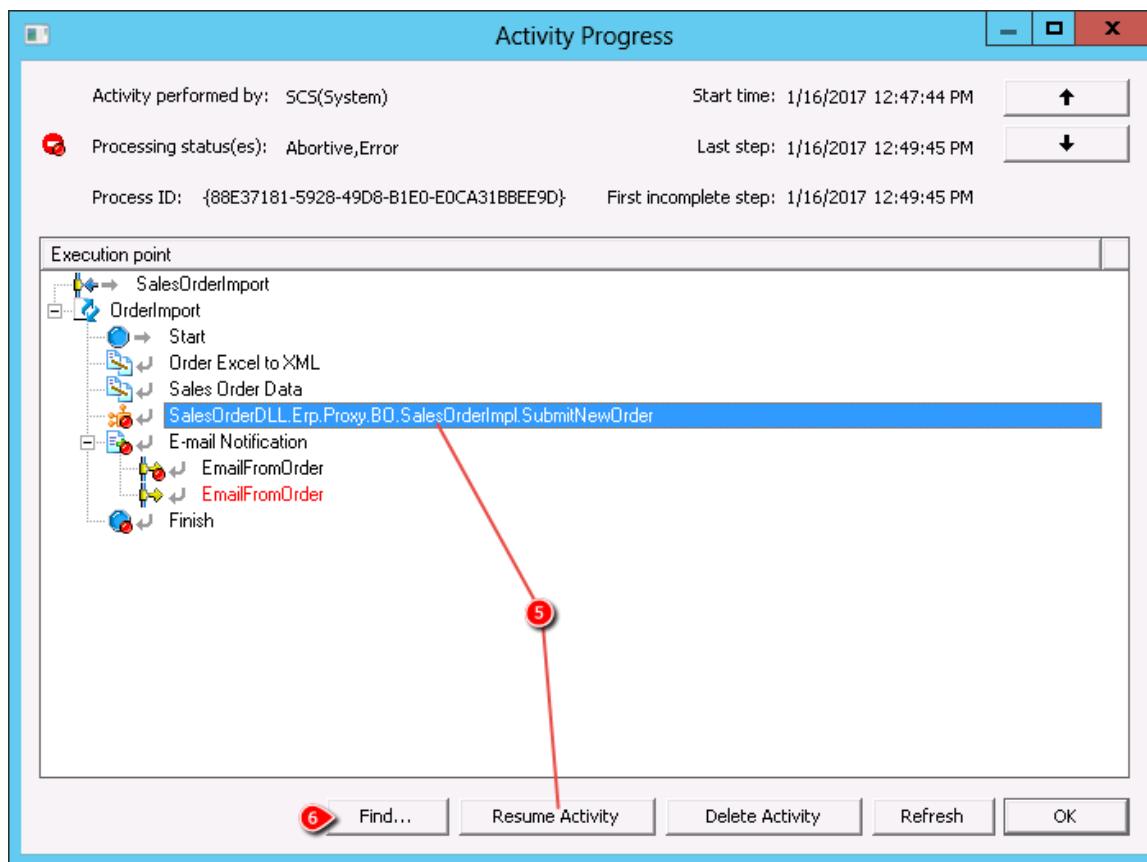
2. In the right pane, view all activities displayed in a list.
3. Double-click an activity to view the details.

4. The **Activity Progress** window displays, with each execution point listed.



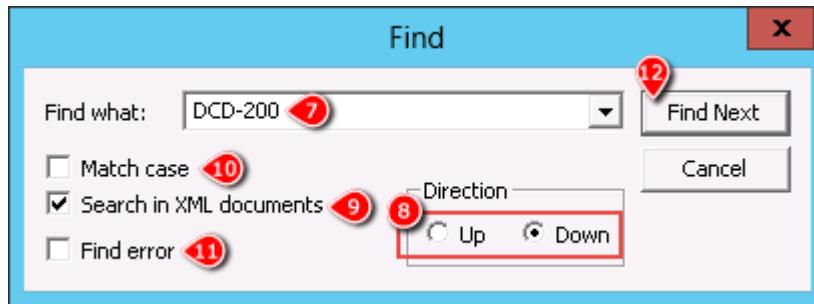
Incomplete activities display in red.

5. 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.



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

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



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

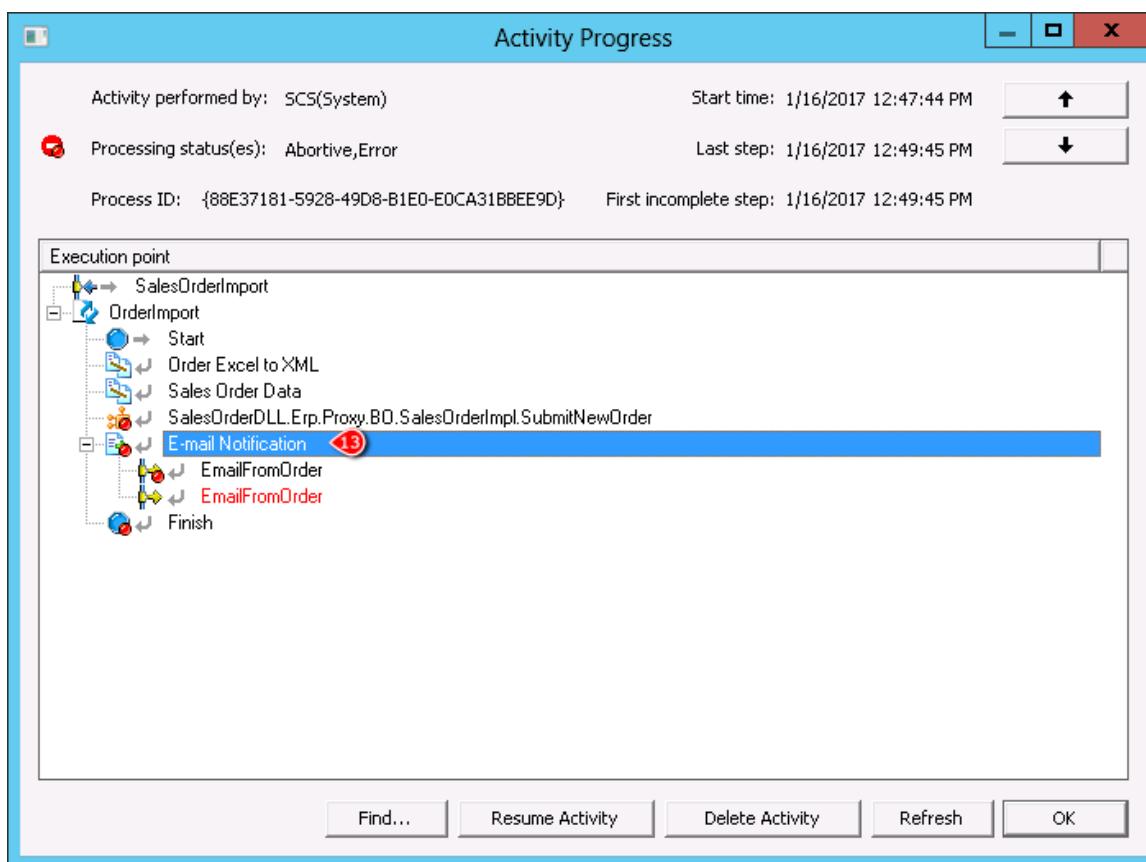
9. You can include the xml documents in the search by selecting the **Search in XML documents** check box.

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

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

12. Click **Find Next**.

13. In the **Activity Progress** window, double-click the highlighted execution point to view the details

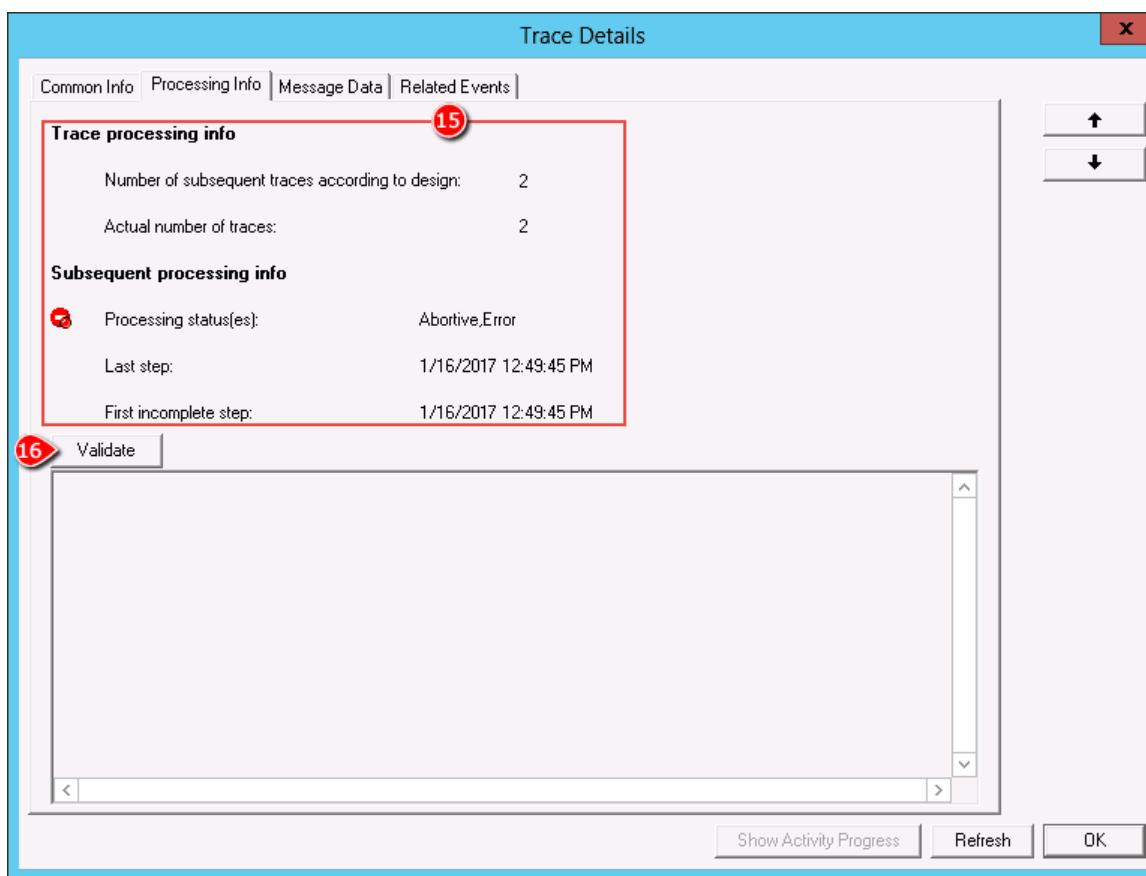


14. Detailed data displays on four tabs. Click the **Common Info** tab to display the common processed document attributes.

The screenshot shows the 'Trace Details' window with a red '14' badge in the top-left corner. The window has a title bar 'Trace Details' and a close button 'X'. Below the title bar is a tab bar with 'Common Info' (highlighted in blue), 'Processing Info', 'Message Data', and 'Related Events'. On the left, there's a toolbar with an up arrow, a down arrow, and a refresh button. The main area contains a table with two columns: 'Field' and 'Value'. The 'Field' column uses a tree-view expand/collapse icon. The 'Value' column contains XML-style paths and their corresponding values. The table is scrollable, indicated by a vertical scrollbar on the right. At the bottom are buttons for 'Show Activity Progress', 'Refresh', and 'OK'.

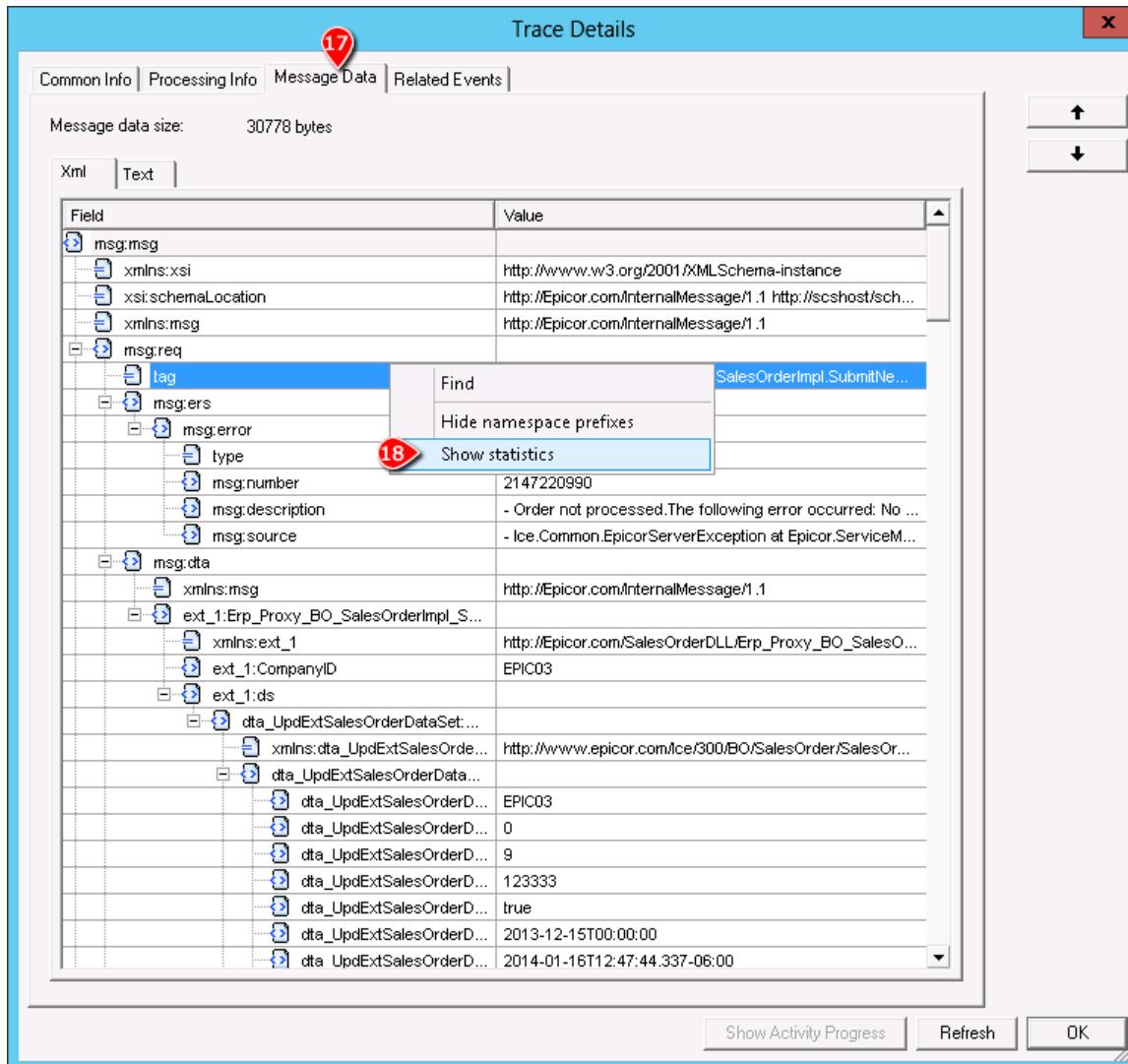
Field	Value
msg:msg	
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://Epicor.com/InternalMessage/1.1 http://scshost/sch...
xmlns:msg	http://Epicor.com/InternalMessage/1.1
msg:req	
tag	SalesOrderDLL.Erp.Proxy.BO.SalesOrderImpl.SubmitNe...
msg:ers	
msg:error	
type	unexpected error
msg:number	2147220990
msg:description	- Order not processed.The following error occurred: No ...
msg:source	- Ice.Common.EpicorServerException at Epicor.ServiceM...
msg:dta	
xmlns:msg	http://Epicor.com/InternalMessage/1.1
ext_1:Erp_Proxy_BO_SalesOrderImpl_S...	
xmlns:ext_1	http://Epicor.com/SalesOrderDLL/Erp_Proxy_BO_SalesO...
ext_1:CompanyID	EPIC03
ext_1:ds	
dta_UpdExtSalesOrderDataSet:...	
xmlns:dta_UpdExtSalesOrde...	http://www.epicor.com/Ice/300/BO/SalesOrder/SalesOr...
dta_UpdExtSalesOrderData...	
dta_UpdExtSalesOrderD...	EPIC03
dta_UpdExtSalesOrderD...	0
dta_UpdExtSalesOrderD...	9
dta_UpdExtSalesOrderD...	123333
dta_UpdExtSalesOrderD...	true
dta_UpdExtSalesOrderD...	2013-12-15T00:00:00
dta_UpdExtSalesOrderD...	2014-01-16T12:47:44.337-06:00

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



16. 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.

17. 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.

18. To view the statistic information, right-click anywhere on the Xml tab and select **Show statistics**.

The screenshot shows the 'Trace Details' window with the 'Message Data' tab selected. At the top, it displays 'Message data size: 30778 bytes'. Below this is a tabs section with 'Xml' and 'Text' (which is currently selected). A red callout bubble with the number '18' points to the 'Statistics' column header in the main table.

Field	Value	Statistics
msg:msg	http://www.w3.org/2001/XMLSchema#anyType	2 children; 30666 B
xmlns:xsi	http://www.w3.org/2001/XMLSchema#anyURI	
xsi:schemaLocation	http://Epicor.com/InternalMessage/M.1.xsd	
xmlns:msg	http://Epicor.com/InternalMessage/M.1	
msg:req		4 children; 29778 B
tag	SalesOrderDLL.Erp.Proxy.BO.Sales...	
msg:ers		1 children; 982 B
msg:error		3 children; 916 B
type	unexpected error	
msg:number	2147220990	0 children; 170 B
msg:description	- Order not processed.The followin...	0 children; 330 B
msg:source	- Ice.Common.EpicorServerException...	0 children; 446 B
msg:dta		1 children; 14550 B
xmlns:msg	http://Epicor.com/InternalMessage/M.1	
ext_1:Erp_Proxy_B...		2 children; 14412 B
xmlns:ext_1	http://Epicor.com/SalesOrderDLL/Erp...	
ext_1:CompanyID	EPIC03	0 children; 276 B
ext_1:ds		1 children; 14096 B
dta_UpdExt...		5 children; 13860 B
xmlns:d...	http://www.epicor.com/Ice/300/BO/...	

At the bottom of the window are buttons for 'Show Activity Progress', 'Refresh', and 'OK'.

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.

19. 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 development.

The screenshot shows the 'Trace Details' window with the 'Related Events' tab selected. There are 19 events listed in the table below. The columns are 'Type' and 'Category'. The 'Type' column shows various log levels: ERROR, INFORMATION, and UNKNOWN. The 'Category' column shows all entries as 'System'. The table includes standard grid controls like sorting arrows and a scroll bar.

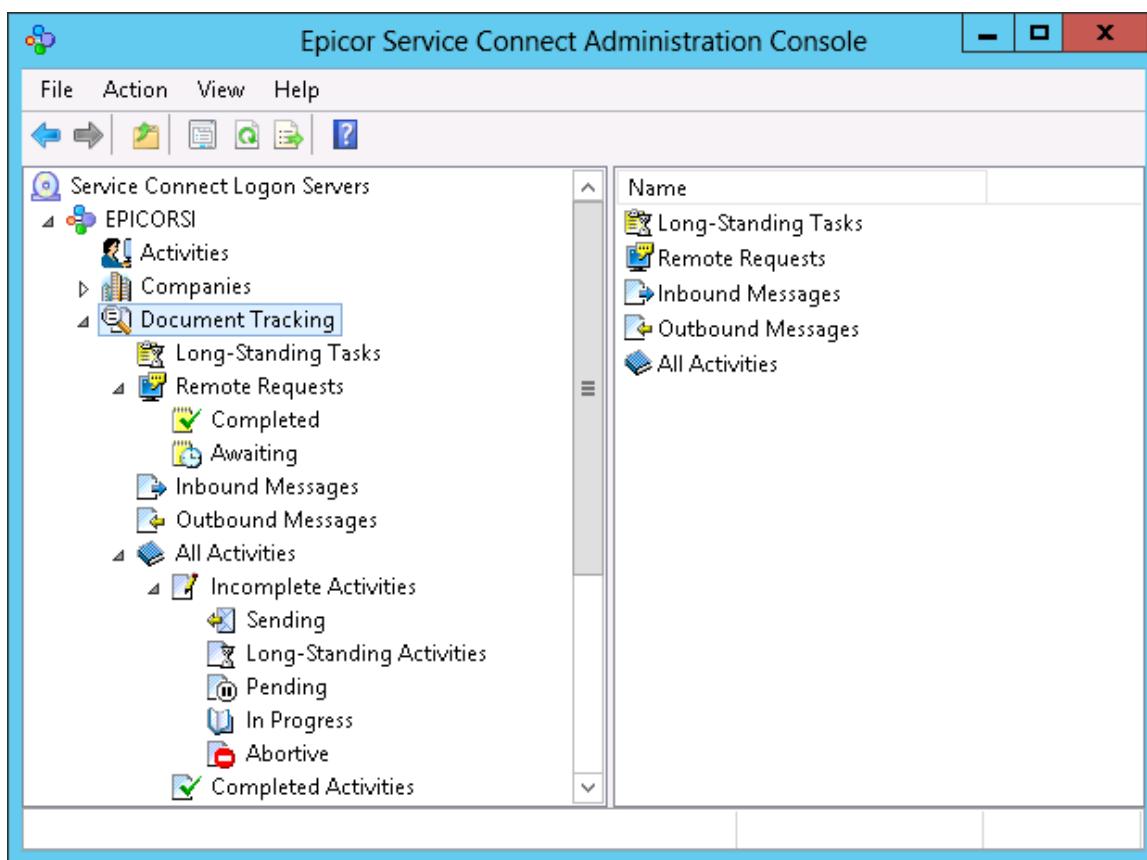
Type	Category
UNKNOWN	System

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.

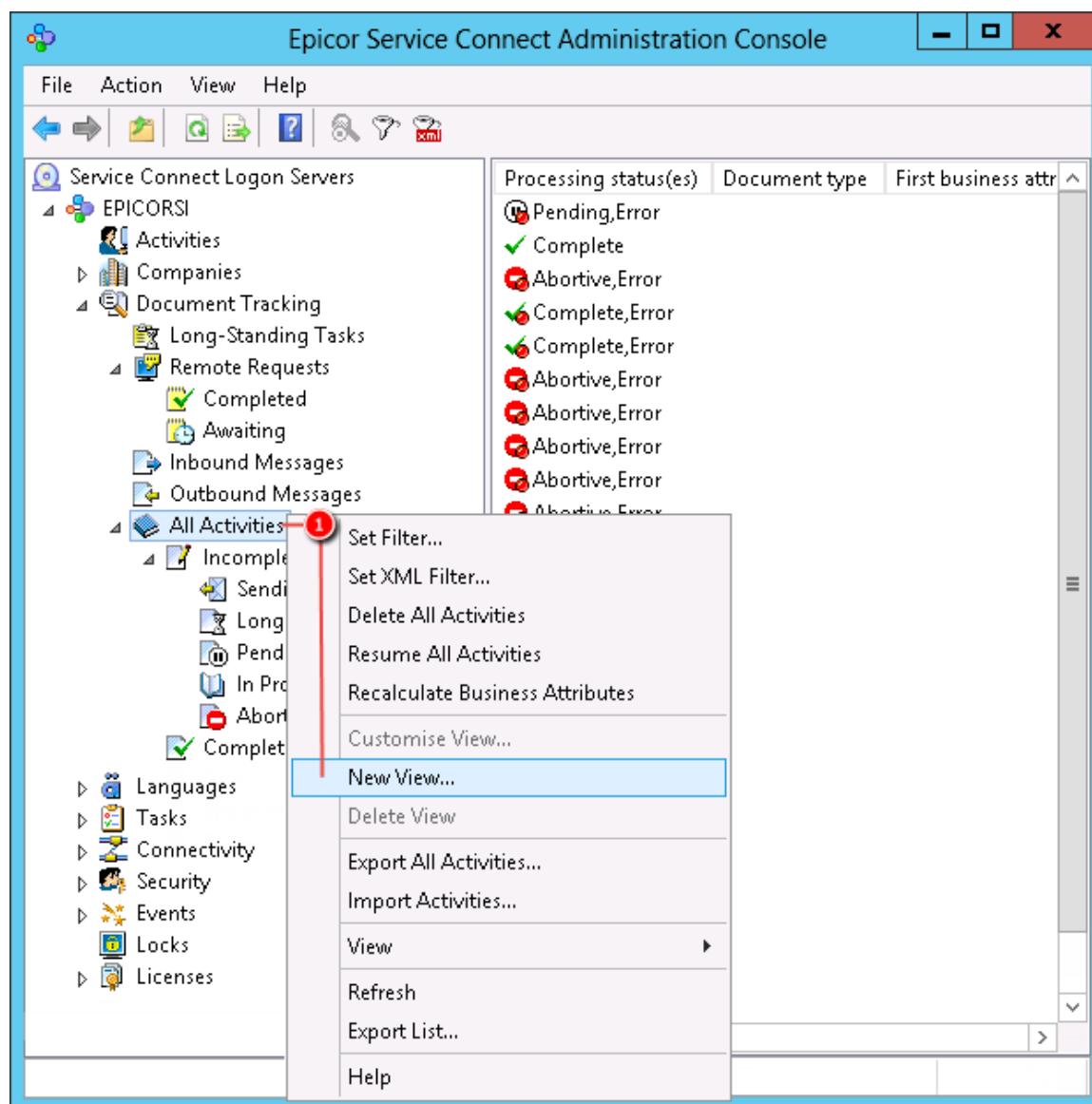


Add a New View

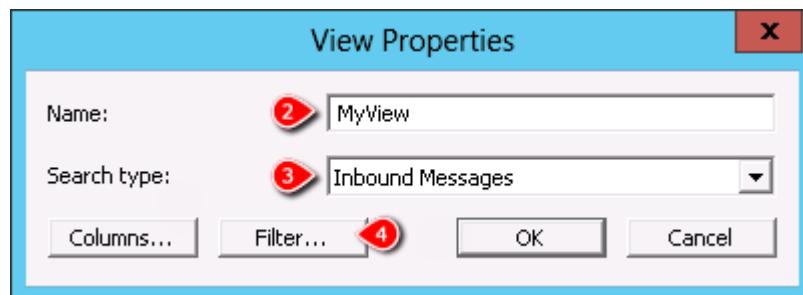
To create a new view:

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.

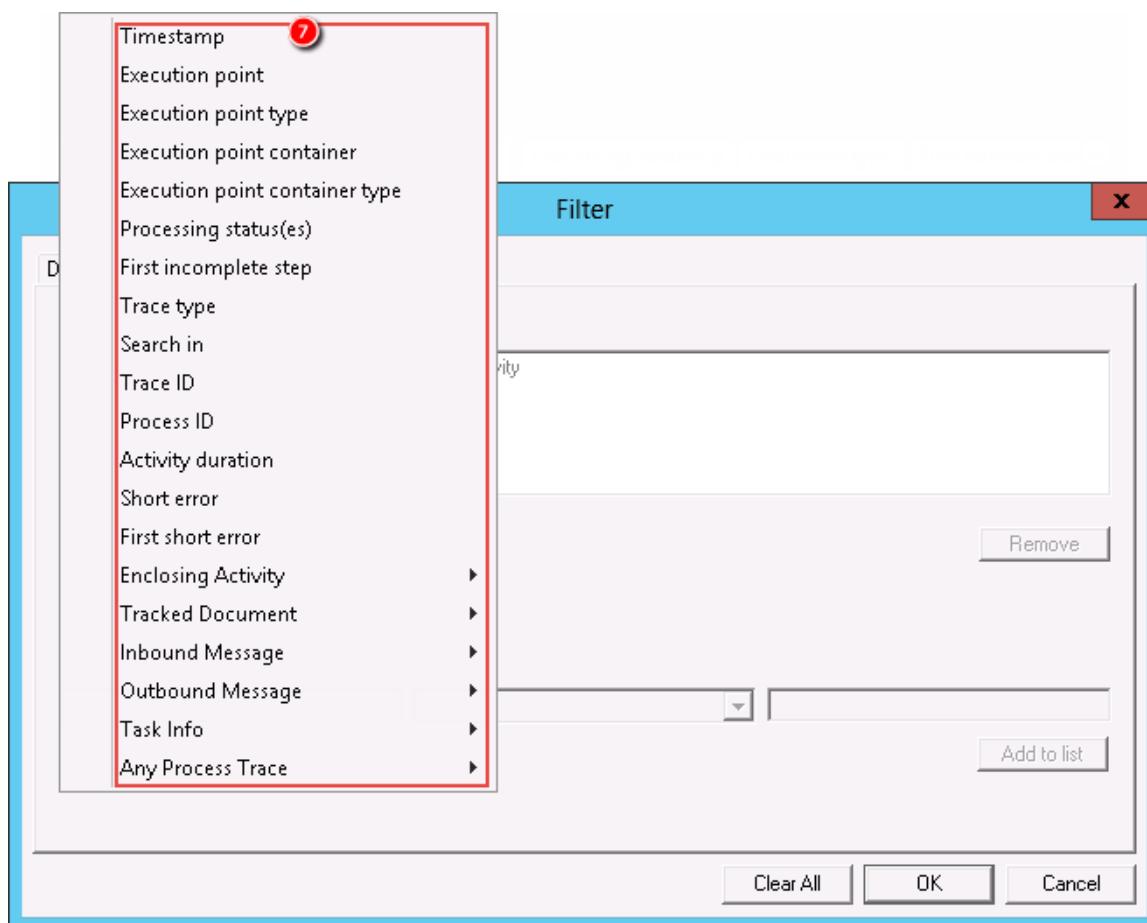
4. Click **Filter**.

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



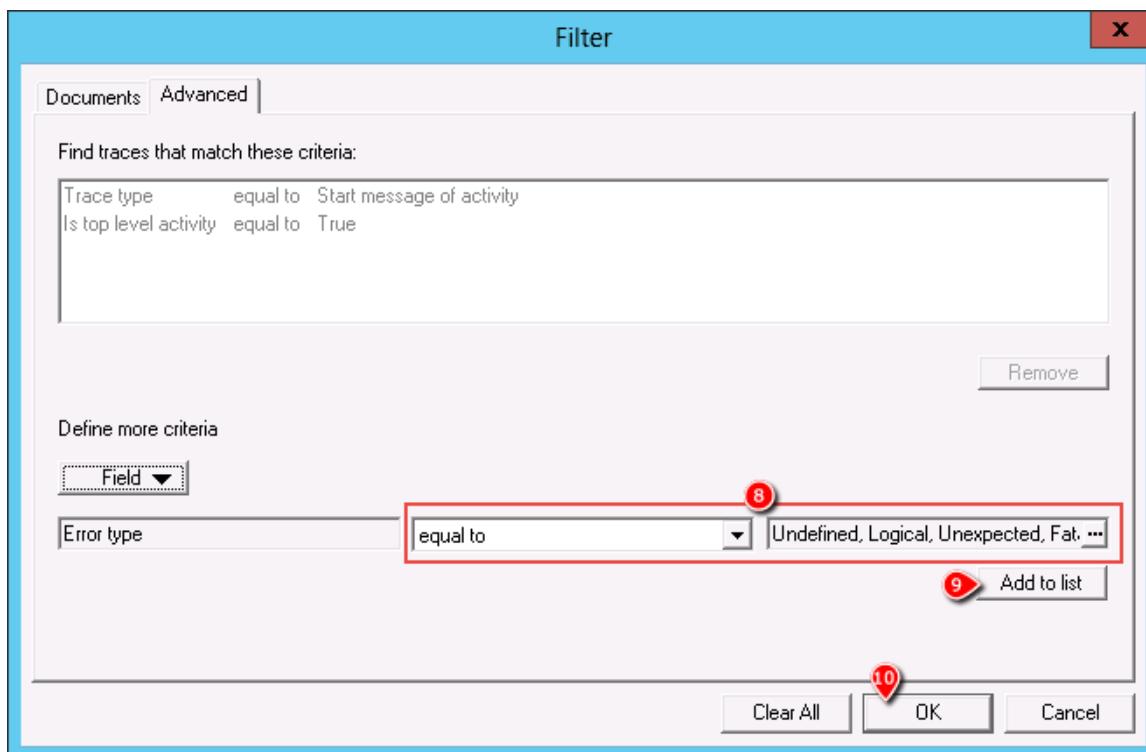
6. Click **Field**.

7. From the list of available fields, select a message field to use in a criteria statement.

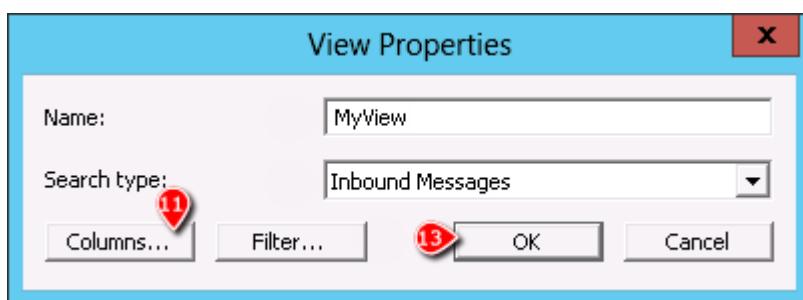


To view the list of message attributes you can use to set up a filter for the Document Tracking views, see Epicor Service Connect Help.

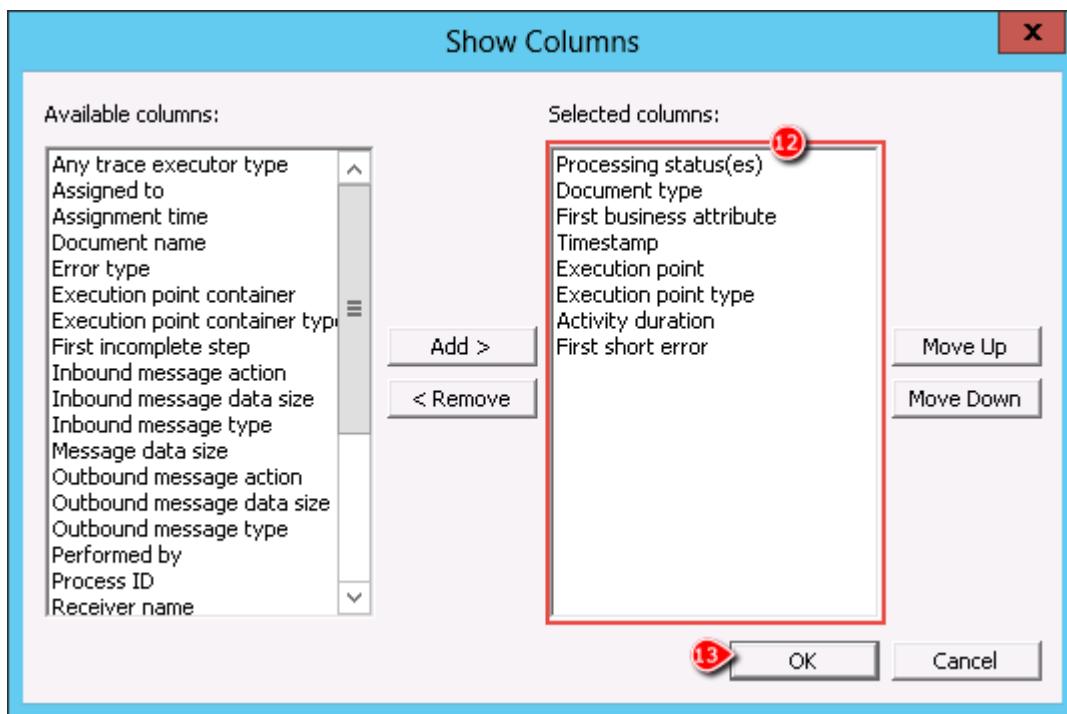
8. Enter filter criteria to complete your statement.



9. Click **Add to list**.
10. Click **OK**.
11. In the **View Properties** window, click **Columns**.



12. 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.

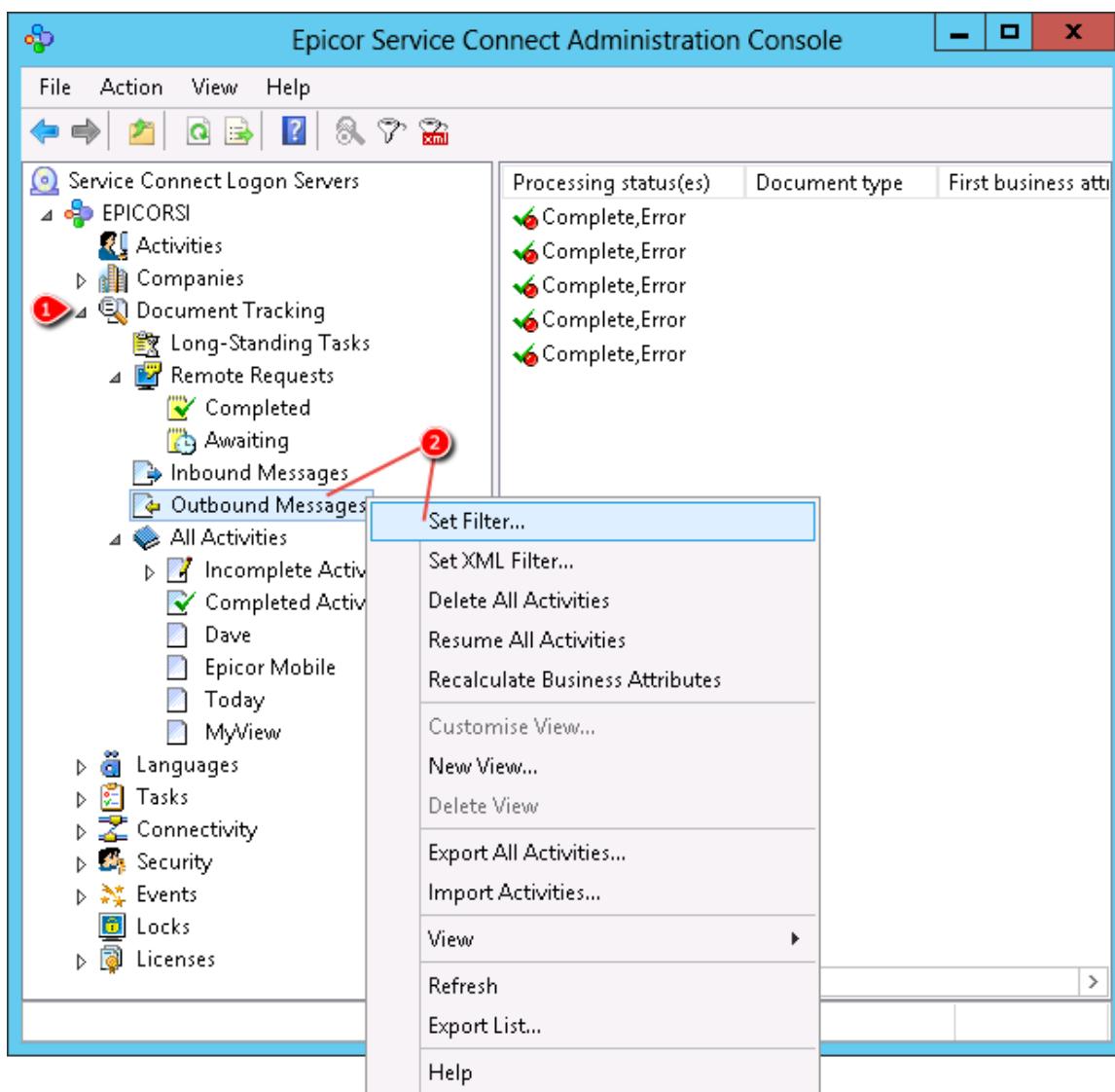
13. Click **OK** until you exit all dialog boxes.

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

Filter an Existing View

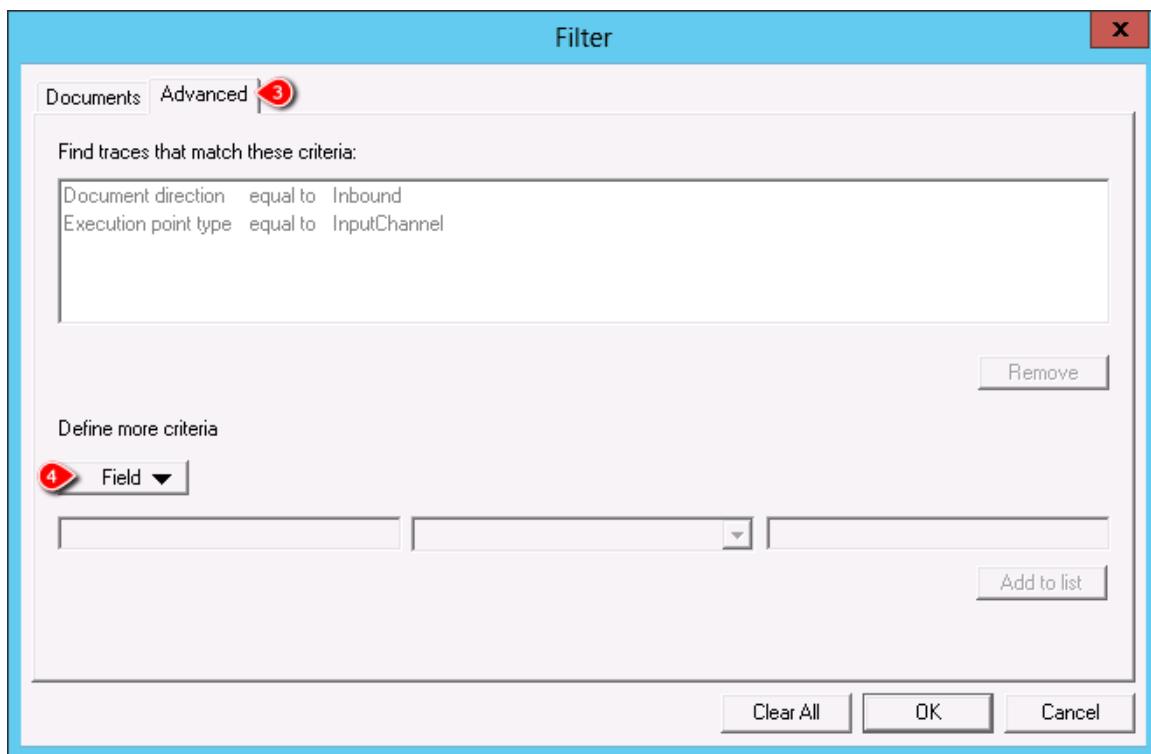
To change the filter for an existing view:

1. In the **tree view**, expand the **Document Tracking** node.



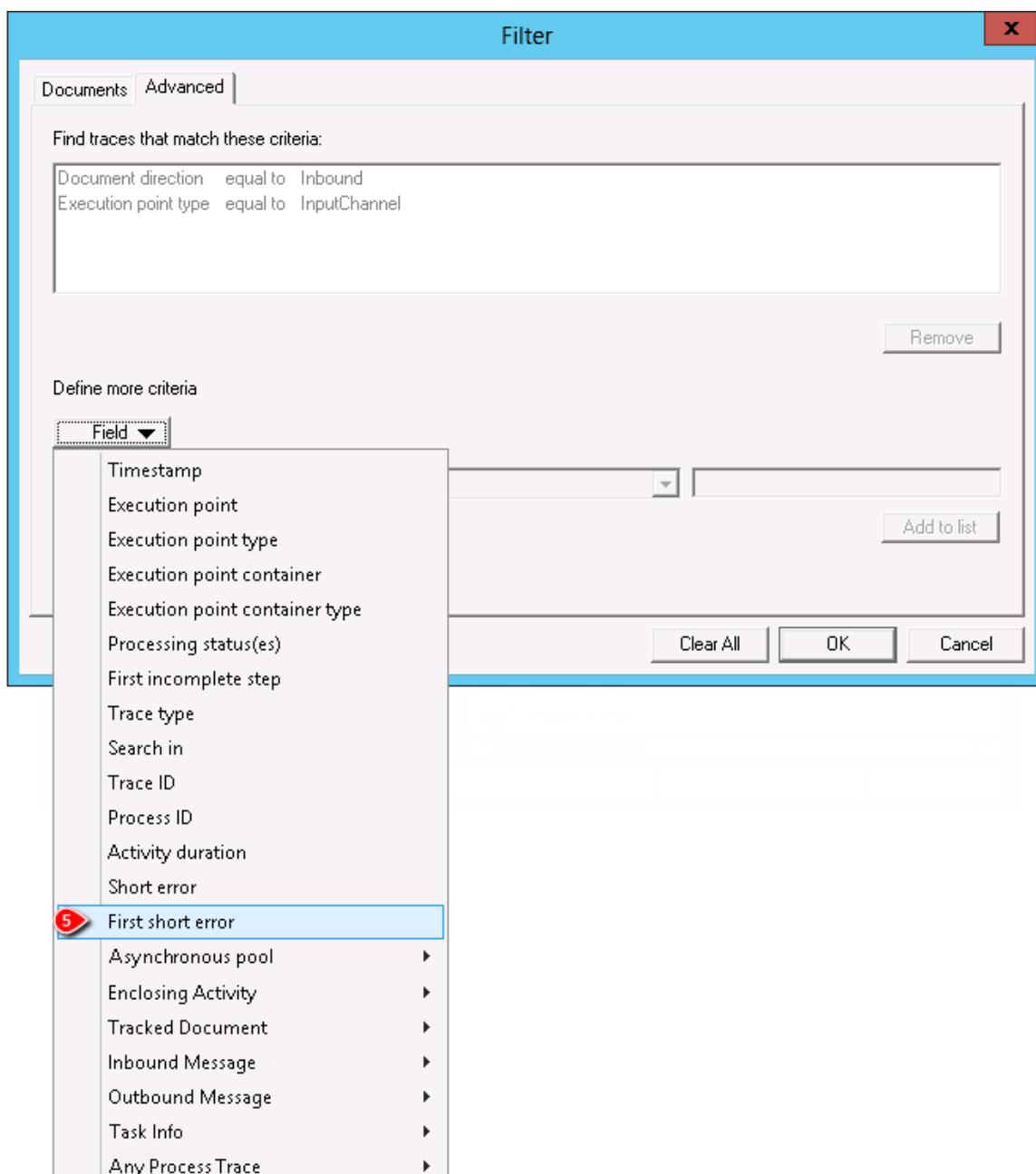
2. Right-click the view you want to use as your starting point and select **Set Filter**.

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

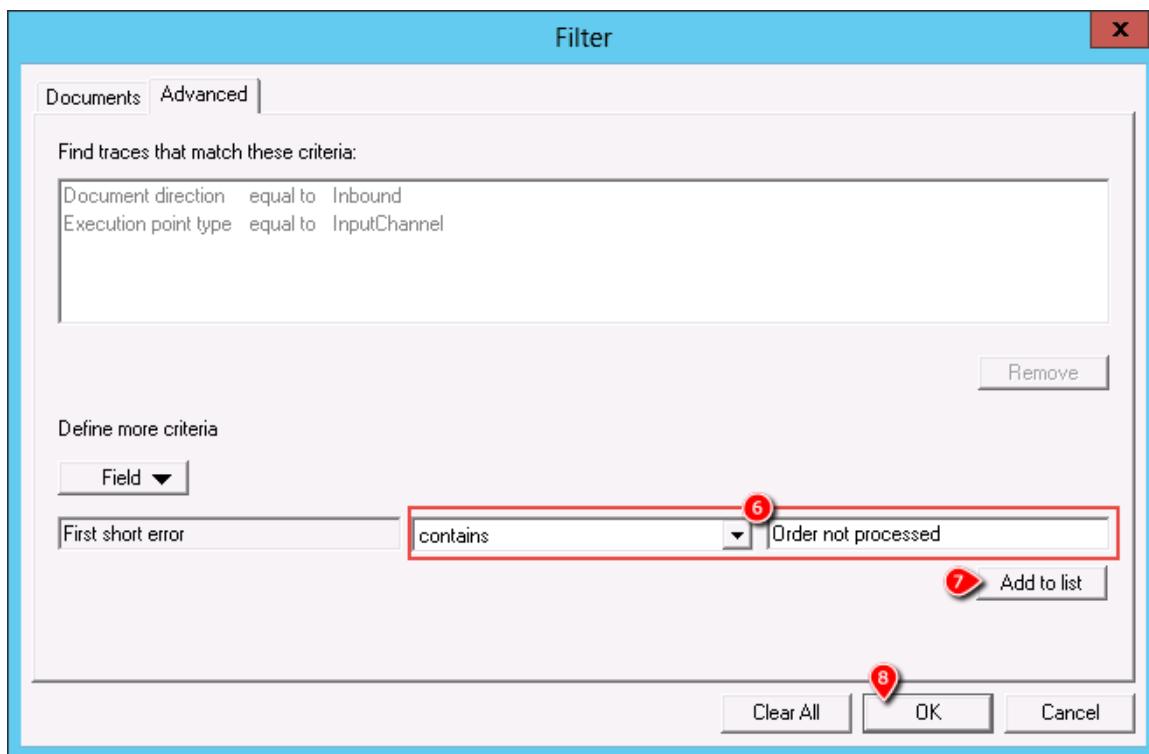


4. Click **Field**.

5. Select a message field to use in a criteria statement.



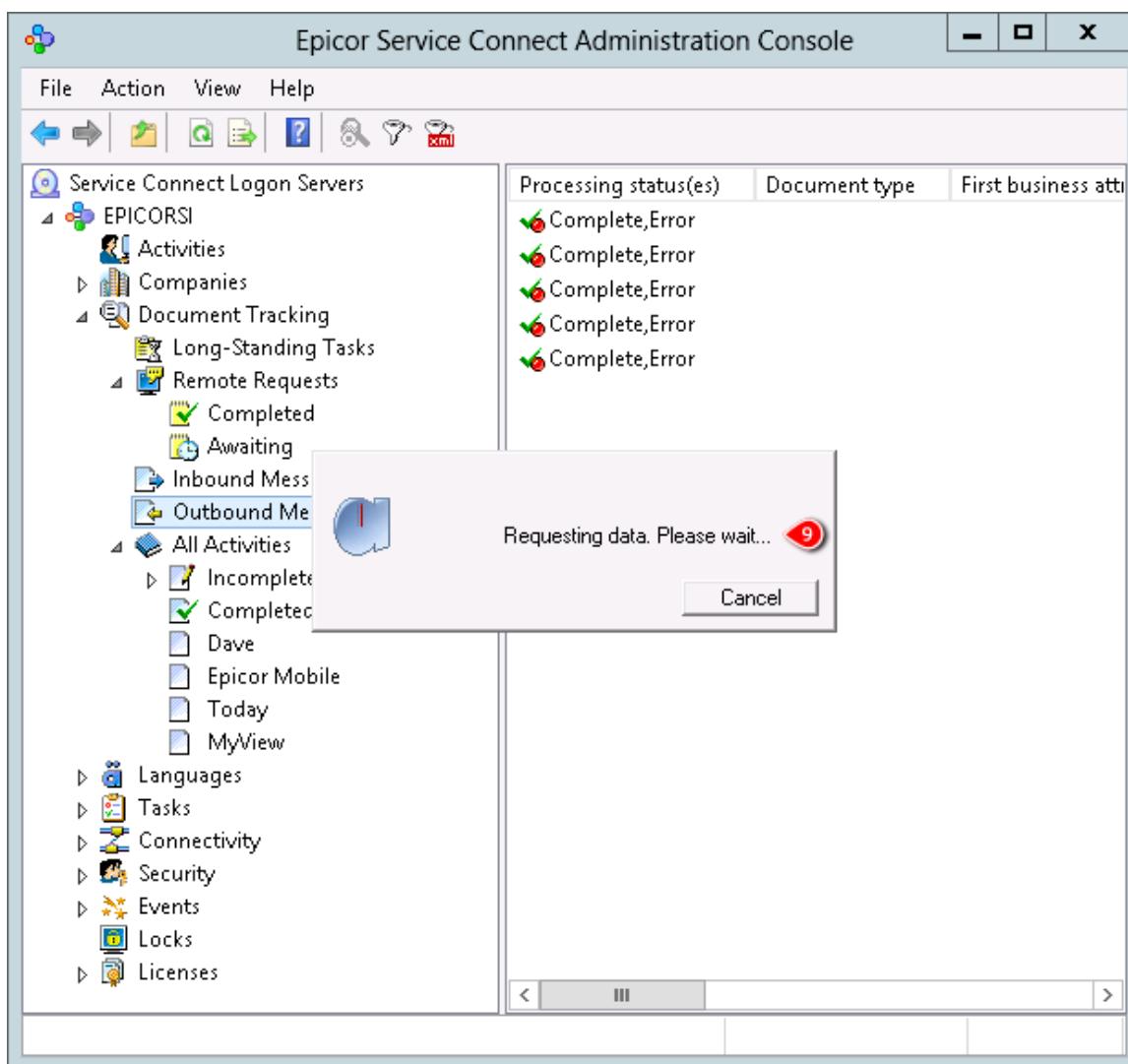
6. Enter filter criteria to complete your statement.



7. Click **Add to list**.

8. Click **OK**.

9. The **Requesting data. Please wait** message displays while the view is updated.



10. See the document tracking view 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

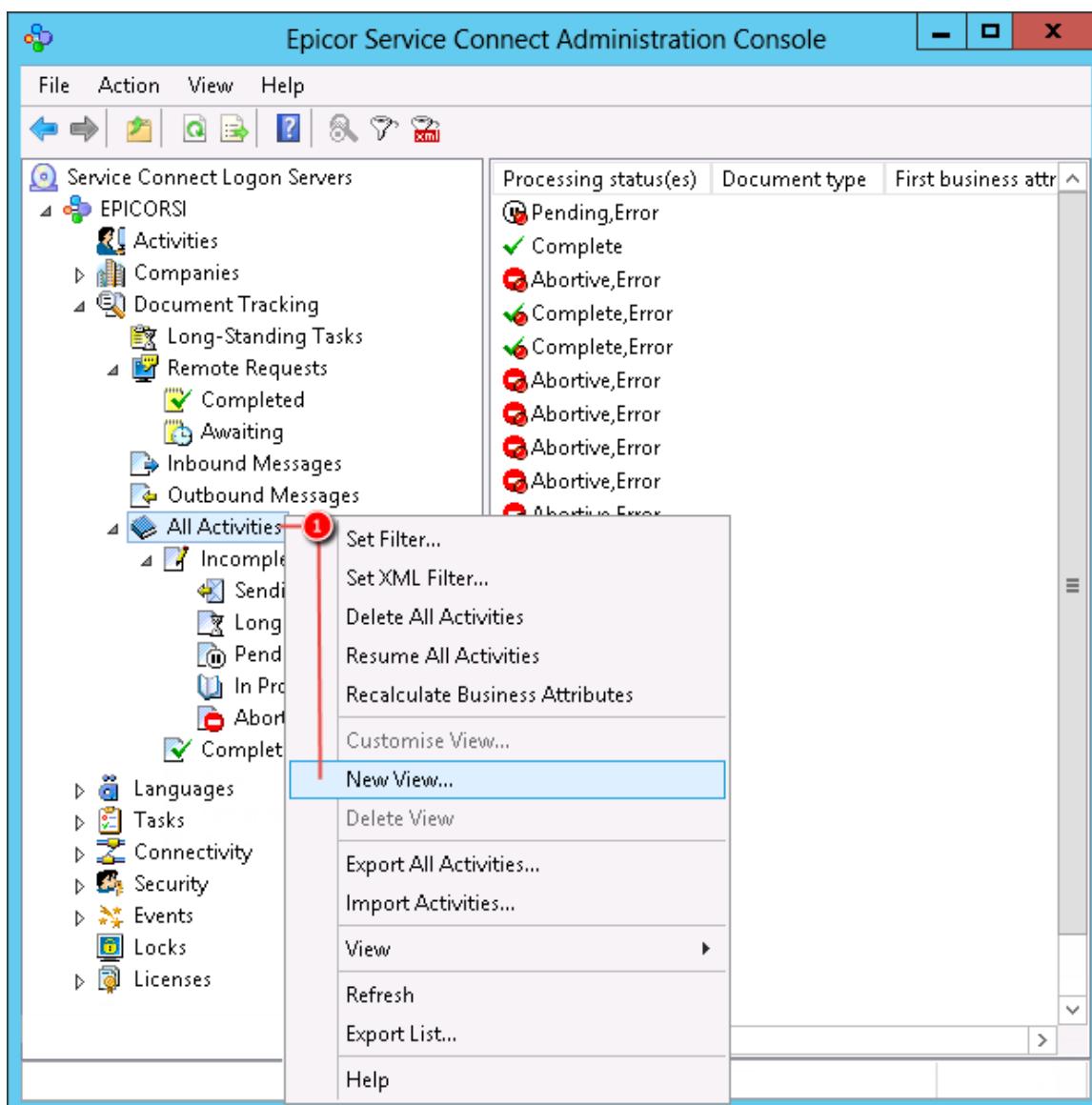
You can define document types that are assigned to messages that enter Service Connect. In turn, you can use the document type as a view filter, so the view only displays the processing status of documents of a specific type, such as an Engineering Change Order or a Customer.

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 Epicor Service Connect Administration chapter, or use the Schemas node in the Administration Console, or use the Generate Schema from Sample Data tool in Workflow Designer.

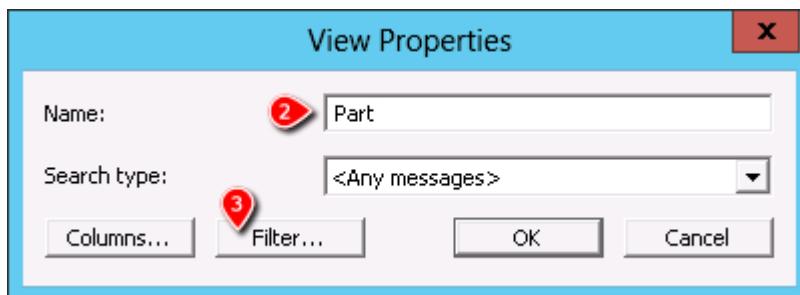
To use a document type as a filter:

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.

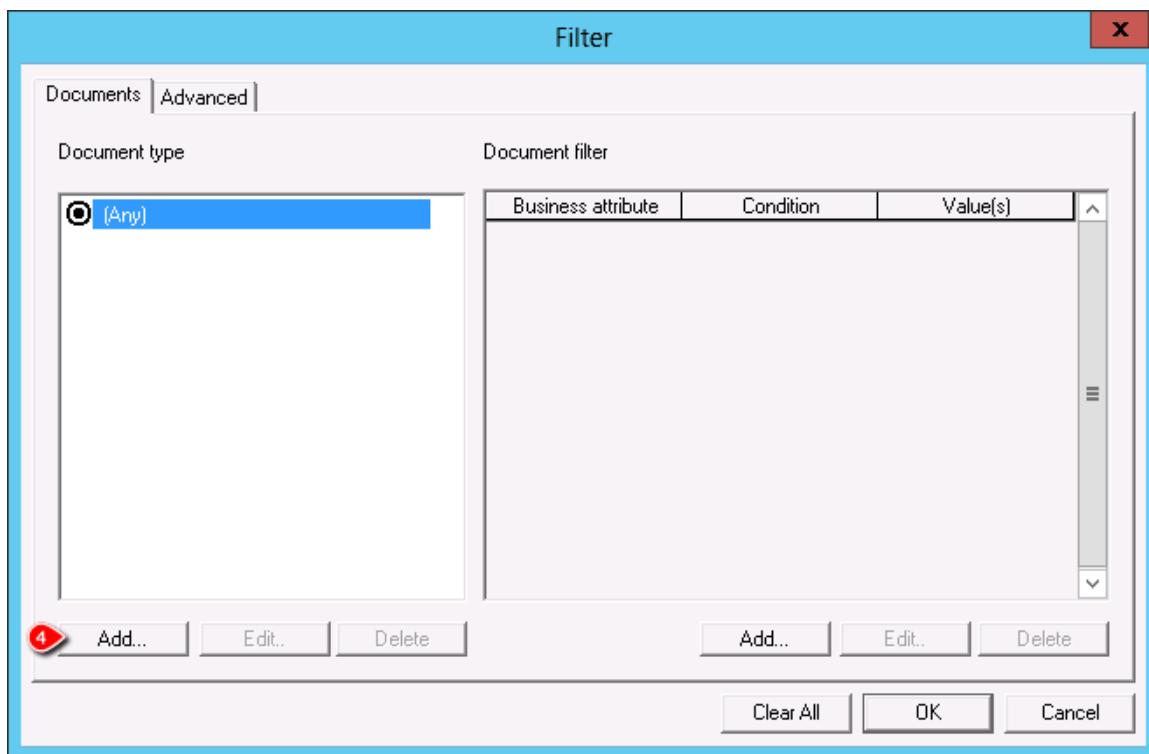


2. The **View Properties** window displays. Enter a view **Name**.

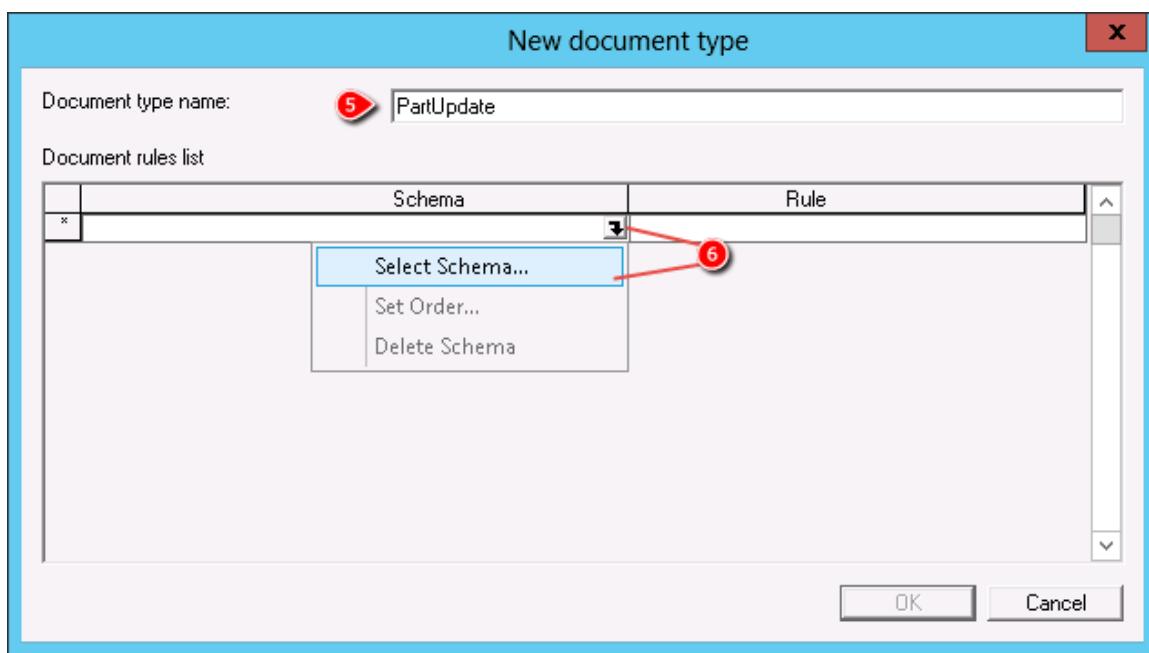


3. Click **Filter**.

4. The **Filter** window displays. Under the **Document type**, click the **Add** button.

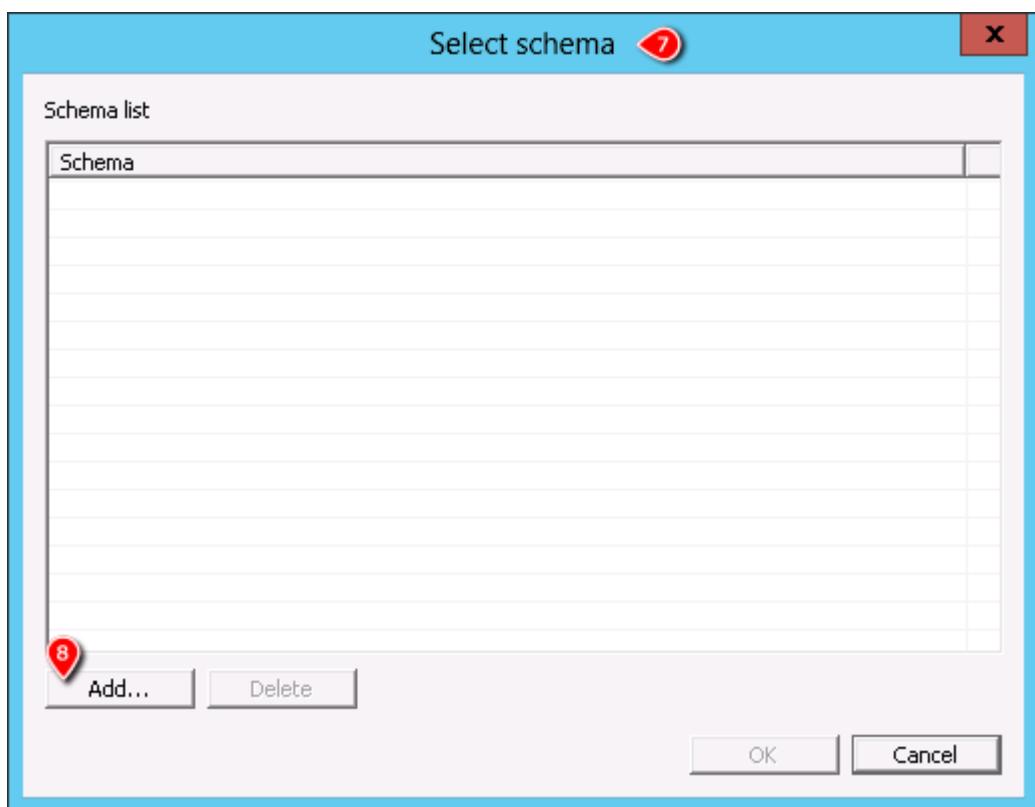


5. The **New document type** window displays. Enter a **Document type name**.

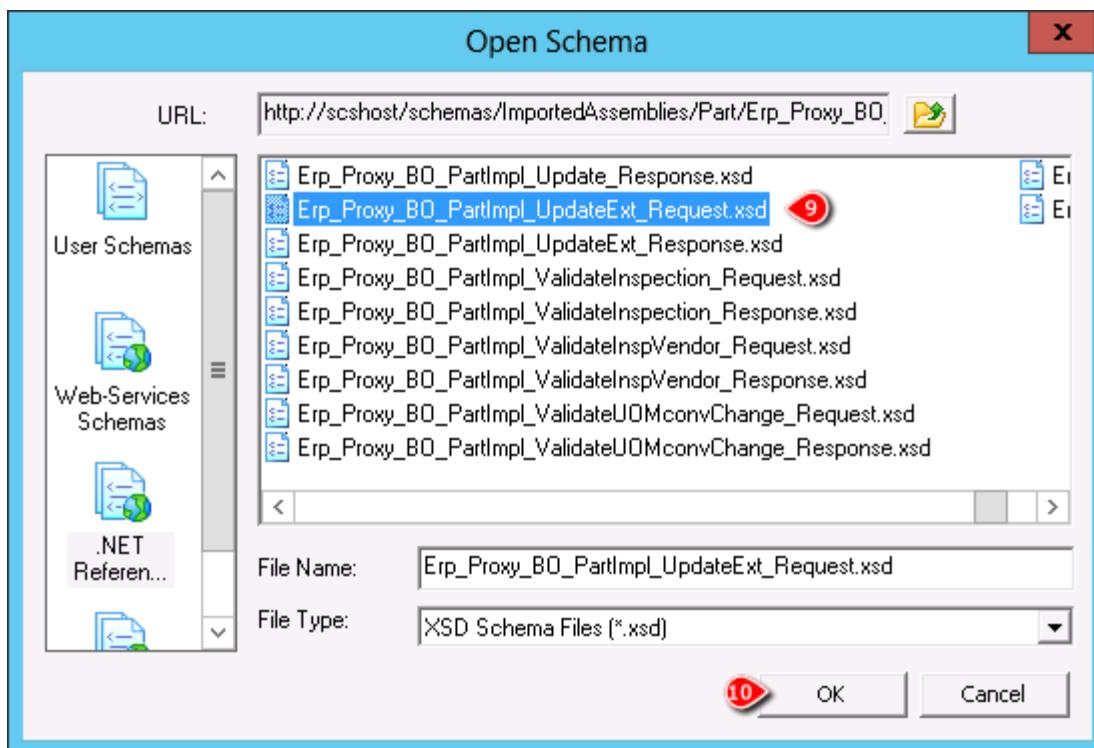


6. Click the button in the **Schema** field and click **Select Schema**.

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



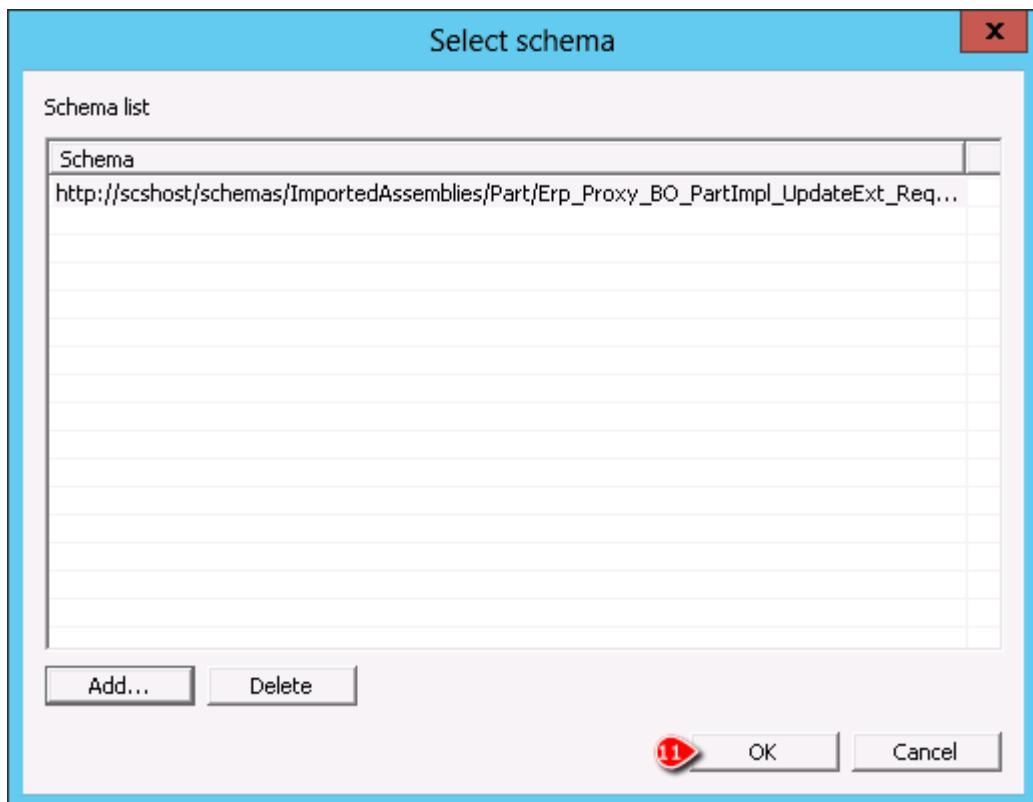
8. If a schema is not defined for the document type, click **Add**.
9. Browse to the service references or user-defined schemas, and select a schema.



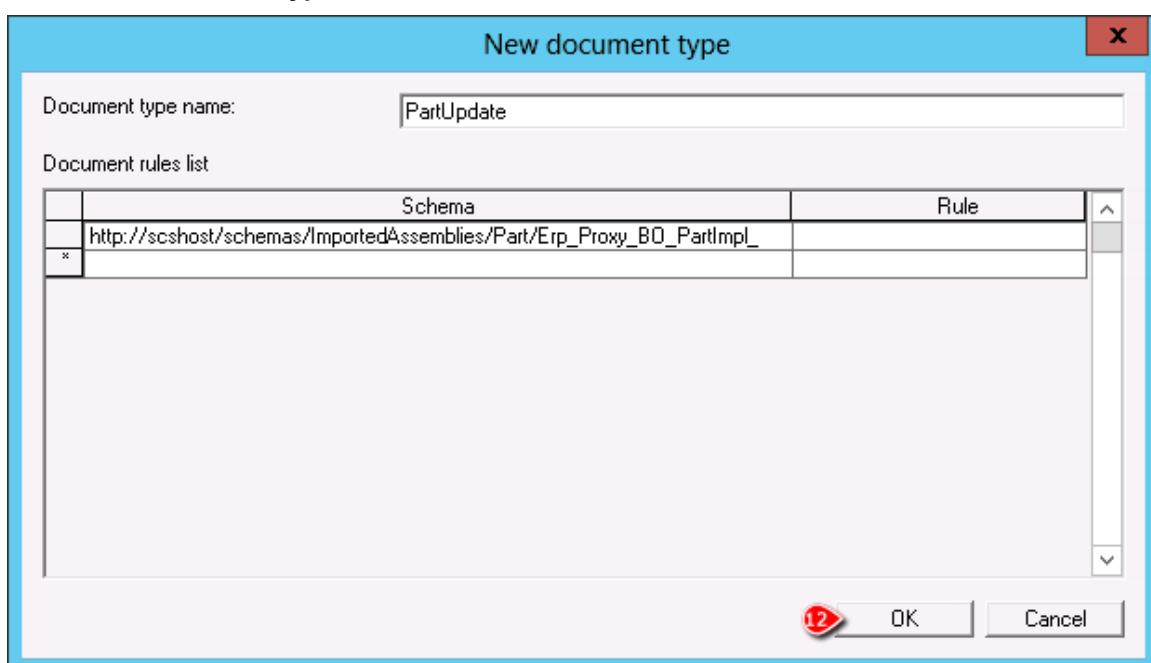
Service Connect evaluates documents that enter the system and assigns a document type to any messages. The messages that use the selected schema display in the view. Optionally, you can define a rule for the schema that must also be met in order for a message to display in the view.

10. Click **OK**.

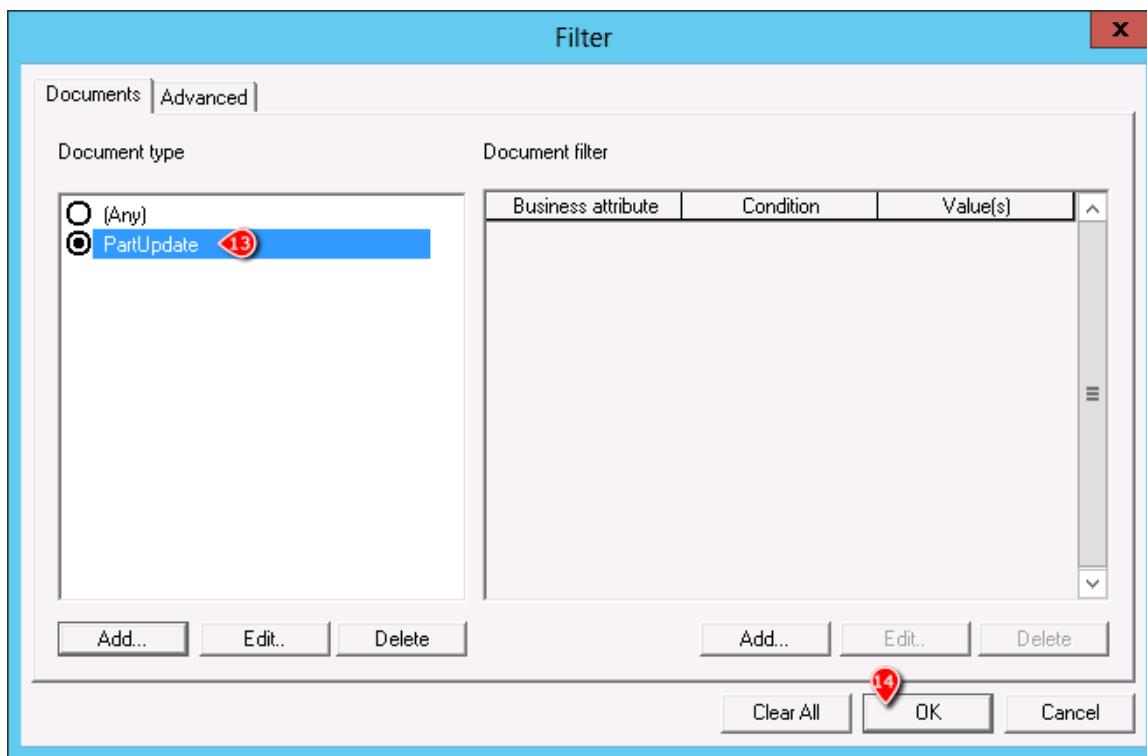
11. In the **Select schema** window, click **OK**.



12. In the **New document type** window, click **OK**.



13. In the **Filter** dialog box, select the **Document type**.



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

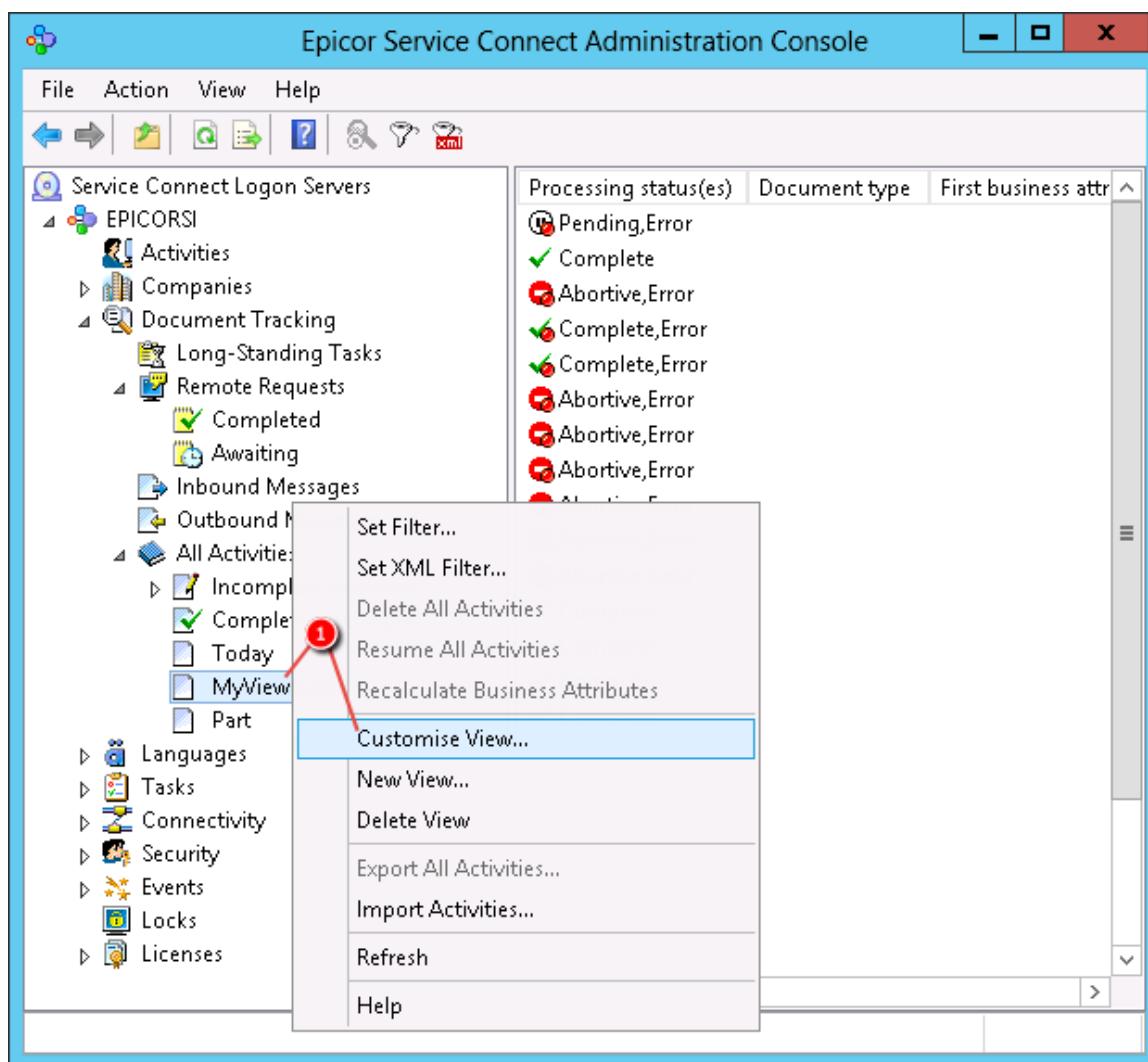
The new view displays only the messages assigned the document type you defined.

Filter a View Using Business Attributes

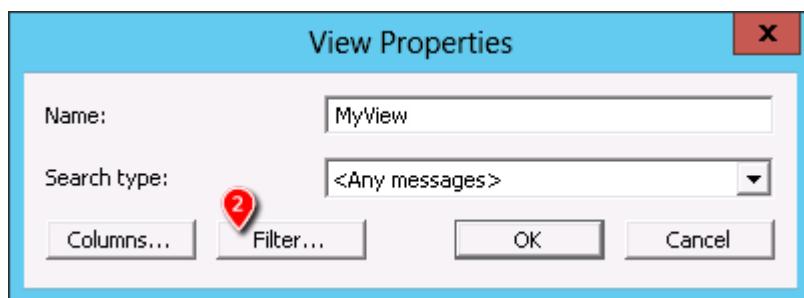
You can define business attributes that read the business data in Service Connect messages and use those attributes to further control the number of records that display in a Document Tracking view. For example, if you have a view that displays the status of Customer documents, you can add a business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

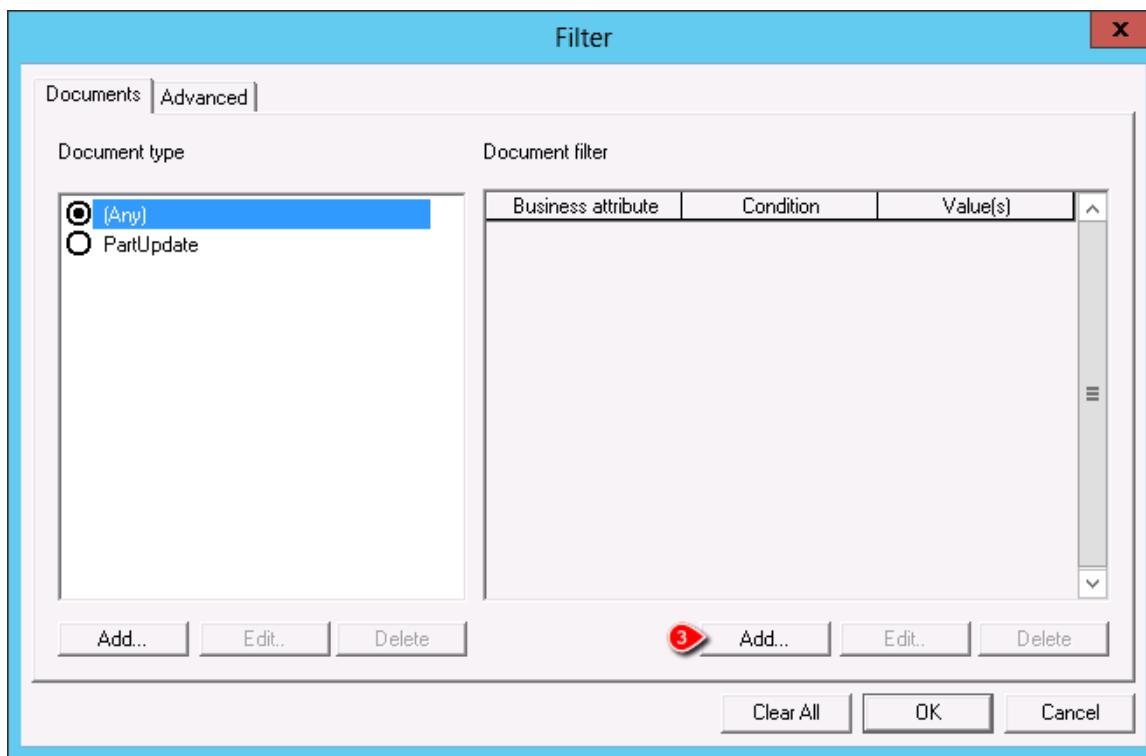
1. In the **tree view**, expand the **Document Tracking** node, right-click the view you want to use as your starting point and select **Customize View**.



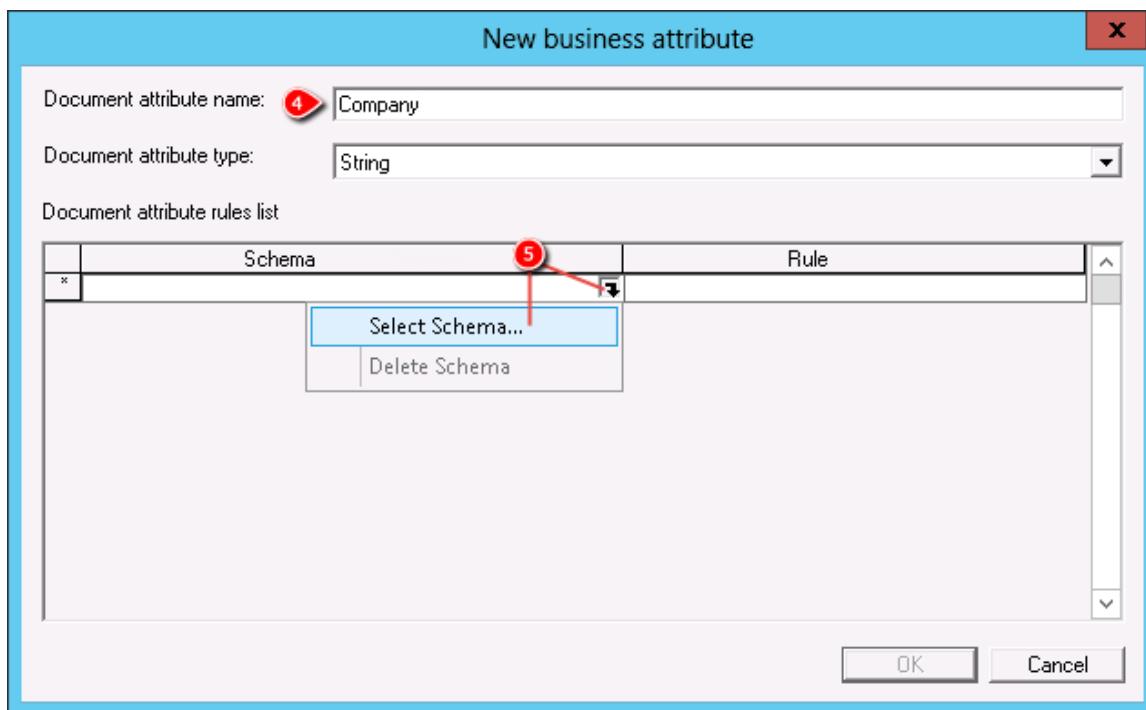
2. The **View Properties** window displays. Click **Filter**.



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

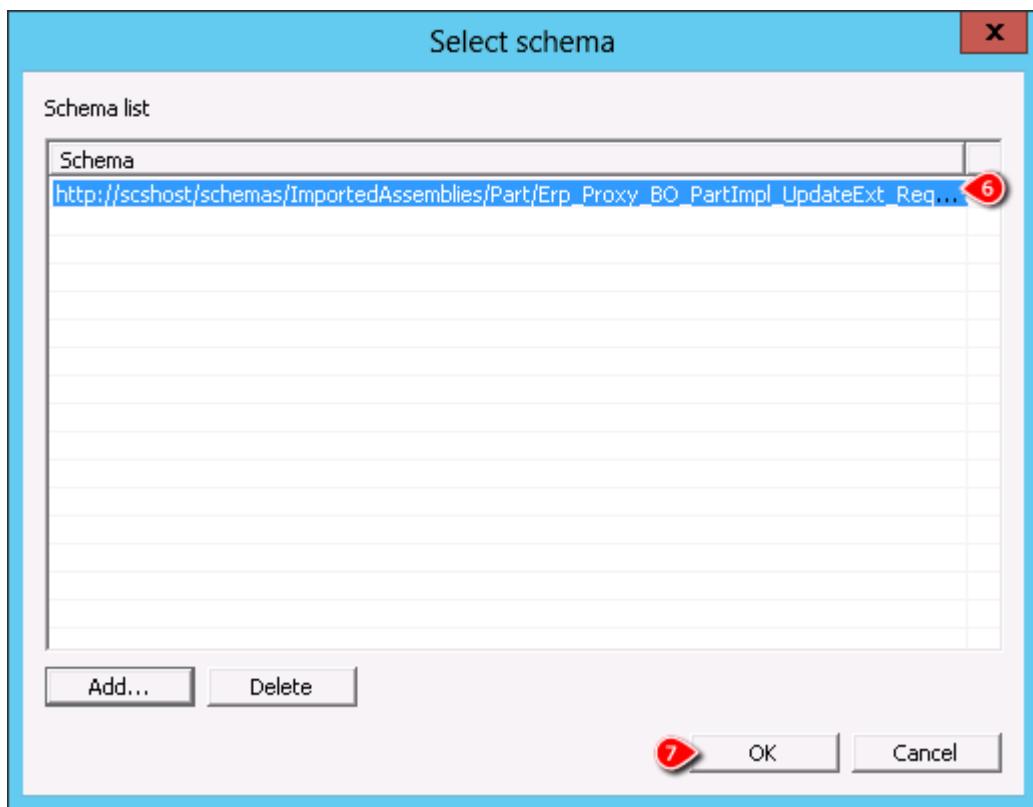


4. The **New business attribute** window displays. Enter a **Document attribute name**.



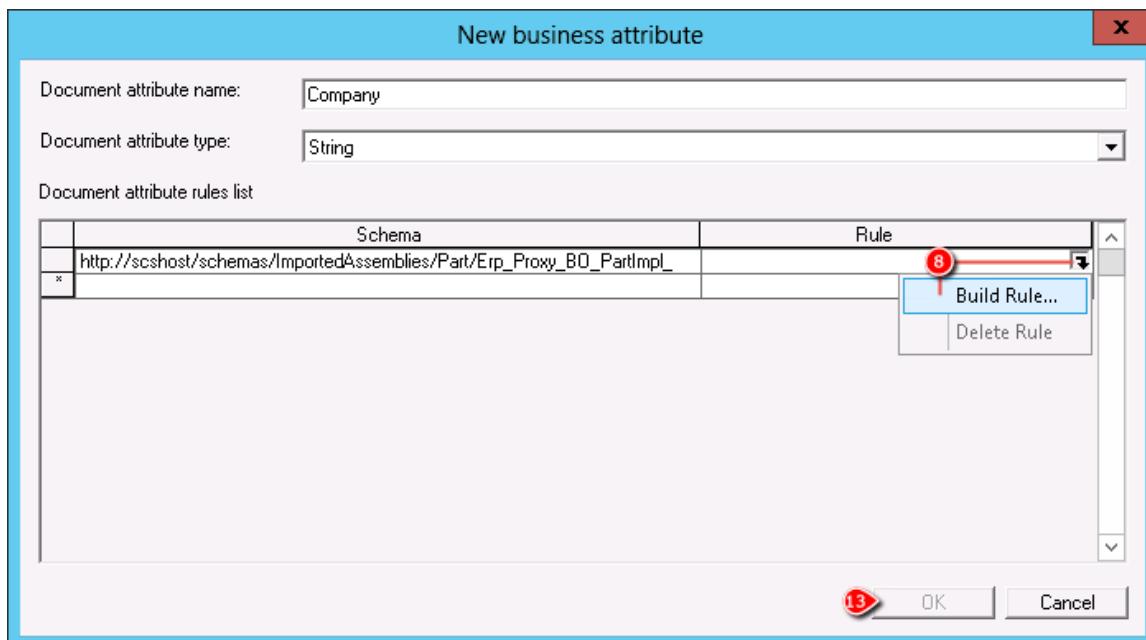
5. Click the **Schema** field, then click the arrow button in the **Schema** field, and select **Select Schema**.

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

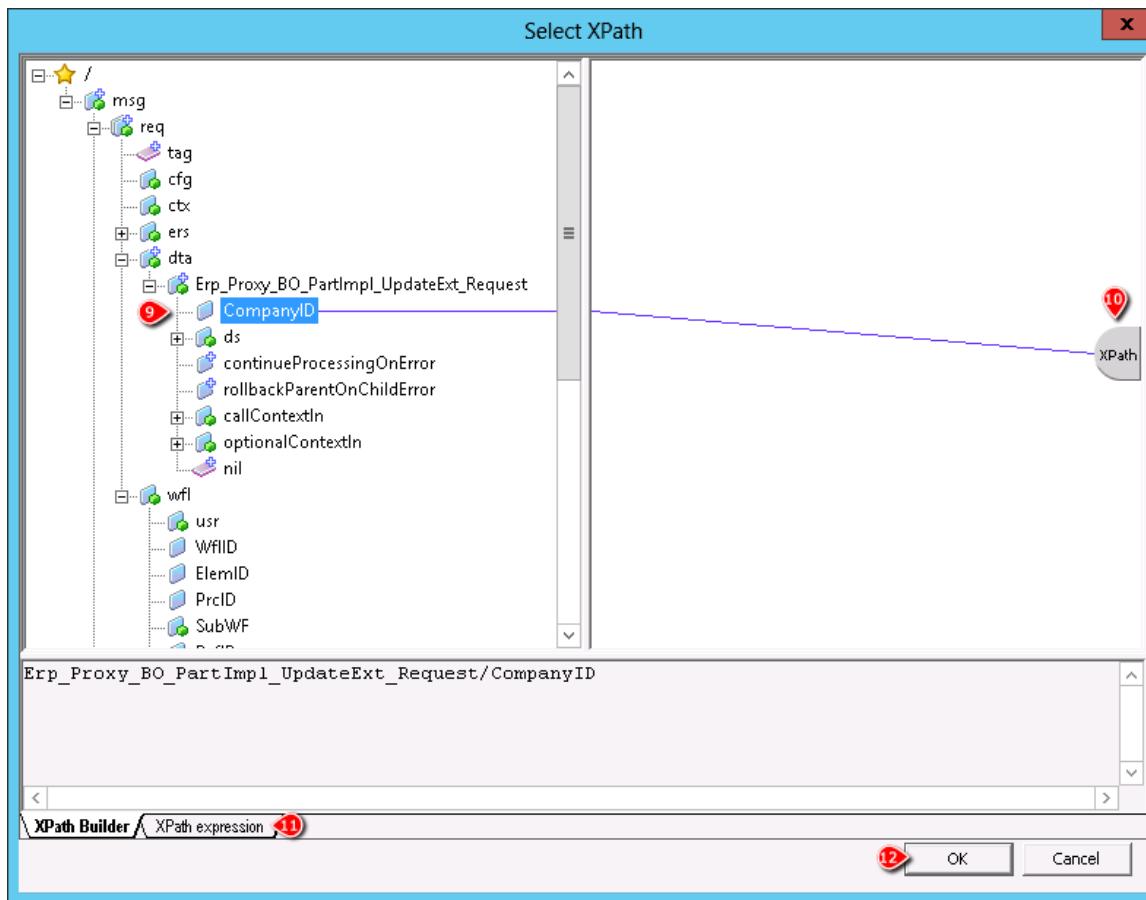


7. Click **OK**.

8. The **New business attribute** window displays all the schemas associated with the document type. Click the **Rule** field, then click the arrow button in the **Rule** field, and select **Build Rule**.



9. The **Select XPath** window displays. On the left, 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 on 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 CompanyID node will be used to filter the document tracking view.

11. You can switch to the **XPath expression** tab to view the resulting XPath statement. It is displayed in simplified form: namespaces are omitted and only the xml elements names are displayed.

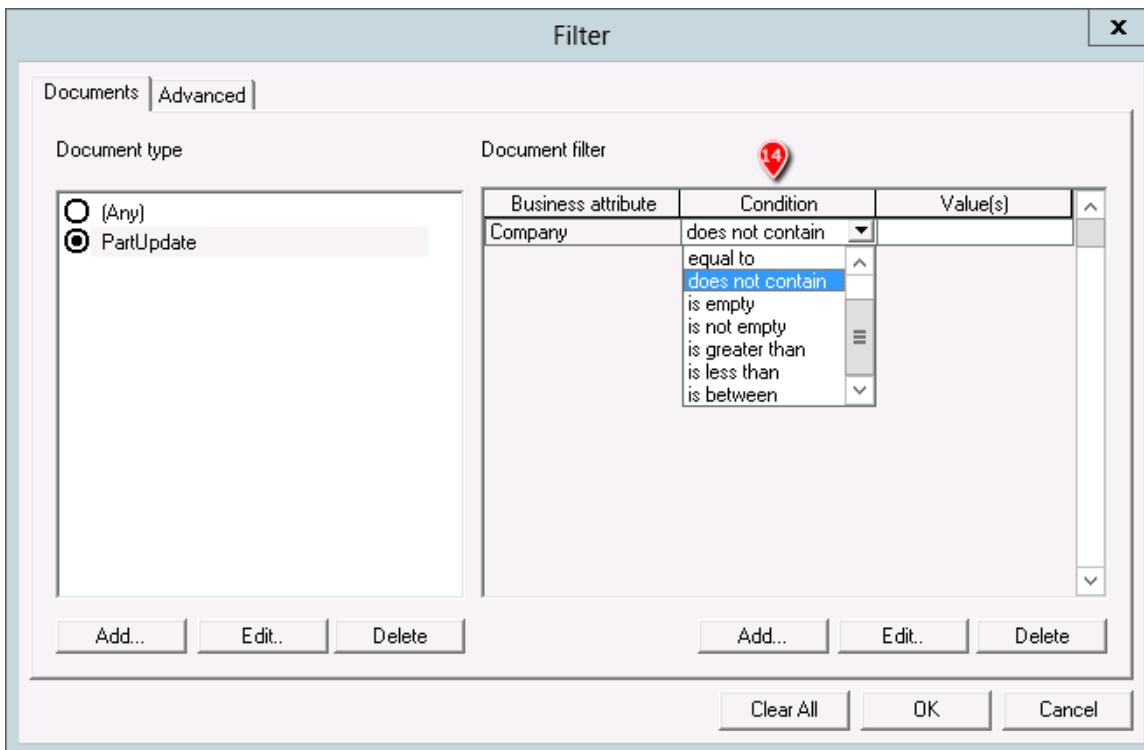
12. Click **OK**.

13. In the **New business attribute** window, click **OK**.

14. In the **Filter** dialog box, select a **Condition** for the rule. Available options:

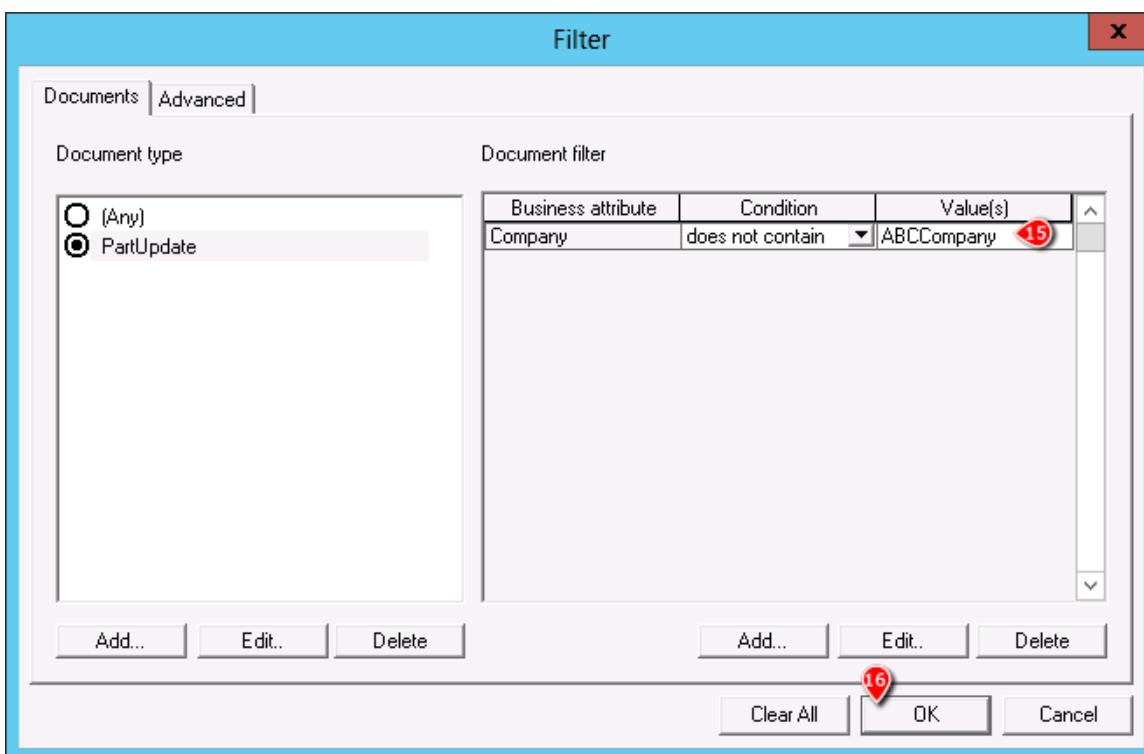
- Any
- Contains
- Equal to
- Doesn't contain
- Is empty
- Is not empty
- Is greater than
- Is less than

- Is between



15. 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.



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

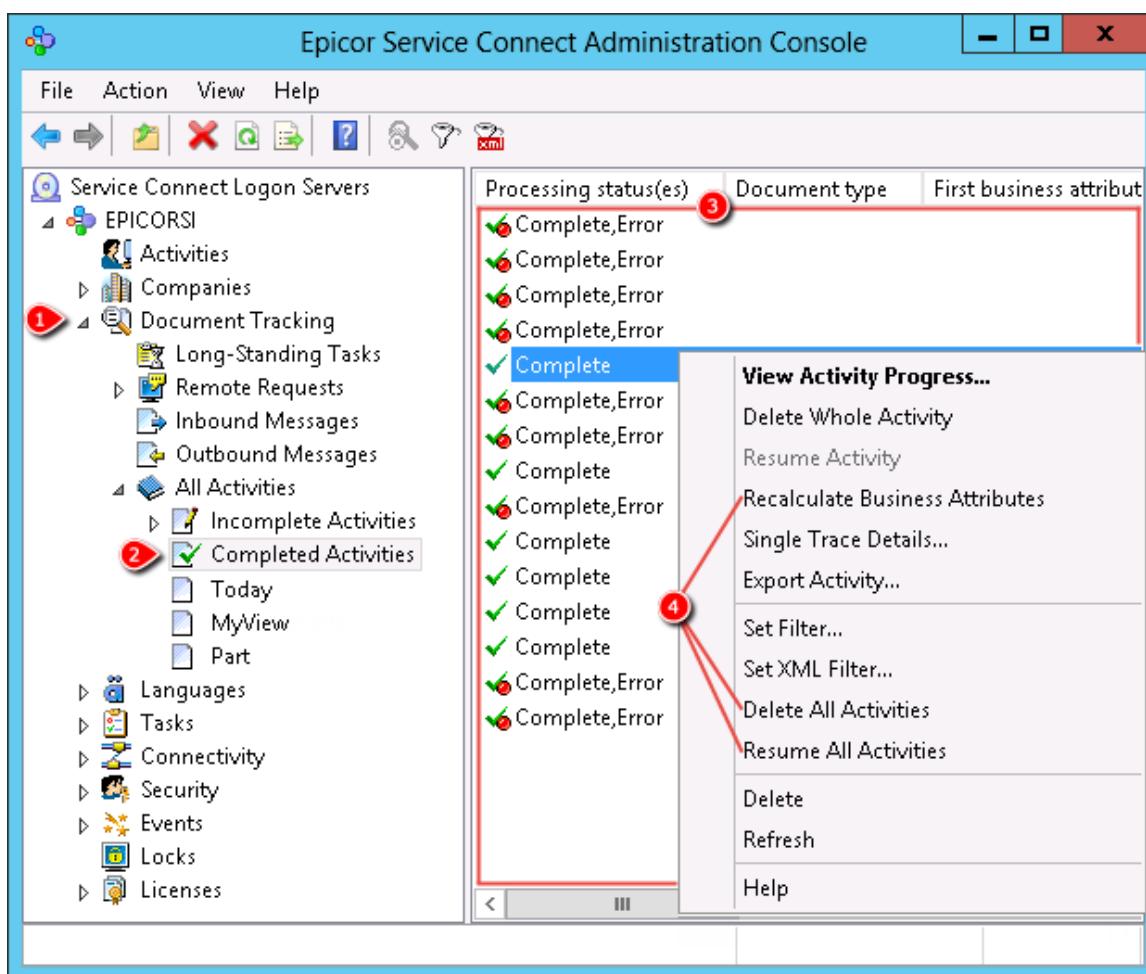
The resulting view is filtered by the rule according to the data in the message.

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.



2. Select one of the Document Tracking folders.
3. Right-click the display pane where the activities are listed.
4. Select **Recalculate Business Attributes**, **Resume All Activities**, or **Delete All Activities**.

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 context menu items are selected: Delete All Activities,

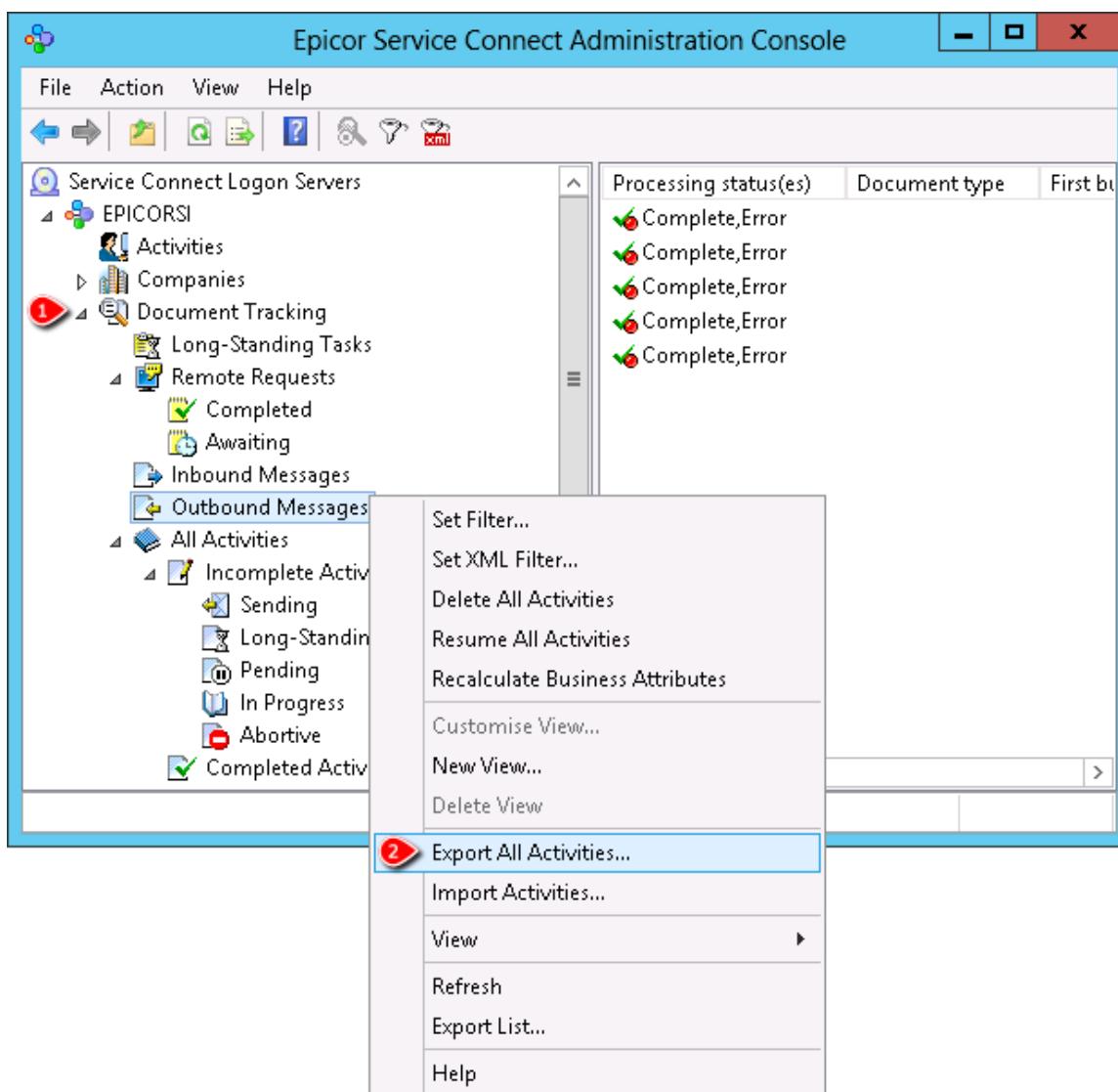
Resume All Activities, Recalculate Business Attributes, Export All Activities and Import Activities.

Import to and Export from Document Tracking

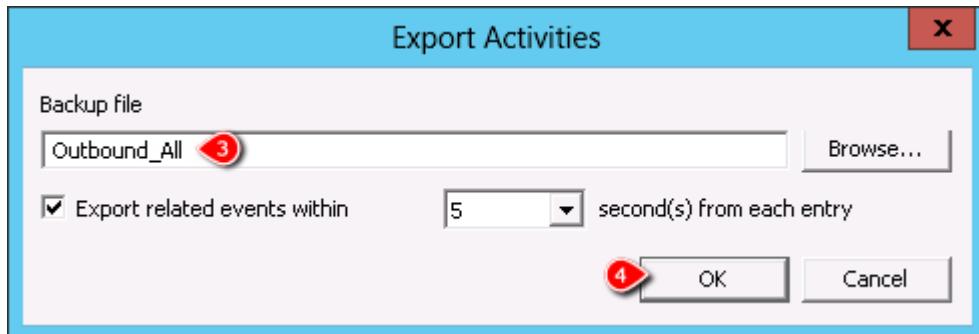
The Document Tracking system supports the export and import functionality, which benefits support teams that evaluate issues and share information.

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**.



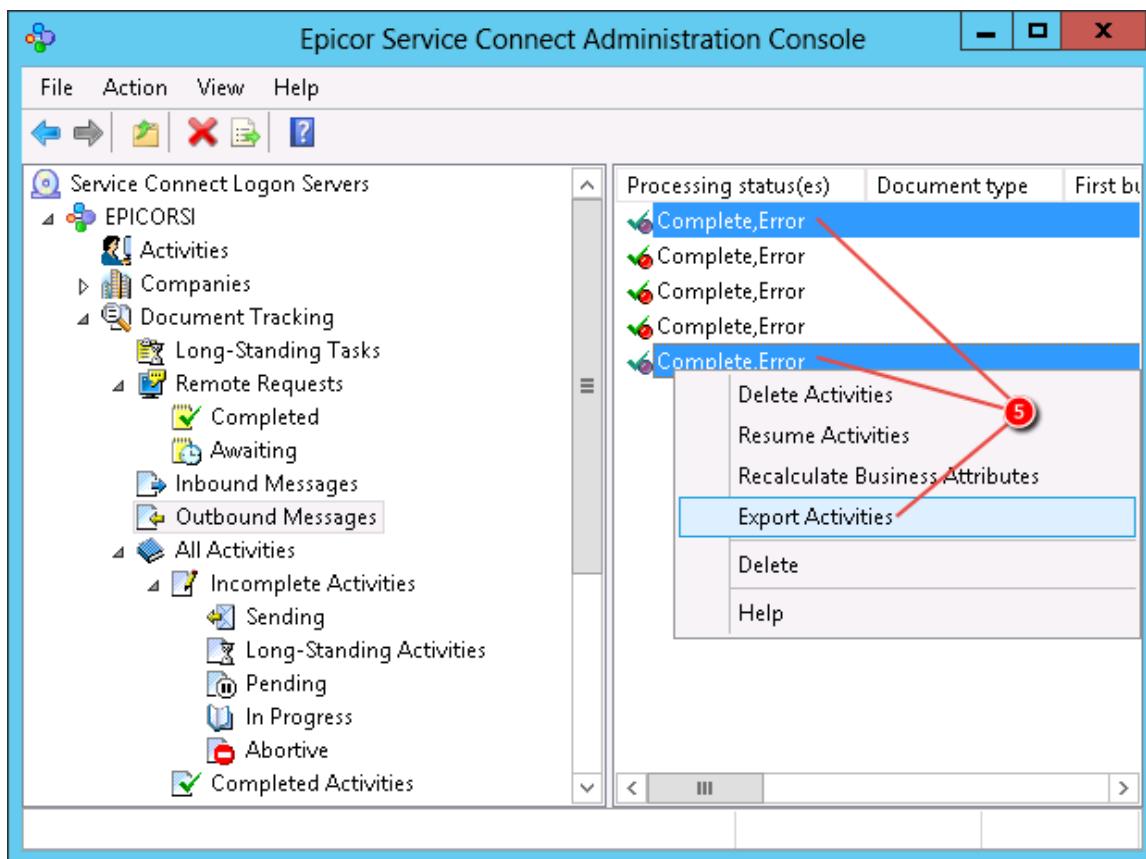
3. Enter a **File name** for the backup file.



4. Click **Save**.

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

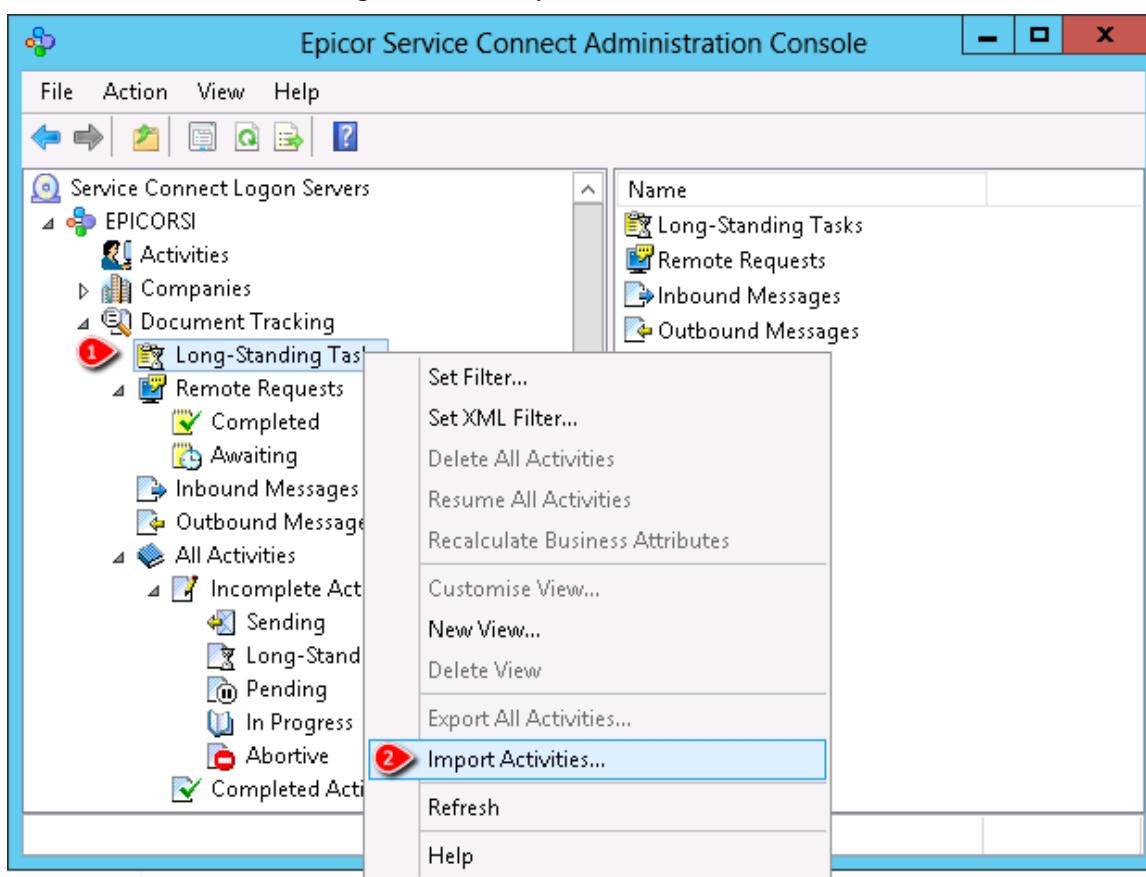
5. To export a single activity or a selection of activities, right-click the activity or select and right-click activities and select **Export Activity** or **Export Activities**.



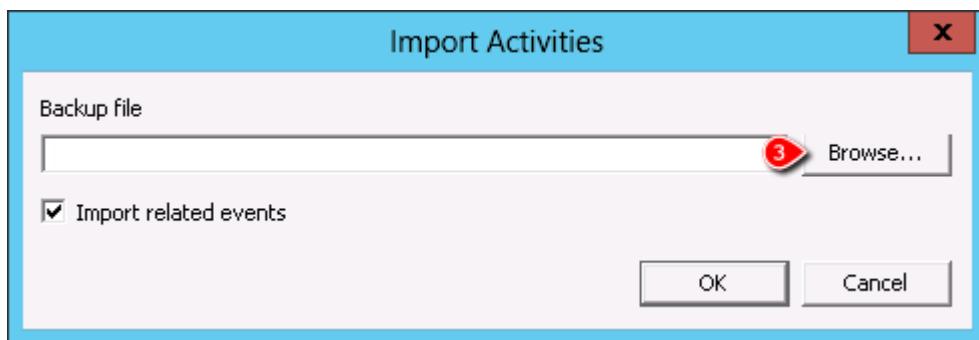
Import Activities into ESC Administration Console

To import activities into the ESC Administration Console:

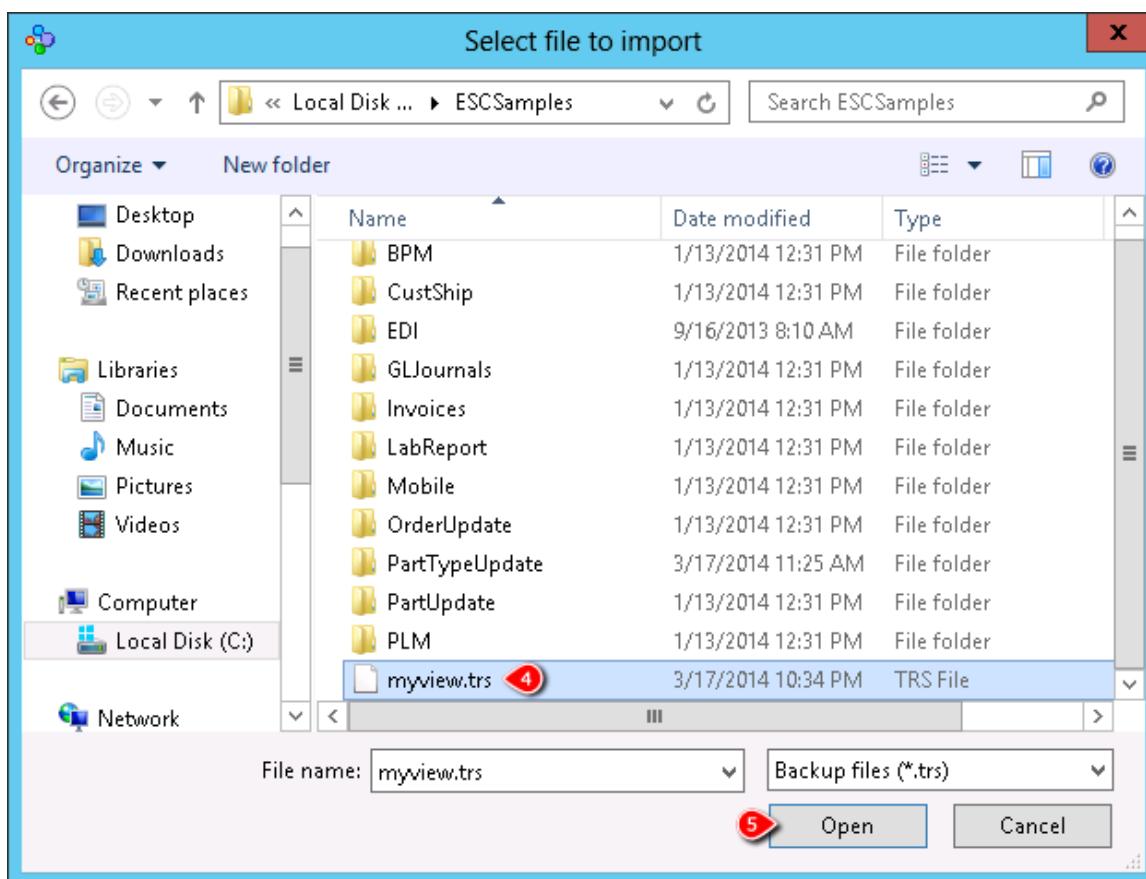
- Under the **Document Tracking** node, click any view.



- Right-click the view and select **Import Activities**.
- In the **Import Activities** window, click the **Browse** button.



4. In the **Select file to import** window, navigate to and select a .trs backup file.



5. Click **Open**.

6. In the **Import Activities** window, click **OK**.



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.

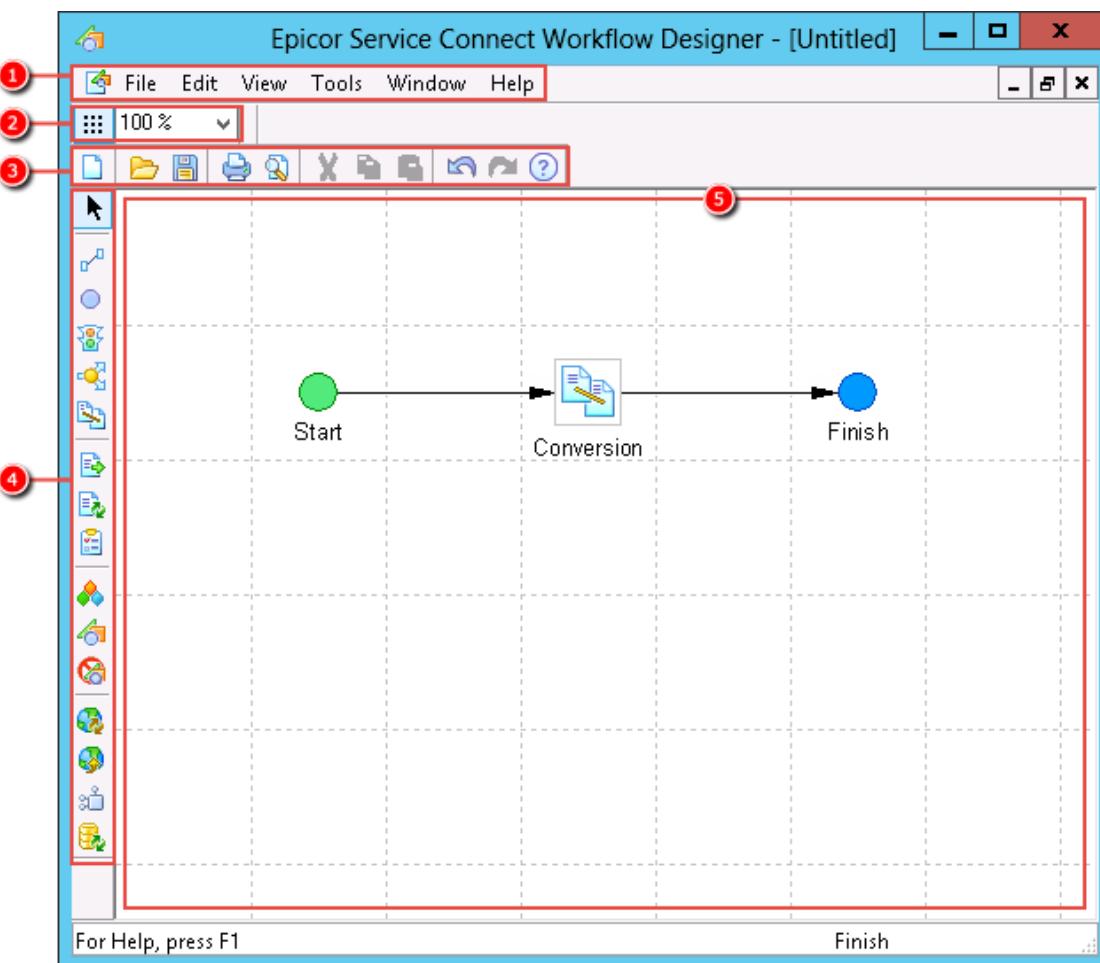
Chapter 4: Workflow Designer

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.
Validate Process	Check the workflow you created for consistency. For more details on the Validate Process functionality refer to the Process Validation section later in this chapter.
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.

Select	To
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.
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. 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.

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.
<input type="button" value="100 %"/> ▾	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.

Click	To
	

Items Toolbar

The Items toolbar contains the Selector tool and all the elements you can use in a workflow. The following is the complete list of workflow elements 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 elements. All the elements in a workflow must be connected, and at least one element must be connected to the Finish element, before a workflow will function. Connections can have a Caption. The Caption is used by Choice and Task elements, 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 elements.

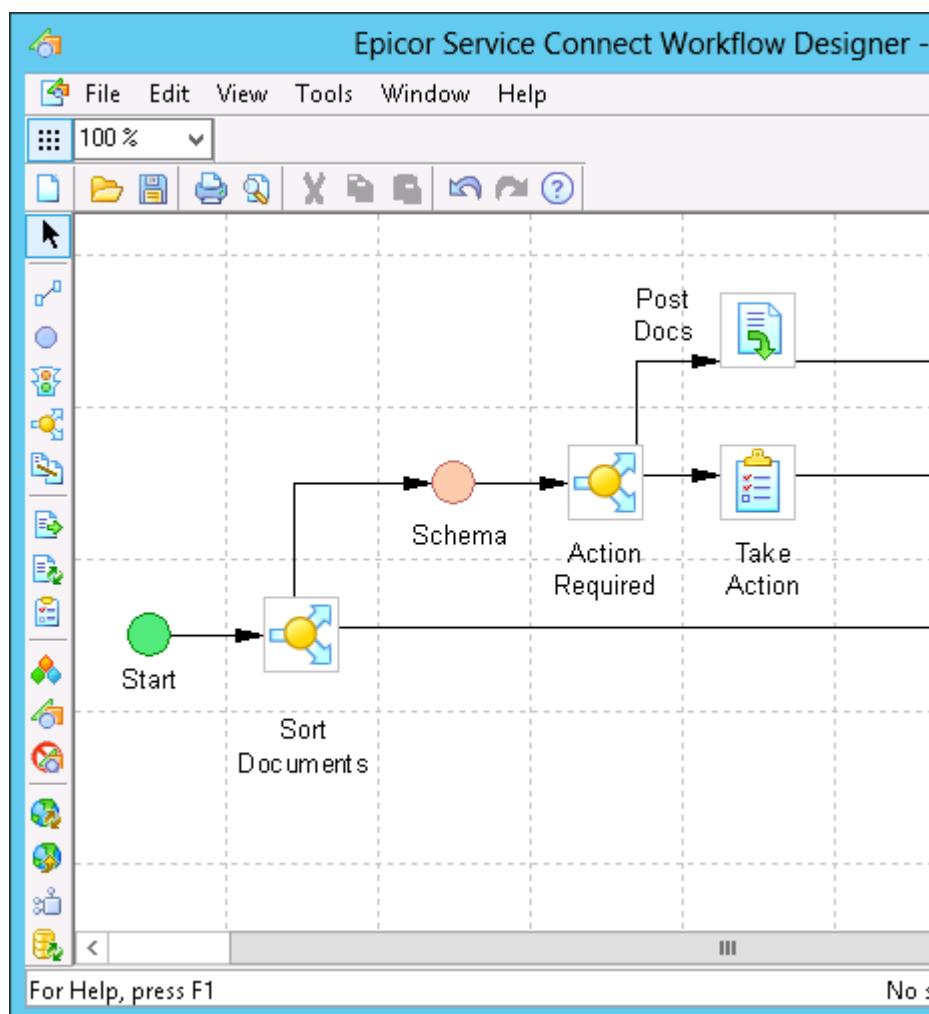
Splitter



Use this workflow element 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 element.

Example A workflow accepts documents in two different formats. A Choice element routes incoming documents based

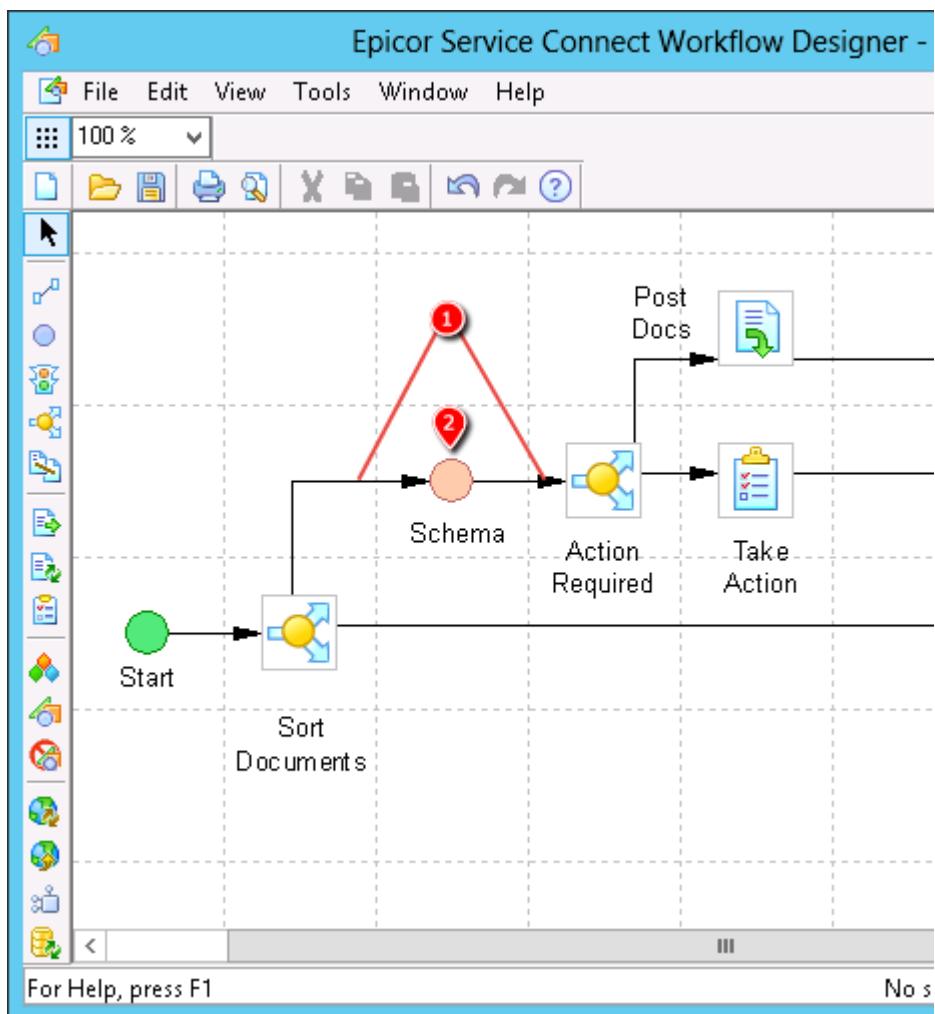
on type. You can use a Splitter after the Choice to apply a schema to the document.



Set Up a Splitter

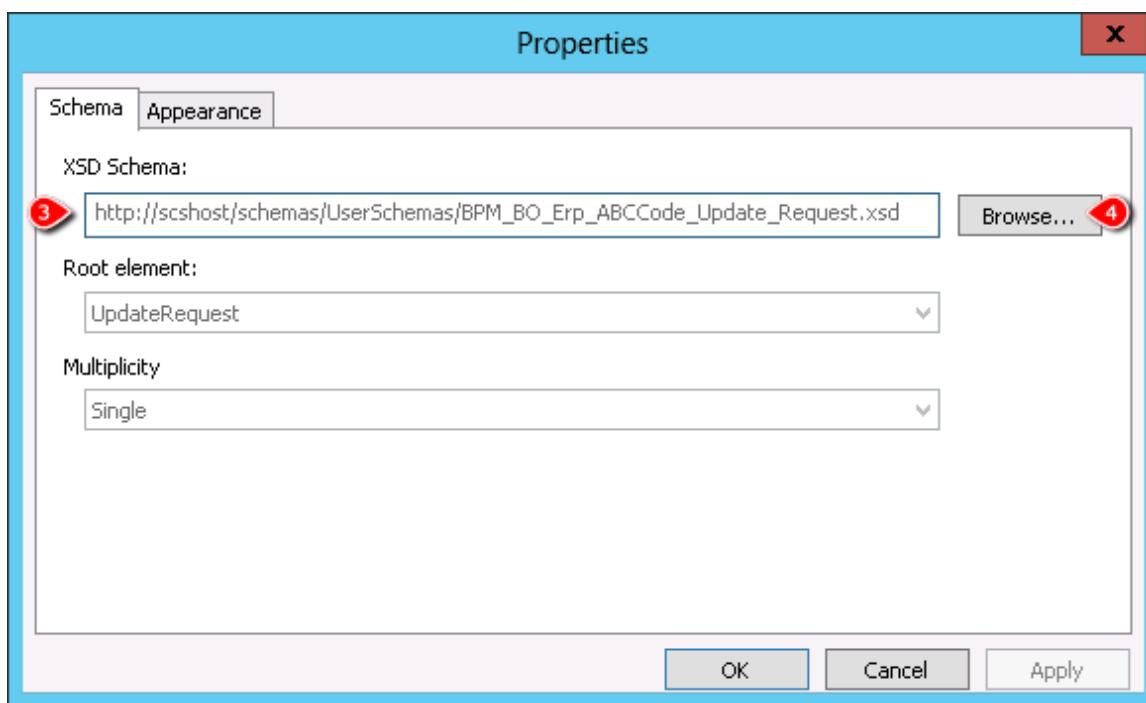
To set up a Splitter activity:

1. In the **Workflow design area**, add the **Connections** to and from the **Splitter**.



2. Double-click the **Splitter** to open the **Properties** window.

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



4. Click the **Browse** button to find and select a different schema. The **Open Schema** window displays.

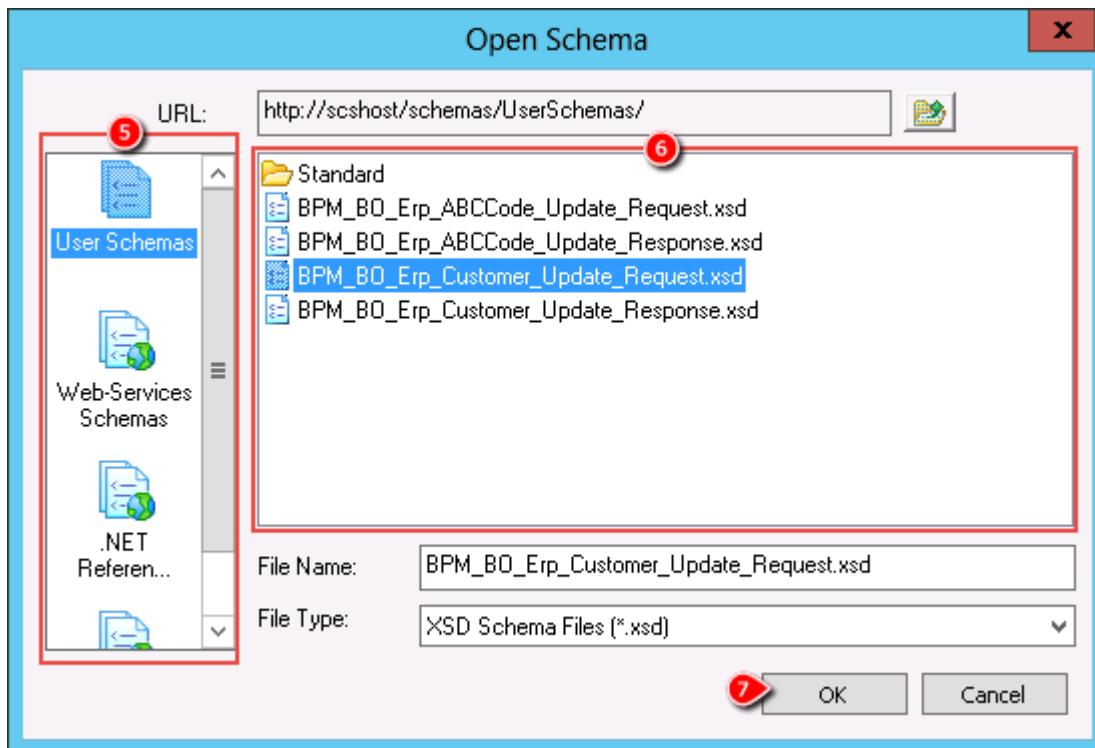
5. In the left pane, select a schema type. Available options:

- User Schemas

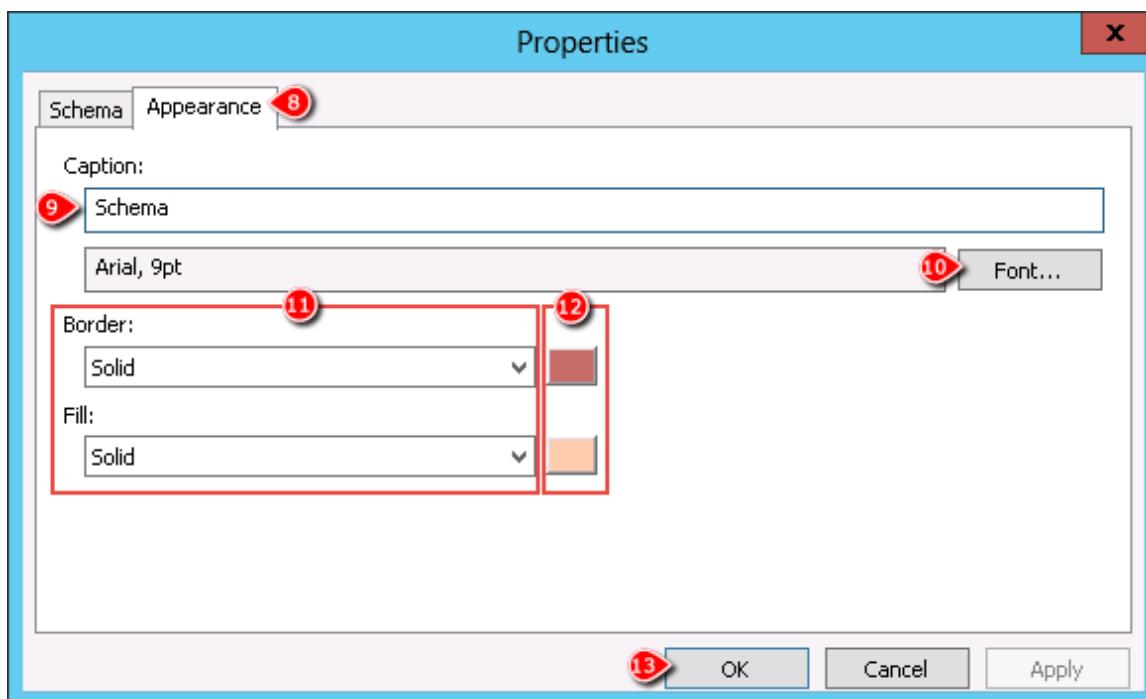
These are the schemas generated in the Schemas node of Service Connect Administration Console, or in the Generate Schema from Sample Data tool in the Workflow Designer, or the schemas generated in the Workflow Designer when you click the Create sub-workflow schema button on the Cycling tab of Sub-workflow properties dialog.

- Web-Services Schemas
- .NET Reference Schemas

- REST-Services Schemas



6. In the right pane, select a schema.
7. Click **OK**.
8. In the **Properties** window, open the **Appearance** tab.



9. In the **Caption** field, enter the Splitter caption to display in the workflow schema.

10. Click the **Font** button to adjust the font settings (style, size, and so on) for the Splitter caption.
11. Click the **Border** and **Fill** drop-down lists to select border and fill styles.
12. Use color buttons to the right of the lists to set the border and fill colors.
13. Click **OK**.

Task



Use this workflow element 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.

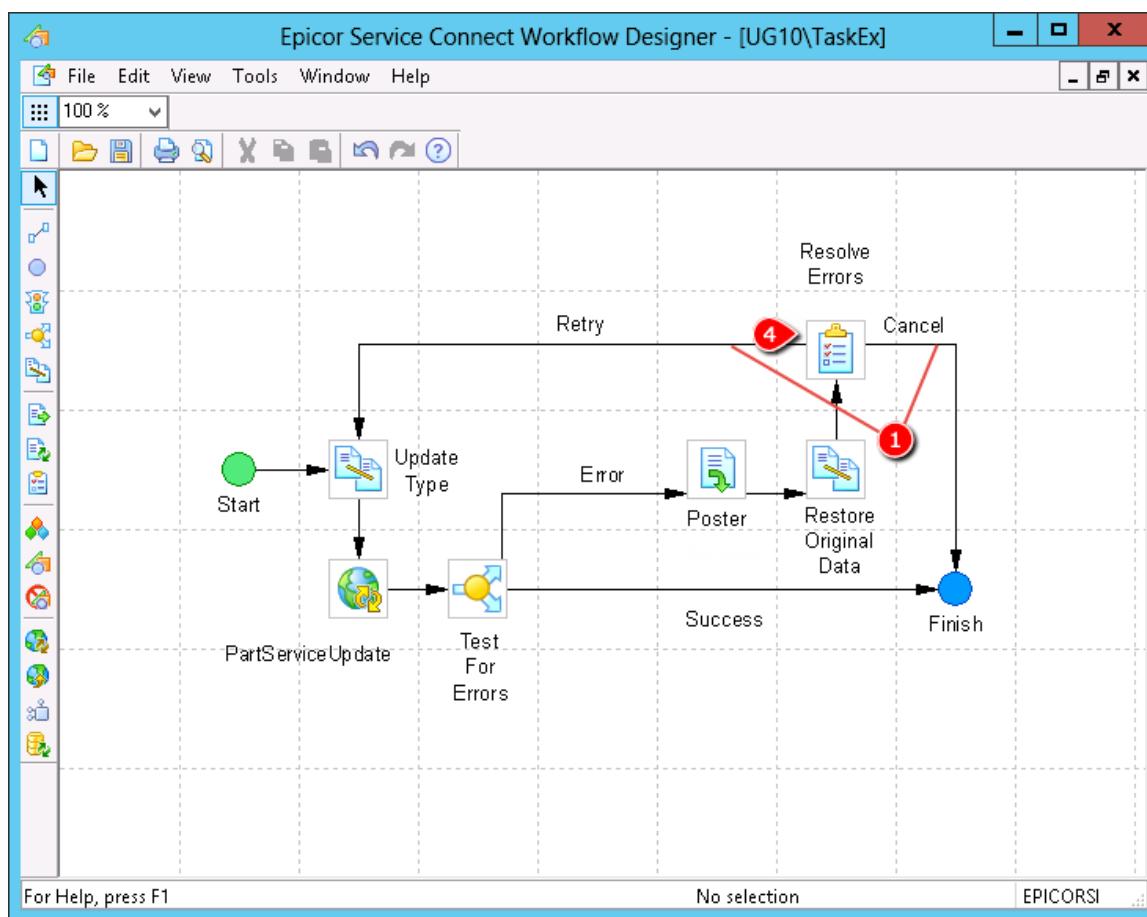
Task Example

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 workflow element. 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.

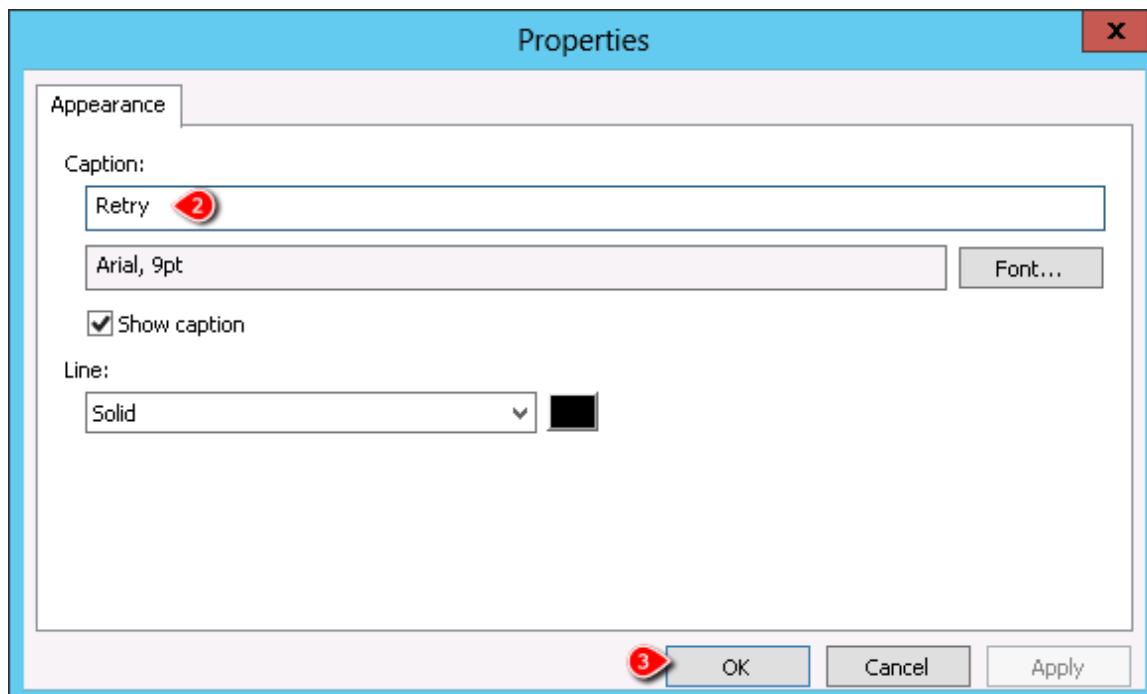
Set Up a Task

To set up a Task:

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

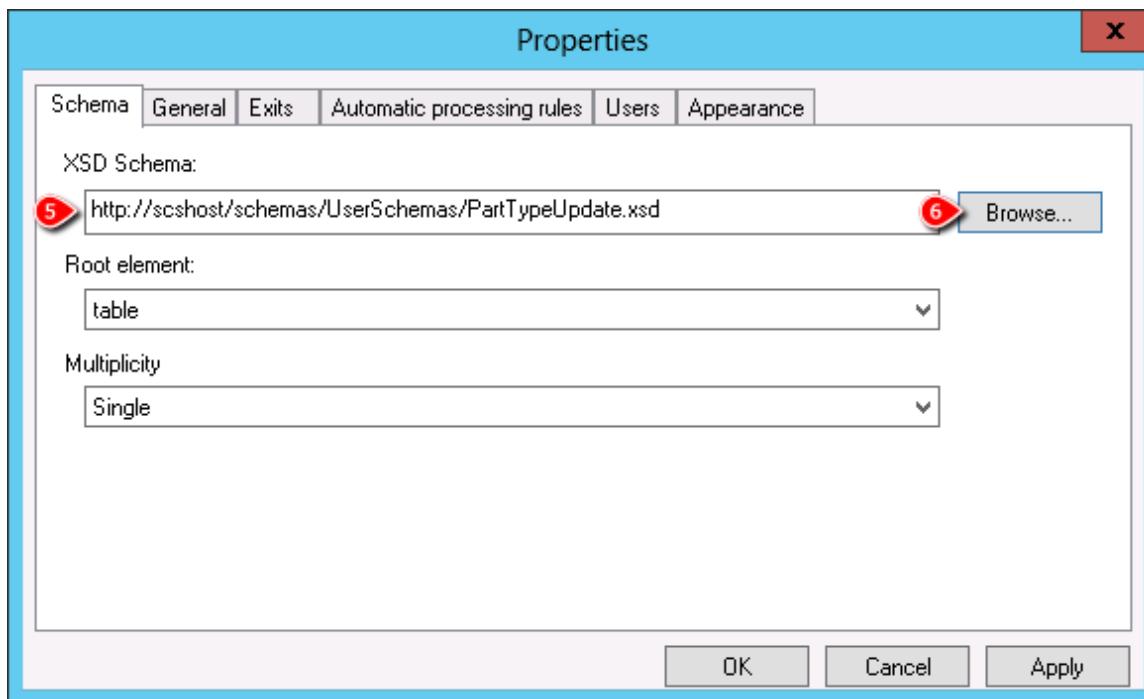


- Double-click each Connection to define its **Caption** in the **Properties** dialog.



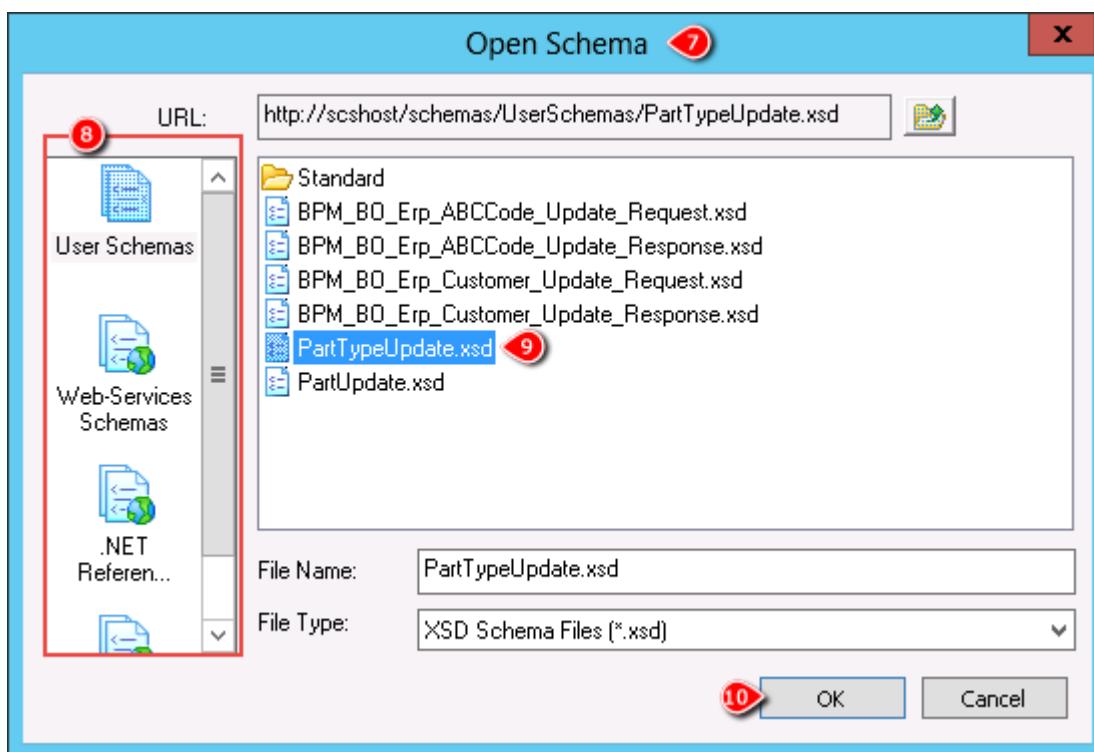
Enter a logical Caption for each outbound Connection, to easily identify them when you configure the Tasks exits.

3. Click **OK**.
4. In the **Workflow design area**, double-click the **Task** to modify its properties.
The **Properties** window displays.



5. On the **Schema** tab, 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. The **Open Schema** window displays.



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

- Web-Services Schemas
- .NET Reference Schemas
- REST-Services Schemas
- User Schemas

These are the schemas generated in the Schemas node of Service Connect Administration Console, or in the Generate Schema from Sample Data tool in the Workflow Designer, or the schemas generated in the Workflow Designer when you click the Create sub-workflow schema button on the Cycling tab of Sub-workflow properties dialog

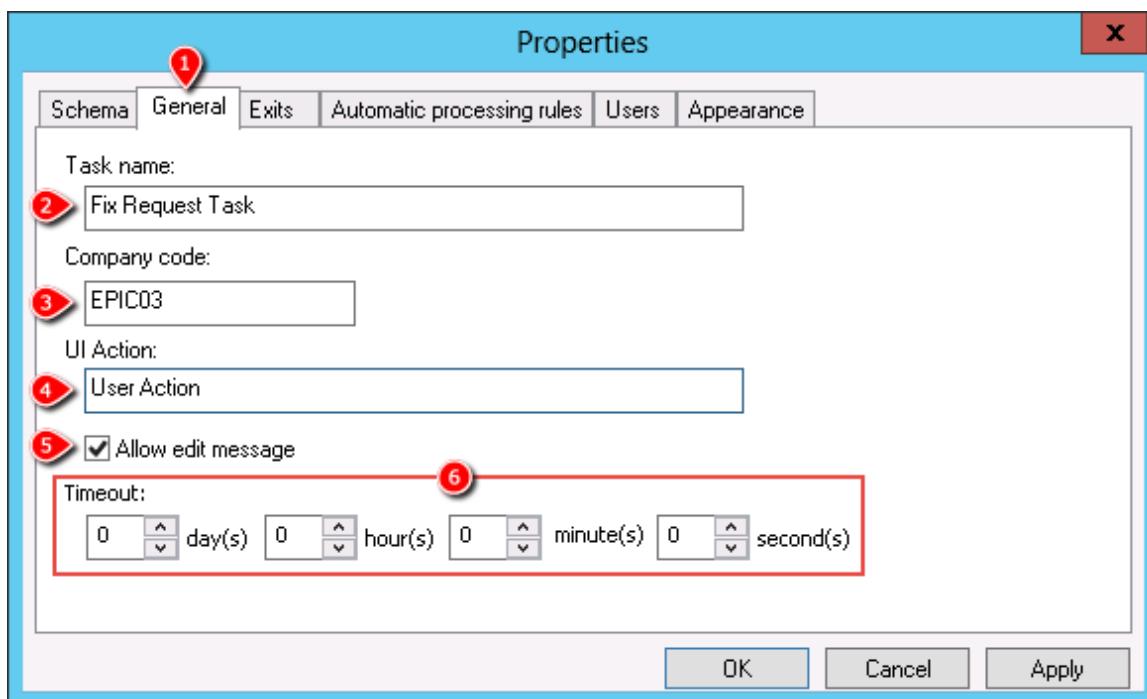
9. In the right pane, select a schema.

10. Click OK.

Remain in the **Properties** window.

Set Up Task General Properties

1. Open the **General** tab.



2. Enter the **Task name**.
3. In the **Company code** field, enter the company code.
4. In the **UI Action** field, enter the user action associated with the Task.
5. If you want to make the Task outgoing XML message available for editing in the Task Monitor, select the **Allow edit message** check box.
6. 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.

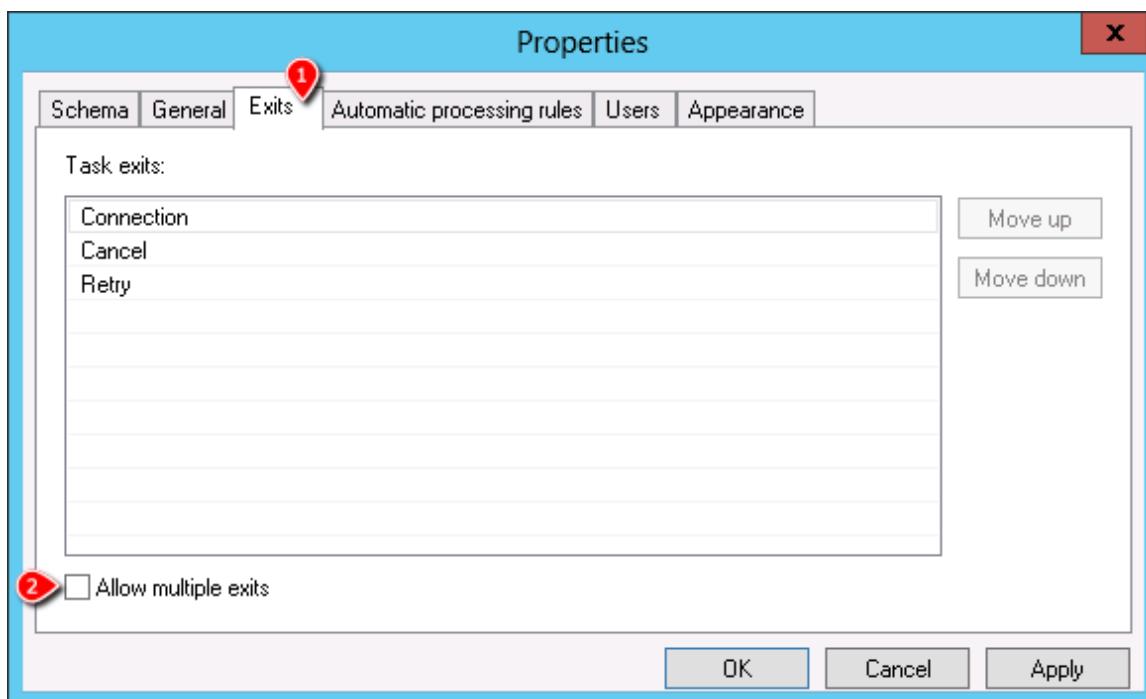
When the specified period elapses, the Task is considered expired, and the Process is resumed automatically using the default option for further action.

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

Remain in the **Properties** window.

Set Up Task Exits

1. Open the **Exits** tab to set the Task exits rules.



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

2. 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.

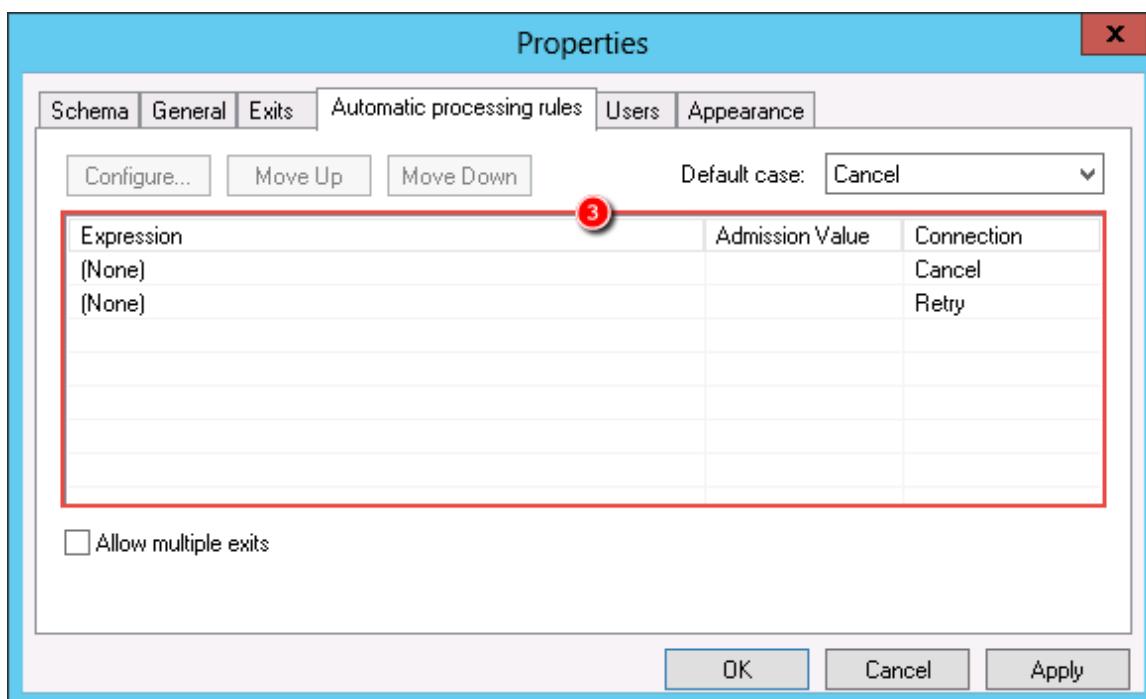
Each automatic rule is an XPath expression evaluated against the incoming document as either true or false. You can enter one rule for each outbound Connection leaving the Task activity. Documents can progress through each outbound Connection for which they satisfy the rule.

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.

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

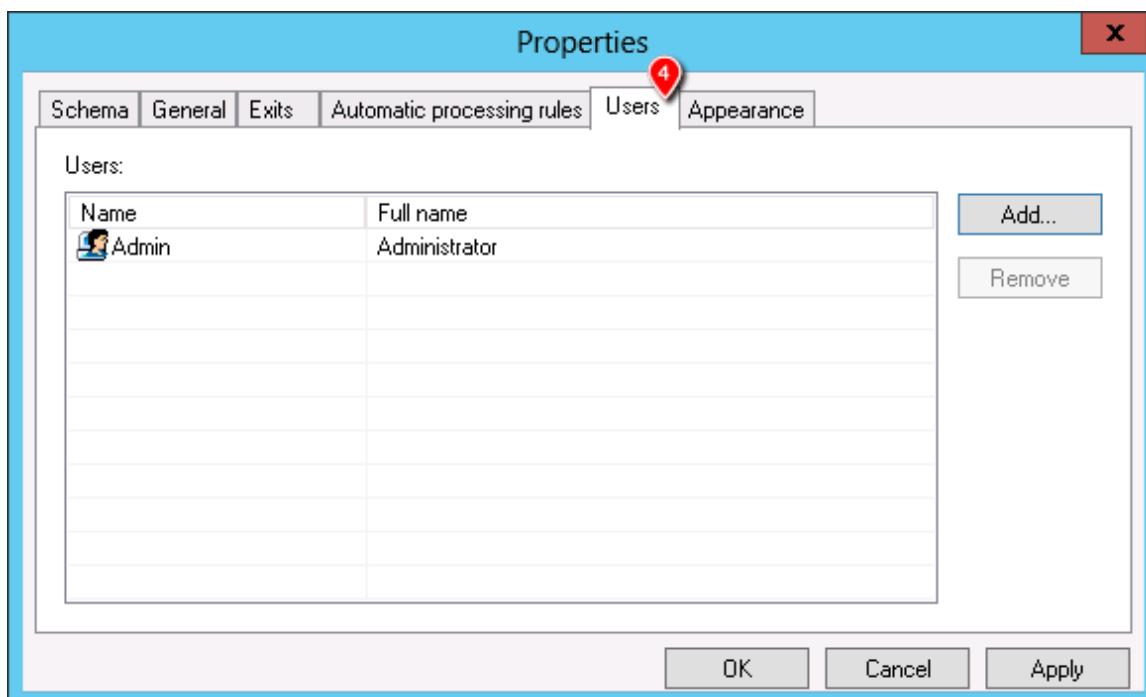
- **Expression** — The simplified form of XPath statement that is evaluated as logical expression. In the simplified form, namespaces are omitted and only the xml elements names are displayed.
- **Admission Value** — The value the rule expression evaluated value is compared with. If the two values are 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.

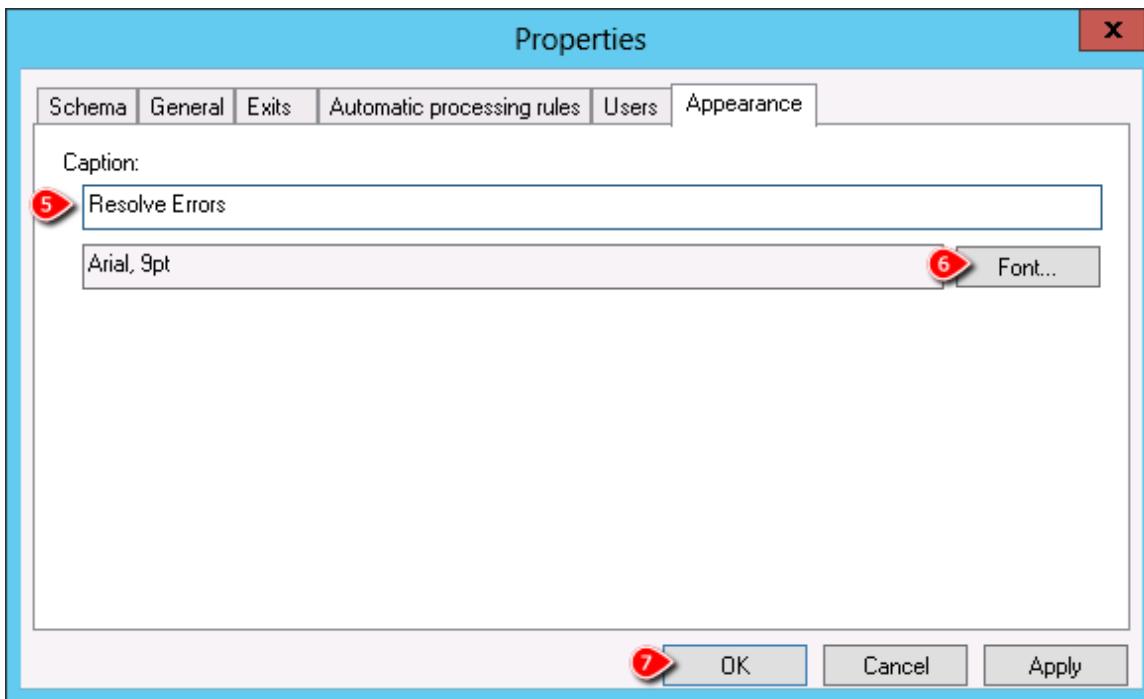


4. On the **Users** tab, set the user or user group to which the Task will be assigned.

The Task will display in the Task Monitor queue for the designated user or for a member of the designated group.



5. On the **Appearance** tab, enter the Task **Caption**.



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

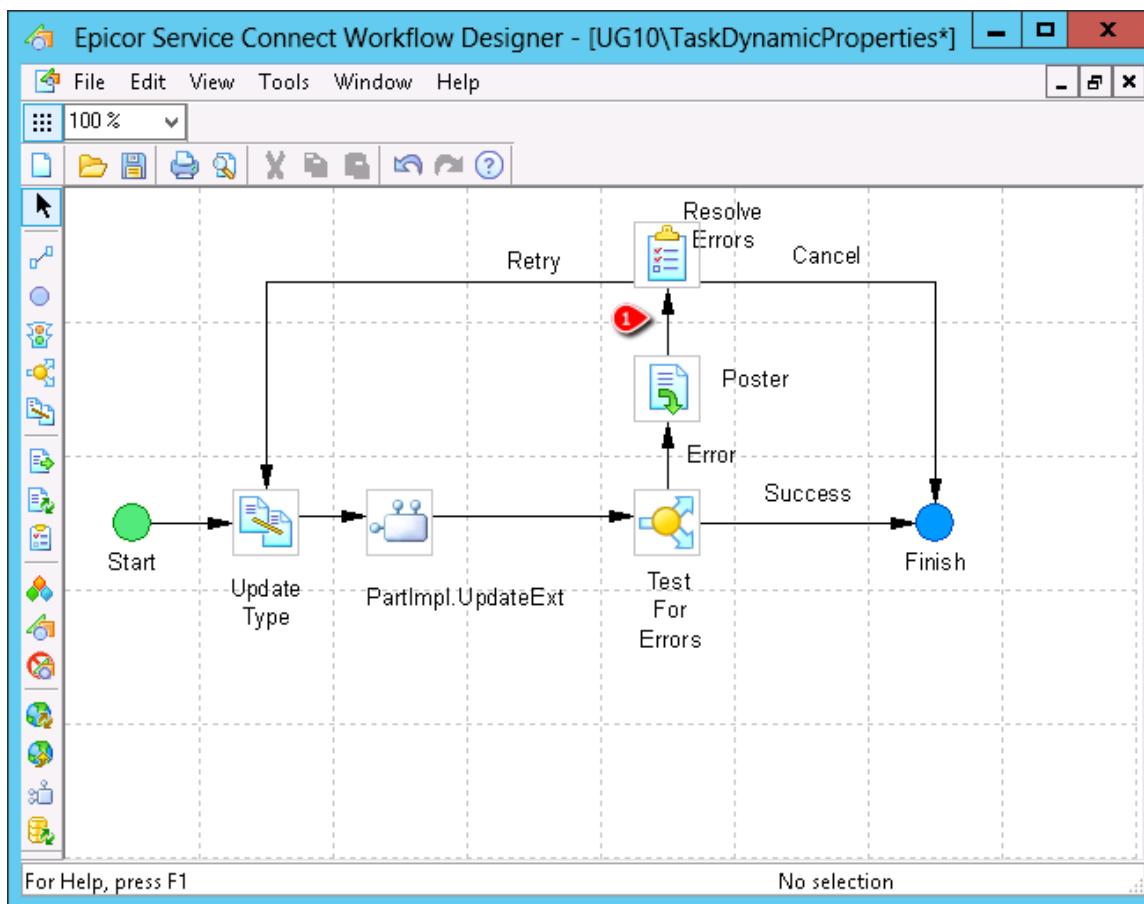
7. Click **OK**.

Set Up Task Dynamic Properties

You can define values for all the Task properties when you design your workflow in the Workflow Designer. This is **static** way to configure a Task. You can also configure Task element in a **dynamic** way. That means that some of Task properties can be redefined during process execution. In this case, elements in incoming XML message determine the values of Task properties. These elements are found in **ctx** section of the message. Task Dynamic properties are initialized via the Conversion element.

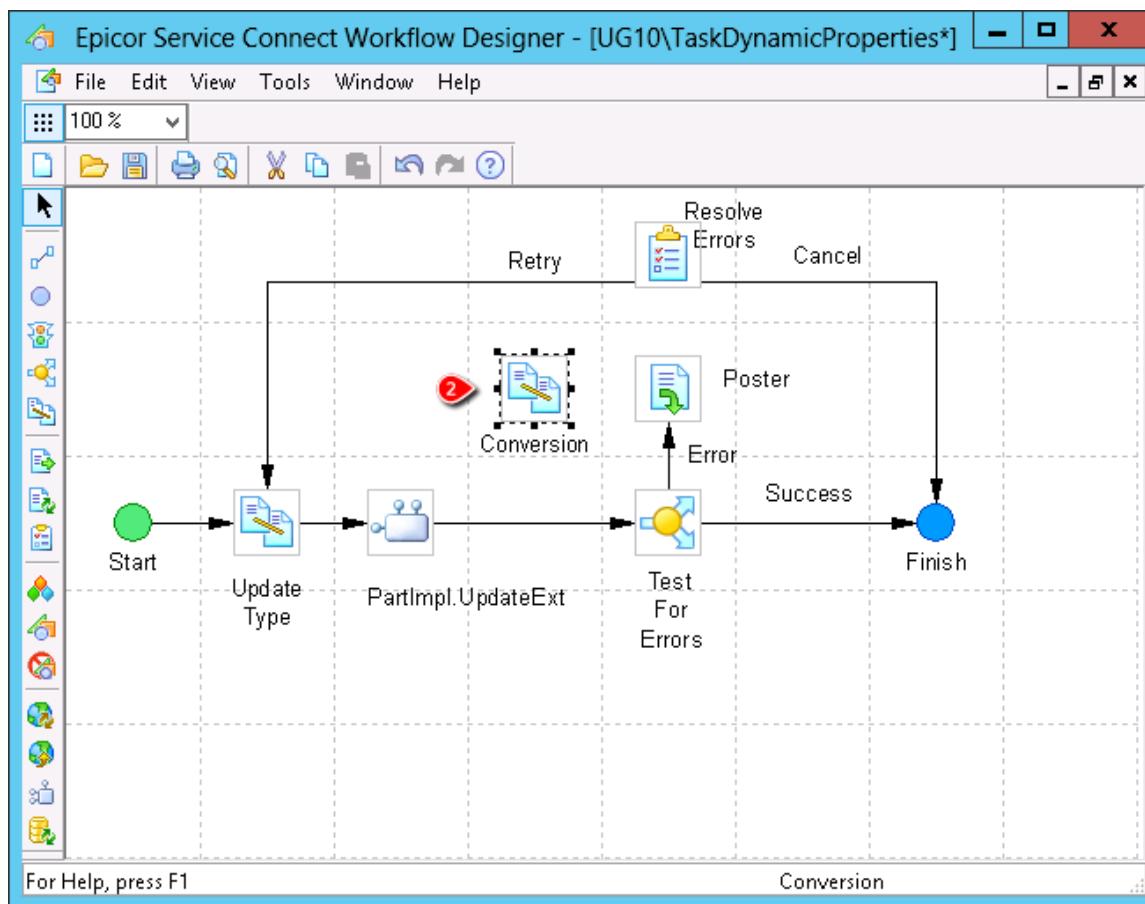
To configure Task element dynamically:

1. In the **Workflow design area**, remove the Connection between the Task and the previous workflow element.

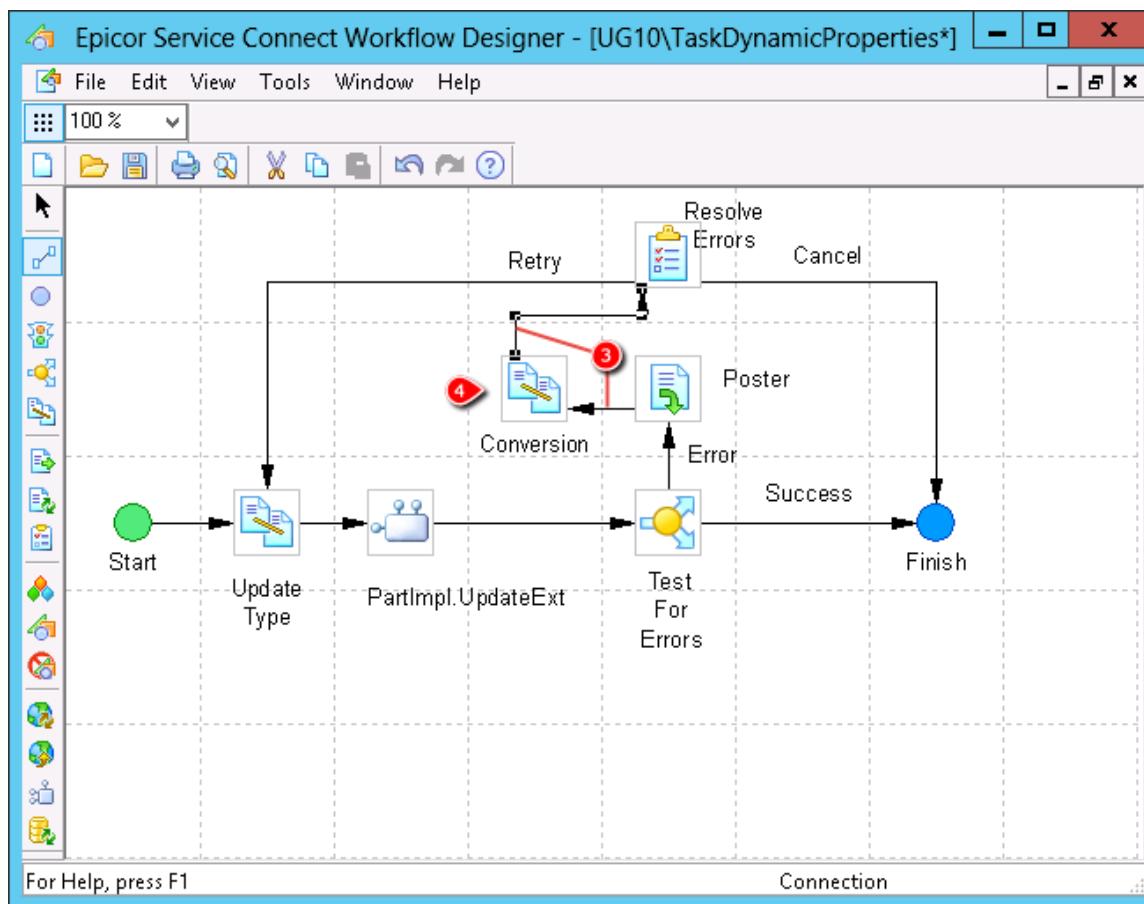


2. Insert the Conversion element into the workflow before the Task element.

A Conversion can find a Task element with dynamic properties even if it is divided from the Task by the Splitter, Condition or Choice element.

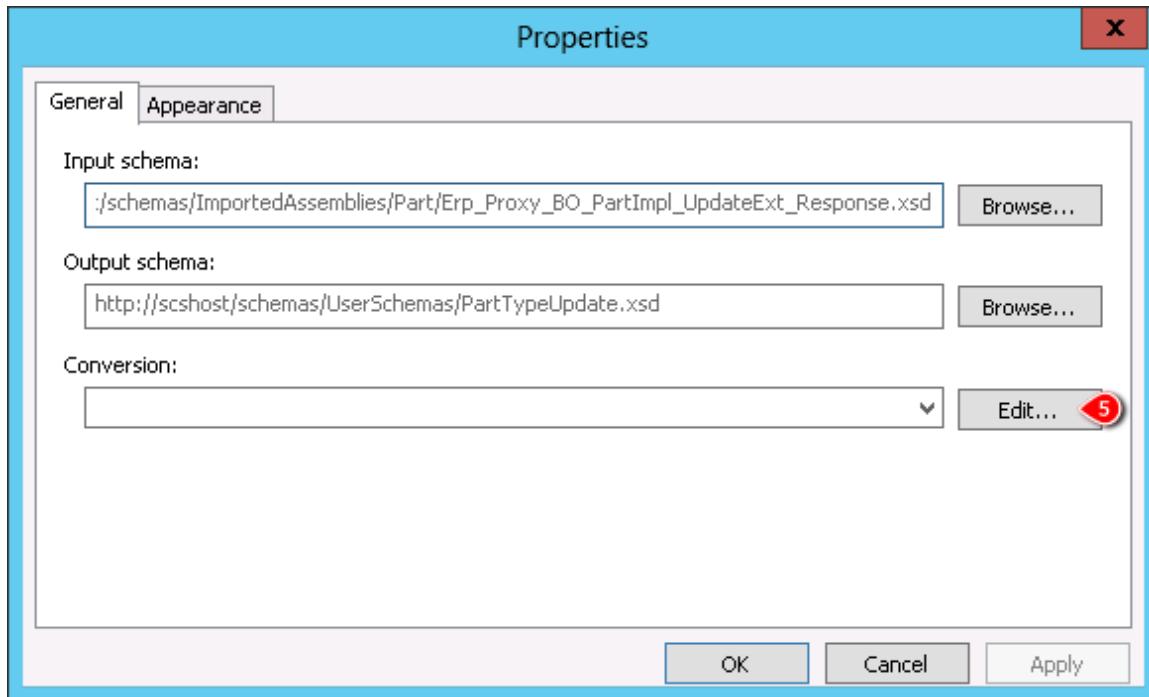


3. Draw Connections from the previous workflow element to the Conversion and from the Conversion to the Task.



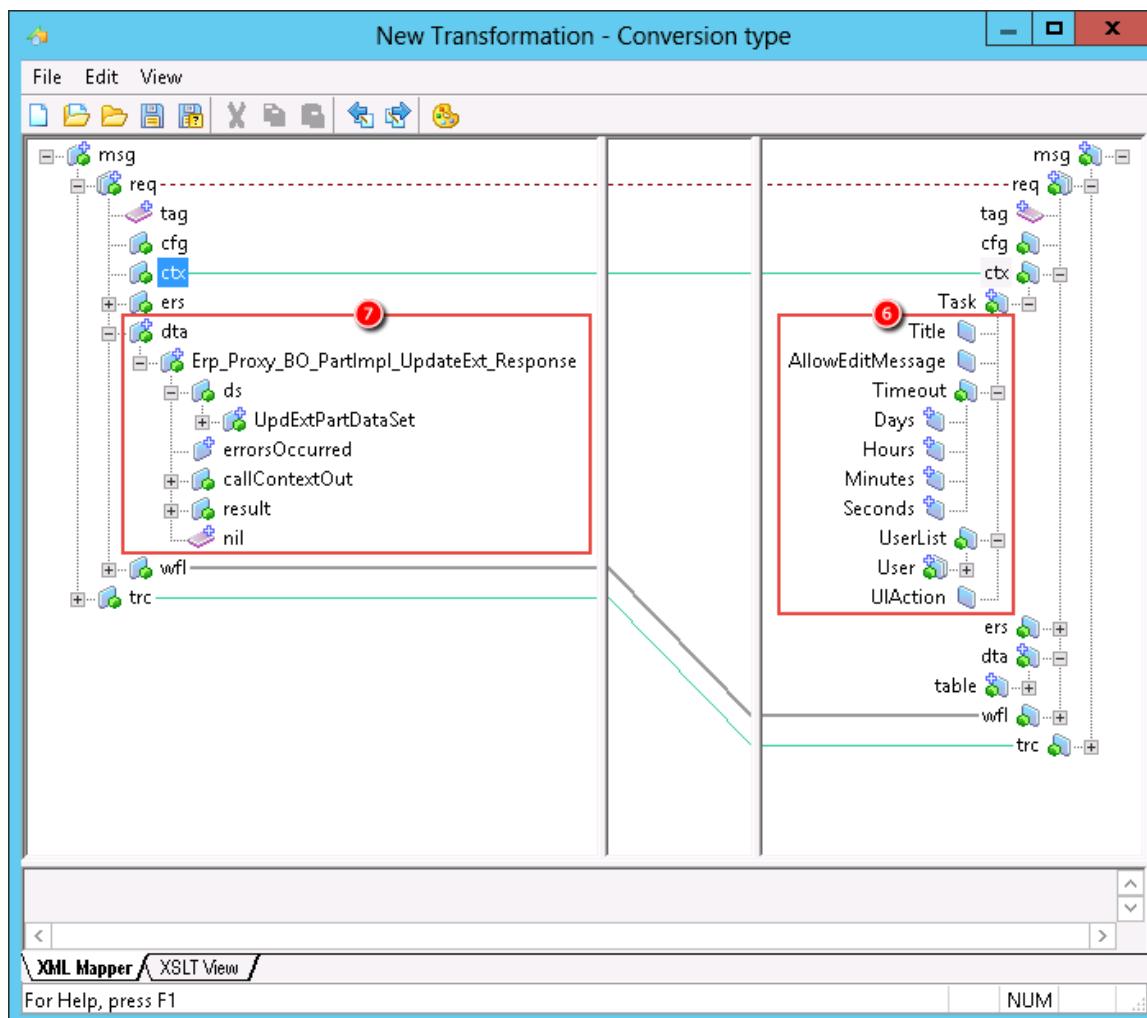
4. Double-click the Conversion.

The **Properties** dialog is displayed.



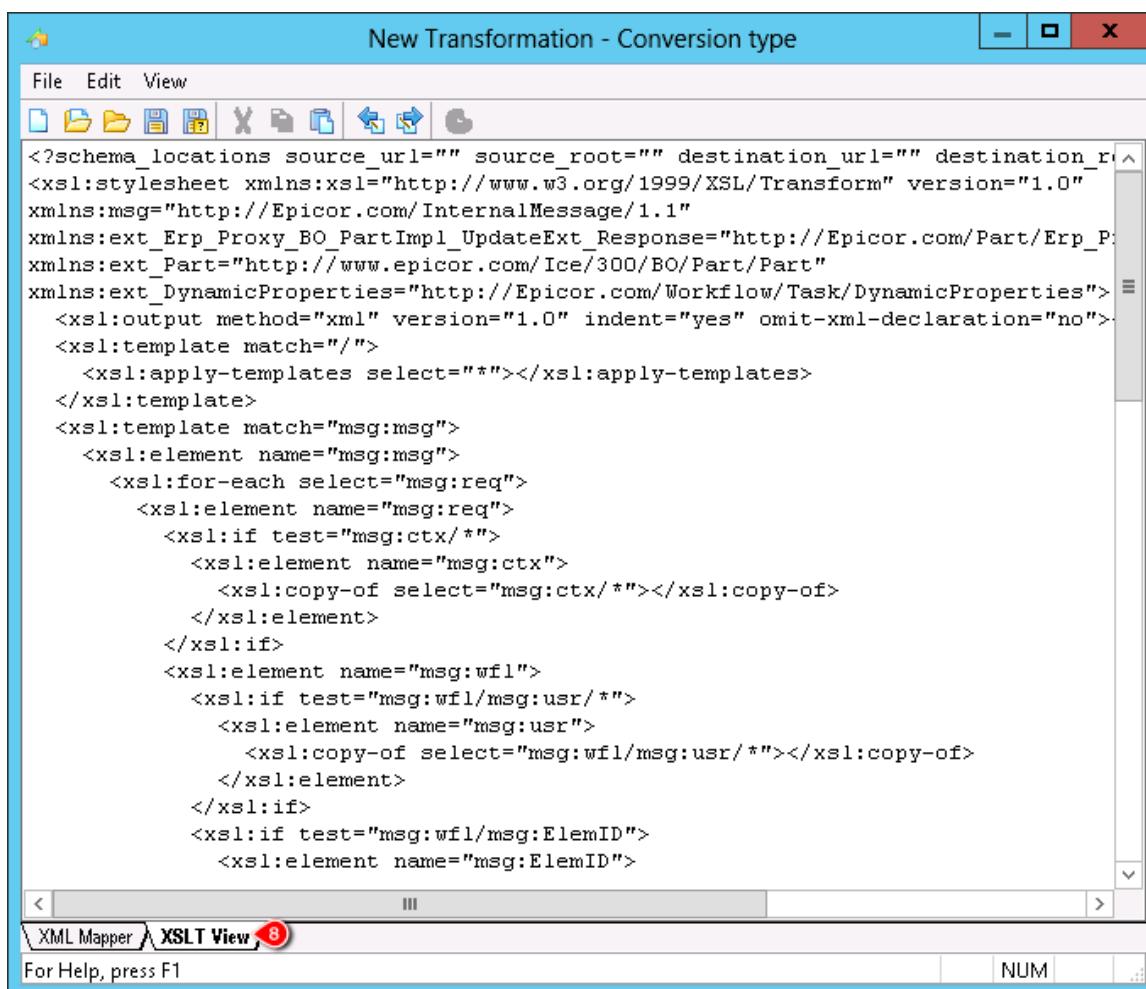
5. Click the **Edit** button next to the **Conversion** field.

XML Mapper window is displayed.



6. In the Mapper right pane, unfold msg > reg > ctx > Task nodes.
All the Task properties that can be set dynamically are displayed.
7. Map message data or assign literal values onto the Task properties.

8. You can edit the xml in XSLT view as well.



For instance, the following format is accepted for user group:

Task Configuration:

```

<UserList>
<User qualifier="USER_GROUP_ID">
<ID>10</ID>
<Name>ADMINs</Name>
<FullName>Root/ADMINs</FullName>
</User>
</UserList>

```

The following format is accepted for a single user:

```

<User qualifier="SINGLE_USER_ID">
<ID>2</ID>
<Name>epicor</Name>
<FullName>epicor</FullName>
</User>

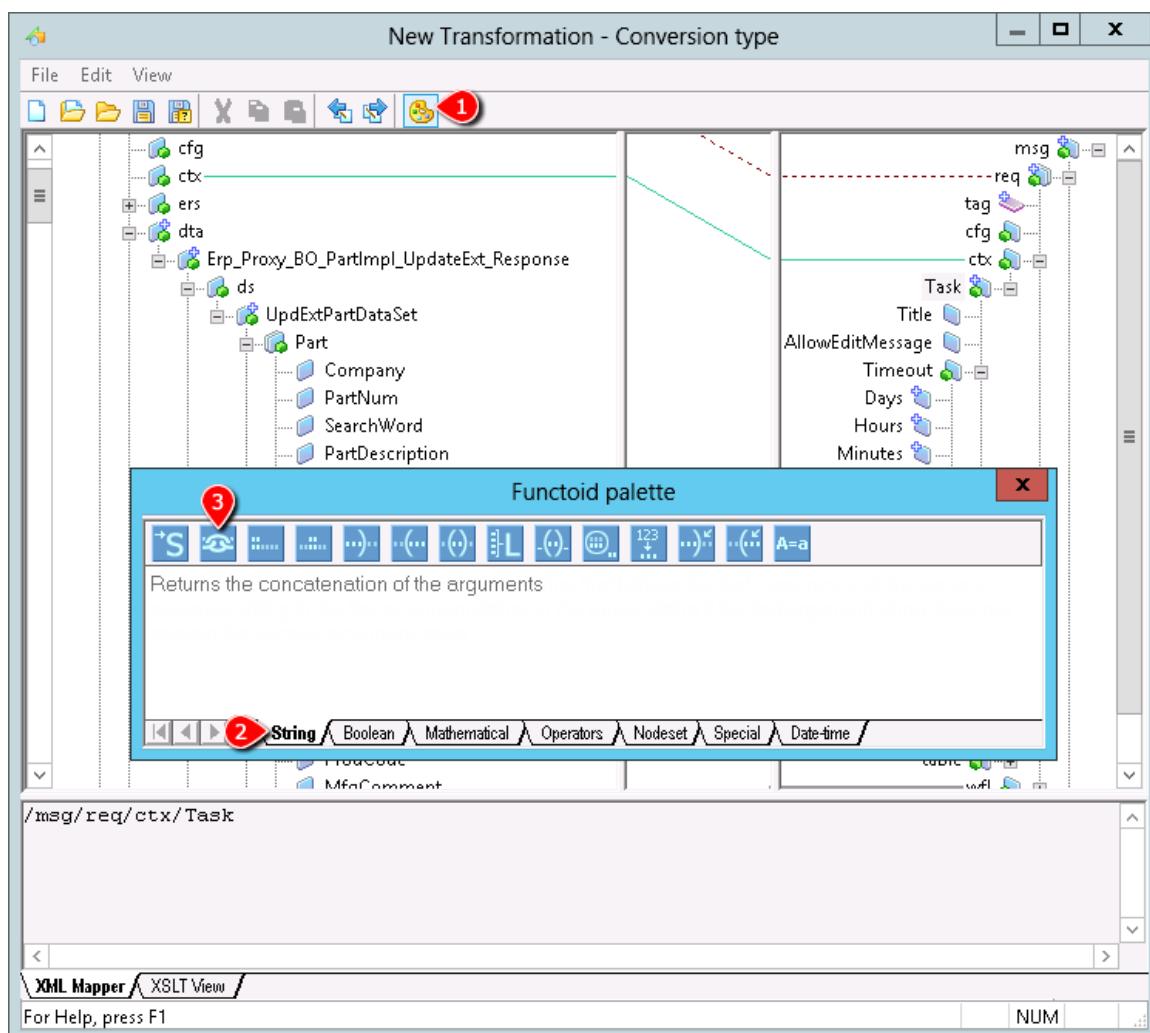
```

ID, Name, and FullName parameters for a user or user group can be found in User Properties under **Security** > **Users** node of Administration Console.

Task Dynamic Configuration Example

Following is an example of how to configure the Task title to show part number to be updated.

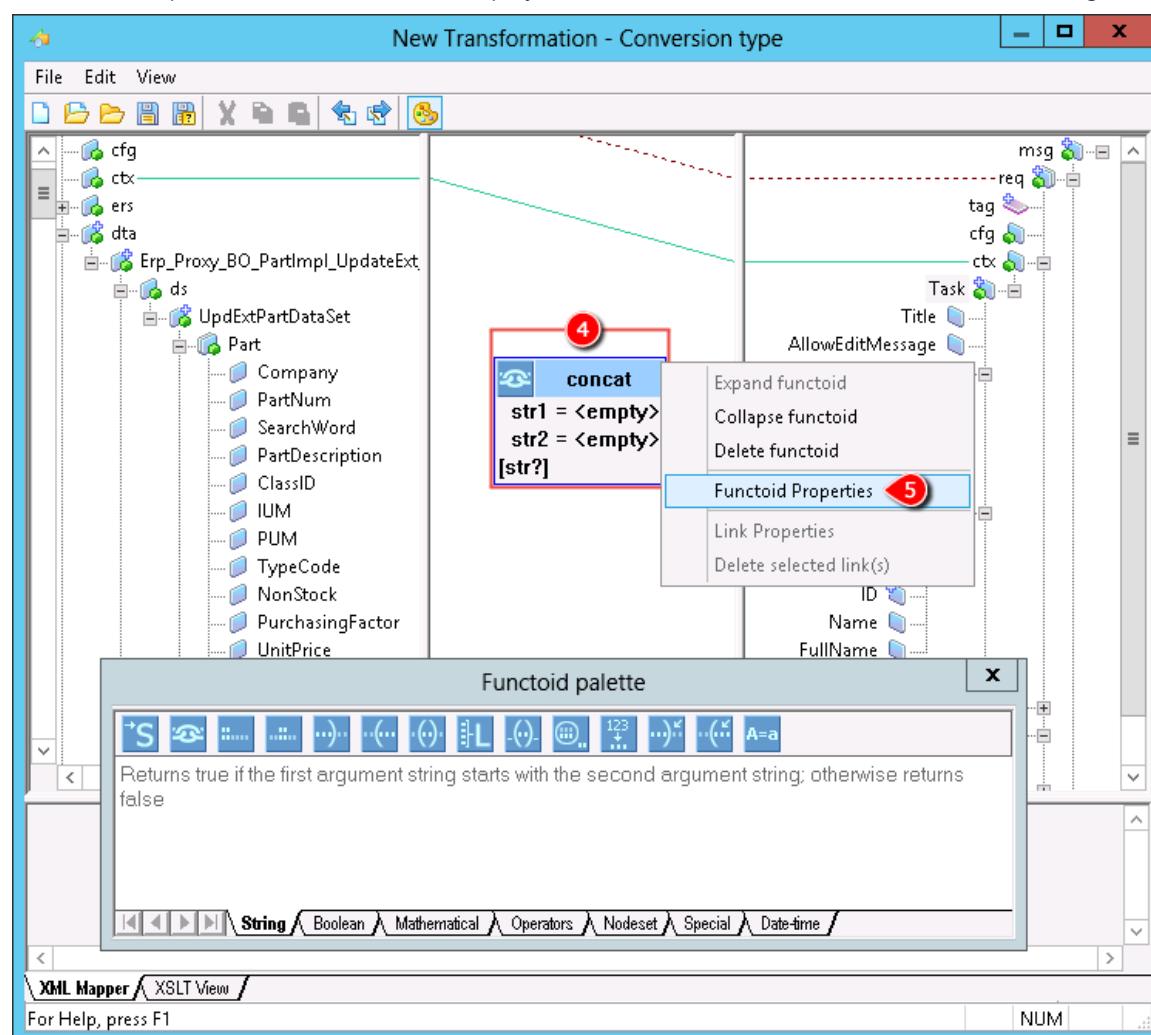
1. In the XML Mapper, on the toolbar, click the **Functoid Palette** button.



The **Functoid Palette** is displayed.

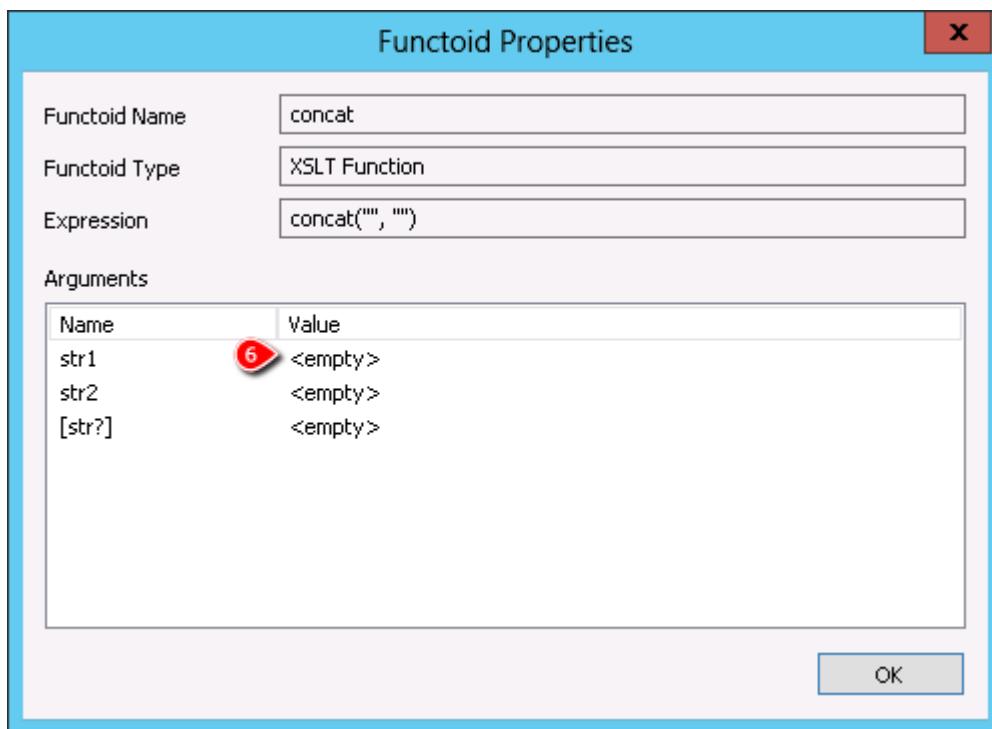
2. On the **Functoid Palette**, verify the **String** tab is open.
3. Select the **concat** functoid.
4. Drag the functoid icon from the palette to the Mapper middle pane.

In the middle pane, the functoid box is displayed. It contains the functoid name and a list of arguments.

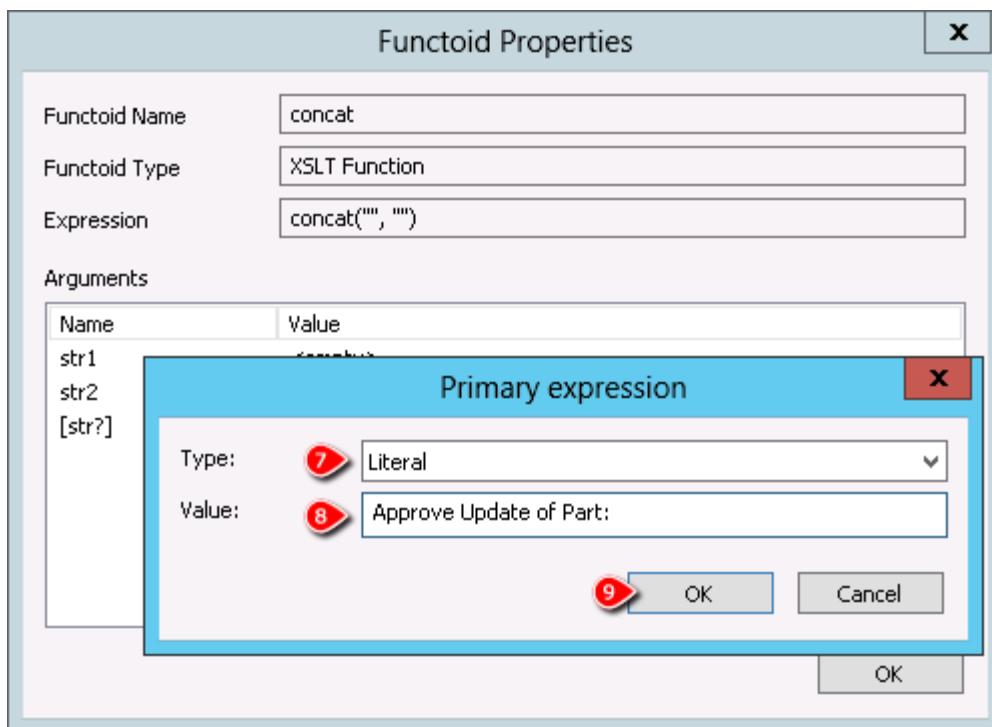


5. Right-click the functoid, and select **Functoid Properties**.

6. In the **Functoid properties** window, in the **Arguments** table, double-click **str1** line (the first argument of the functoid) to assign a constant value to it.



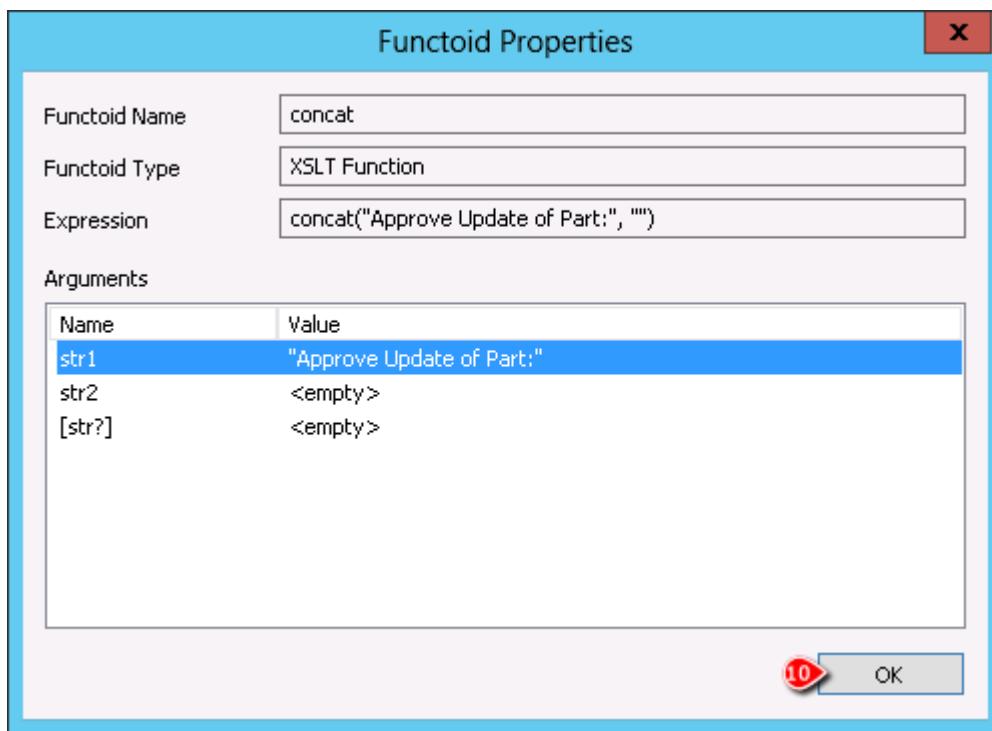
7. In the **Primary expression** window, verify the **Type** field displays **Literal**.



8. In the **Value** field, enter **Approve Update of Part:**

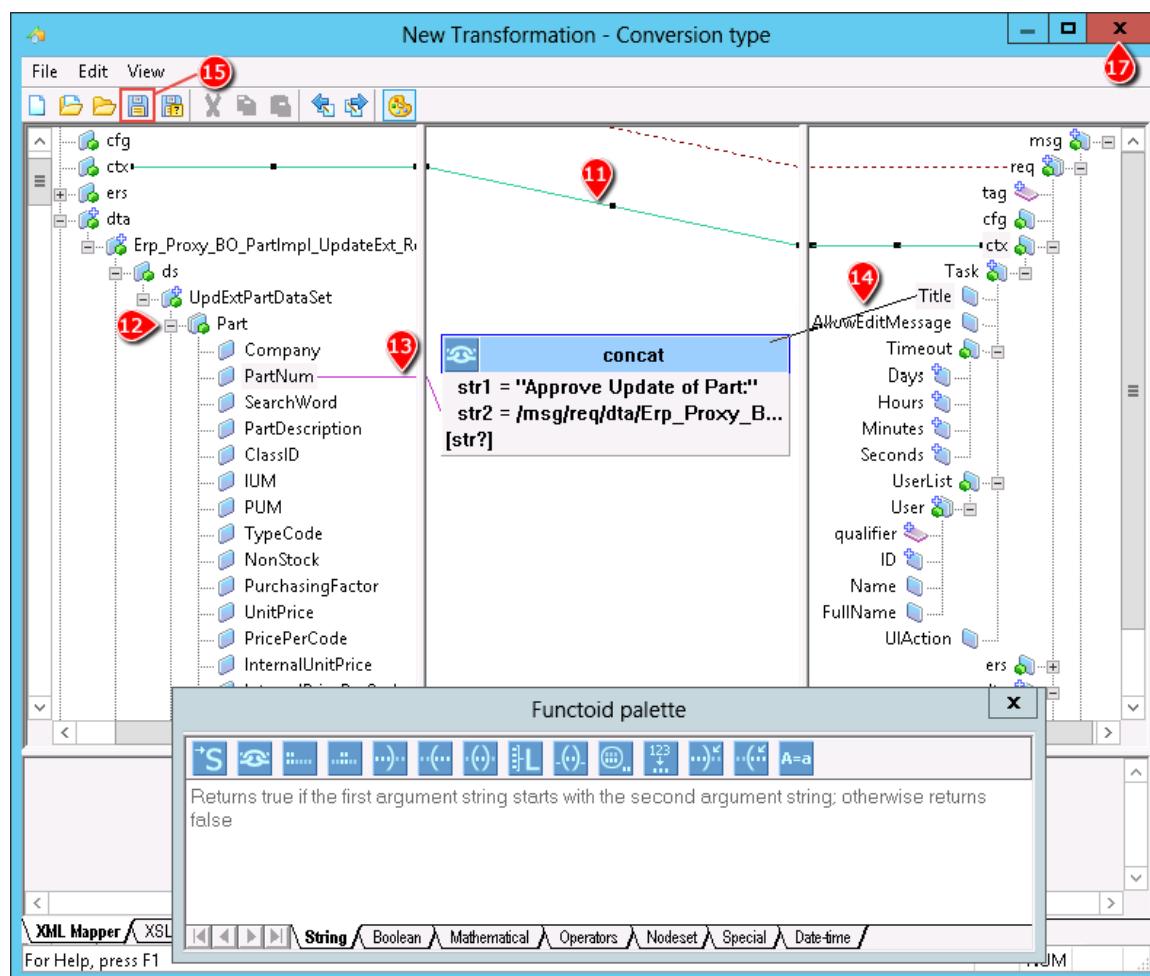
9. Click **OK**.

10. In the **Functoid Properties** window, click **OK**.



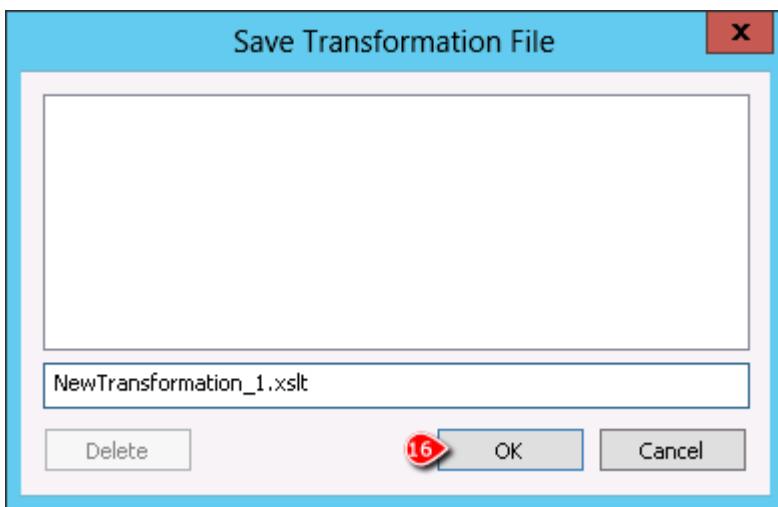
11. In the XML Mapper, select the link between the **ctx** nodes, and hit **Delete**.
To the confirmation message, click Yes.

12. In the XML Mapper left pane, expand the **Part** node.



13. Drag the connection from the PartNum node in the left pane to the functoid second argument in the middle pane.
14. Drag the connection from the functoid title to the **Title** node in the right pane.
15. Click **Save**.

16. In the **Save Transformation File** window, click **OK**.



17. Close the XML Mapper window.

Poster



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 workflow elements are functioning as intended.

Poster Example

An incoming document contains a new customer record, which is added to the target database using a .Net 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.

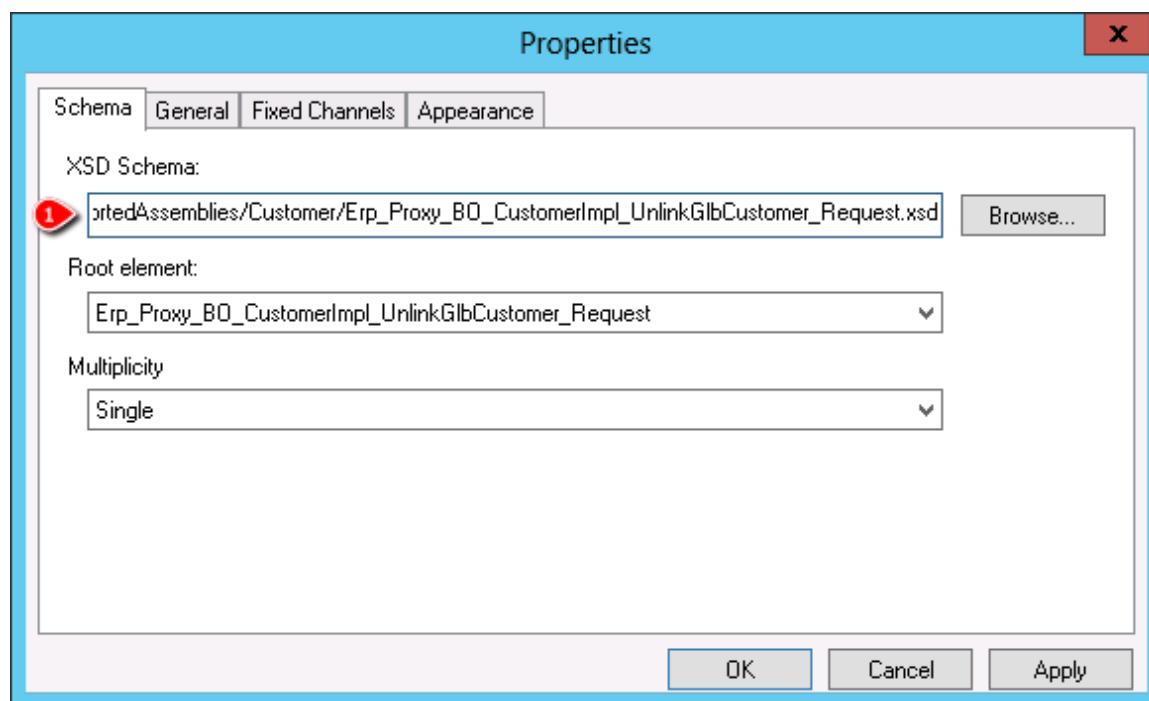
Set Up a Poster

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 Connectivity Components Chapter for more information.

To set up a Poster activity:

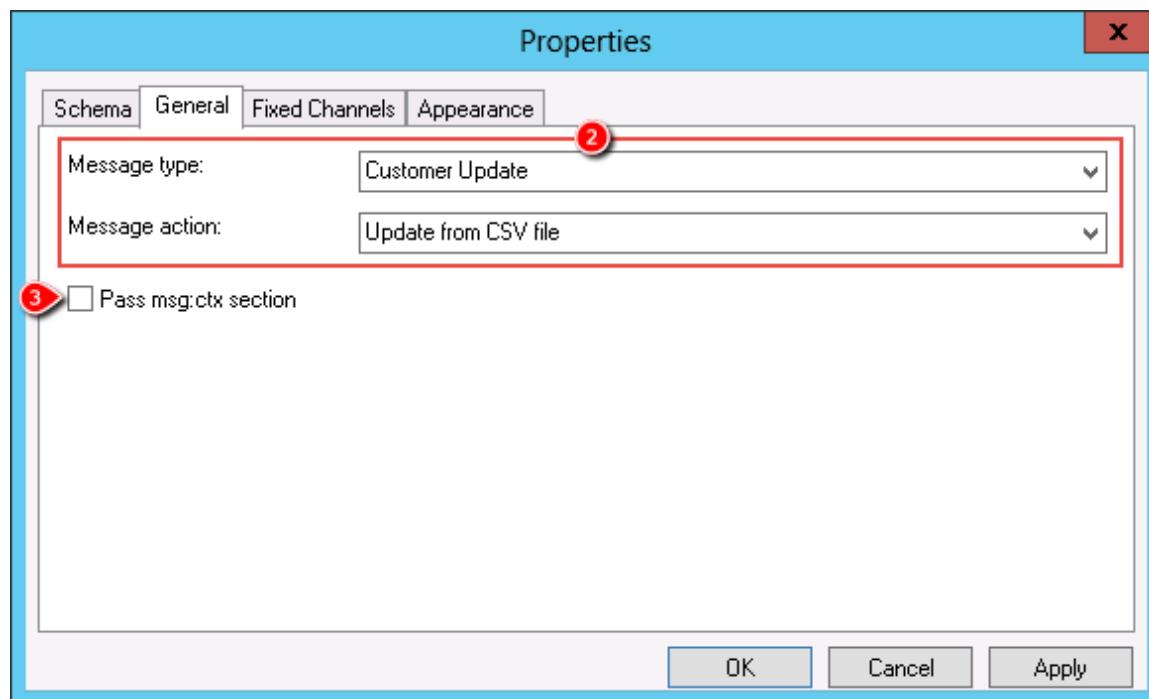
1. Set the **XSD Schema** to indicate which document can be used as part of the email.

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



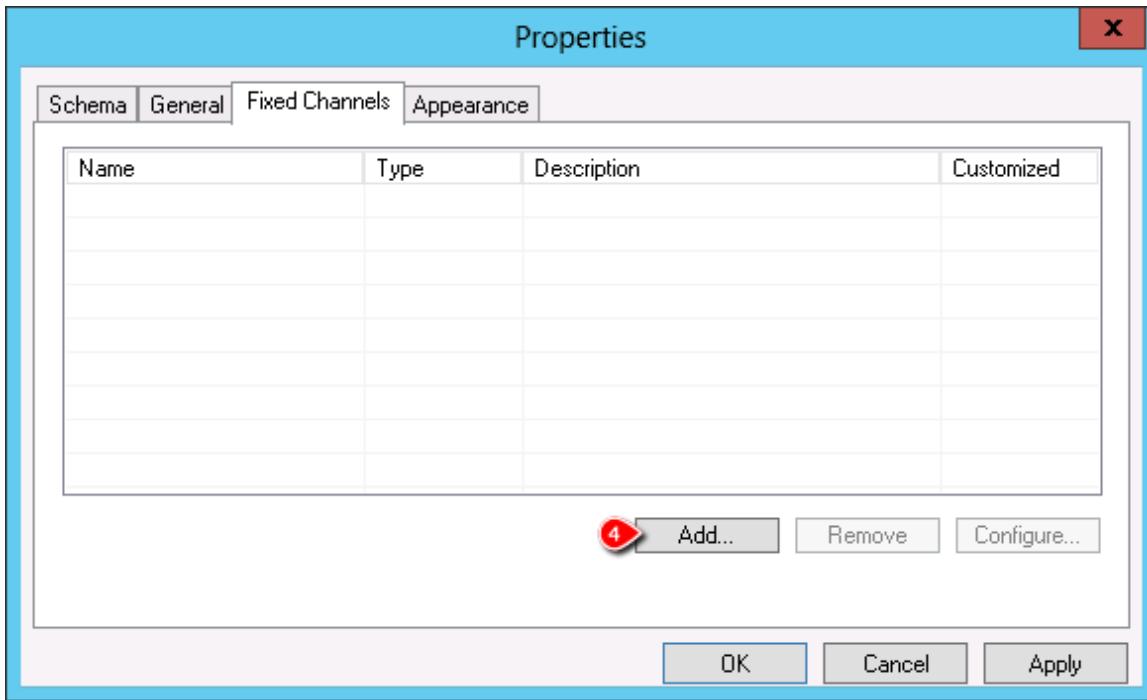
In this example, the XSD schema corresponds to the new customer information.

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

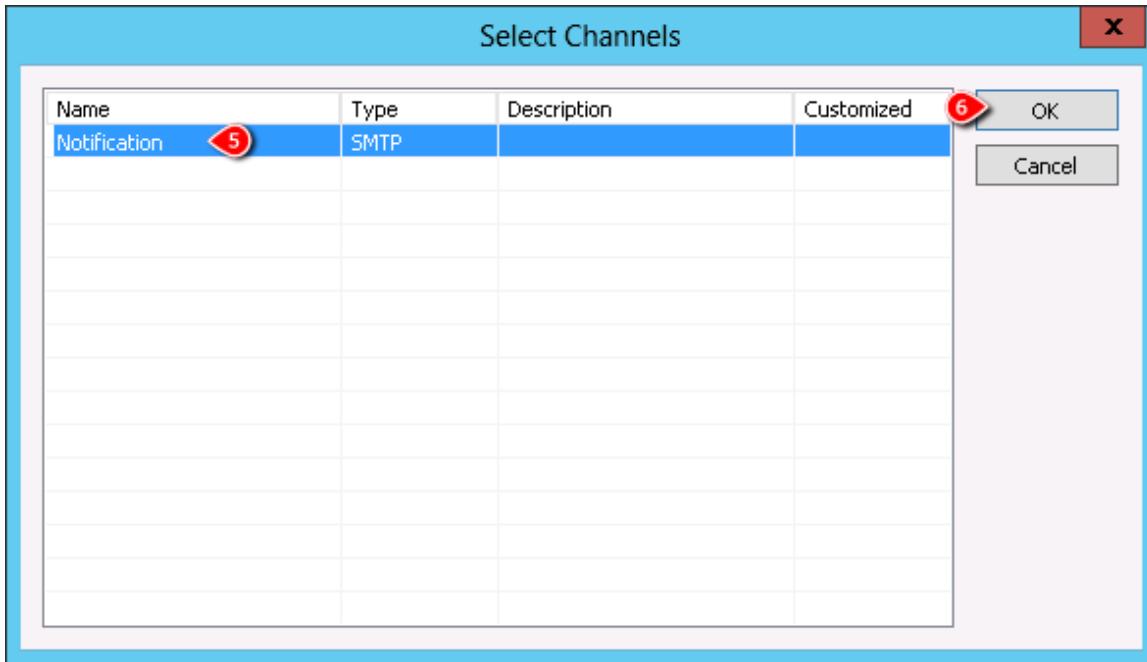


3. Select the **Pass msg:ctx section** check box to send the Poster configuration to the workflow. Settings in this node take precedence over the other workflow element configuration settings.

4. On the **Fixed Channels** tab, click **Add** to add an Output Channel that is configured to use SMTP.

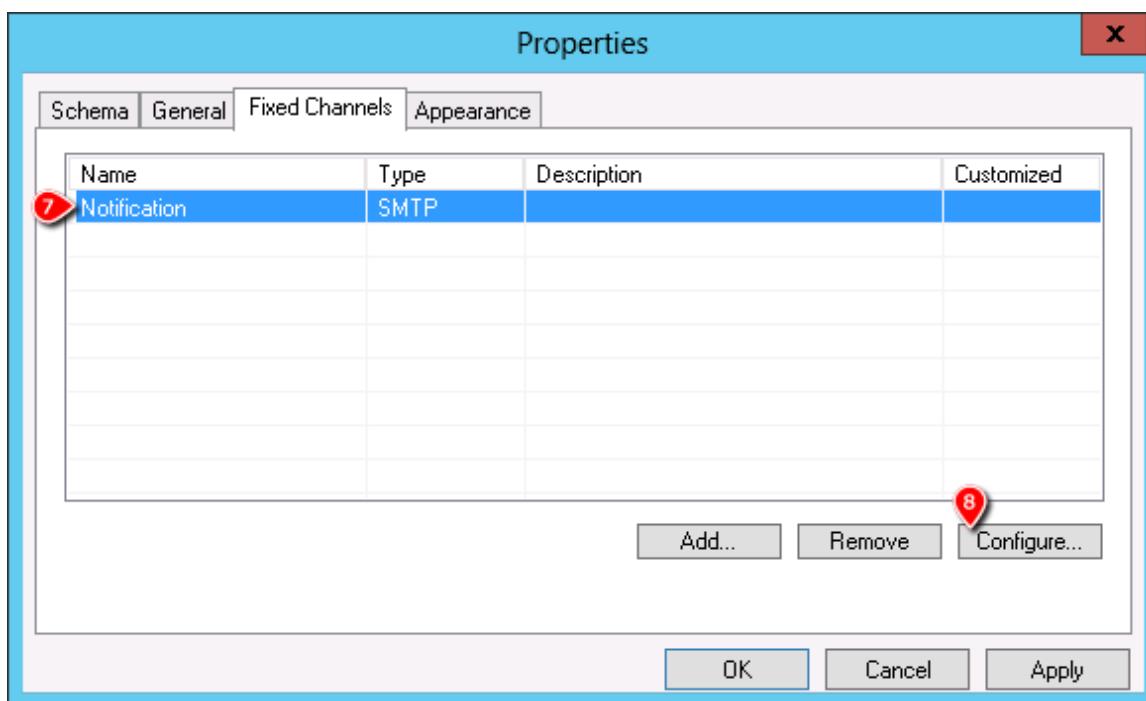


5. In the **Select Channels** window, select the SMTP output channel.



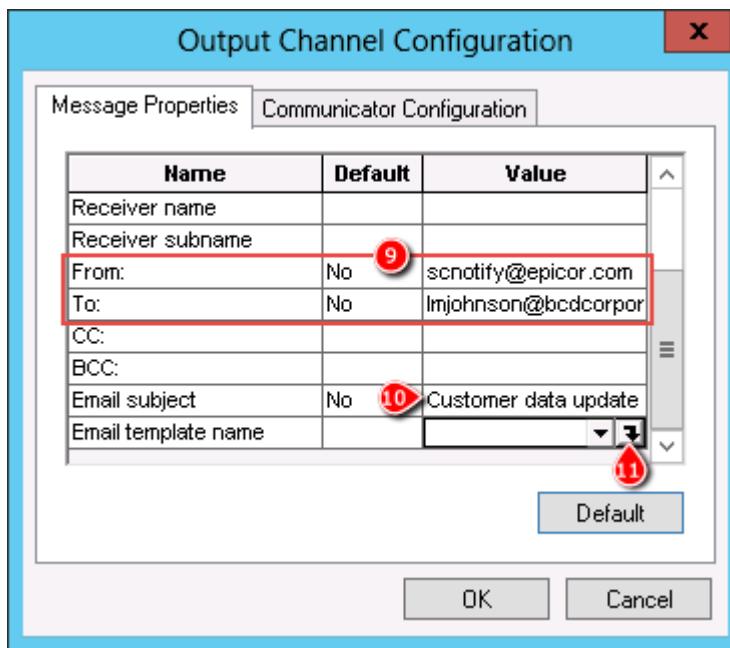
6. Click **OK**.

7. On the **Fixed Channels** tab, select the channel you've just added.



8. Click **Configure**.

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



10. Enter an **Email subject**.

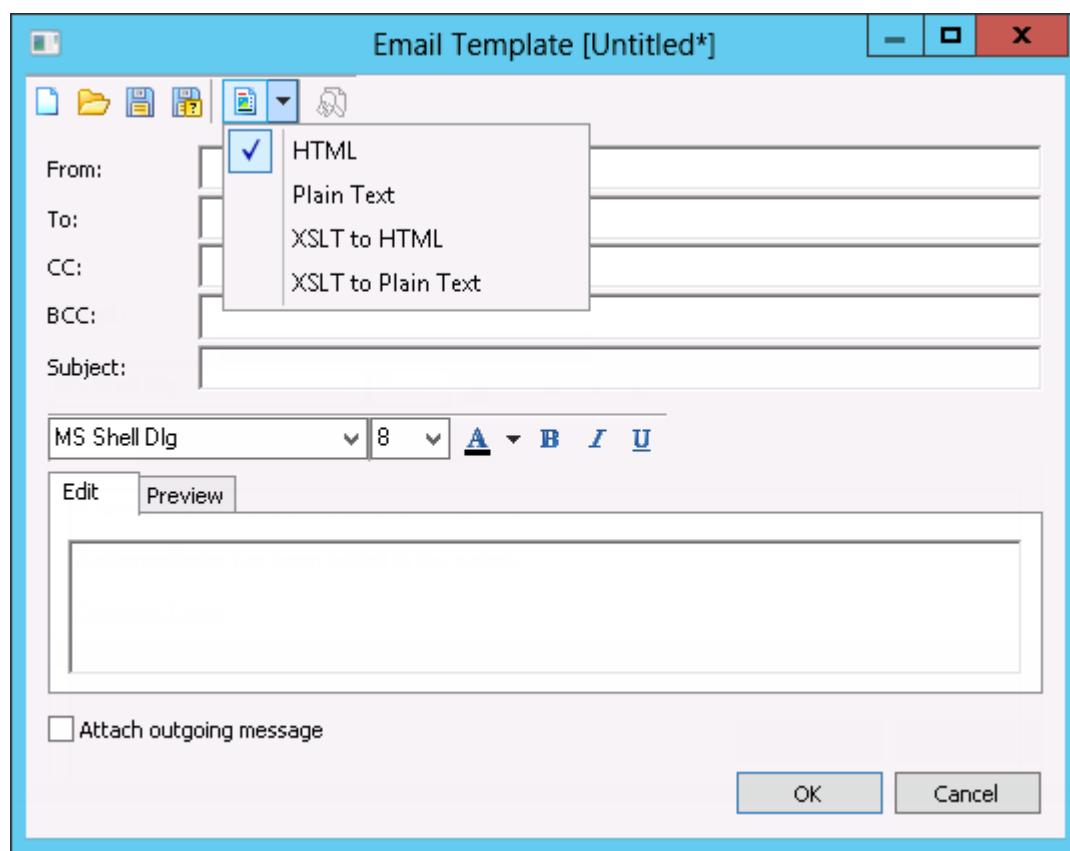
11. Click the **Email template name** field, and then click the **down arrow** button on the far right of the **Email template name** field to open the Email Template builder.

The **Email Template** window displays. Remain in this window.

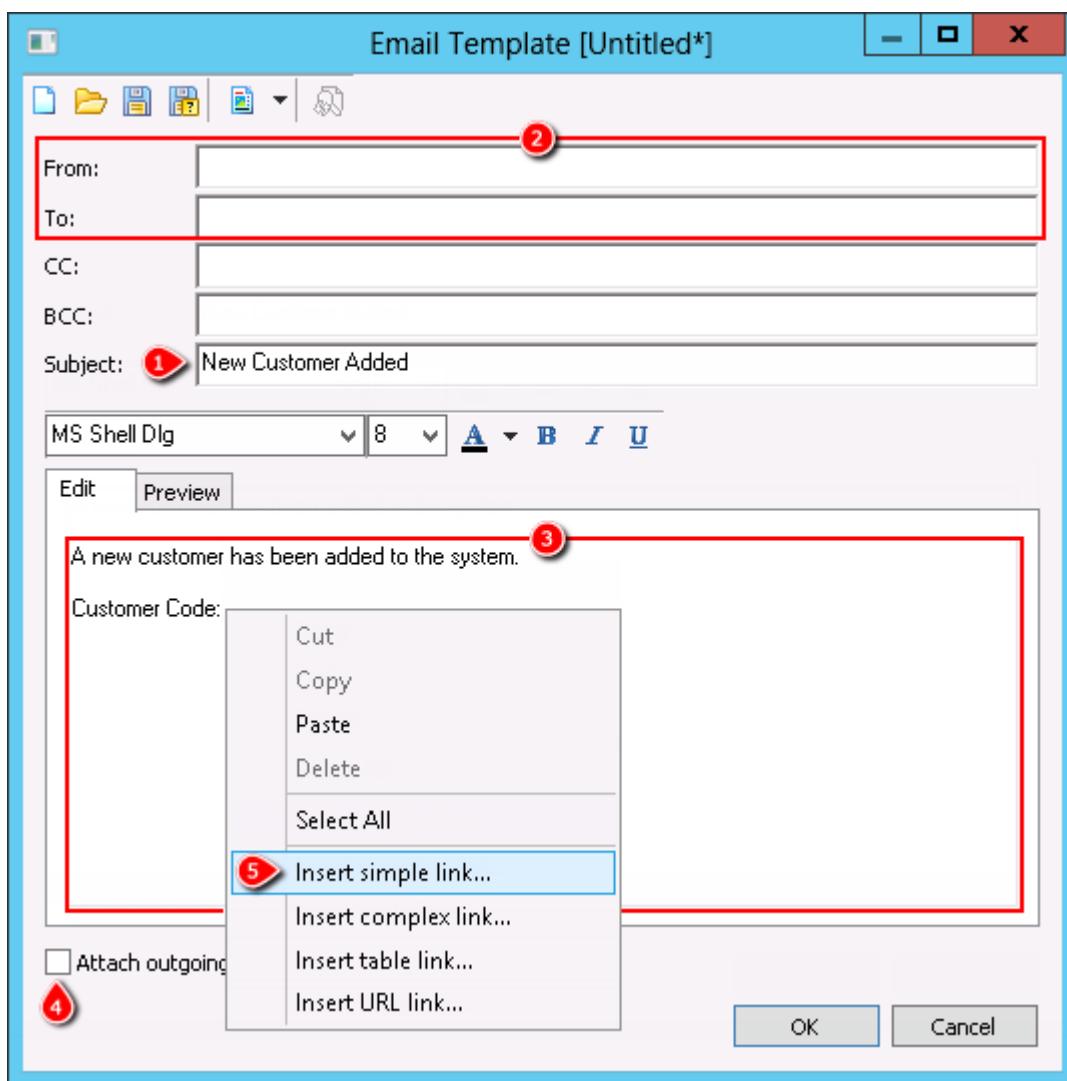
Build Email Template

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.



1. In the **Email Template** window, enter a **Subject**.



2. Notice the **From** and **To** fields are empty.

The information for these 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.

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

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

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

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

In HTML mode, more link types are available for insertion:

- **Table link**

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.

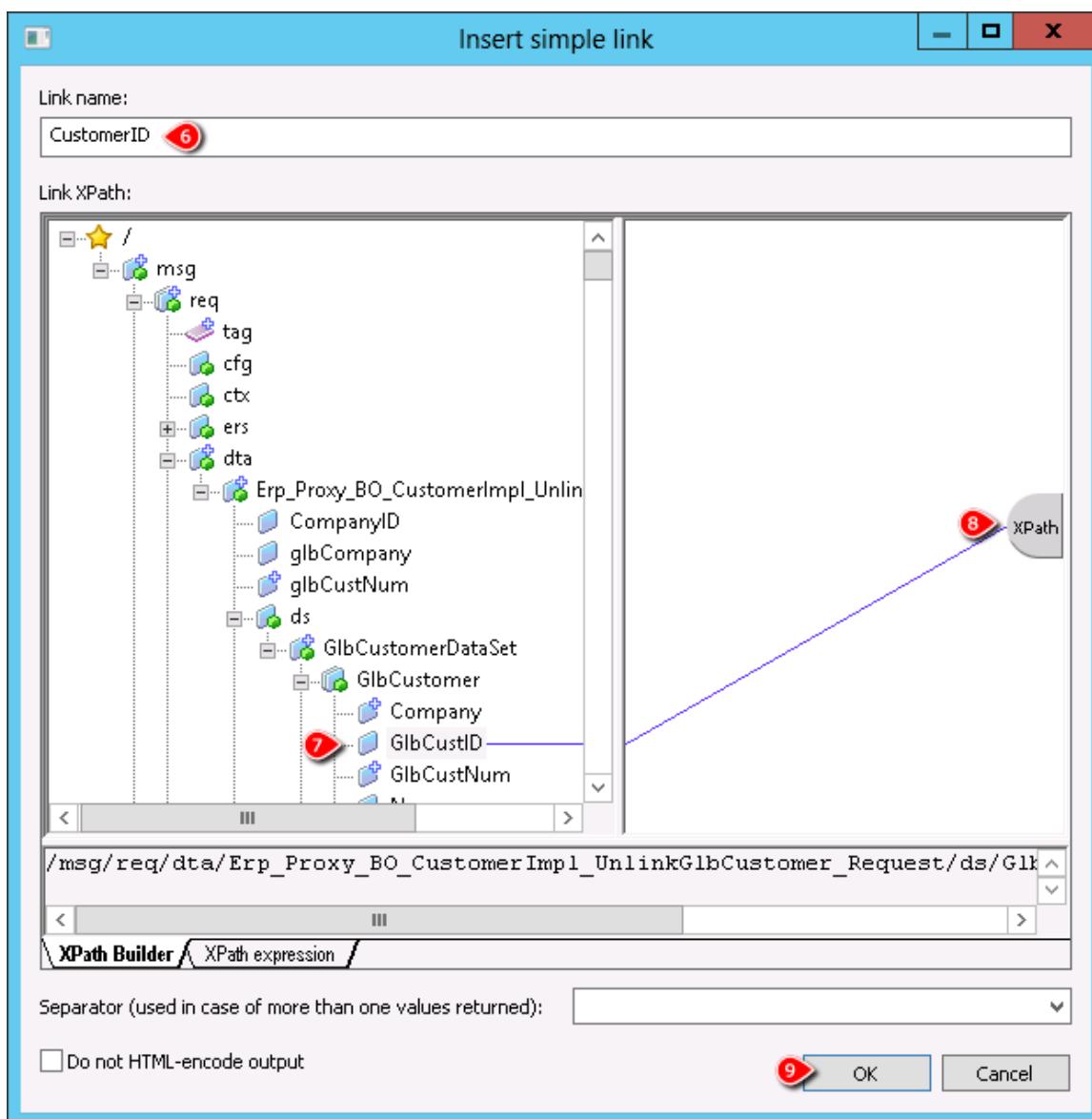
- **Complex link**

Complex links are useful for documents that contain a collection of items. For example, the document can represent 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 example, select a separator to appear between the items in the output. You can select either TAB or CRLF (Carriage Return/Line Feed).

- **URL link**

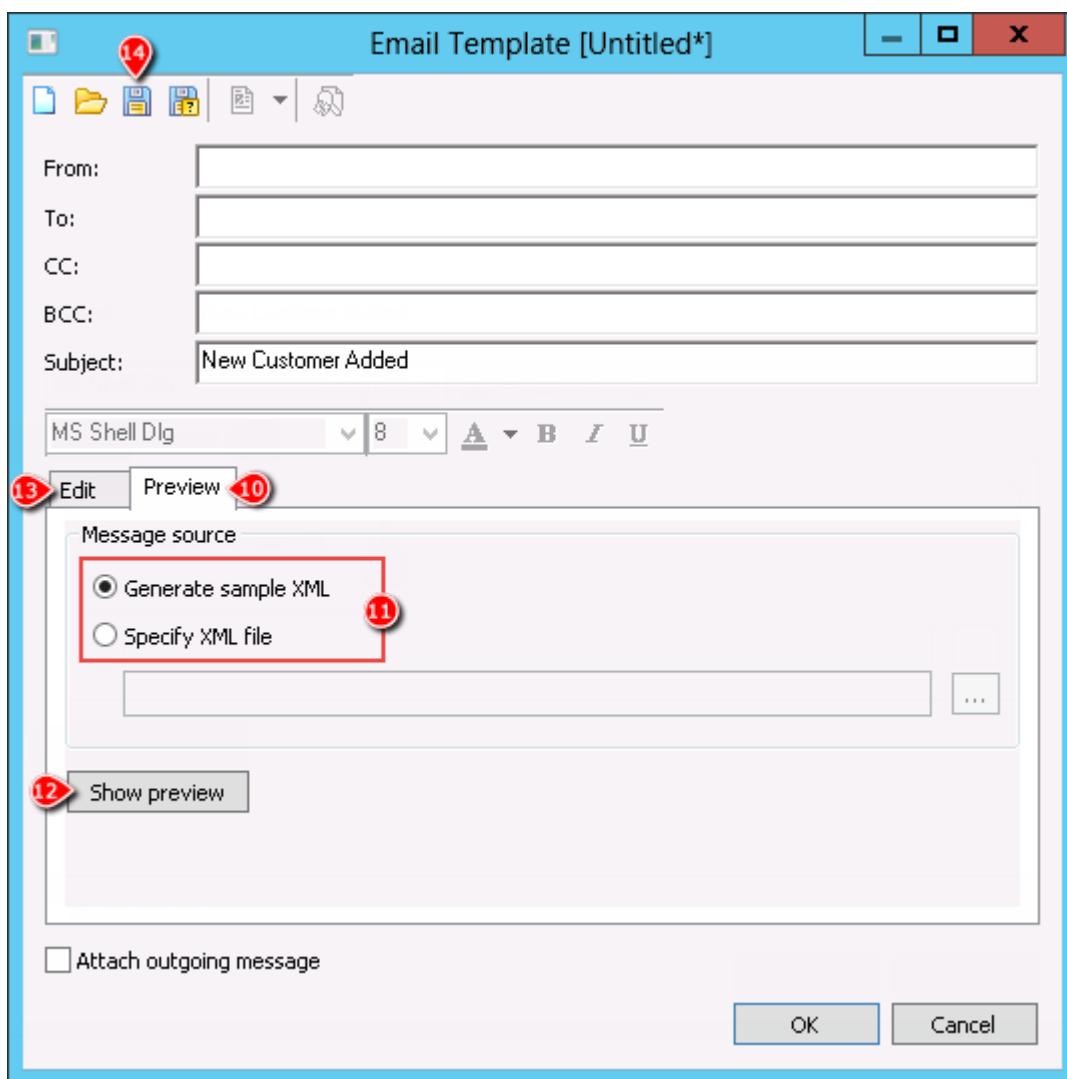
Use the URL link to construct hypertext links when you create an email template for emails based on html. With this type of link you can hide the long URL and display a short word instead. You can define URL link name and value statically at design time, or dynamically if XPath is used to evaluate the value at run time.

6. In the **Insert simple link** window, enter a **Link name**.



7. In the **tree view** in the left pane, expand the schema nodes until you locate the value you want to insert into the body of the email.
8. Drag the link from the selected node and drop it onto the **XPath** marker on the right pane.
9. Click **OK**.

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



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

12. 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.

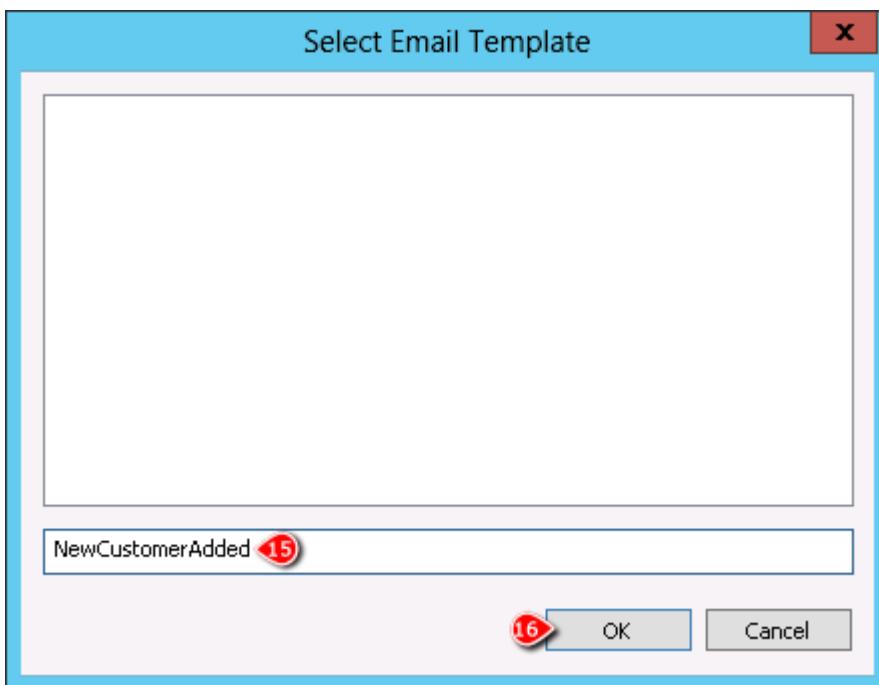
In message preview, all links are replaced with their values, extracted from the source XML provided. In HTML mode, any table links are rendered as tables as well. In XSLT editing modes, preview contains the result of the specified XSLT transformation applied to the source XML provided, either in plain 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.

13. On the **Edit** tab, continue to add information from the workflow to the Email Template.

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

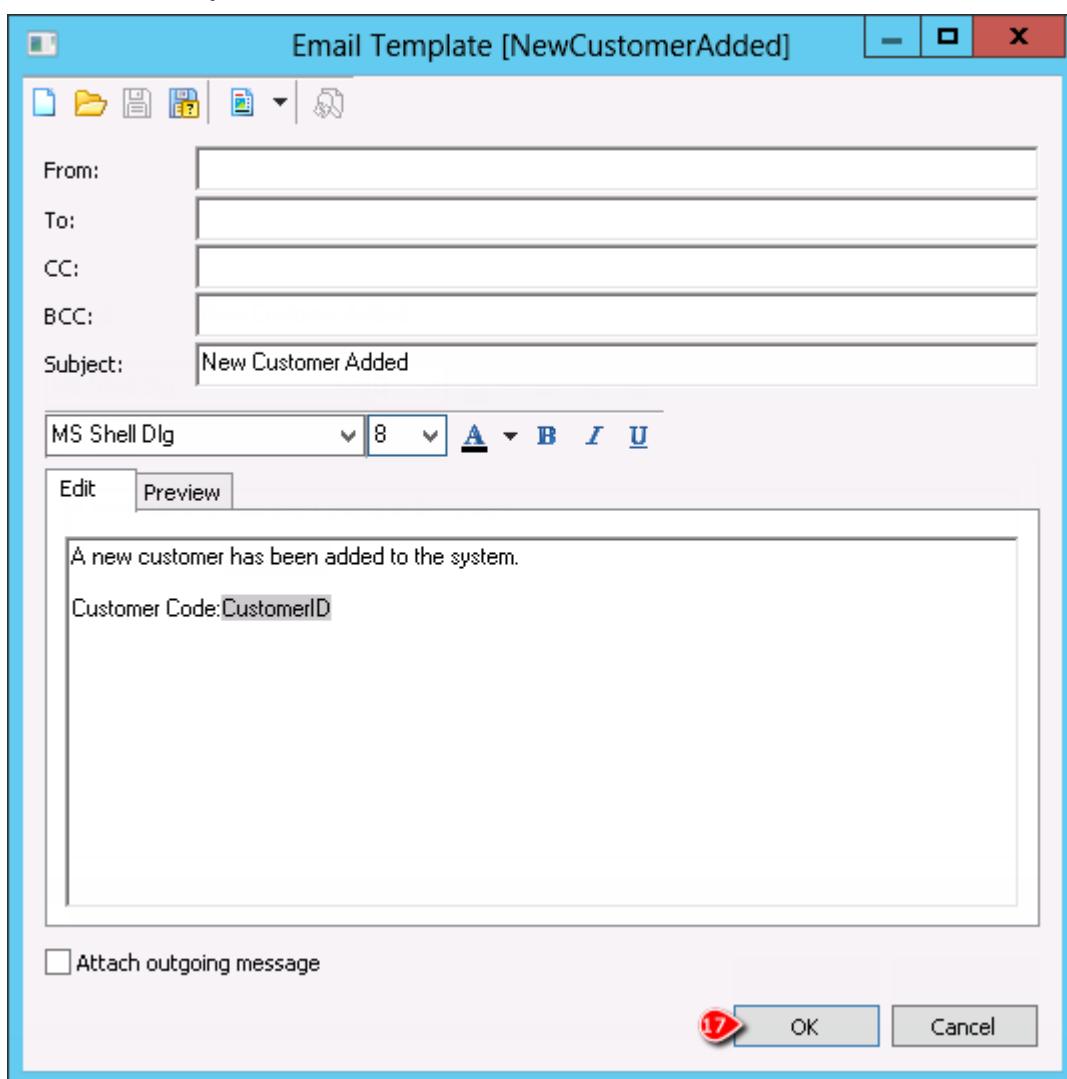
14. Click **Save**.

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

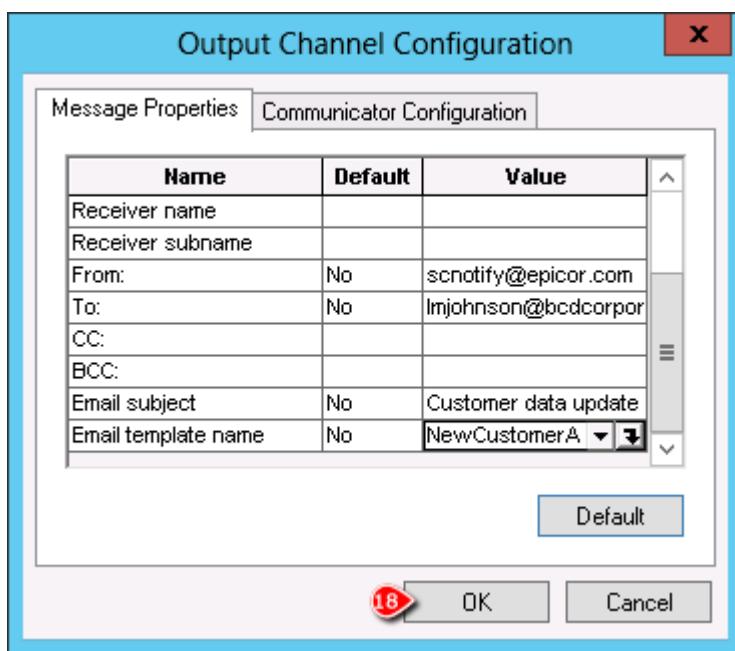


16. Click **OK**.

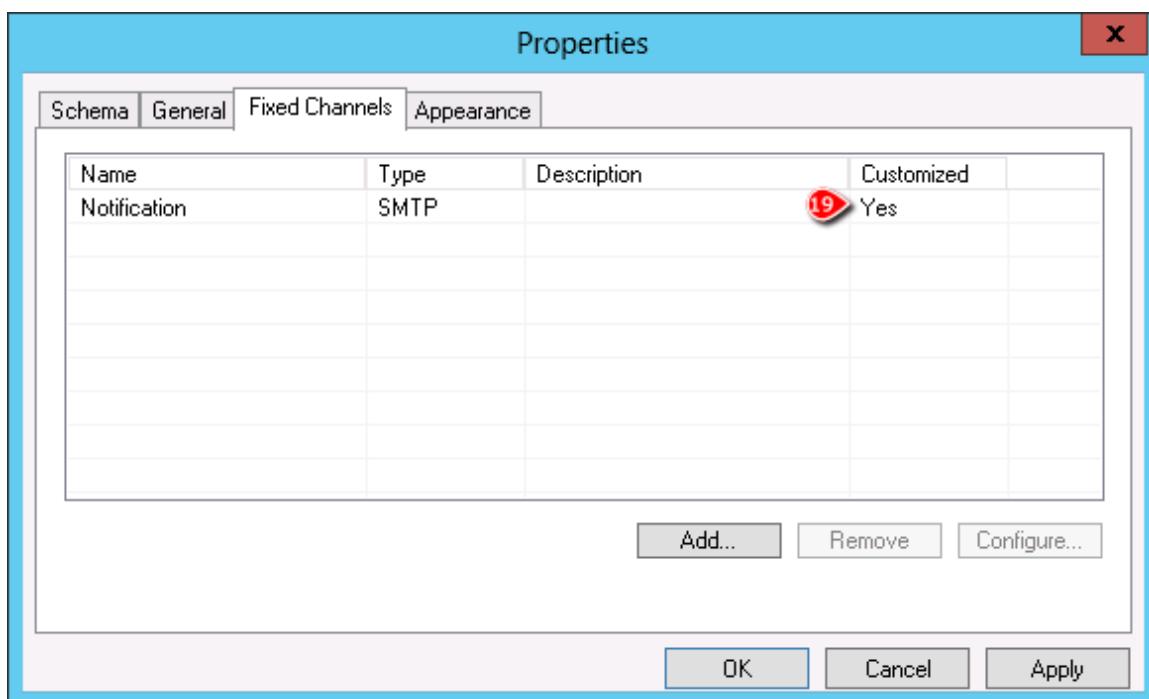
17. In the **Email Template** window, click **OK**.



18. In the **Output Channel Configuration** window, click OK.



19. In the **Properties** window, on the **Fixed Channels** tab, notice the **Customized** column indicates the channel configuration was modified.

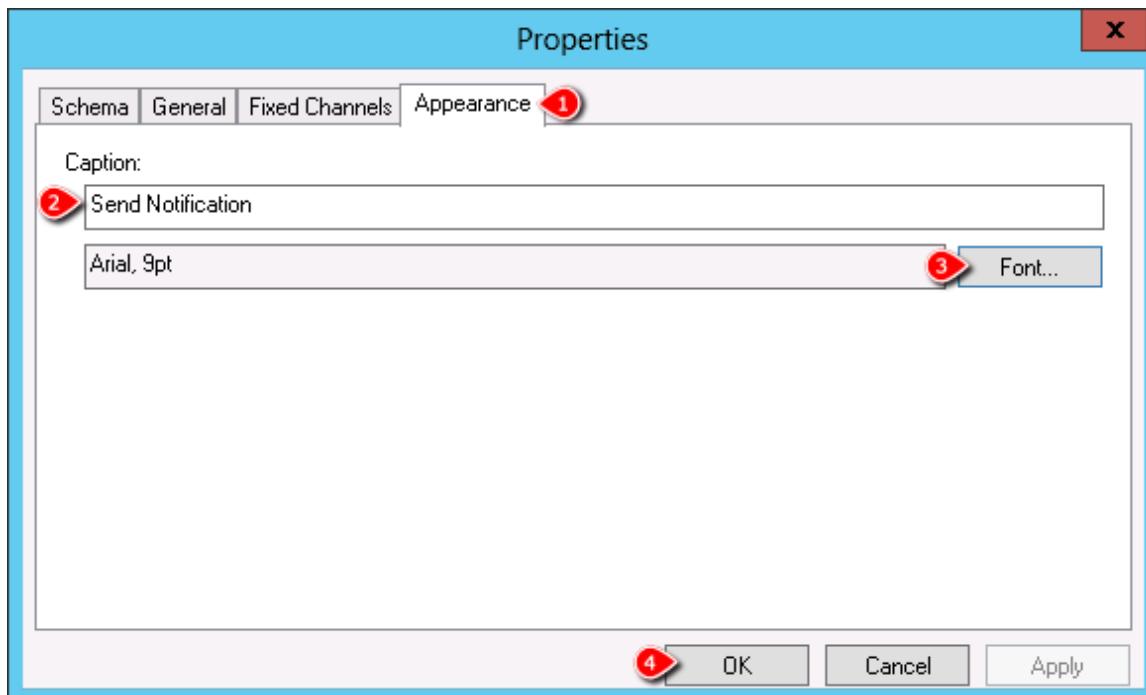


Remain in this window.

Set Up a Poster Appearance

Now define the Poster element appearance in the workflow schema.

1. Open the **Appearance** tab.



2. Enter the Poster **Caption**.
3. You can click the **Font** button to customize the font.
4. Click **OK**.
5. Notice, the Poster icon changed to:



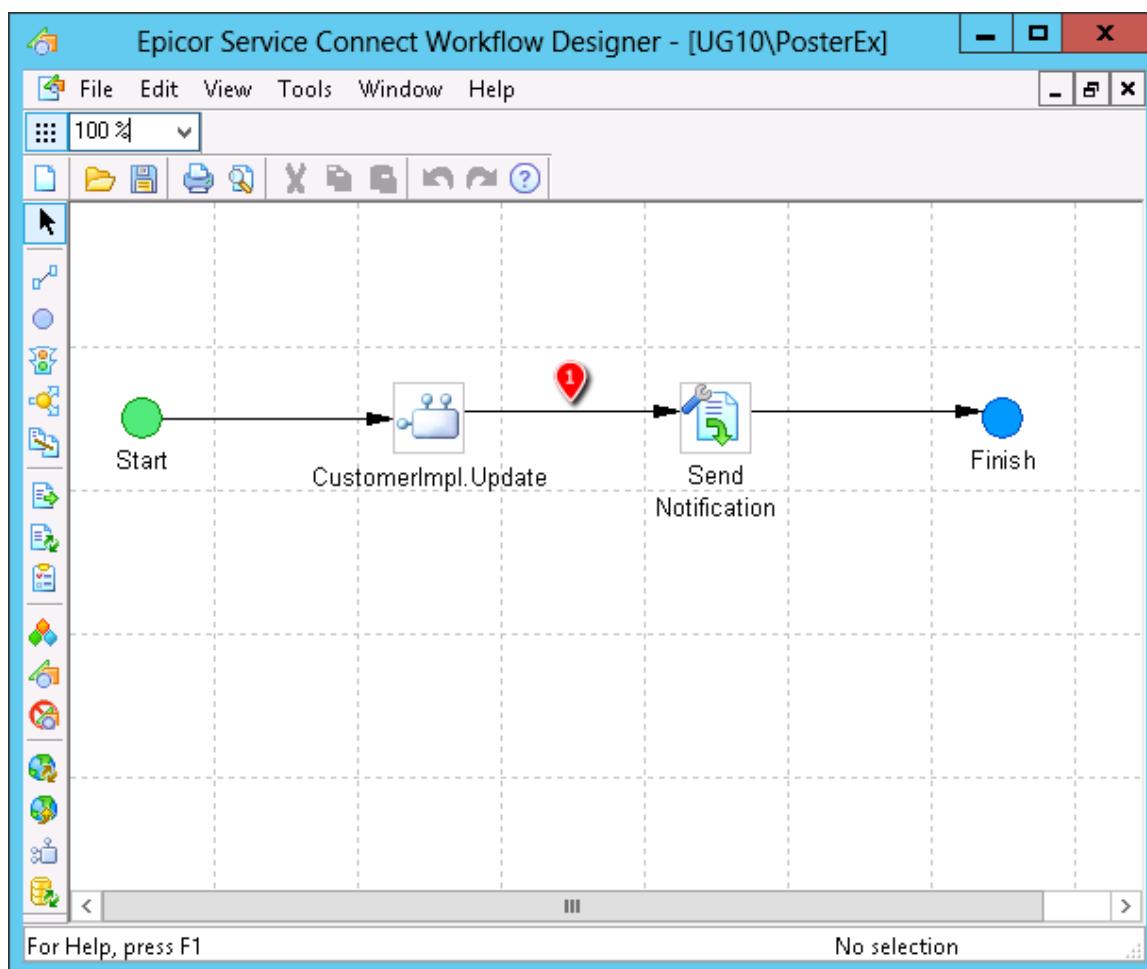
This indicates the Poster channel configuration was modified in the Workflow Designer.

Set Up Poster Dynamic Properties

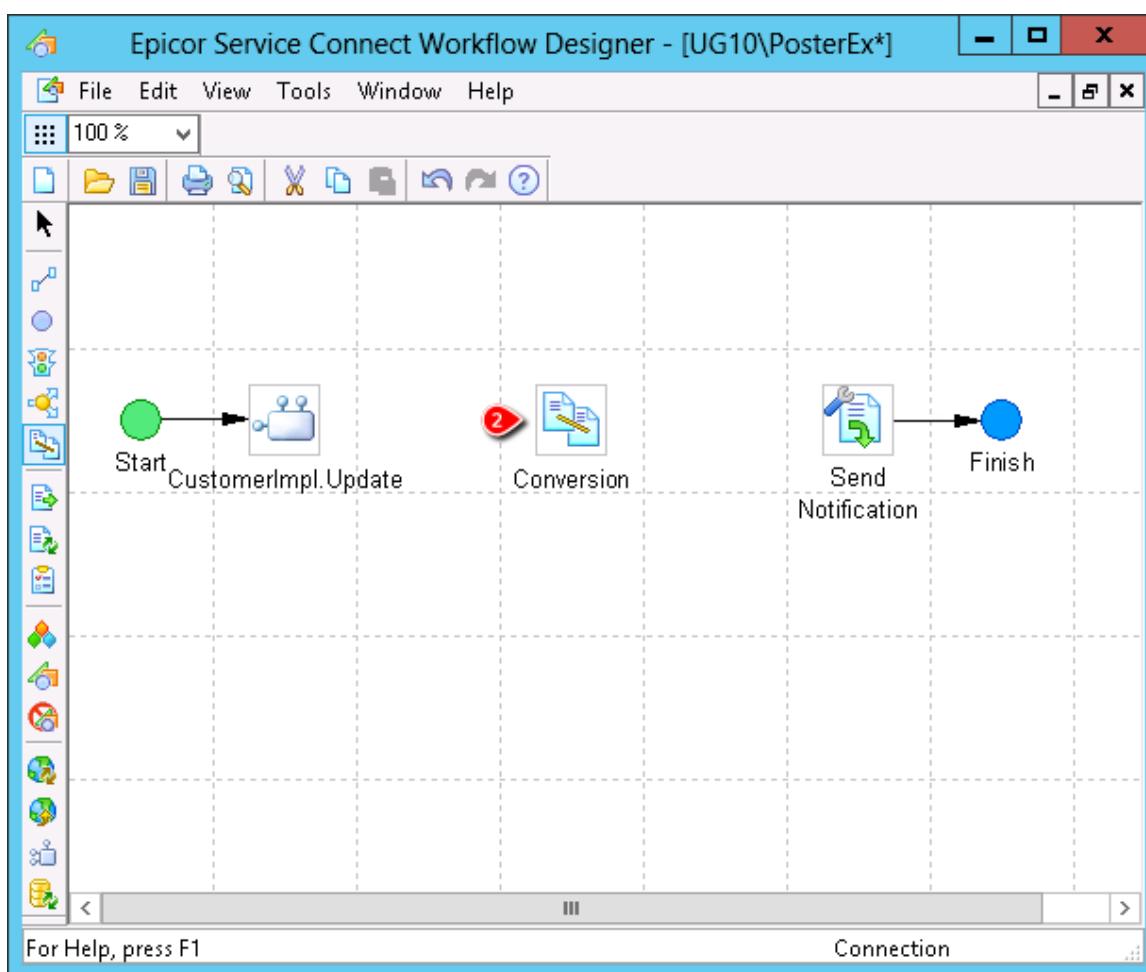
You can define values for all the Poster properties when you design your workflow in the Workflow Designer. This is **static** way to configure a Poster. You can also configure Poster element in a **dynamic** way. That means that some of Poster properties can be redefined during process execution. In this case, elements in incoming XML message determine the values of Poster properties. These elements are found in ctx section of the message. Poster Dynamic properties are initialized via the Conversion element.

To configure Poster element dynamically:

1. In the **Workflow design area**, remove the **Connection** between the **Poster** and the previous workflow element.

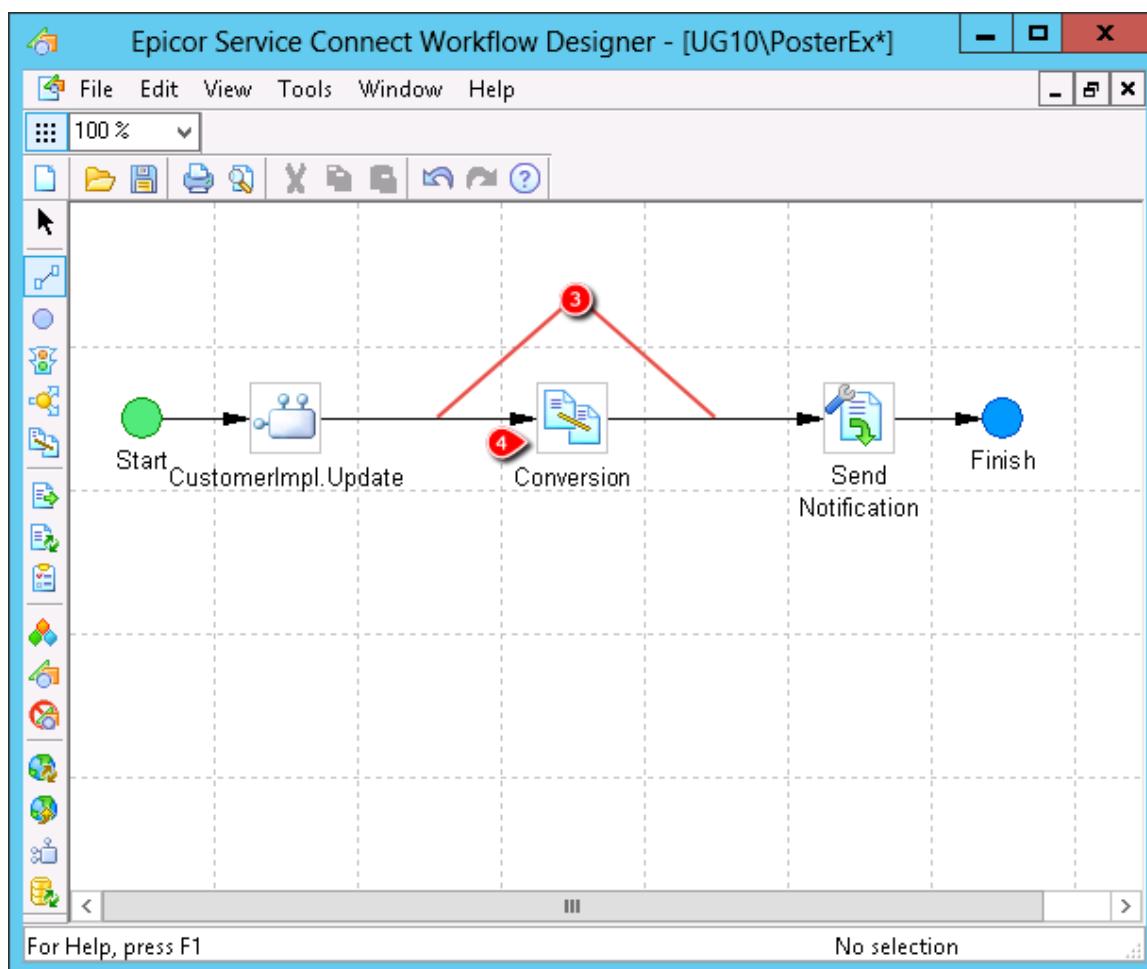


2. Insert the **Conversion** element into the workflow before the **Poster** element.



A Conversion can find a Poster element with dynamic properties even if it is divided from the Poster by the Splitter, Condition or Choice element.

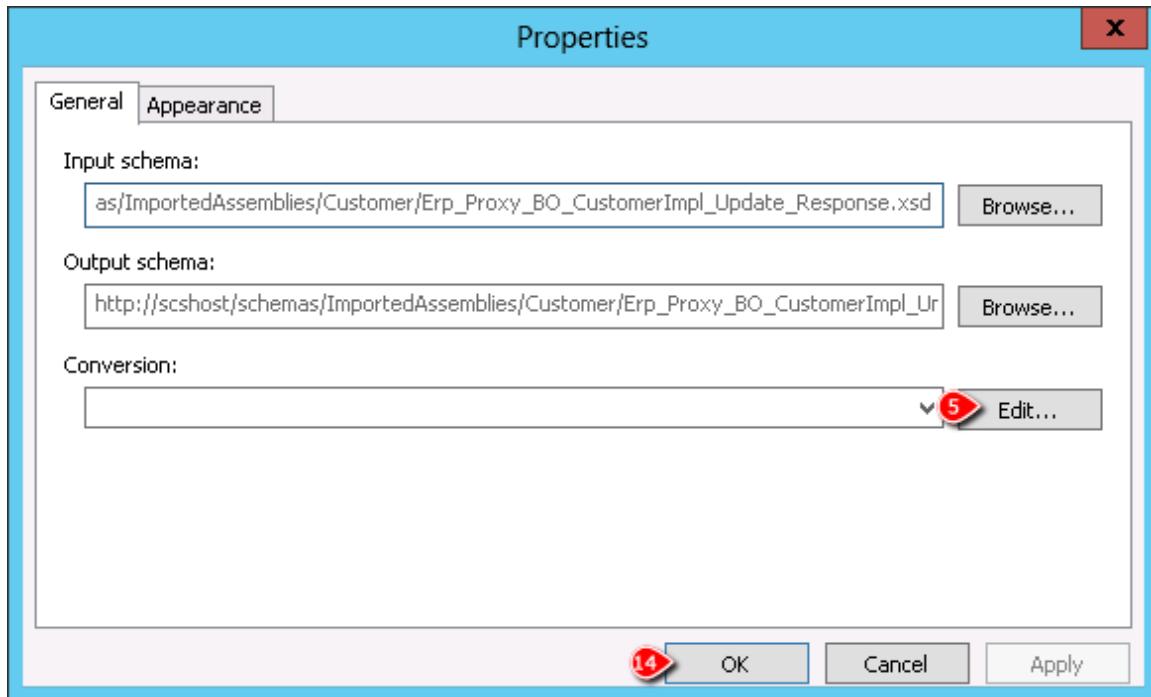
3. Draw **Connections** from the previous workflow element to the **Conversion** and from the **Conversion** to the **Poster**.



4. Double-click the Conversion.

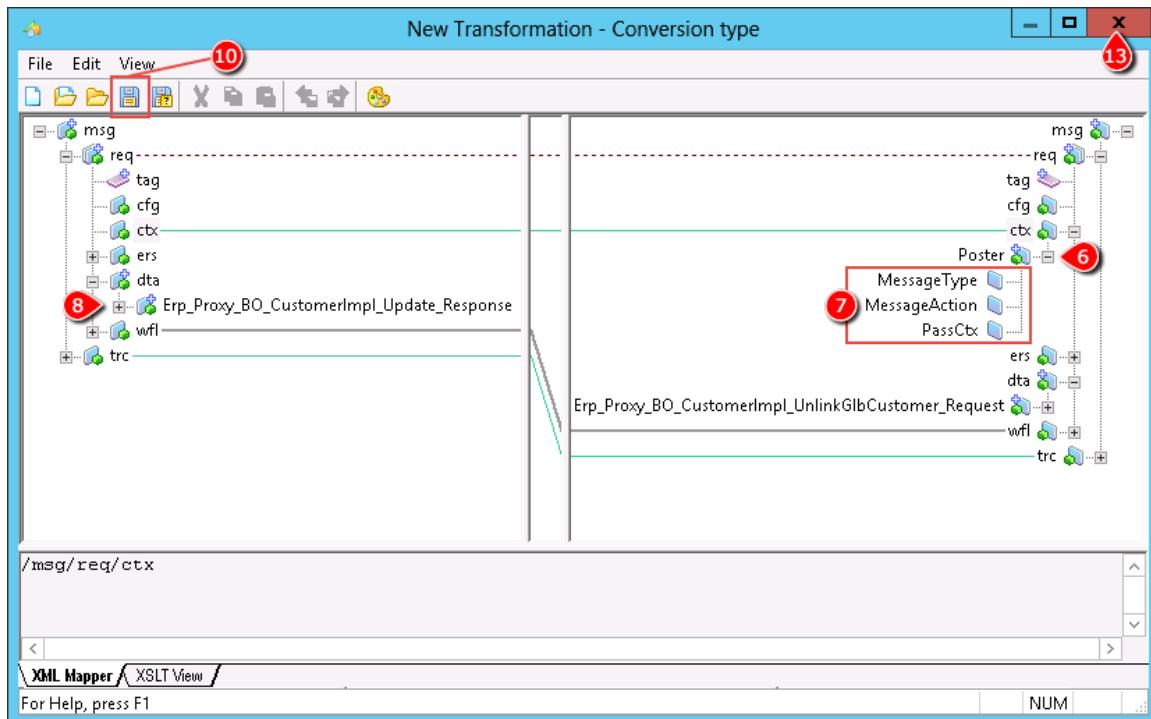
The **Properties** dialog is displayed.

5. On the **General** tab, next to the **Conversion** field, click the **Edit** button.



XML Mapper window is displayed.

6. In the Mapper right pane, unfold **msg > reg > ctx > Poster** nodes.



7. Under the **Poster** node, view all the Poster properties that can be set dynamically.
8. Map the source message data or literal values onto the Poster properties.

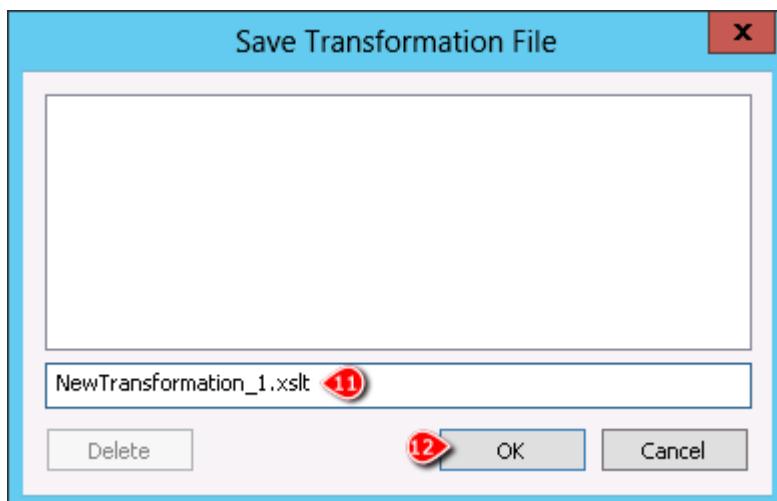
9. You can edit the xml in XSLT view as well.

```
<?schema_locations source_url="" source_root="" destination_url="" destination_root="" version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msg="http://Epicor.com/InternalMessage/1.1">
<xsl:stylesheet>
<xsl:output method="xml" version="1.0" indent="yes" omit-xml-declaration="no"/>
<xsl:template match="/">
<xsl:apply-templates select="*"/></xsl:apply-templates>
</xsl:template>
<xsl:template match="msg:msg">
<xsl:element name="msg:msg">
<xsl:for-each select="msg:req">
<xsl:element name="msg:req">
<xsl:if test="msg:ctx/*">
<xsl:element name="msg:ctx">
<xsl:copy-of select="msg:ctx/*"/></xsl:copy-of>
</xsl:element>
</xsl:if>
<xsl:element name="msg:wfl">
<xsl:if test="msg:wfl/msg:usr/*">
<xsl:element name="msg:usr">
<xsl:copy-of select="msg:wfl/msg:usr/*"/></xsl:copy-of>
</xsl:element>
</xsl:if>
<xsl:if test="msg:wfl/msg:ElemID">
```

The PassCtx node treats "True", "true" or 1 values as enabled "Pass msg:ctx section" setting. The "False", "false" or 0 values indicate disabled "Pass msg:ctx section" setting.

You can find values for MessageType and MessageAction in Message Attributes node of Administration Console.

10. When the mapping is complete, click **Save**.
11. In the **Save Transformation file** window, enter a name for the XSLT file.



12. Click **OK**.
13. Close the XML Mapper window.
14. In the **Properties** window, click **OK**.

Sub-Workflow



Use this workflow element 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 workflow element 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.

Sub-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.

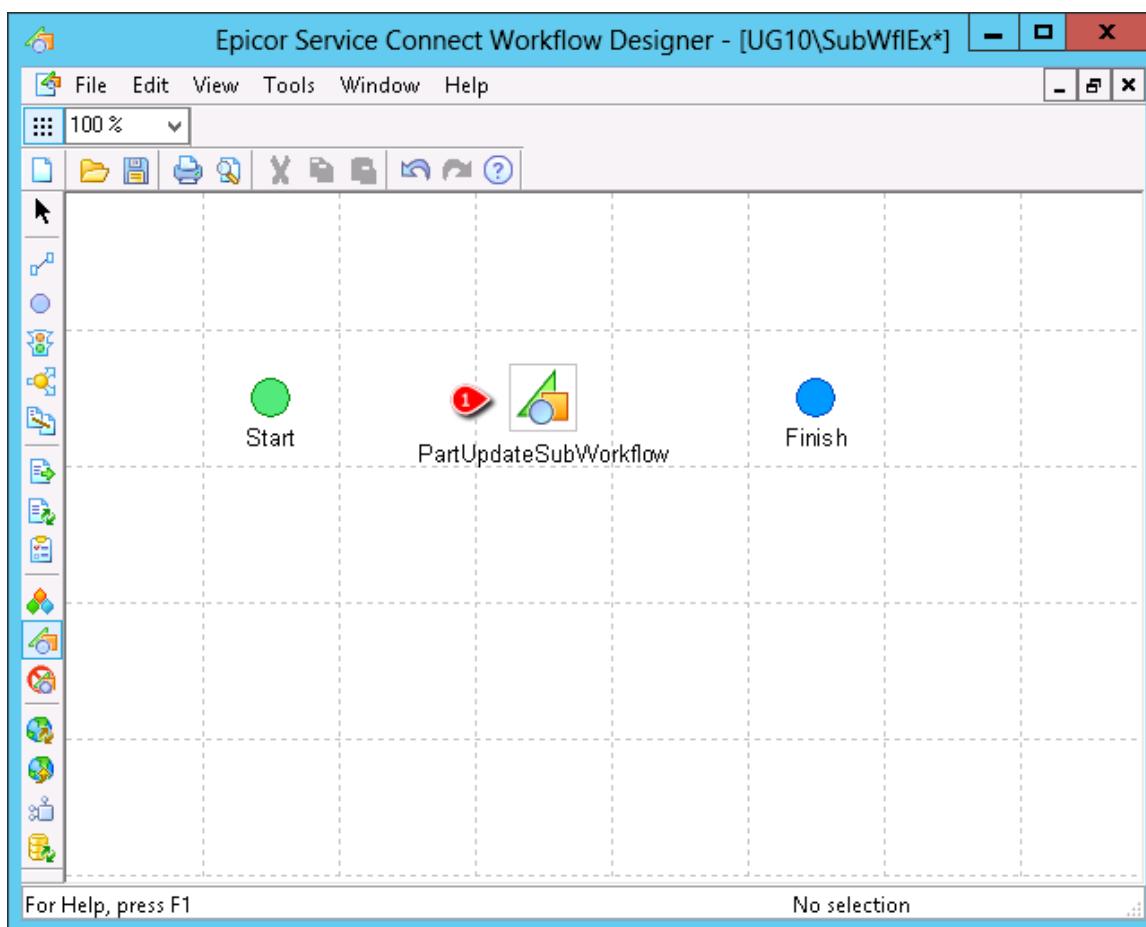
Set Up a Sub-Workflow

Sub-workflow setup involves the following tasks:

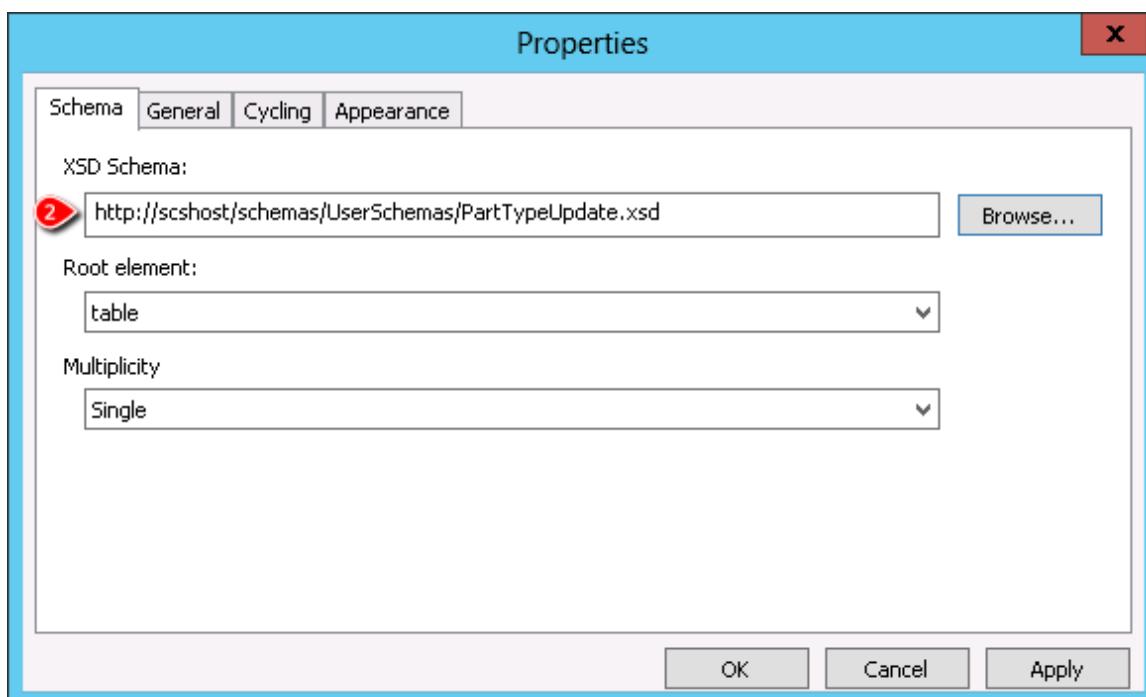
- Create a workflow to serve as a subroutine to the main workflow. Usually, a Sub-workflow is saved to the same workflow package as the main workflow.
- Add the Sub-workflow activity to the main workflow.

To set up a Sub-workflow activity:

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



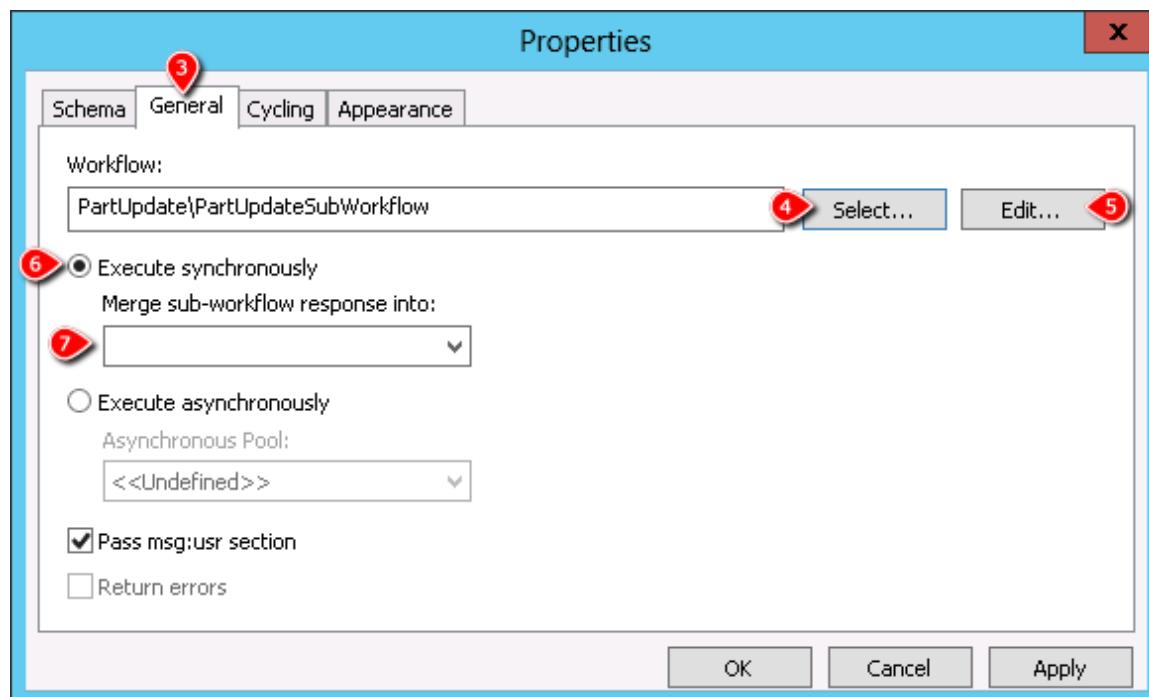
2. The **Properties** window displays. On the **Schema** tab, set the **XSD Schema** of the document that comes to the Sub-workflow element.



You can send the document to the Sub-workflow as a whole 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. Open the **General** tab.



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

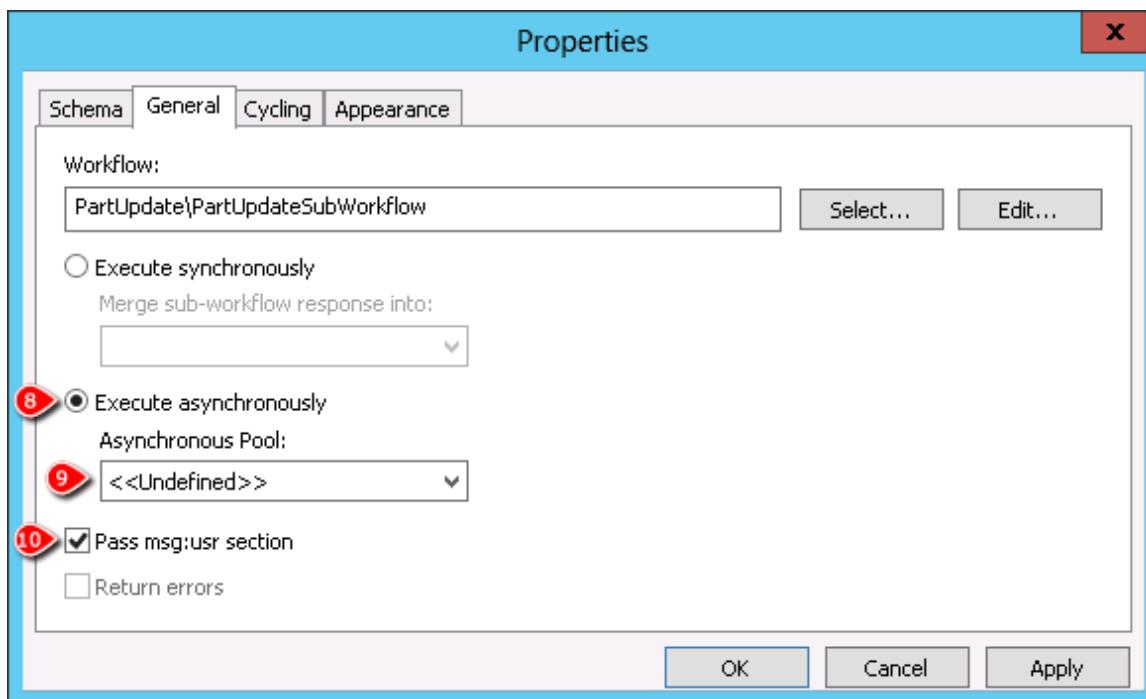
5. You can click **Edit** to launch the second instance of Service Connect Workflow Designer and edit the selected workflow.

6. Select the **Execute synchronously** option 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 displays as a message extension under the user node when a Conversion workflow element 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. If you selected the **Execute asynchronously** option, 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. A pool is defined with the pool name and thread number. The thread number can take 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 Connectivity Components chapter 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.

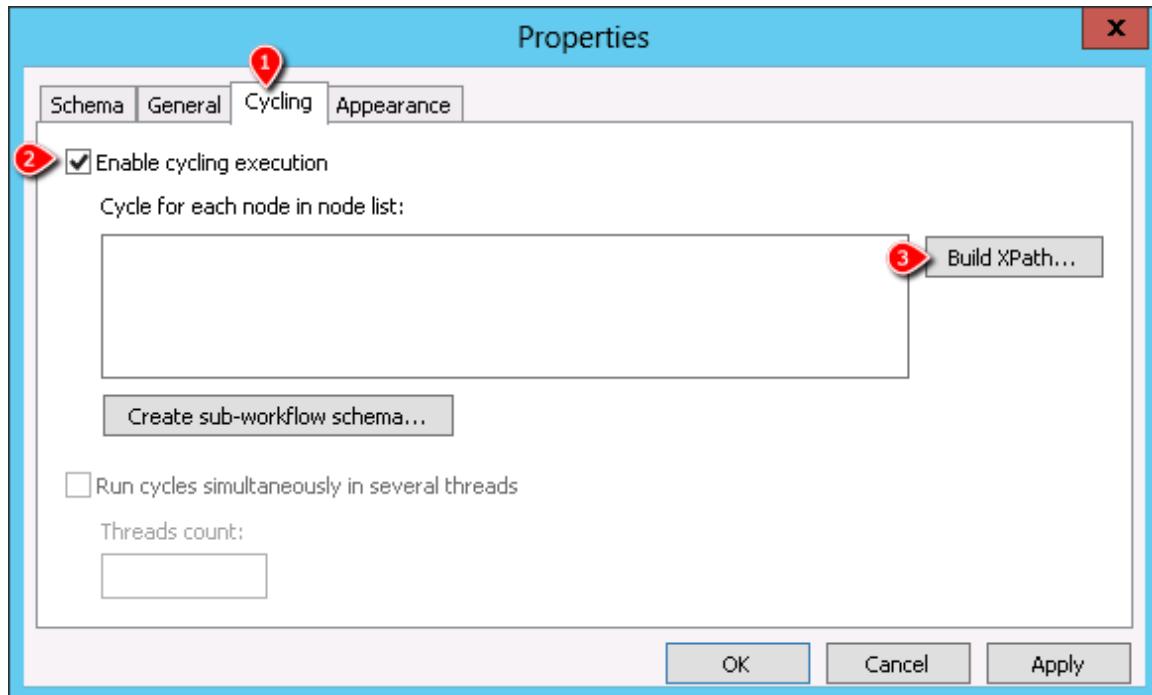
You can pass process variables only in one direction: from the main workflow to a Sub-workflow. Process variables from a Sub-workflow are not passed back to the main workflow.

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.

Remain in the **Properties** window.

Set Up Sub-Workflow Cycling

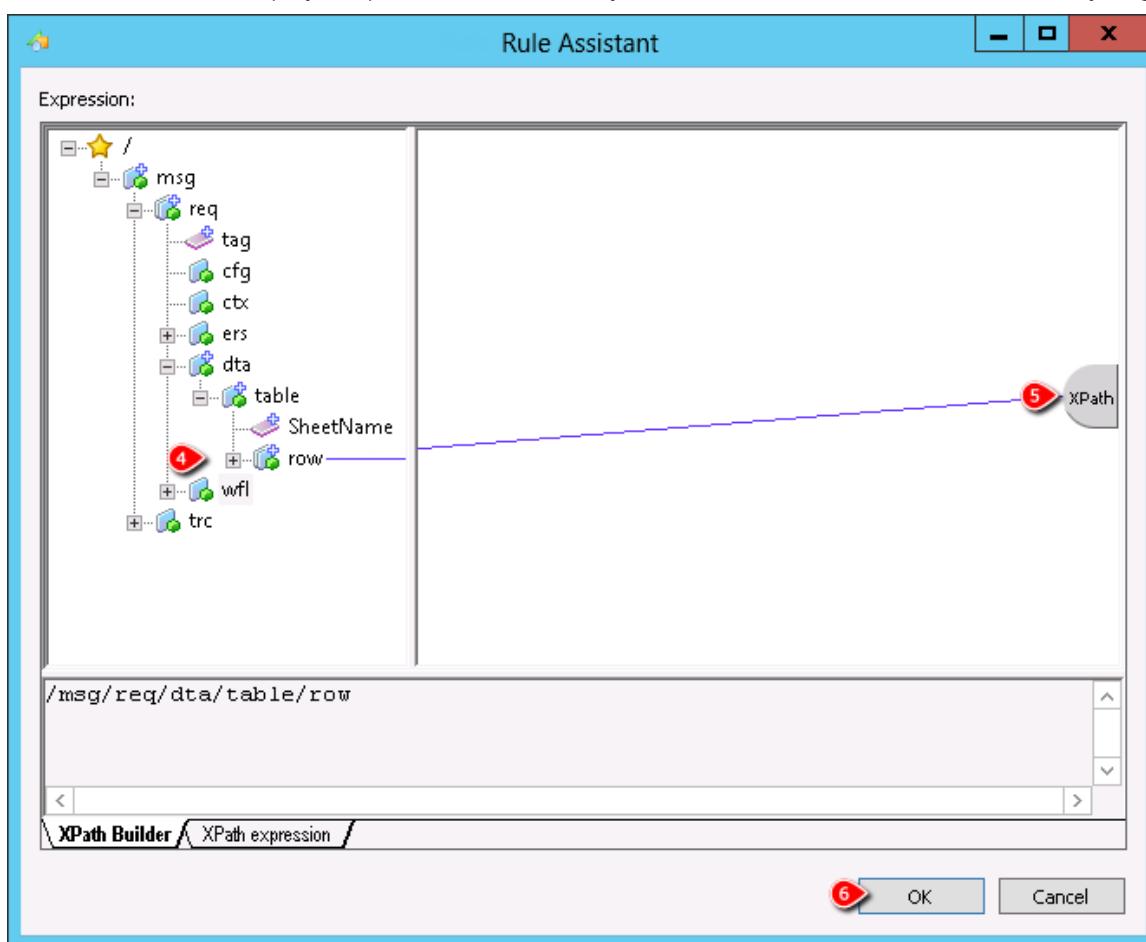
1. Open the **Cycling** tab.



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

2. Select the **Enable cycling execution** check box.
3. Click **Build XPath** to select the part of the document on which the Sub-workflow will cycle.

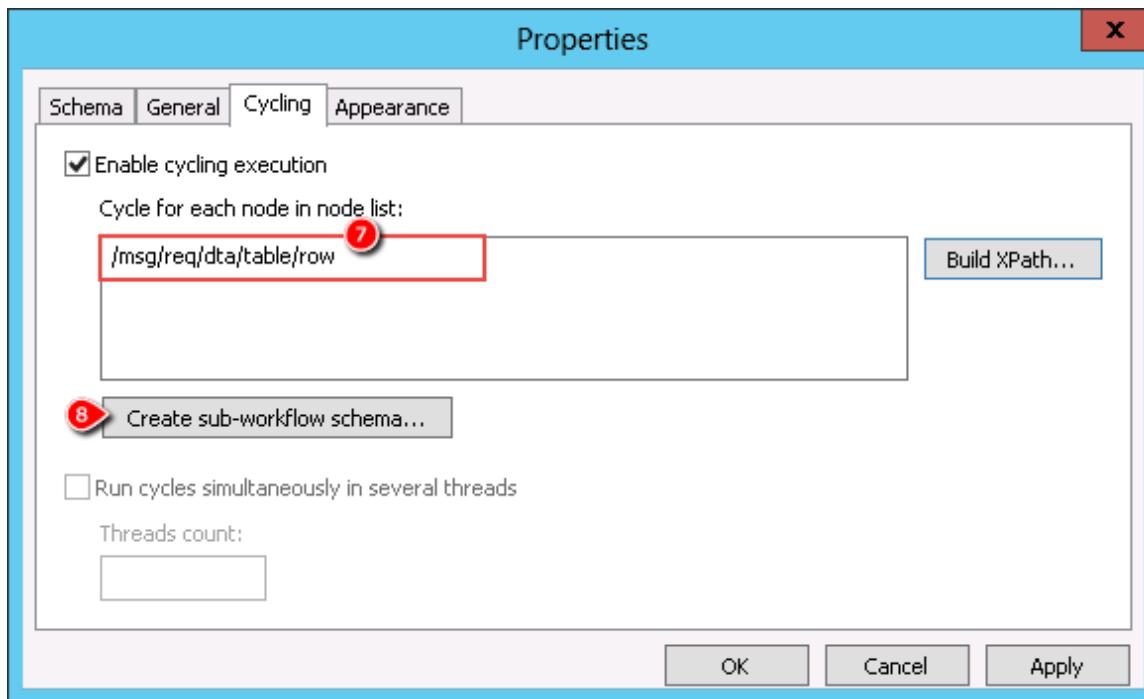
4. The **Rule Assistant** displays. Expand the nodes until you locate the collection node to use for cycling.



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

5. Link the collection node to the **XPath** sign.
6. Click **OK**.

7. In the **Properties** window, notice the **Cycle for each node in node list** field displays XPath statement in simplified form: namespaces are omitted and only the xml element names are displayed.

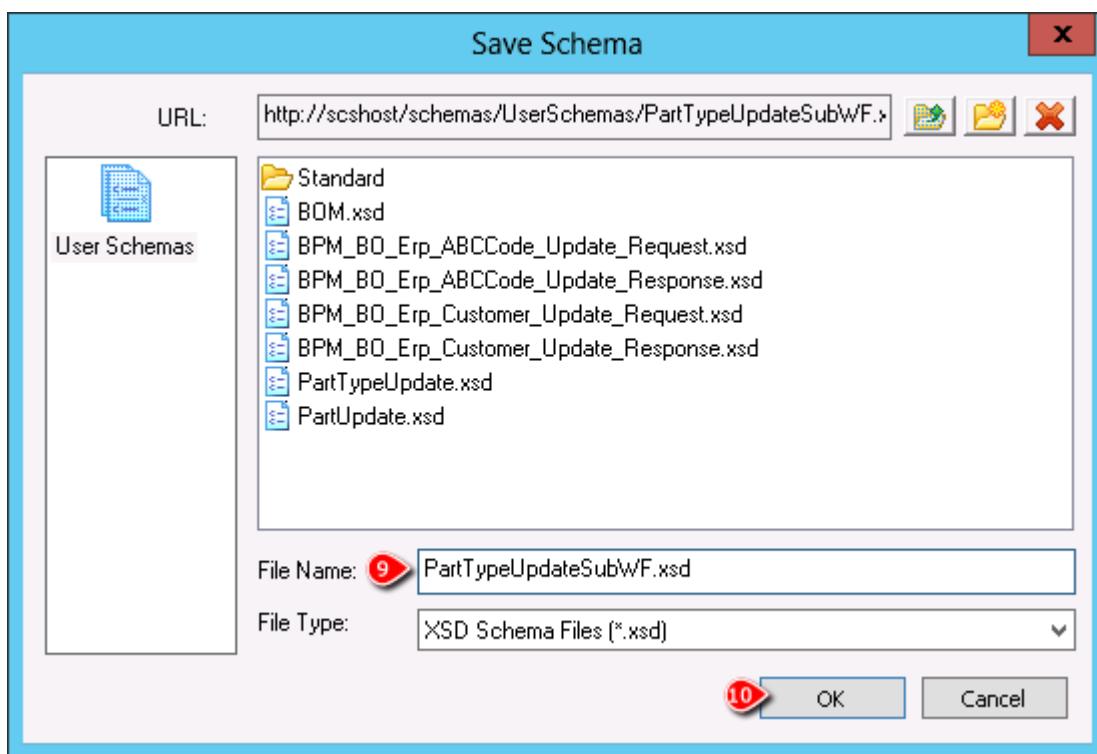


8. Click **Create sub-workflow schema** to create a schema for the Sub-workflow.

This opens the Save Schema dialog.

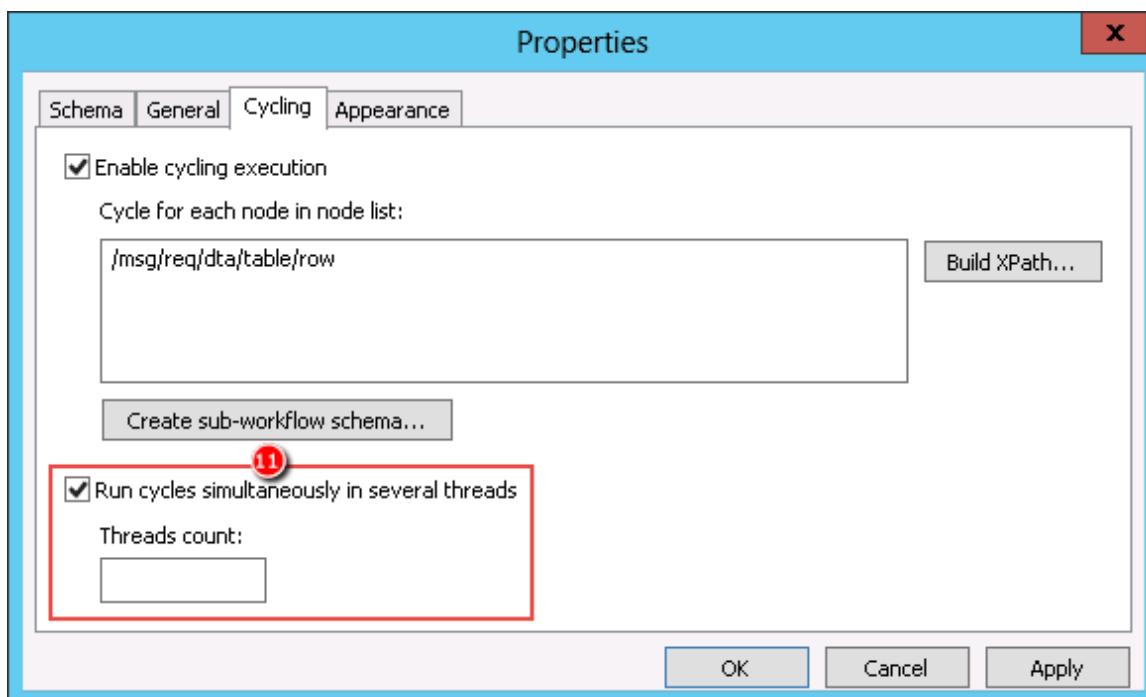
The schema will be used in a Conversion workflow element of the Sub-workflow to specify the document format sent to the Sub-workflow. A schema is not always required for cycling execution; it is possible to use the schema for a Web Method request in a Sub-workflow activity or to use only an XPath expression.

9. The **Save Schema** window displays. Enter the schema name.



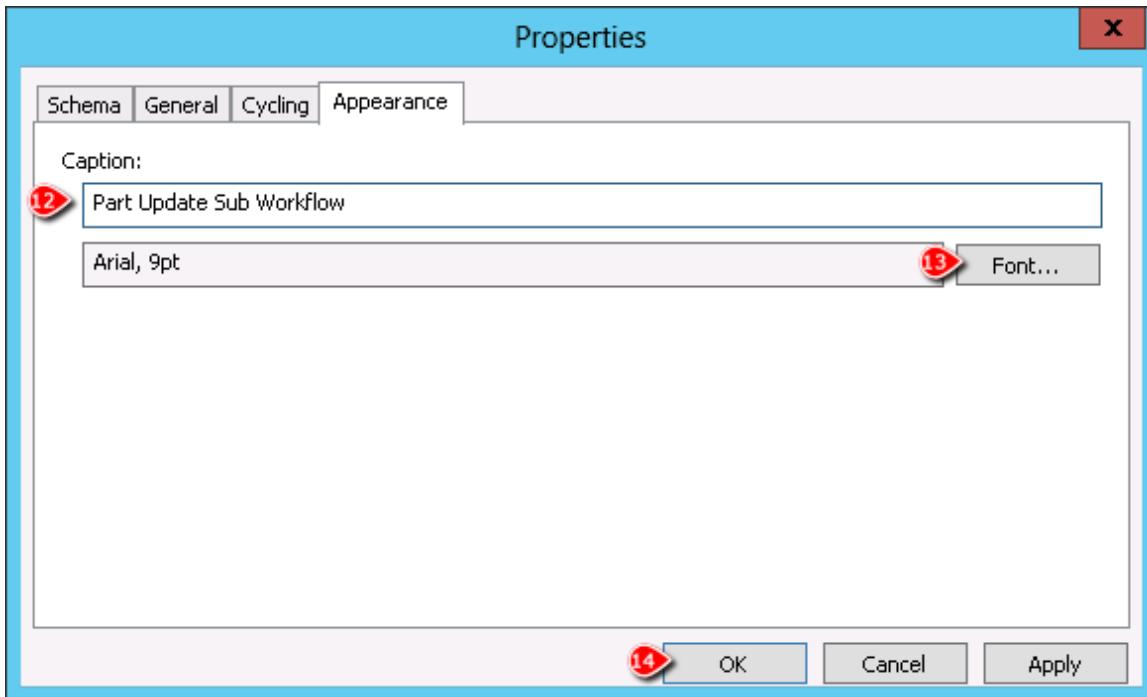
10. Click **OK**

11. If you selected to execute the Sub-workflow synchronously (the Execute synchronously option on the General tab), 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.

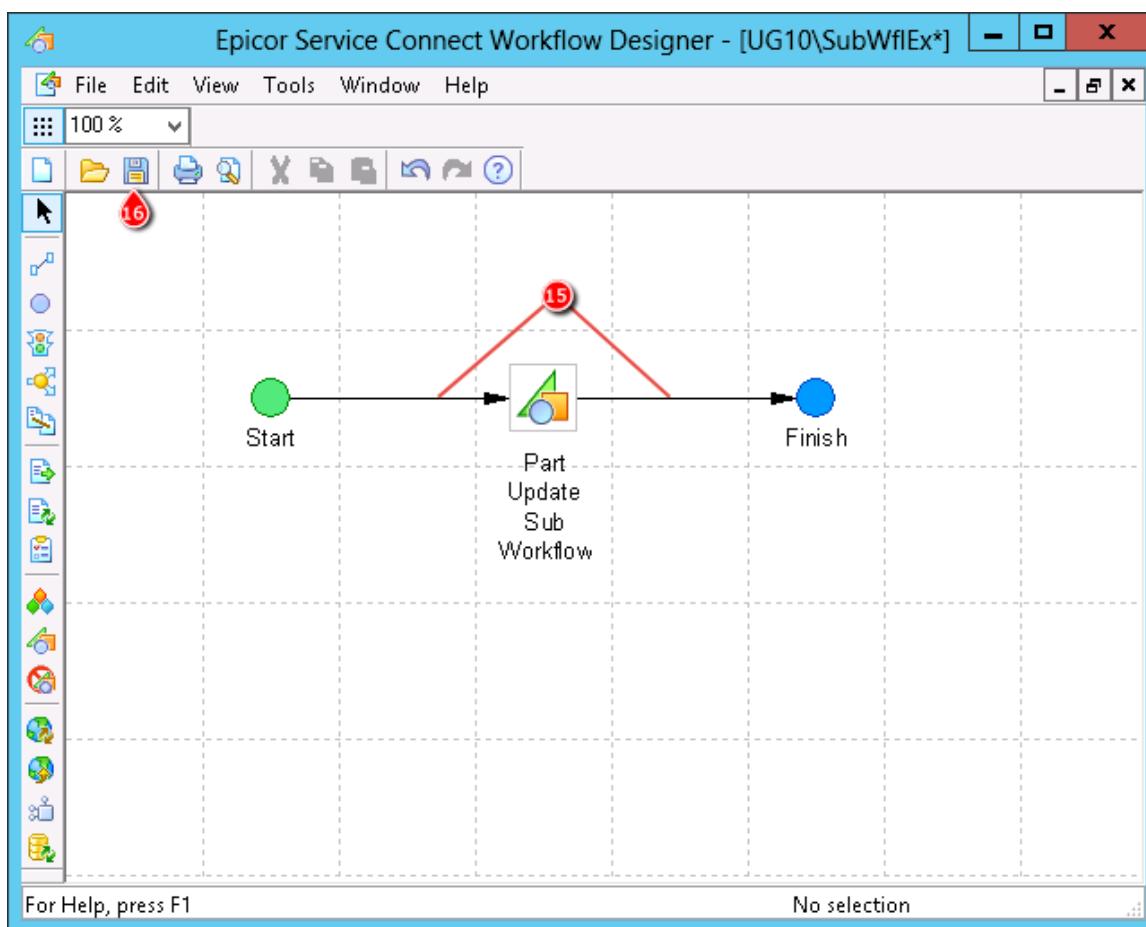
12. On the **Appearance** tab, enter the **Caption**.



13. If you want to customize the font, click **Font**.

14. Click **OK**.

15. In the **Workflow design area**, add connections to and from the **Sub-workflow** element.



16. Save the workflow.

This is the main workflow.

Create a Subroutine Workflow

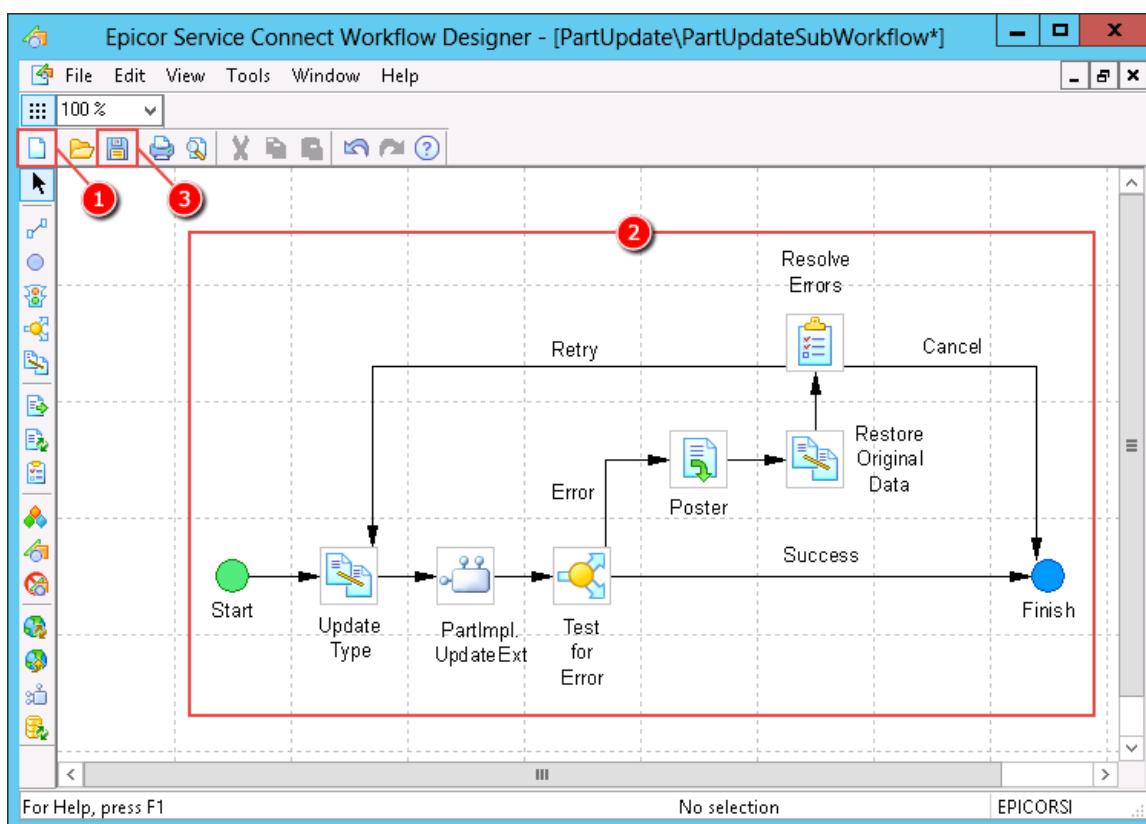
Now create a new workflow that the Sub-Workflow element will reference from the main workflow.

1. Create a blank new workflow.
2. Add **workflow elements** to the workflow to use as the Sub-workflow.

In this example, the Sub-workflow uses two elements: a Conversion and a Web Method to update the part record. The Choice workflow element that follows verifies record accuracy. If a failure occurs, the Poster workflow element 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.



3. Save the workflow.

Windows Workflow Foundation® Call



Use this workflow element 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.
13. You may have to click the Base Class field to display the ... (Ellipse) button.
14. The **Browse and Select a .NET Type** window displays. In the **Tree View**, select **Epicor.ServiceConnect.Core**.
15. In the **Type Name** column, select **BaseWorkflowActivity**.
16. 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 Epicor Service Connect Overview chapter 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 **XmIConversion** 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 XmIConversion WF activity. The XmIConversion 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 **ReturnValue** 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.
`<Process> <Package Name = "Workflow Library1"> <Workflow Name = "Workflow" /> </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 workflow element 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 workflow element 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 workflow element.

Requester Example

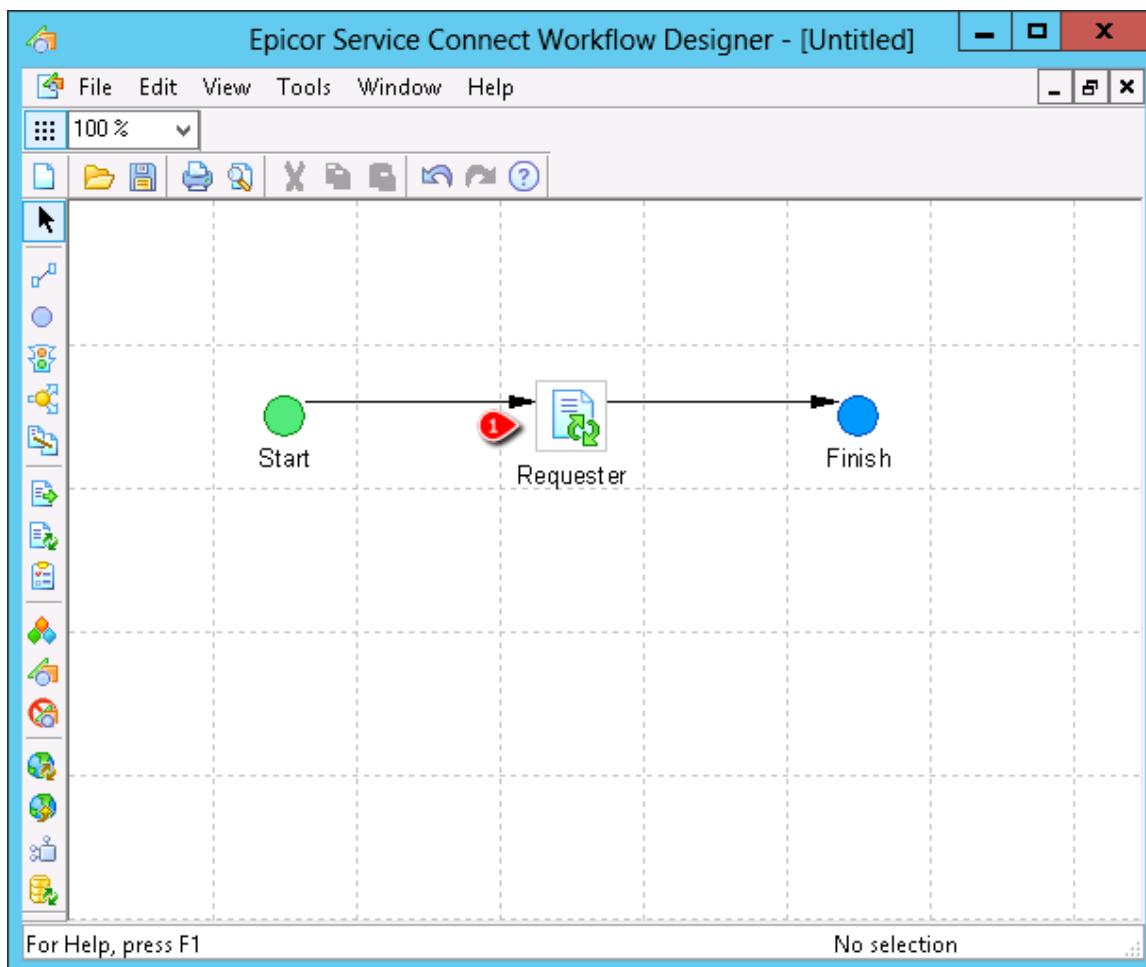
A document that enters a workflow is routed to a Requester, which forwards the document to a COM output channel. The message sent to the output channel contains message attributes that indicate the message is being sent from the Requester. The Correlation Manager uses these attributes to identify the 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.

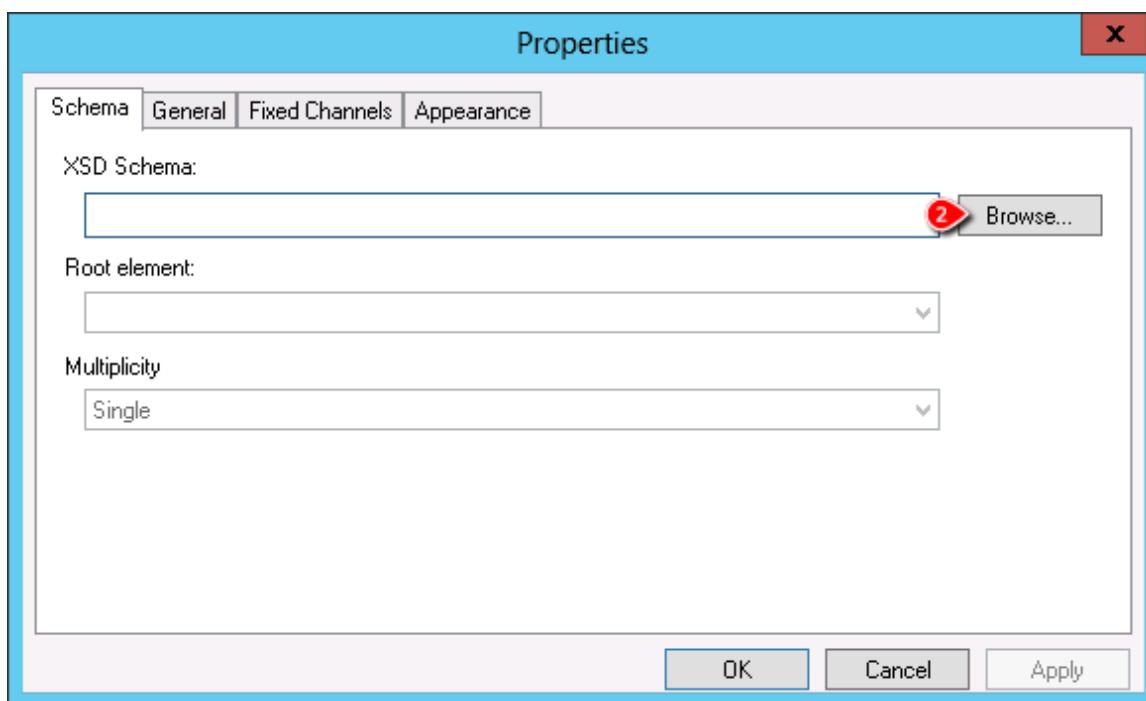
Set Up a Requester

To set up a Requester:

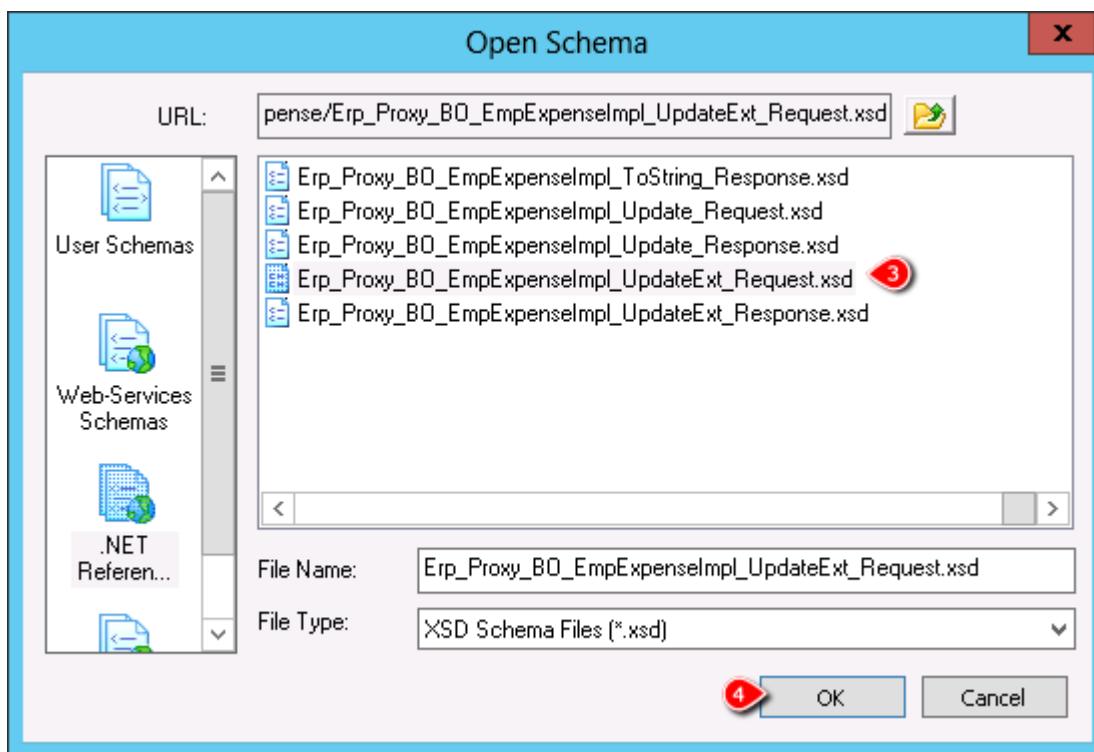
1. In the **Workflow design area**, double-click the **Requester** to open the **Properties** window.



2. On the **Schema** tab, click the **Browse** button.

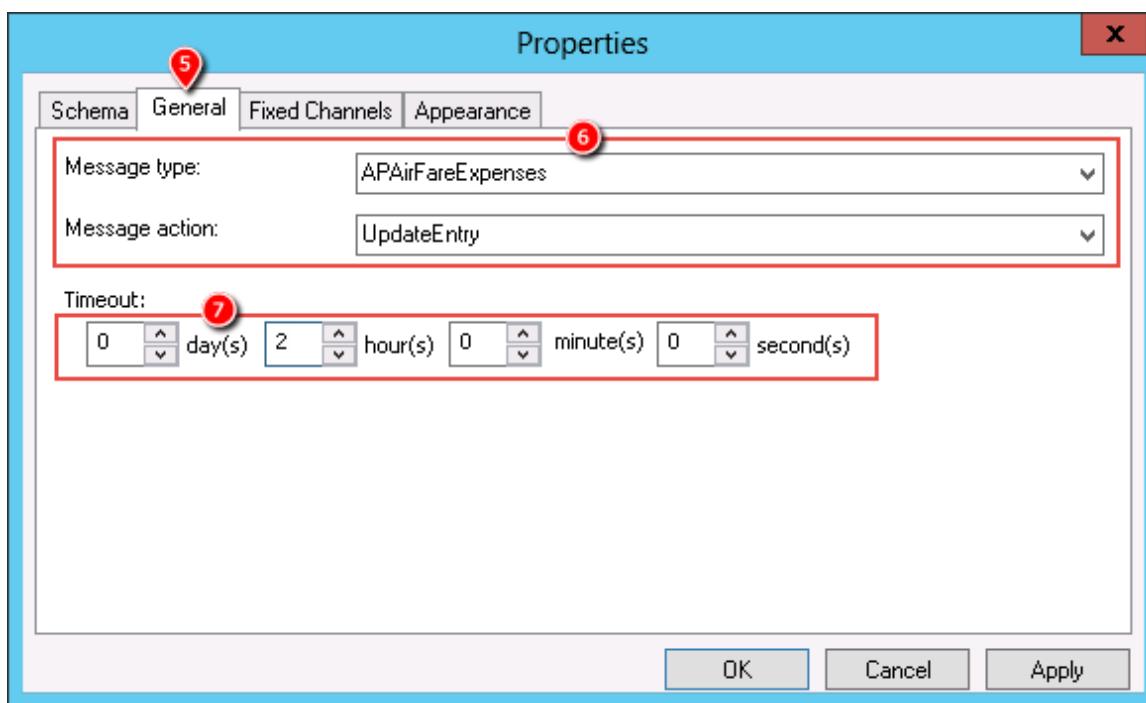


3. In the **Open Schema** window, navigate to and select an **XSD** schema to define the format of the document entering the Requester.



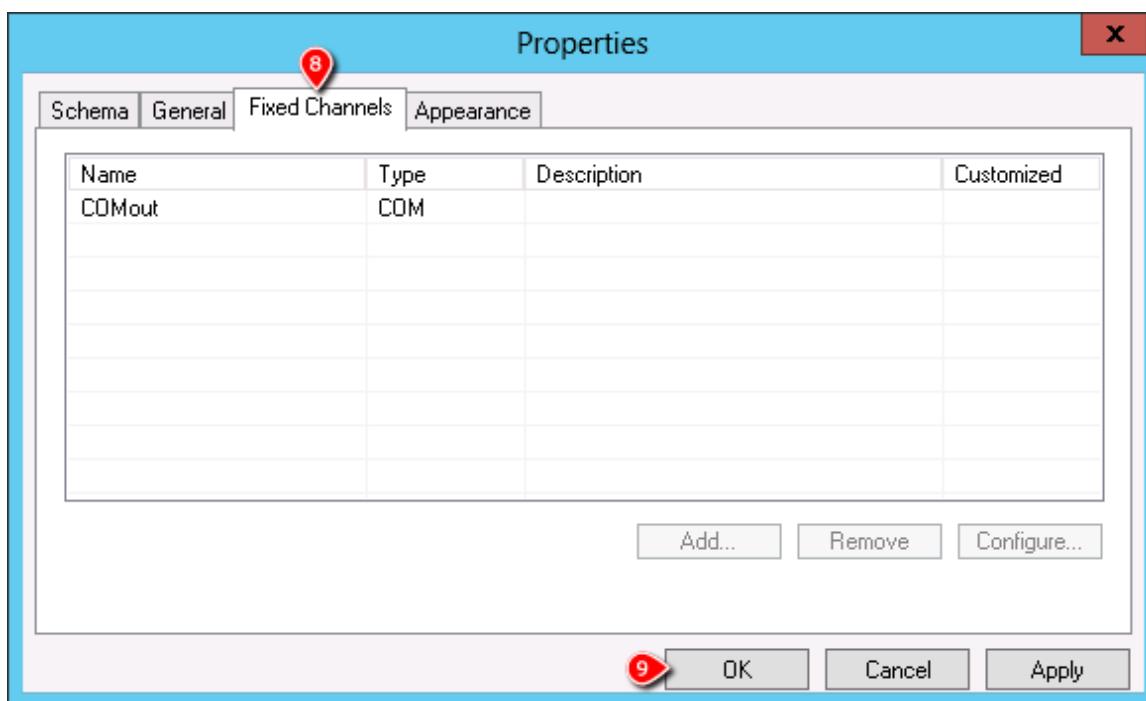
4. Click **OK**.

5. Click the **General** tab.



6. Select the **Message type** and **Message action**.

7. Use the **Timeout** fields to set the time period. The workflow will resume if no response is received after the timeout expires.
8. Open the **Fixed Channels** tab to assign one or more output channels.



Refer to the Poster description earlier in this chapter for details on how to add and configure fixed channels.

9. Click **OK**.
10. If you modify the output channel configuration that was previously set up in the ESC Administration Console, the Workflow design area displays customized Requester icon instead of the default one.



Set Up Requester Dynamic Properties

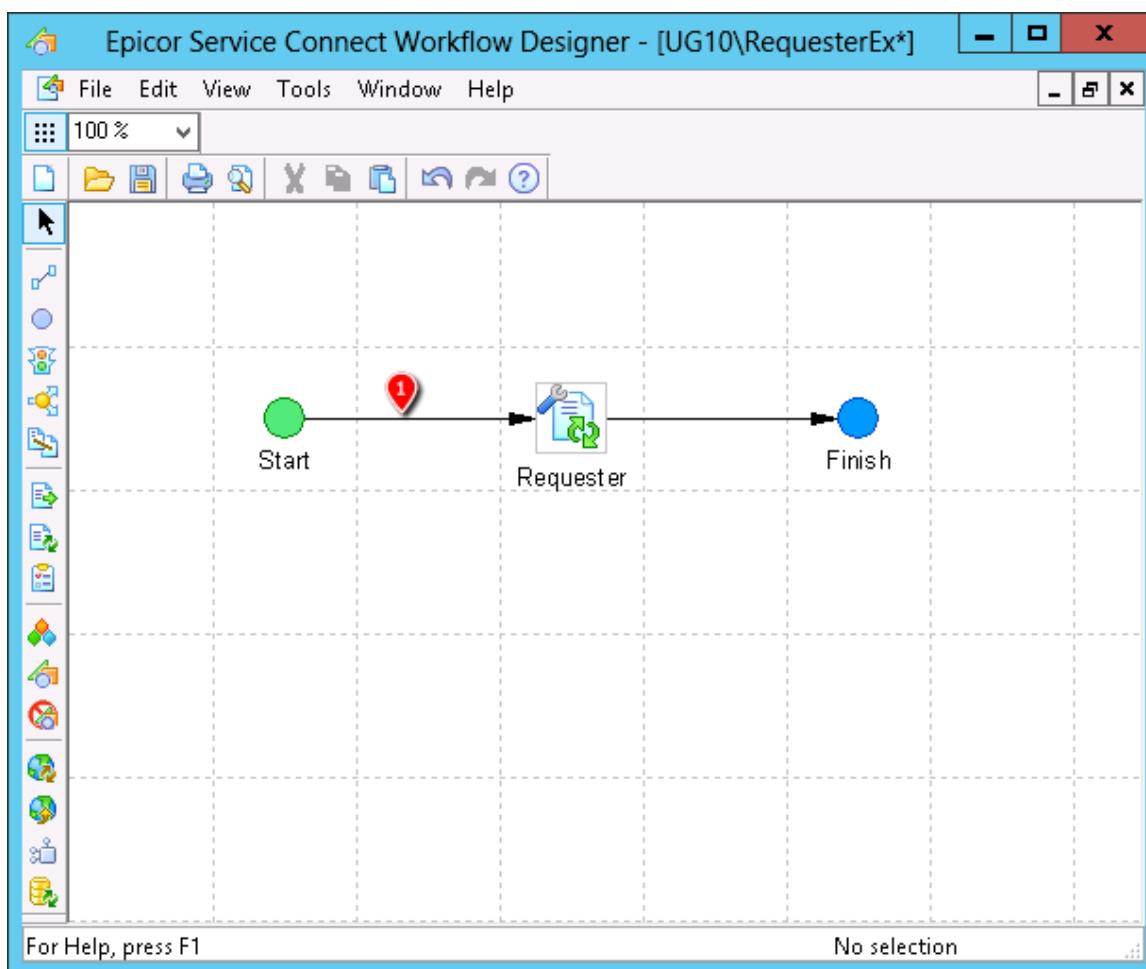
You can define values for all the Requester properties when you design your workflow in the Workflow Designer. This is **static** way to configure a Requester. You can also configure Requester element in a **dynamic** way. That means that some of Requester properties can be redefined during process execution. In this case, elements in incoming XML message determine the values of Requester properties. These elements are found in ctx section of the message. Use the Conversion element to initialize Requester dynamic properties.

You can define values for the following Requester properties at runtime:

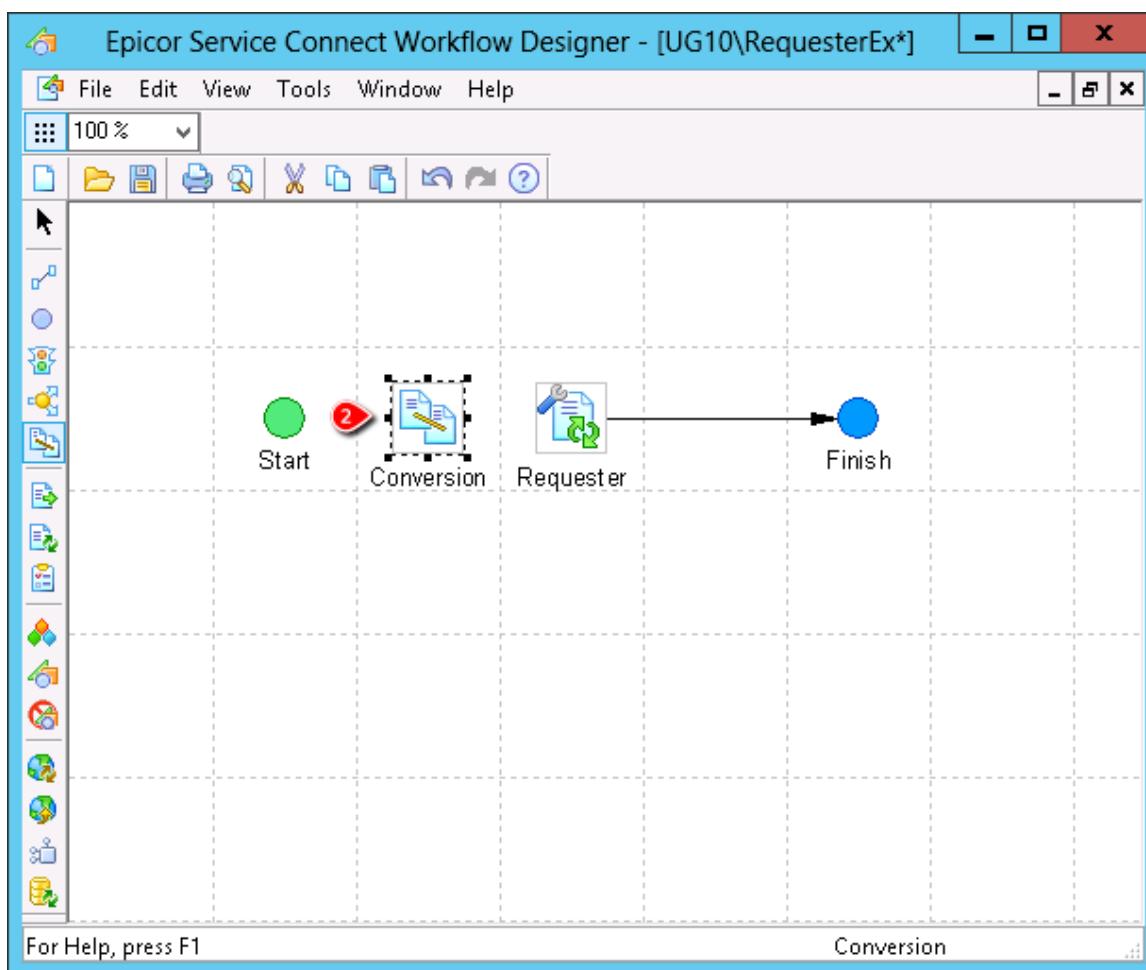
- Message Type
- Message Action
- Timeout

To configure Requester element dynamically:

1. In the **Workflow design area**, remove the Connection between the Requester and the previous workflow element.

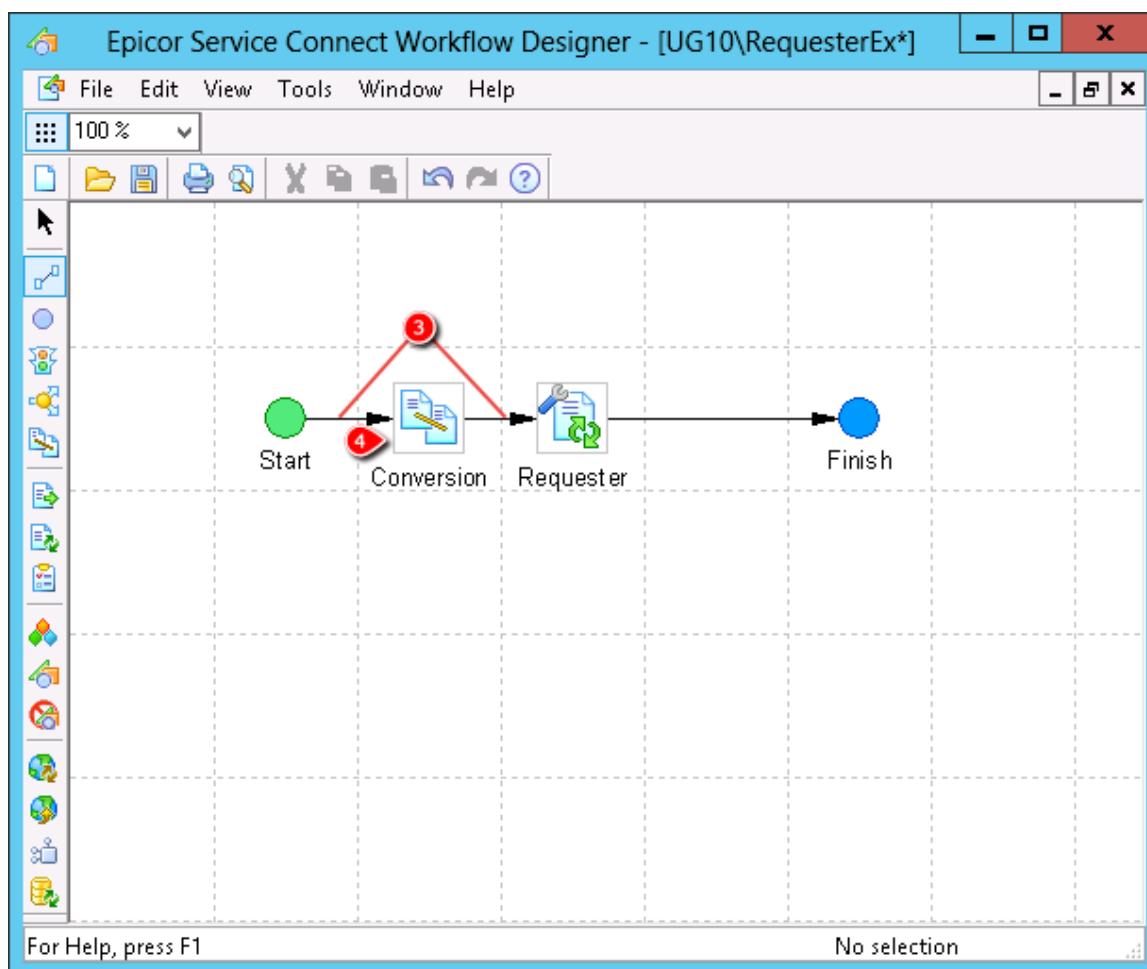


2. Insert the **Conversion** element into the workflow before the Requester element.



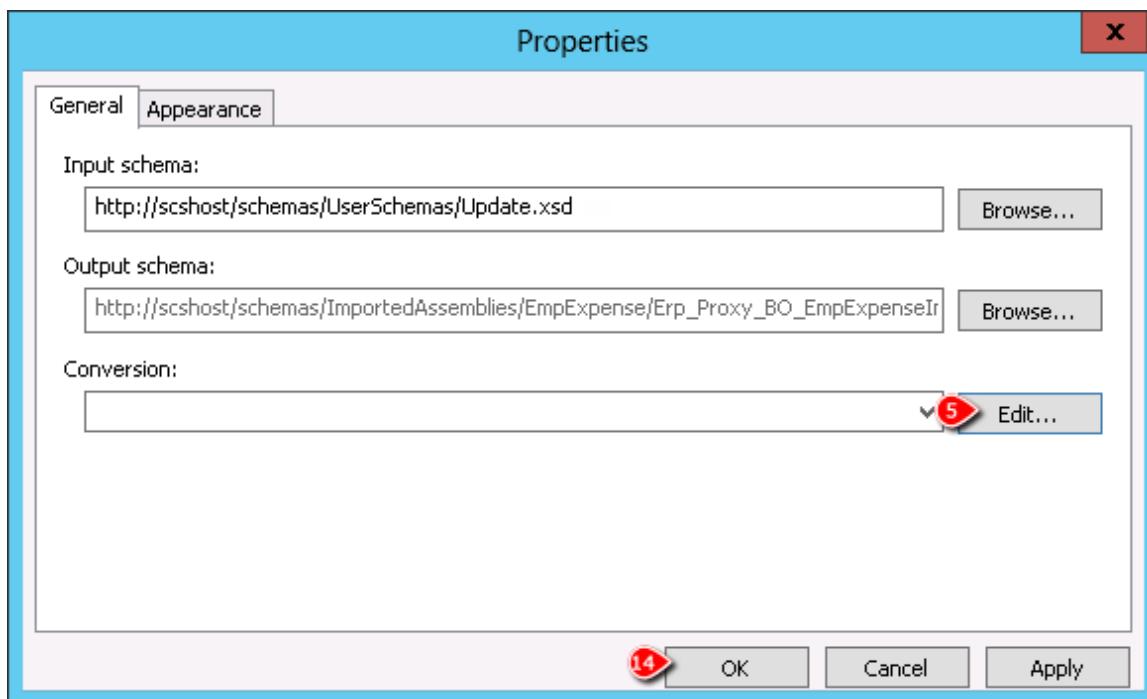
A conversion can find a Requester element with dynamic properties even if it is divided from the Requester by the Splitter, Condition or Choice element.

3. Draw Connections from the previous workflow element to the Conversion and from the Conversion to the Requester.



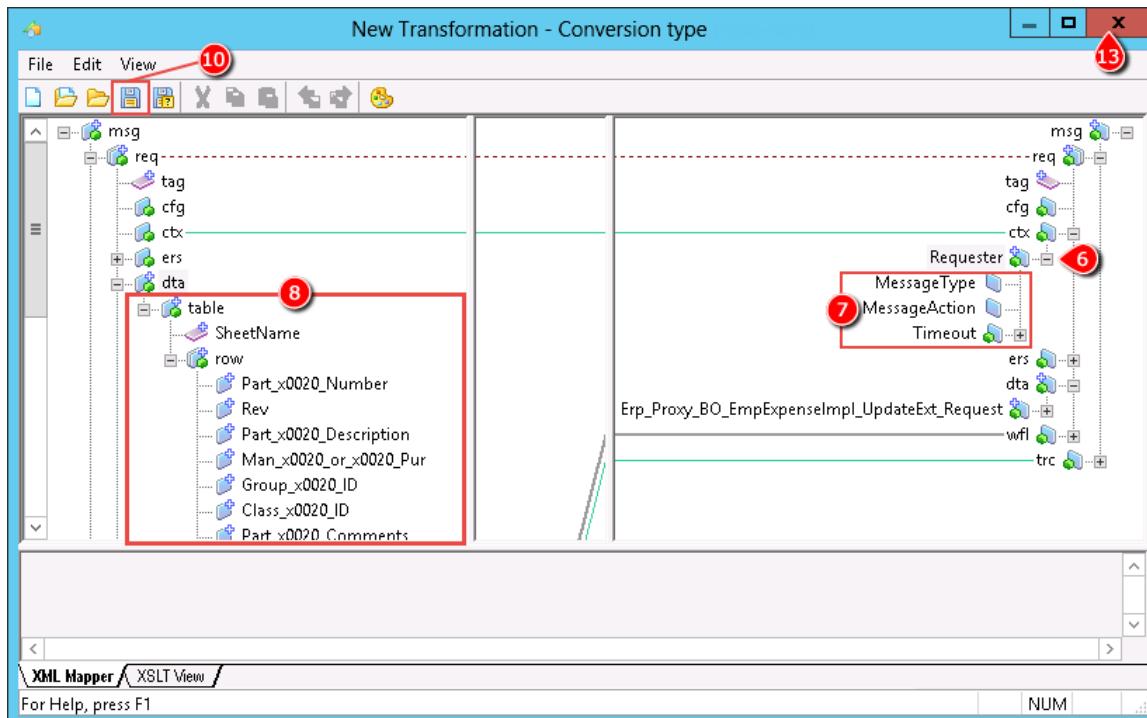
4. Double-click the Conversion.

The **Properties** dialog is displayed.



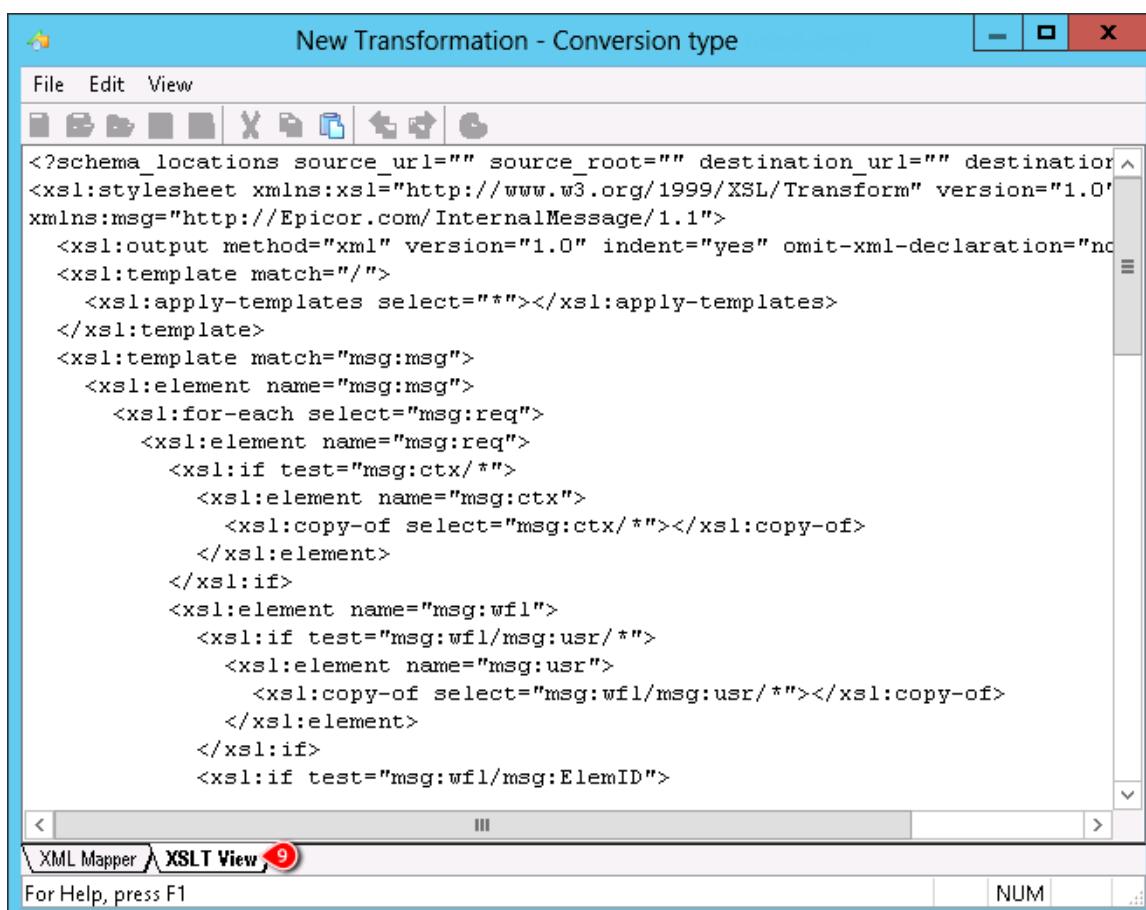
- In the **Properties** dialog next to the **Conversion** field, click the **Edit** button.
XML Mapper window is displayed.

- In the Mapper right pane, expand **msg > reg > ctx > Requester** nodes.



- Under the **Requester** node, view all the Requester properties that can be set dynamically.

8. Map message data from the left pane onto the Requester properties, or assign literal values to the Requester properties.
9. You can edit the xml in XSLT view as well.



The screenshot shows the 'New Transformation - Conversion type' window. The main area displays an XSLT transformation script:

```

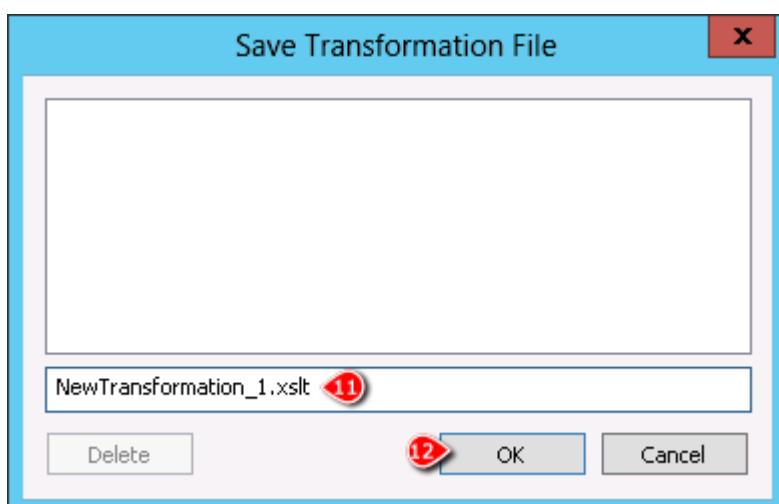
<?schema_locations source_url="" source_root="" destination_url="" destination_type="xml" version="1.0" encoding="UTF-8" standalone="no" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msg="http://Epicor.com/InternalMessage/1.1">
  <xsl:stylesheet version="1.0" indent="yes" omit-xml-declaration="no" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msg="http://Epicor.com/InternalMessage/1.1">
    <xsl:output method="xml" version="1.0" indent="yes" omit-xml-declaration="no" />
    <xsl:template match="/">
      <xsl:apply-templates select="*"/>
    </xsl:template>
    <xsl:template match="msg:msg">
      <xsl:element name="msg:msg">
        <xsl:for-each select="msg:req">
          <xsl:element name="msg:req">
            <xsl:if test="msg:ctx/*">
              <xsl:element name="msg:ctx">
                <xsl:copy-of select="msg:ctx/*"/>
              </xsl:element>
            </xsl:if>
            <xsl:element name="msg:wfl">
              <xsl:if test="msg:wfl/msg:usr/*">
                <xsl:element name="msg:usr">
                  <xsl:copy-of select="msg:wfl/msg:usr/*"/>
                </xsl:element>
              </xsl:if>
              <xsl:if test="msg:wfl/msg:ElemID">

```

The bottom navigation bar shows 'XML Mapper' and 'XSLT View' with a red circled '9' icon, indicating pending changes. The status bar at the bottom says 'For Help, press F1'.

You can find values for MessageType and MessageAction in Message Attributes node of Administration Console.

10. When the mapping is complete, click **Save**.
11. In the **Save Transformation file** window, enter a name for the XSLT file.



12. Click **OK**.
13. Close the XML Mapper window.
14. In the **Properties** window, click **OK**.

Conversion



Use this workflow element to convert a document from one format to another and to store values returned from other workflow elements 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.

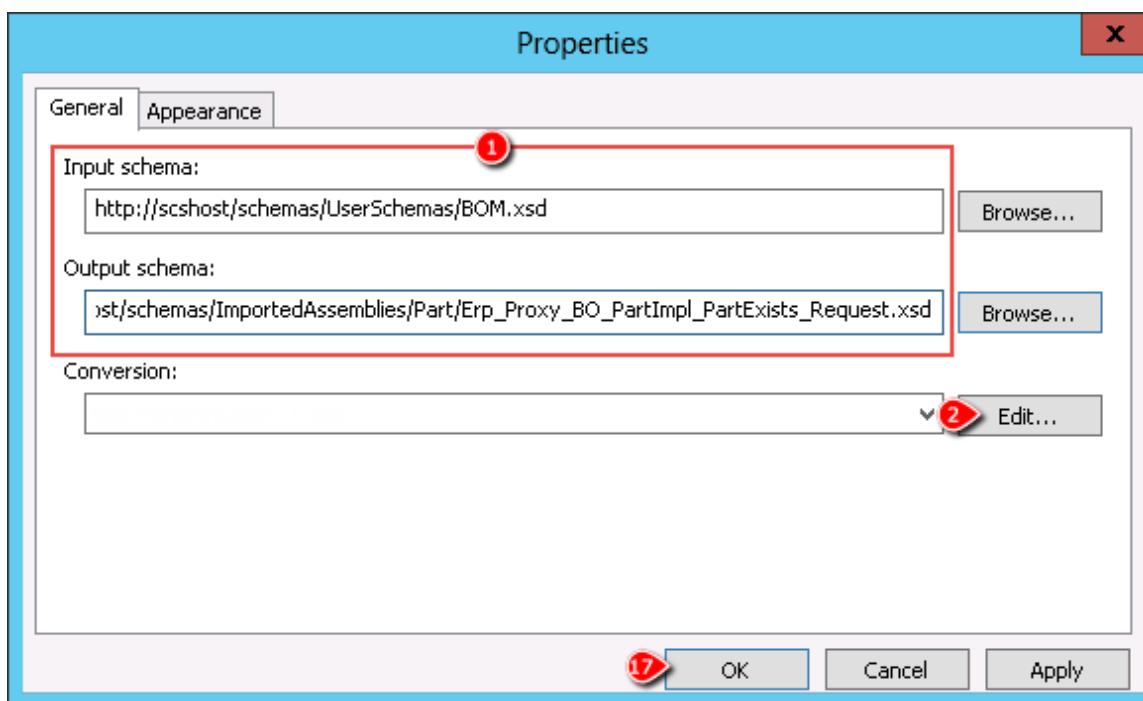
Conversion Example

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 .NET Call to the PartExists .NET 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:

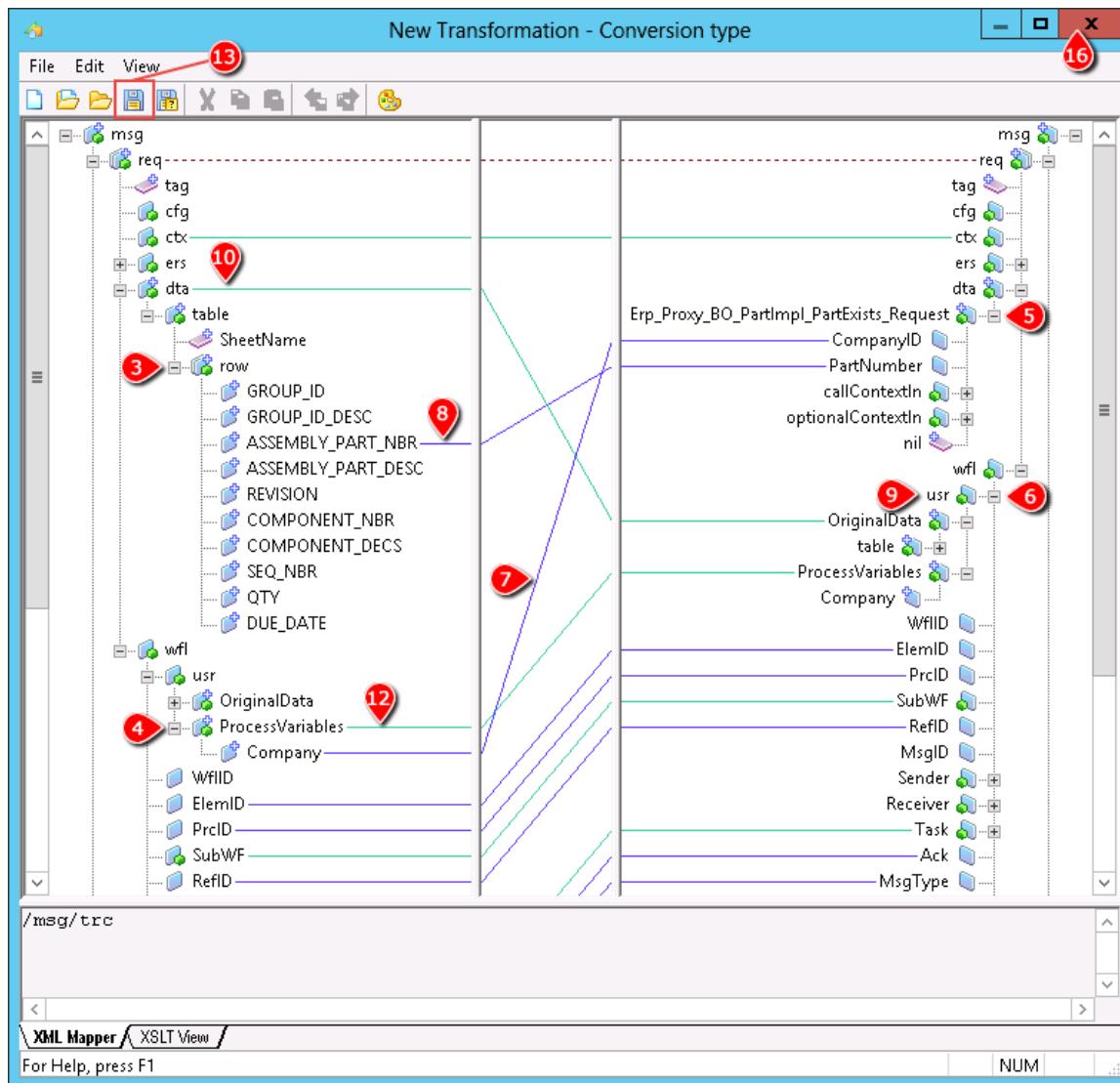
1. In the **Properties** window, select the **Input schema** and **Output schema** to indicate the structure of incoming and target documents.



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 .NET Call workflow element that will follow the conversion.

2. Click **Edit**.

3. In the left pane in the tree view, expand the **req > dta > table > row** node in the incoming document.

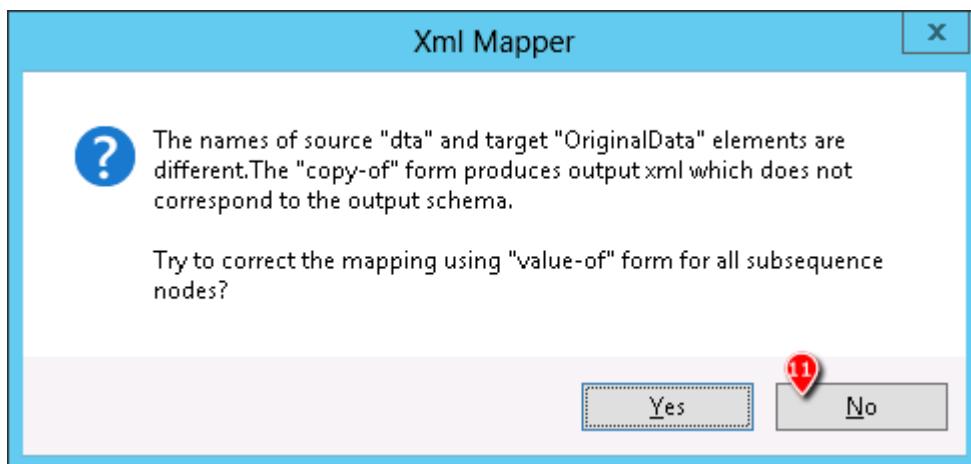


4. Expand the **wfl > usr > ProcessVariables** node in the incoming document.
5. In the target document in the right pane, expand the **req > dta > Erp_Proxy_BO_PartImpl_PartExists_Request** node.
6. In the target document, expand the **wfl > usr** node.
7. Map the **wfl > usr > ProcessVariables > Company** 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. If the **usr** node in the target document is mapped to a node in the source document, 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. To the warning, click **No**.

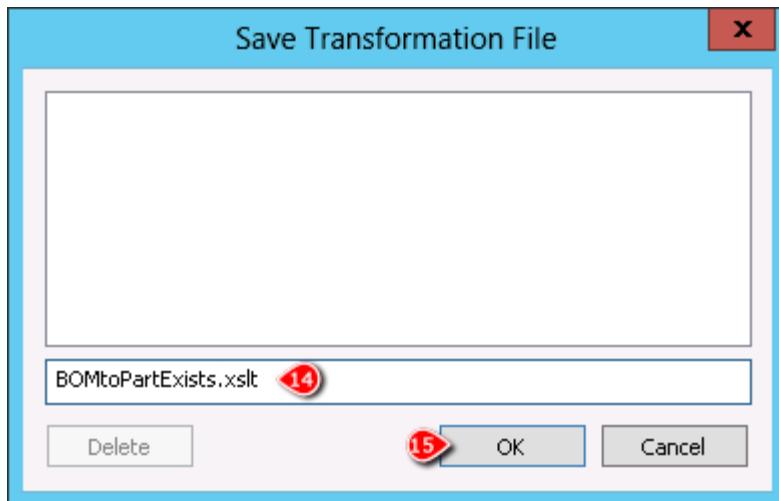


12. Map the **ProcessVariables** node in the incoming document to the **ProcessVariables** node in the target document.

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.

13. Click **Save**.

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



15. Click **OK**.

16. Close the XML Mapper window.

17. In the **Properties** window, click **OK**.

Condition



Use this workflow element 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 workflow element.

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.

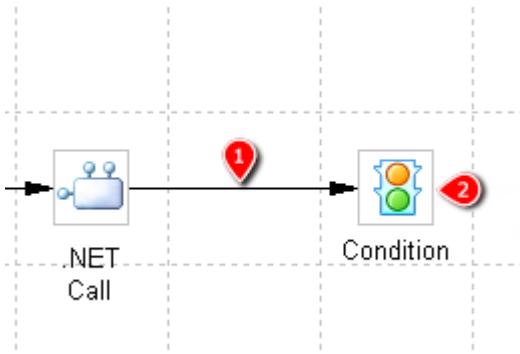
Condition Example

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 workflow element. 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. In the Workflow design area, add the inbound **Connection** to the **Condition**.

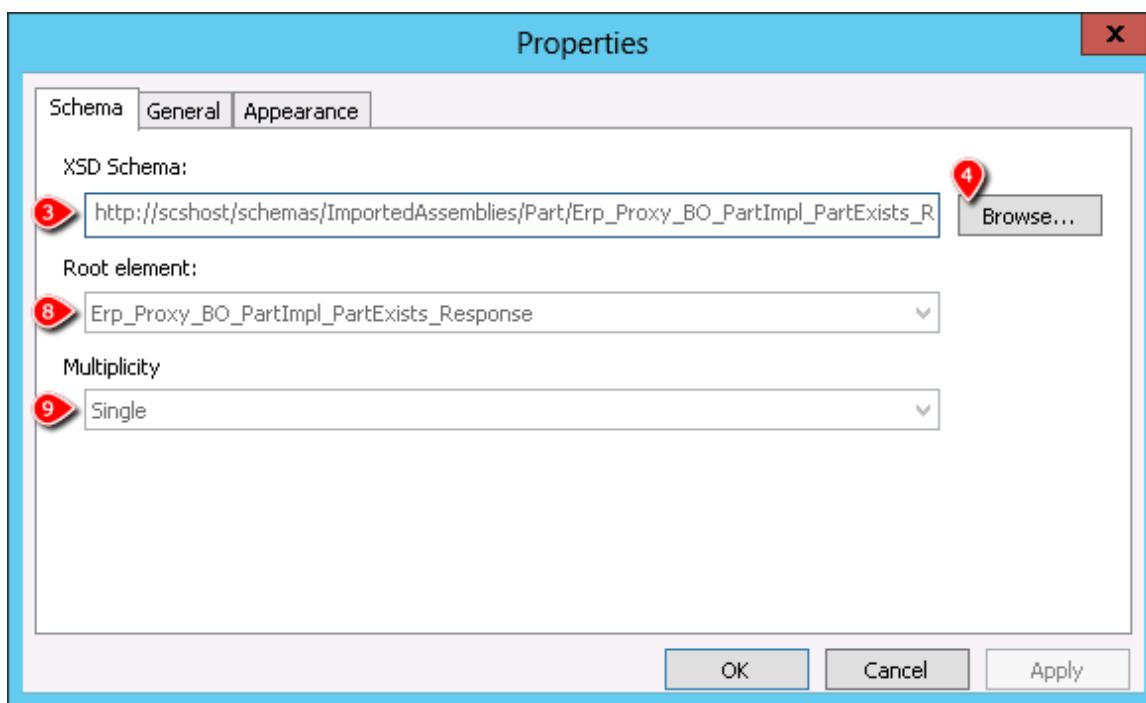


The workflow activity that precedes the inbound Connection supplies the document to be evaluated against the Condition rule.

2. Double-click the **Condition**.

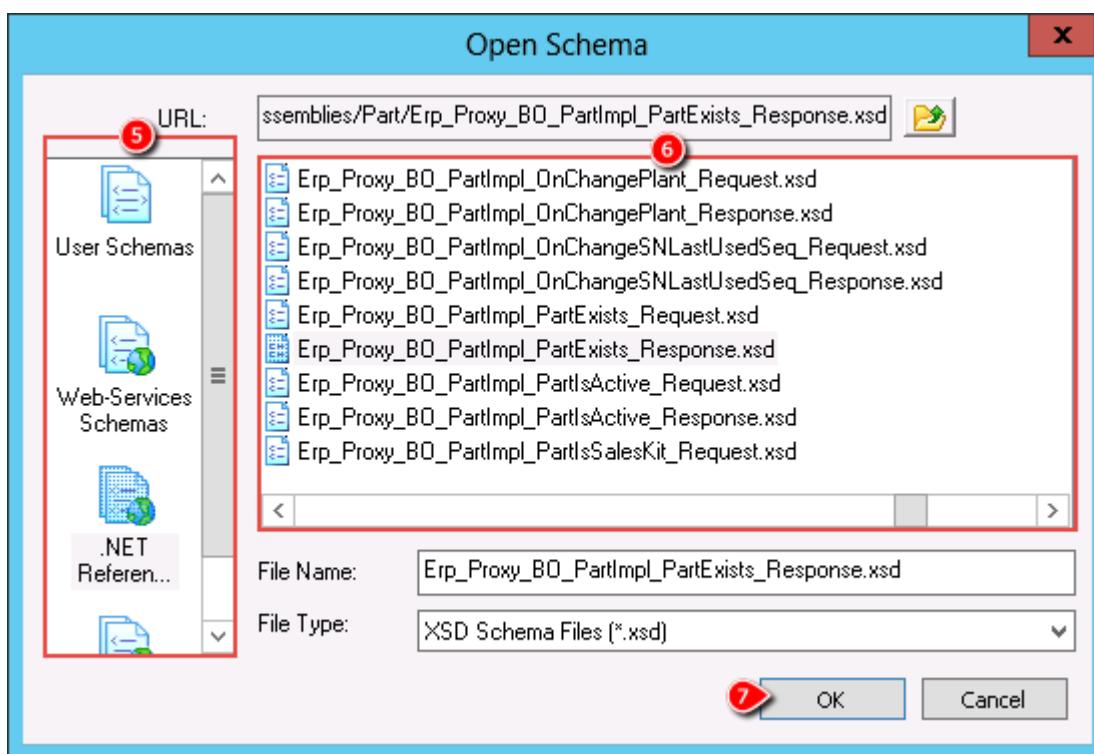
The **Properties** window displays.

3. If the previous workflow element (.NET Call in this example) already has the outbound schema set, this schema is automatically displayed in gray text in the **XSD Schema** field as the Condition schema.



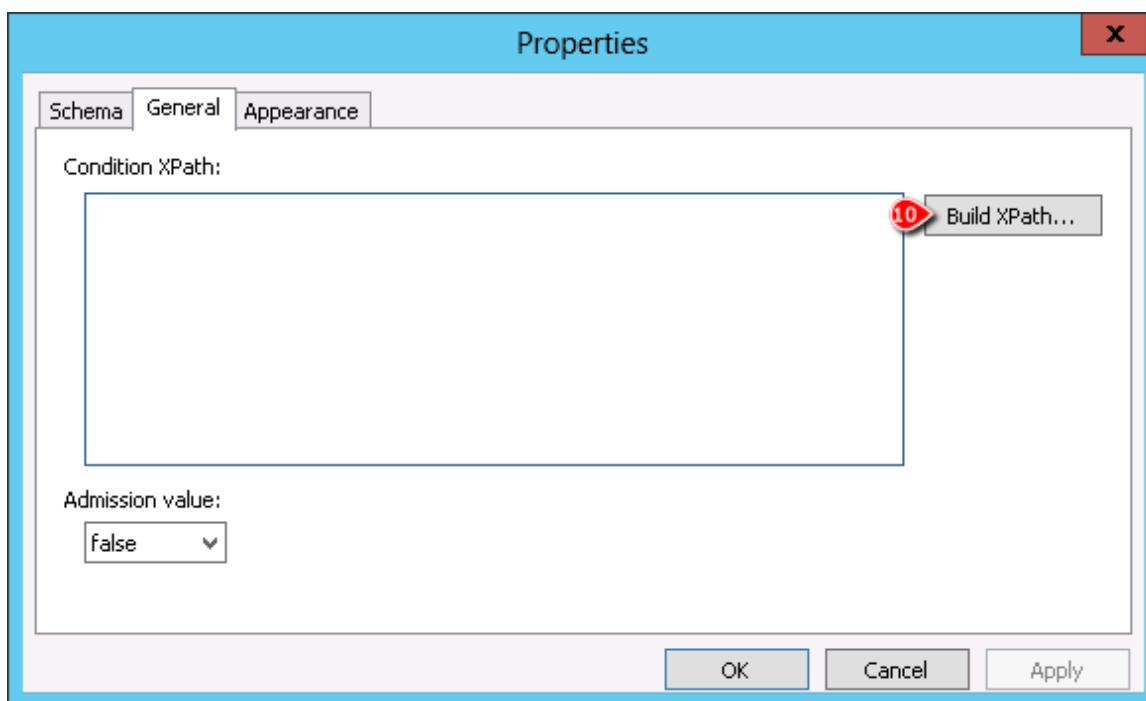
4. Click the **Browse** button to specify a different schema of the incoming document.
5. In the **Open Schema** dialog left pane, you can switch between Web-Services schemas, .NET Reference schemas, REST-Services schemas, or User schemas.

User schemas are generated in the Schemas node of Service Connect Administration Console, or in the Generate Schema from Sample Data tool in the Workflow Designer, or in the Workflow Designer when you click the Create sub-workflow schema button on the Cycling tab of Sub-workflow properties dialog.

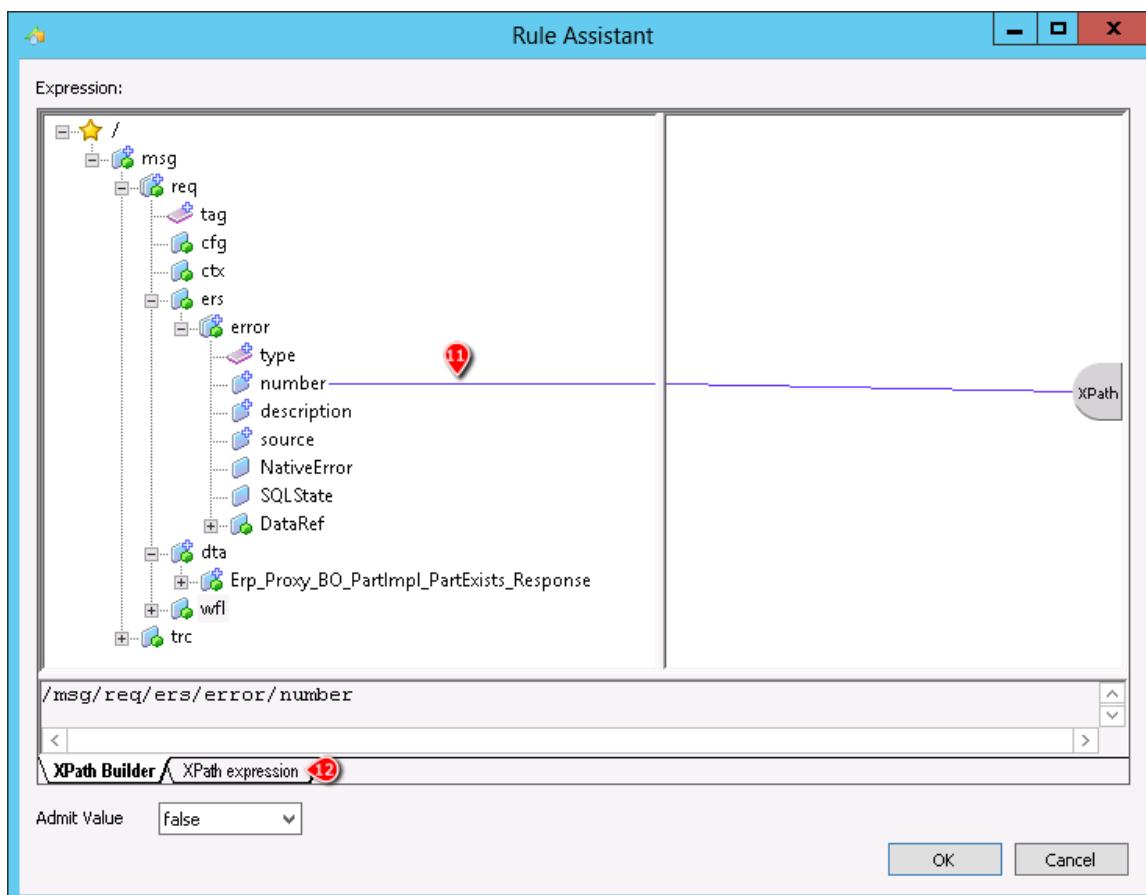


6. In the **Open Schema** dialog right pane, navigate to and select a schema.
7. Click **OK**.
8. Select a **Root Element** for the selected schema.
9. Select the **Multiplicity** for the selected schema.
Select **Single** to use just one element under the dta element, or select **Multiple** to utilize a collection under the dta element.

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

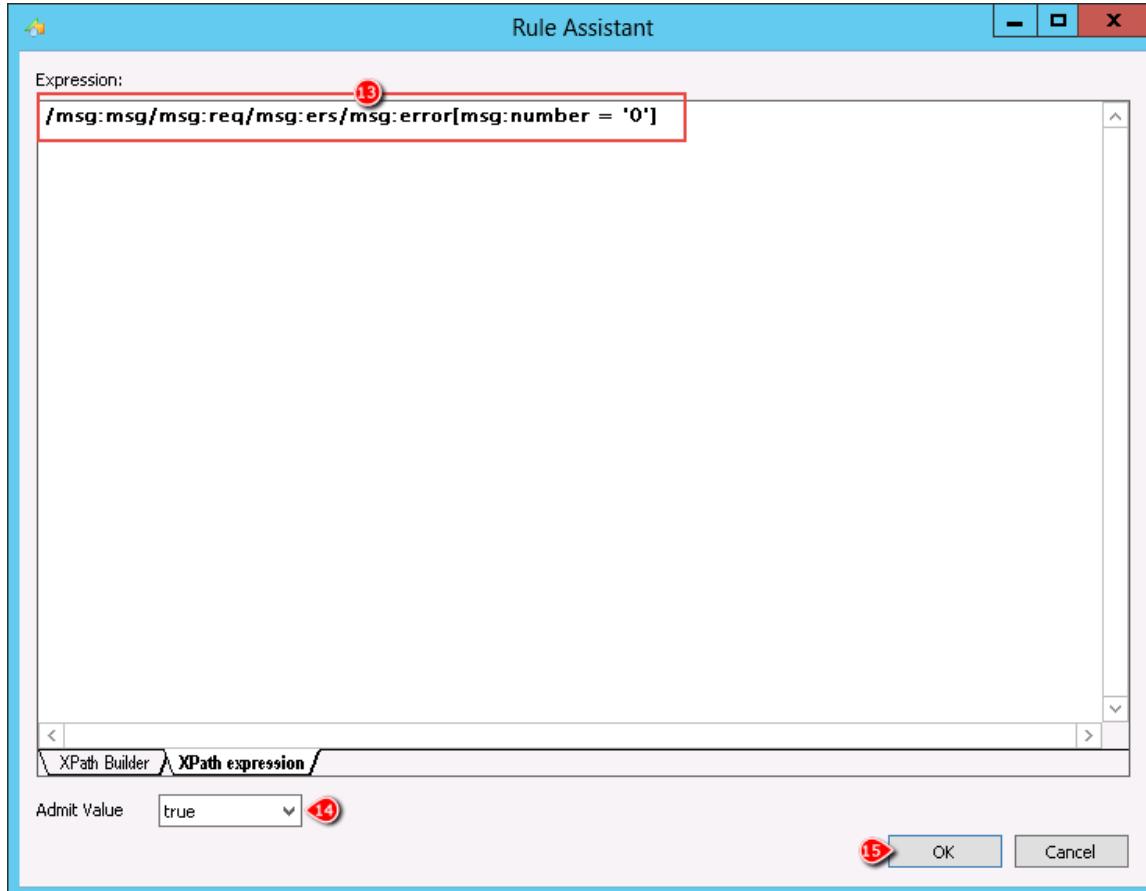


11. The **Rule Assistant** displays. To define a rule, expand the nodes in the document and map the node you want to test to the **XPath** marker on the right.



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.

12. Click the **XPath expression** tab.
13. Add a filter expression to the XPath so the rule tests the value of the number node.

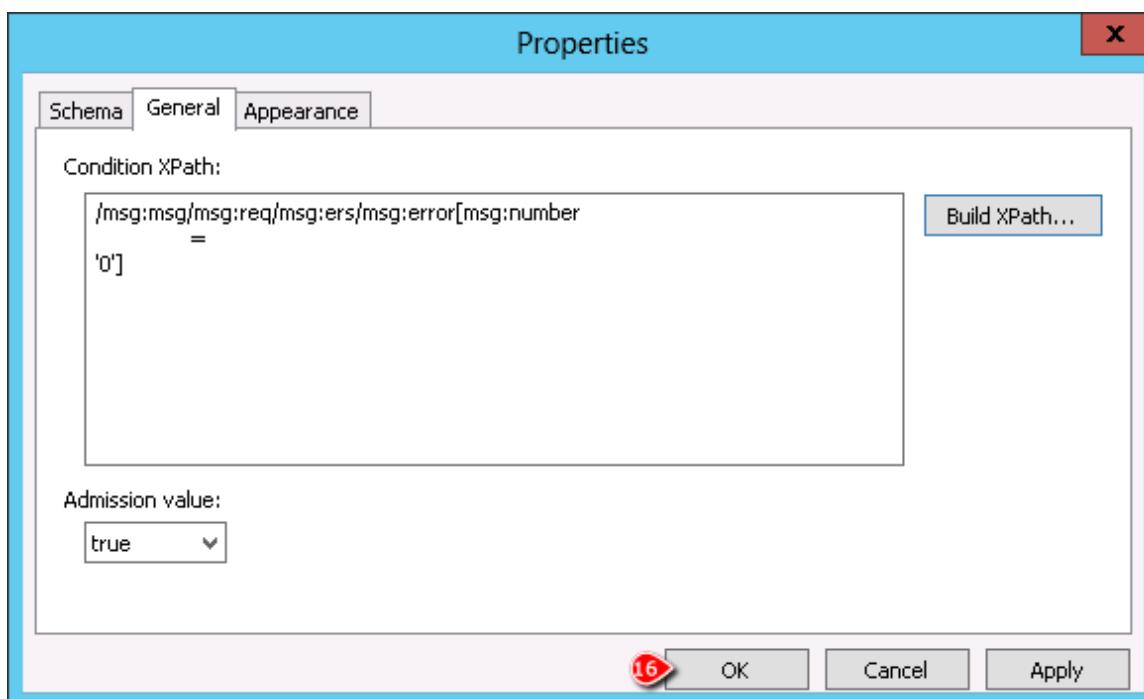


This XPath expression will evaluate to true when the value of the msg:number node equals zero.

The XPath expression is displayed in simplified form: namespaces are omitted and only the xml elements names are displayed.

14. 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.
15. Click **OK**.

16. In the **Properties** window, click **OK**.



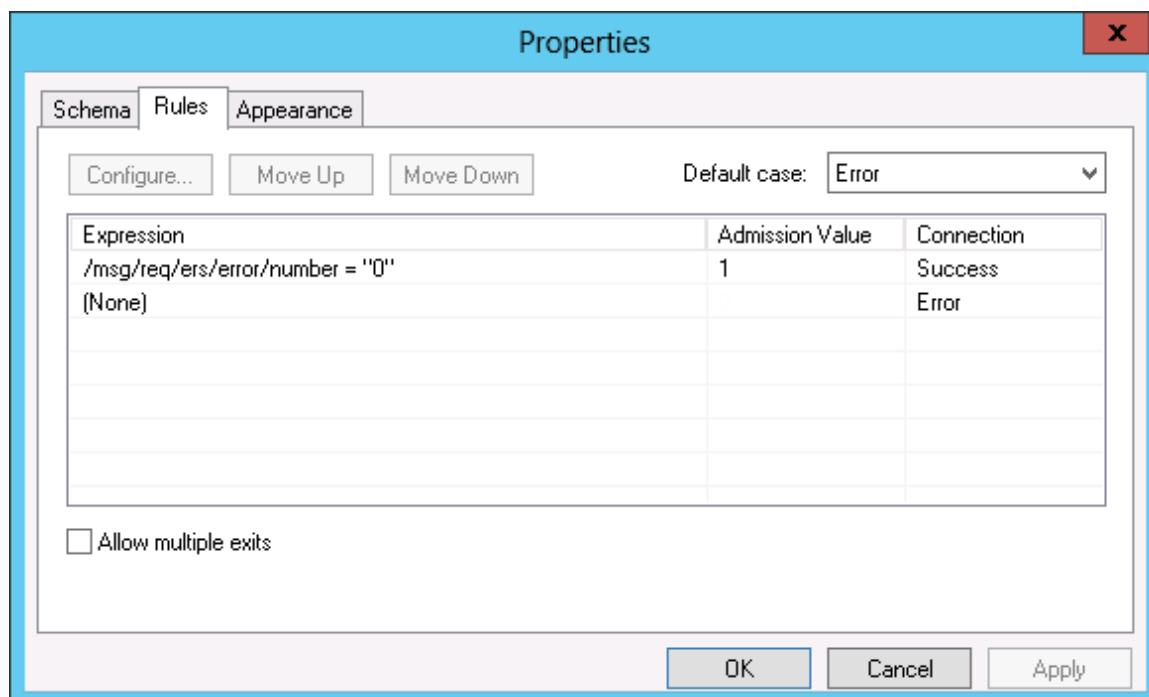
Choice



Use this workflow element 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 workflow element is evaluated against the rules to determine which of the outbound Connections the document will follow. You can configure Choice 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.

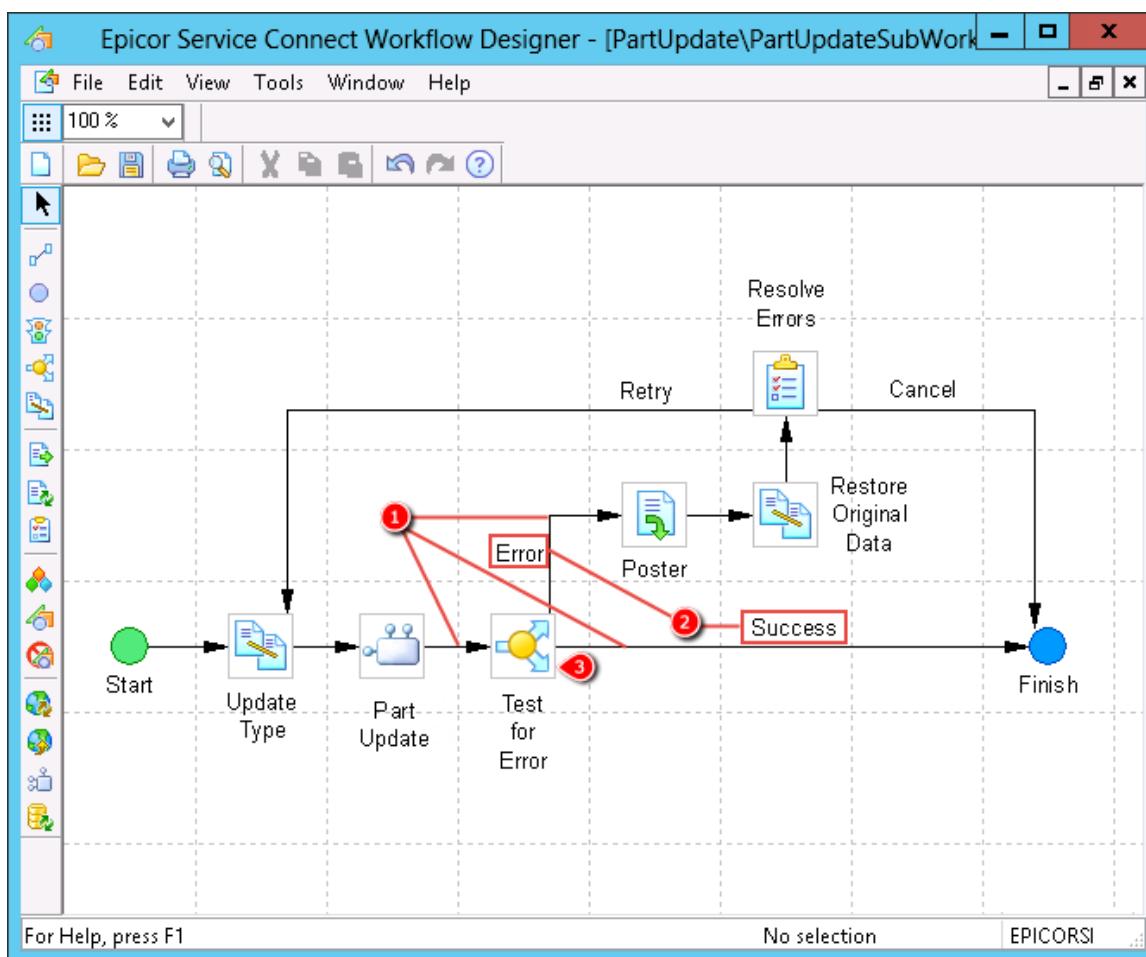
Choice Example

An incoming document contains part update information. The workflow uses a .NET Call to update the Epicor application. The .NET Call output is passed to a Choice workflow element. If the .NET BO method returns an error, the Choice routes the document to the Poster element that notifies a user about an error. If the .NET Call 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**.



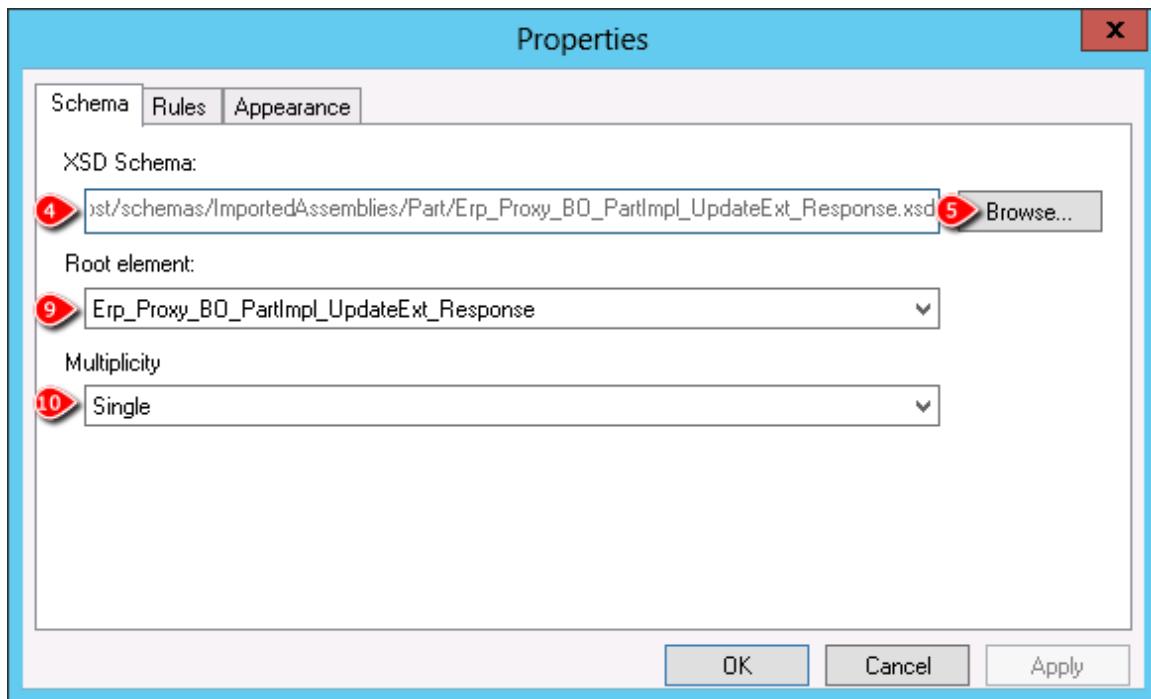
The workflow element that precedes the inbound Connection supplies the document that is evaluated against the Choice rules. In this example, the preceding element is a .NET Call titled Part Update.

2. Set the **Caption** on each outbound Connection.

Give each Connection a logical name to help you when defining the Choice rules.

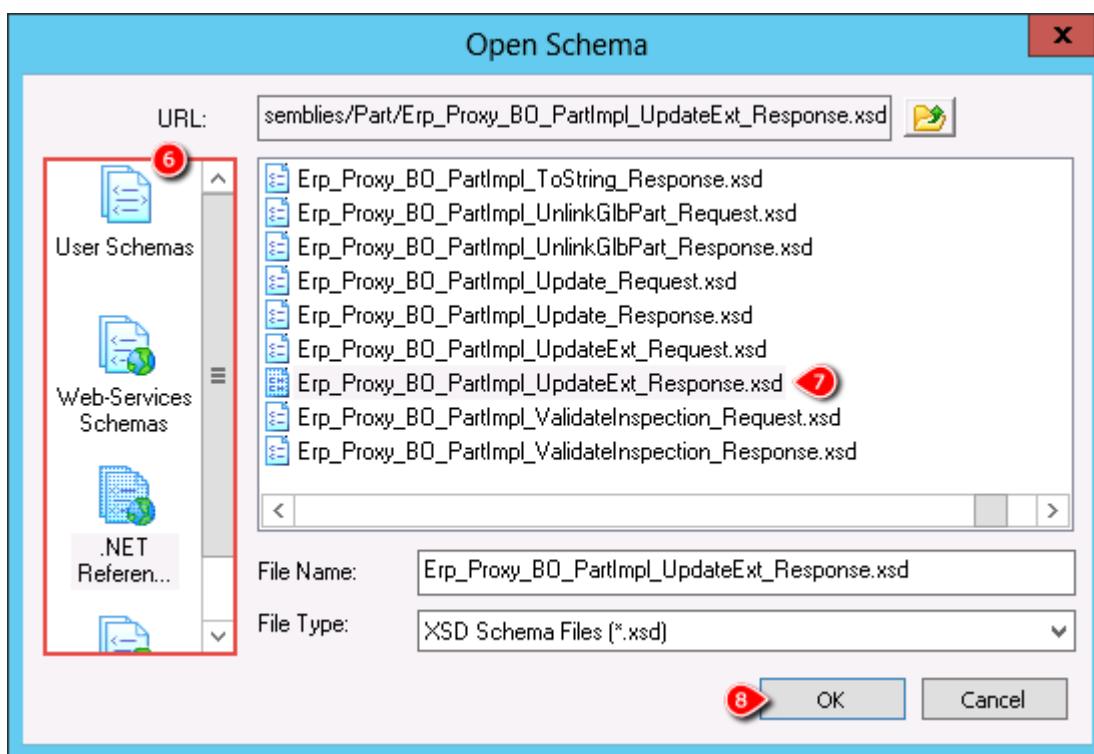
3. Double-click the **Choice**.

4. The **Properties** window displays. If the previous workflow element already has the outbound schema set, this schema is automatically displayed in gray text as the Choice XSD Schema.



5. To select a different incoming document schema, click the **Browse** button.
6. In the **Open Schema** dialog left pane, you can switch between Web-Services schemas, .NET Reference schemas, REST-Services schemas, or User schemas.

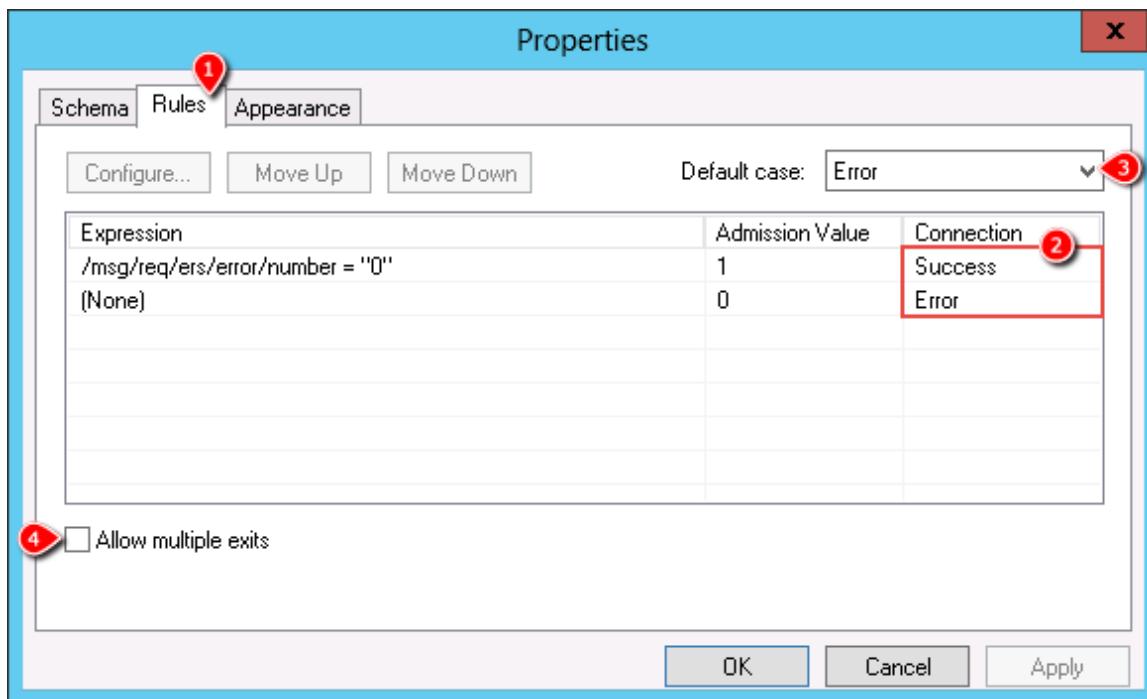
User schemas are generated in the Schemas node of Service Connect Administration Console, or in the Generate Schema from Sample Data tool in the Workflow Designer, or in the Workflow Designer when you click the Create sub-workflow schema button on the Cycling tab of Sub-workflow properties dialog.



7. In the **Open Schema** dialog right pane, navigate to and select the schema.
8. Click **OK**.
9. In the **Properties** window, select a **Root Element** for the selected schema.
10. 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.

Set up Choice Rules

1. In the **Properties** window, navigate to the **Rules** sheet.



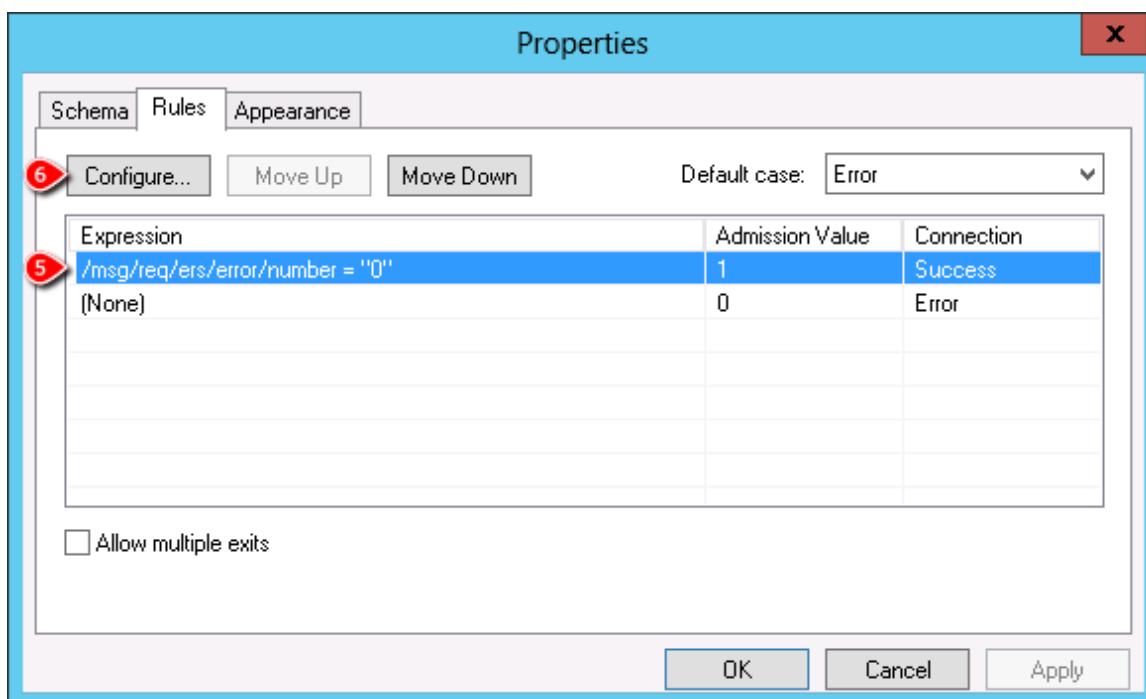
2. Notice the two outbound Connections are listed by name in the **Connection** column.

3. Select the appropriate **Default case**.

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

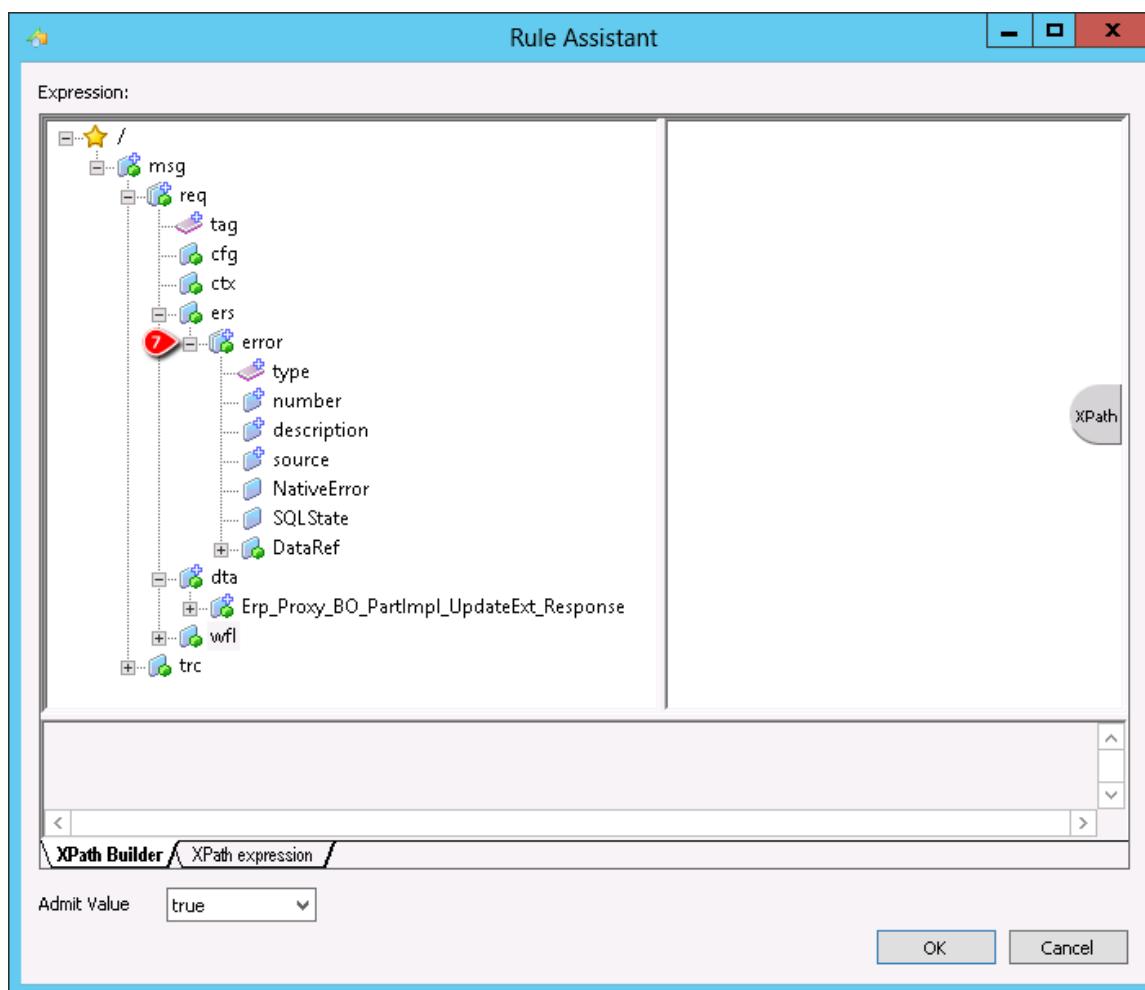
If the Allow multiple exits check box is selected, the document will be evaluated against each rule in the grid even after it has satisfied a rule, and this can be sent to more than one outbound Connection. If the check box is clear, then the document will be sent to the first outbound Connection for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

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

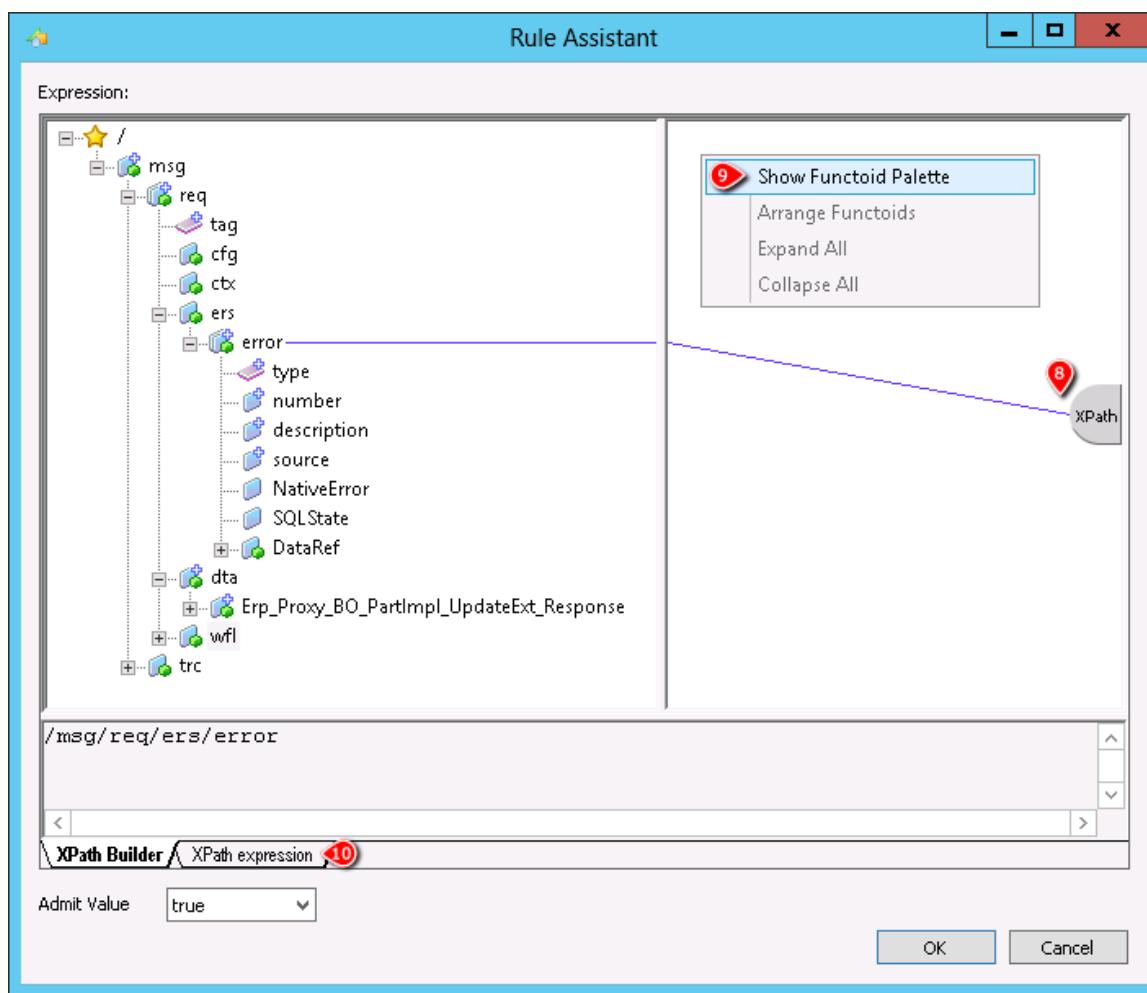


6. Click **Configure**.

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



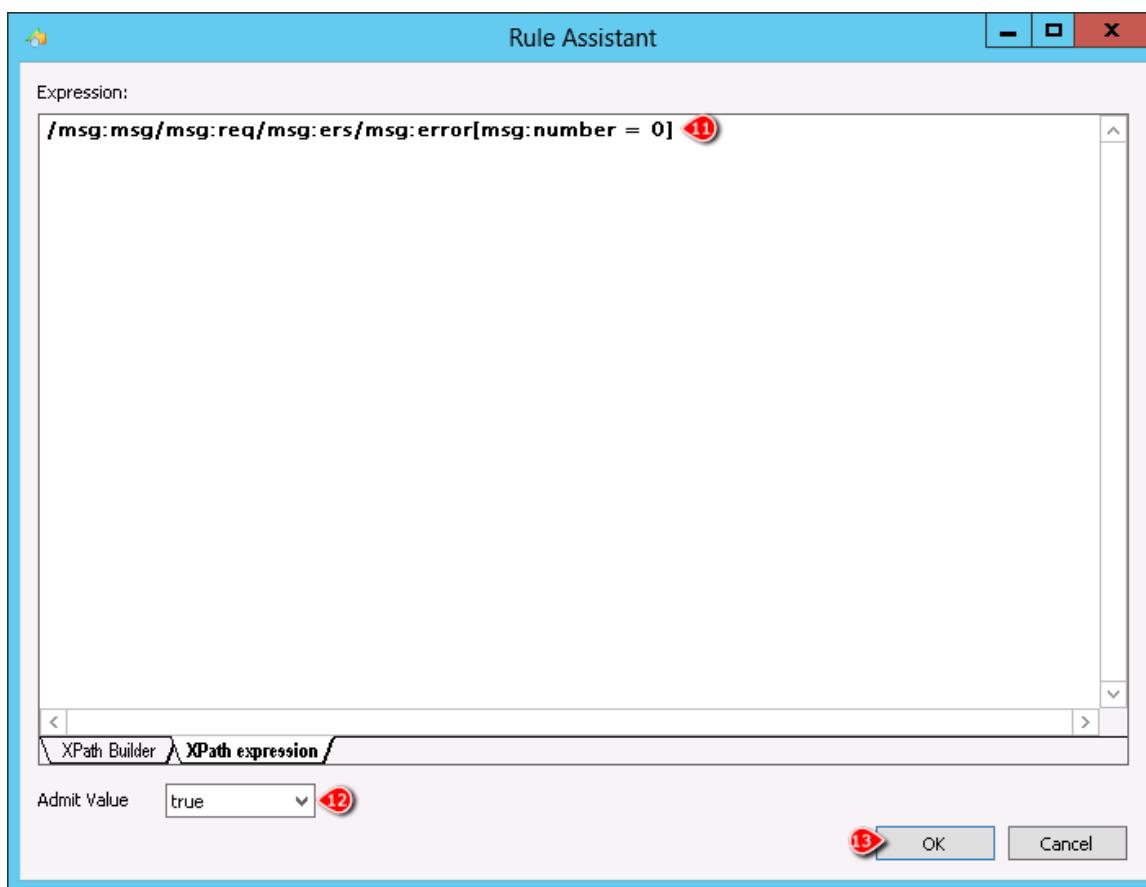
8. Map the node you want to use for the rule to the **XPath** marker on the right.



9. Right-click anywhere in the right pane and select **Show Functoid Palette**.

You can use functoids to visually build the rule customizations. For detailed description of the functoids refer to the Functoids section later in this chapter.

10. Click the **XPath expression** tab if you need to view and edit the XPath expression you built.



In this example, the rule has been customized to test the value of the msg>req>ers>error>number node instead of testing for the presence of the node.

11. View the XPath expression.

It is displayed in simplified form: namespaces are omitted and only the xml elements names are displayed.

12. 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.

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

Web Method



Use this workflow element 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.

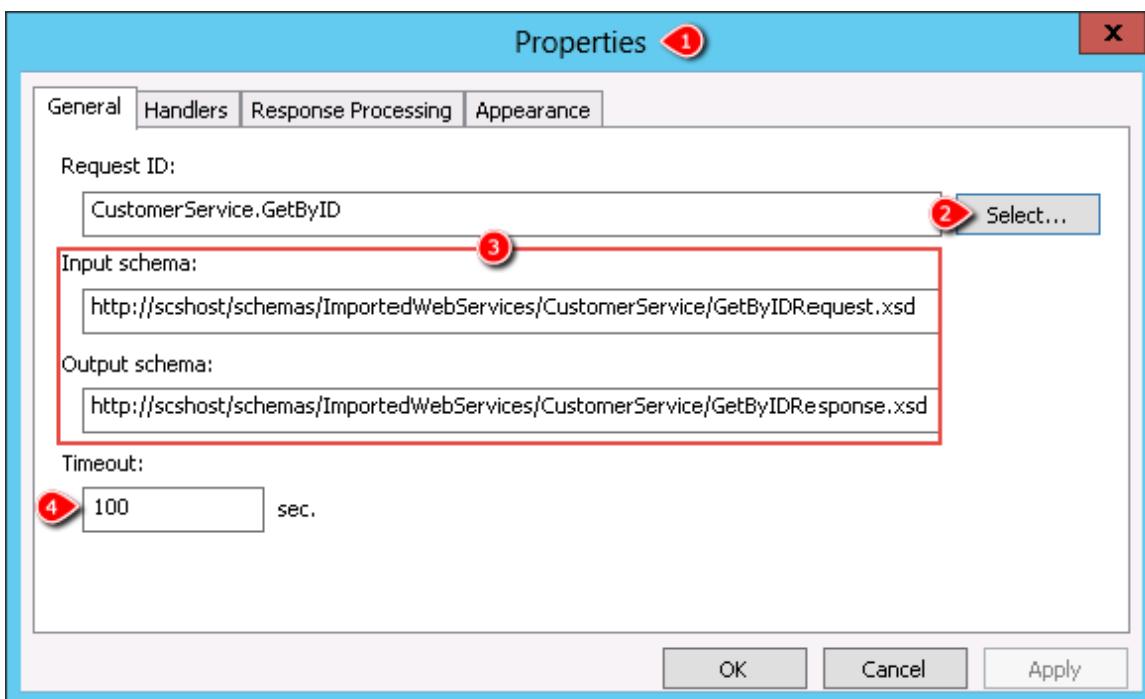
Web Method Example

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

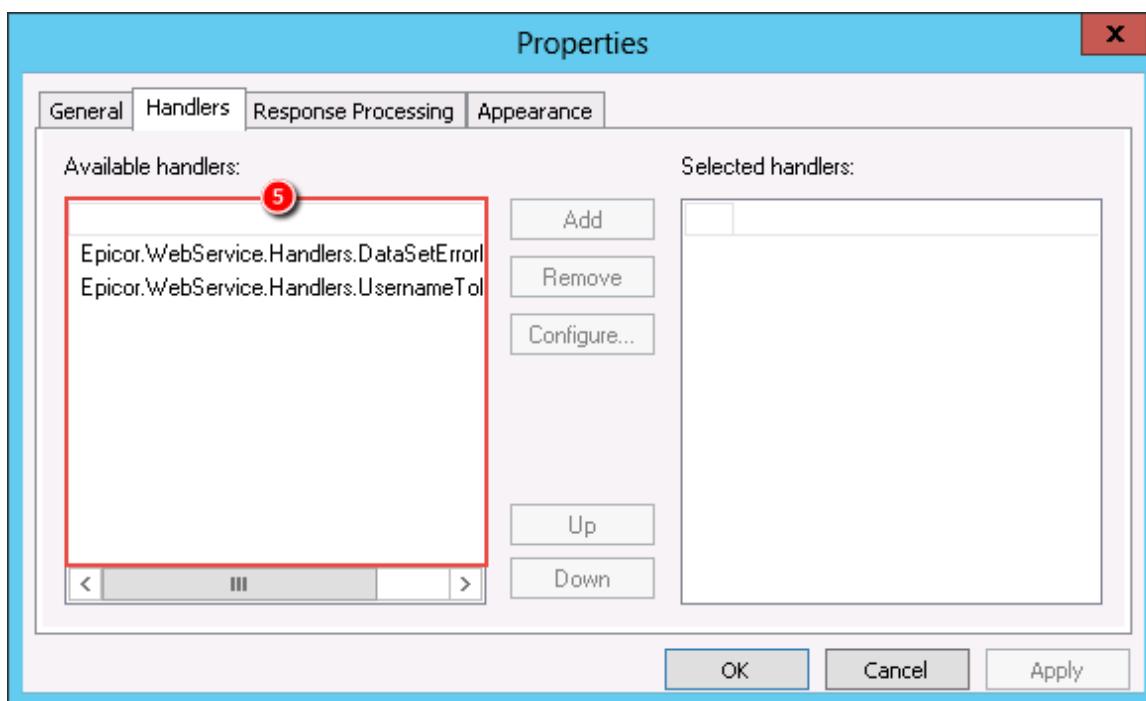
To set up a Web Method:

1. In the Workflow design area, double-click the **Web Method** to open the **Properties** window.



2. In the **Properties** window, on the **General** tab, click **Select** to choose the method you want to call in the Request ID field.
3. Notice, 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 workflow element, 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 workflow element that follows the Web Method must use, or be ready to accept, a document formatted according to the Web Service response schema.
4. To set the time period the Web Method waits for request to complete, use the **Timeout** field. The default value is 100 seconds. If the field is empty, the default value is used. Setting the field value to -1 indicates that the request does not time out.
This field is ignored for E9.05 WCF and E10 WCF Web Methods.

5. On the **Handlers** tab, select handlers from the list of available handlers.



Default settings allow overriding set of handlers and their settings for this particular workflow element. These settings take precedence over the definitions for the Web Service and Web Method.

6. If the handlers configuration was customized in the Workflow Designer, Web Method icon in the workflow diagram changes to display a spanner.



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:



When response processing is enabled, the Web Method icon changes from the standard icon



to the customized one



Response Processing

The calls of a Web Method, .NET Call, or iScala Manager from Service Connect workflows, or the SQL statements that a DBOperation workflow element runs 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 the **Response Processing** tab to configure the number and frequency of loop repetitions and the workflow behavior when the maximum number of loop repetitions is reached.

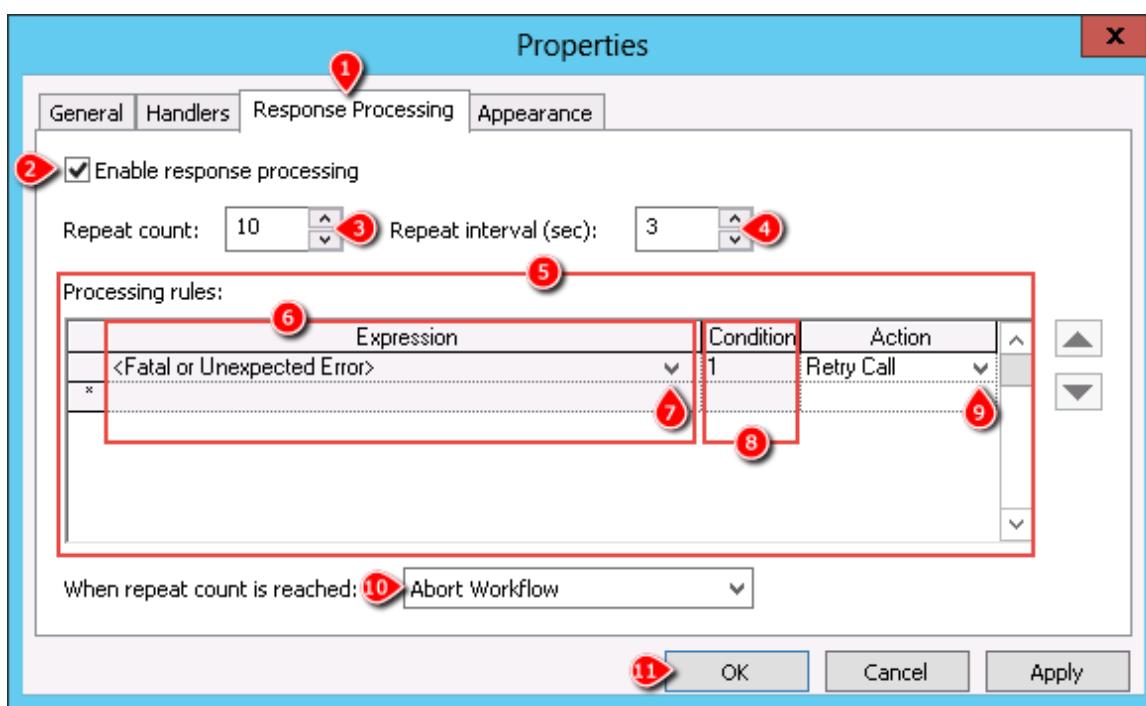
Response Processing tracking

Every time a workflow performs a Web Method, .NET call, iScala Manager, DBOperation, this event is recorded in the Event Log. The single trace is created or updated for the element with configured response processing. Repeat count is added to the execution point name. If there were failed attempts (with fatal or unexpected error) prior to the successful one, the trace is shown with a warning.

In the Document Tracking in Administration Console, the response processing functionality always shows the last response from the last Web Method, .NET call, iScala Manager, DBOperation call execution attempt. Although each attempt logs an error to Event view. Sometime error from the first attempt may differ from subsequent attempts. In the Trace Details window, open the Related events tab, and take a look at the first error by date.

To set up response processing:

1. In the Web Method **Properties** window, open the **Response Processing** tab to configure repetitive loops for the Web Method.



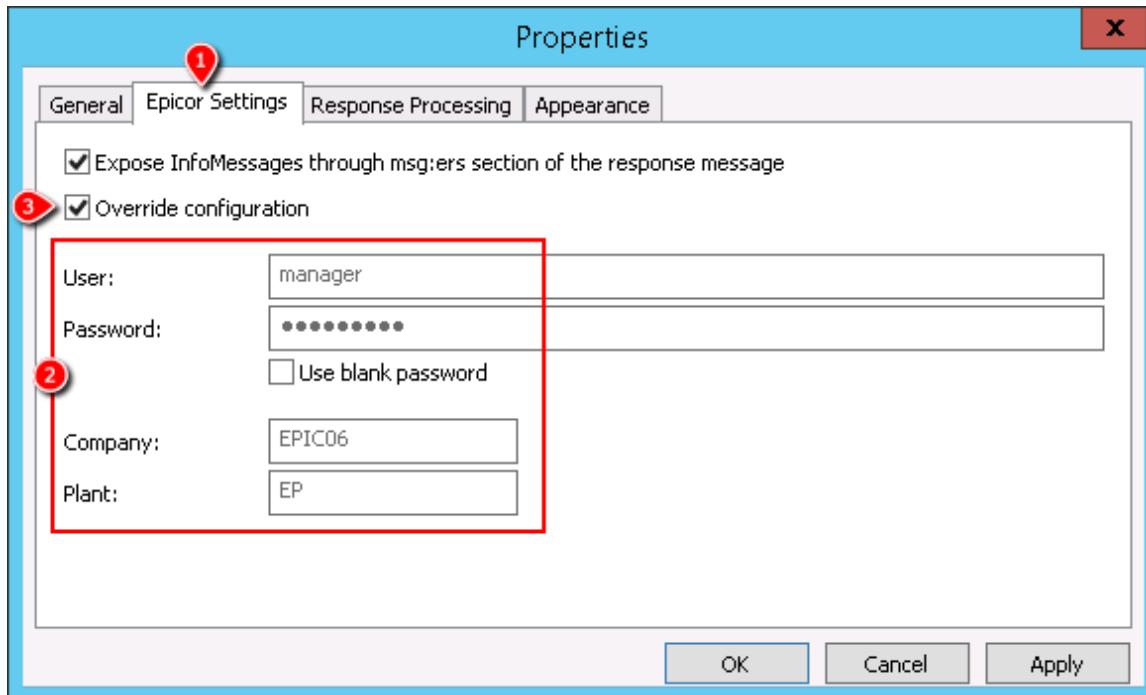
2. Select the **Enable response processing** check box to enable the response mechanism.
3. Specify the **Repeat count** - the number of attempts to repeat Web Method calls. If the call is successful and the specified number of repetitions is not reached, the workflow continues to execute. If the call is not successful, but the specified number of repetitions is reached, the workflow behaves as specified in the **When repeat count is reached** field.

4. Specify the **Repeat interval**, in seconds, between the repeated Web Method calls.
The maximum interval is 999 seconds. By default, the interval is 3 seconds.
5. In the **Processing rules** grid, you can add, edit, and delete the errors and expressions used to handle Web Method responses.
6. 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.
7. To perform an action on the selected expression, click the arrow 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.
8. 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**.
9. 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.
10. 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**.
11. Click **OK** to apply the selected parameters.

Epicor Credentials Setup for Epicor 10 WCF Web Methods

If you configure an Epicor 10 WCF Web Method, the **Properties** dialog displays an additional **Epicor Settings** tab. Use this tab to override the Epicor logon credential, Company and Plant.

1. In the Web Method **Properties** window, open the **Epicor Settings** tab.



2. In the **User**, **Password**, **Company** and **Plant** fields, the default configuration displays as grayed text.

The values you can see were provided when the corresponding Service Reference was imported in the Service Connect Administration Console.

3. To override the Web Method logon details, select the **Override configuration** check box and define new logon parameters.

In ESC, you can define Username, Password, Company and Plant for a Web Method in several ways. The settings override each other in the following order:

1. /msg/req/dta/{RefName}Request/CompanyId node in the request message in workflow Conversion Element.

Syntax:

```
Company=EPIC06;Plant=MfgSys
```

This is the highest priority that overrides the Company and Plant settings specified in any other way.

2. Incoming message to workflow

This is the highest priority that overrides the User name and password settings in all other places.

3. Workflow Process Properties

4. Workflow Element Properties

5. E10 WCF Web Service Reference Properties in Administration Console. They can be defined during reference import or reimport.

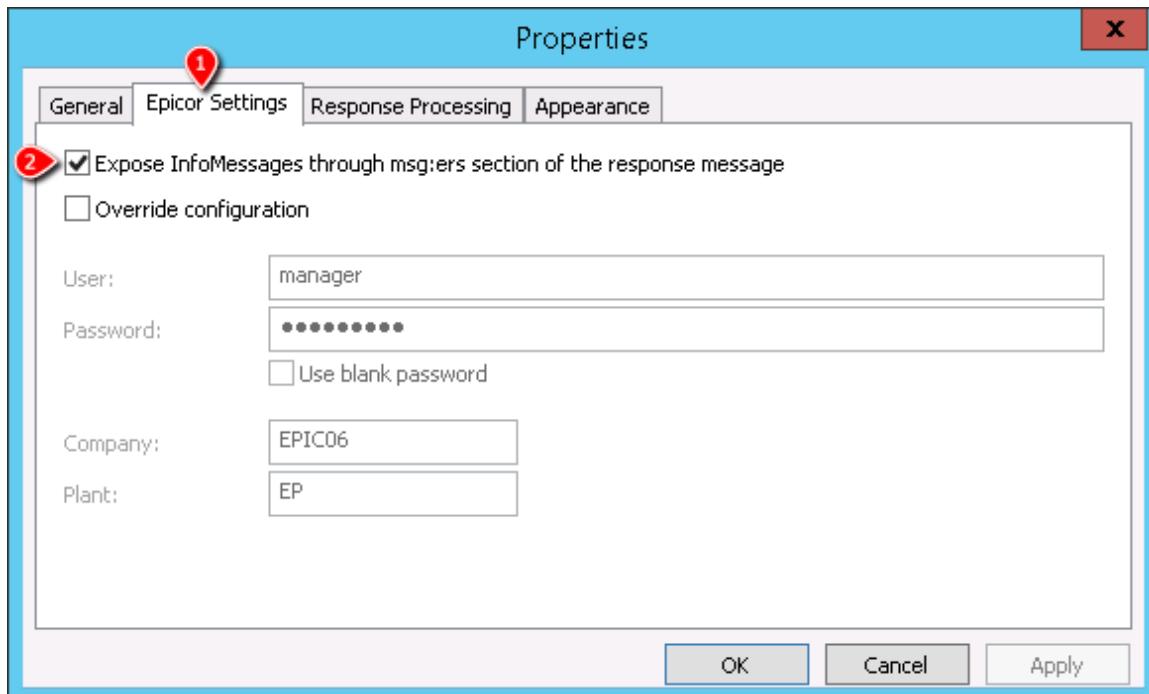
4. If the Epicor logon configuration was customized in the Workflow Designer, the Web Method icon changes to the following:



InfoMessage Setup for Epicor 10 WCF Web Methods

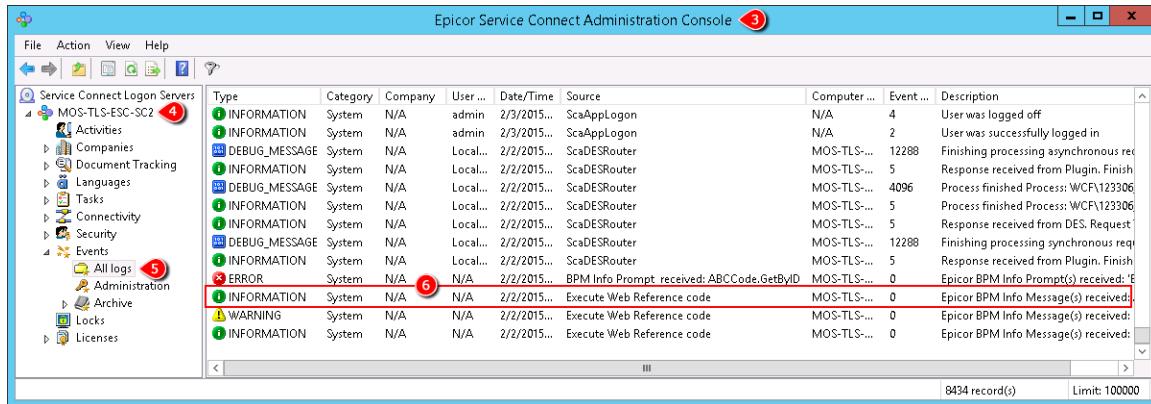
If you configure an Epicor 10 WCF Web Method, the **Properties** dialog displays an additional **Epicor Settings** tab. The tab contains InfoMessage-specific details.

1. In the Web Method **Properties** window, open the **Epicor Settings** tab.



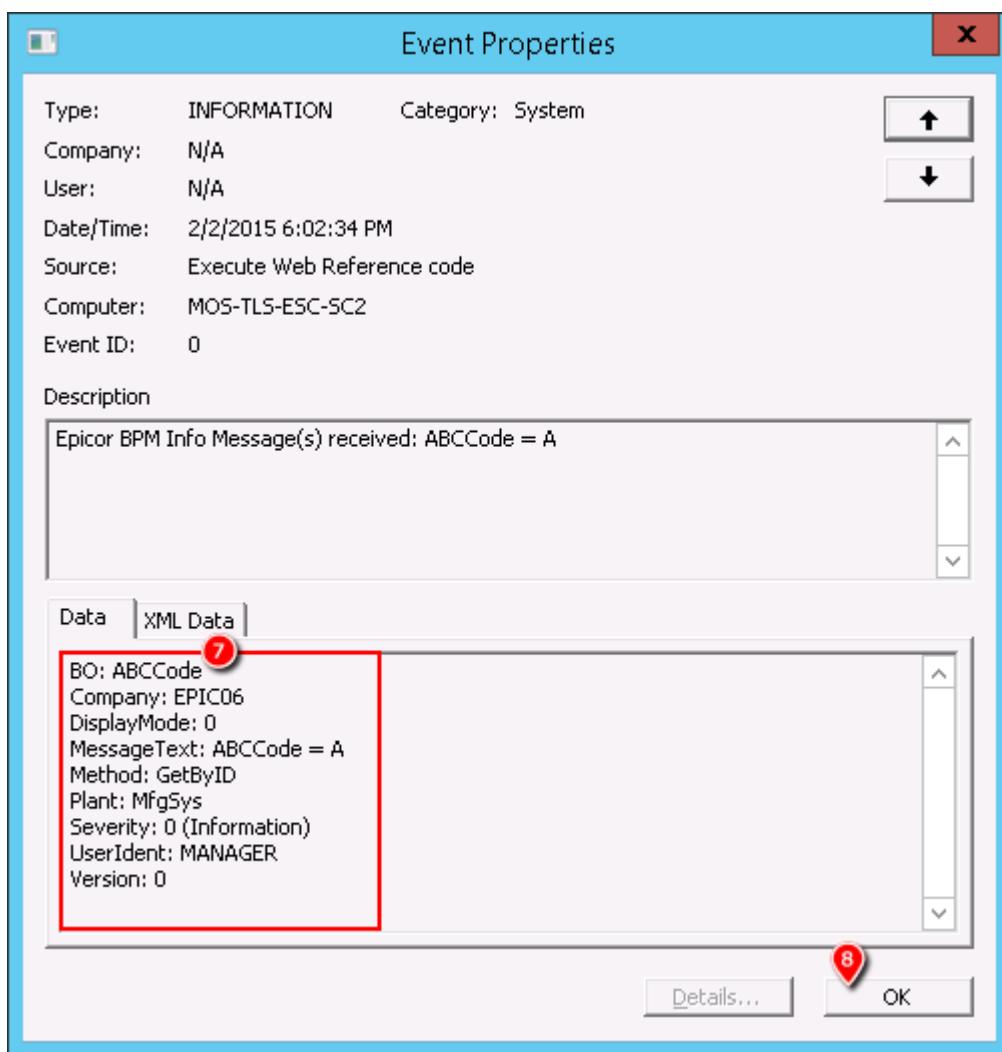
2. On the **Epicor Settings** tab, select the **Expose InfoMessages through msg:ers section of the response message** check box to be able to work with InfoMessages.
If you enable this option, the details of each InfoMessage are added to a separate error section. In this case response will keep data of BO method processing and additional error nodes. You can view this information in the Epicor Service Connect Administration Console.

3. Start the **Epicor Service Connect Administration Console**.



4. In the Tree View, expand the computer name node of your Service Connect installation.
5. Expand the **Events > All logs** node.
The right pane displays the list of recent events and their details.
6. In the right pane, double click one of the BPM-related events.
Error nodes can have Info, Warning, Error and Exception types according to severity of Info Messages.

7. In the **Event Properties** window, review the BPM data.

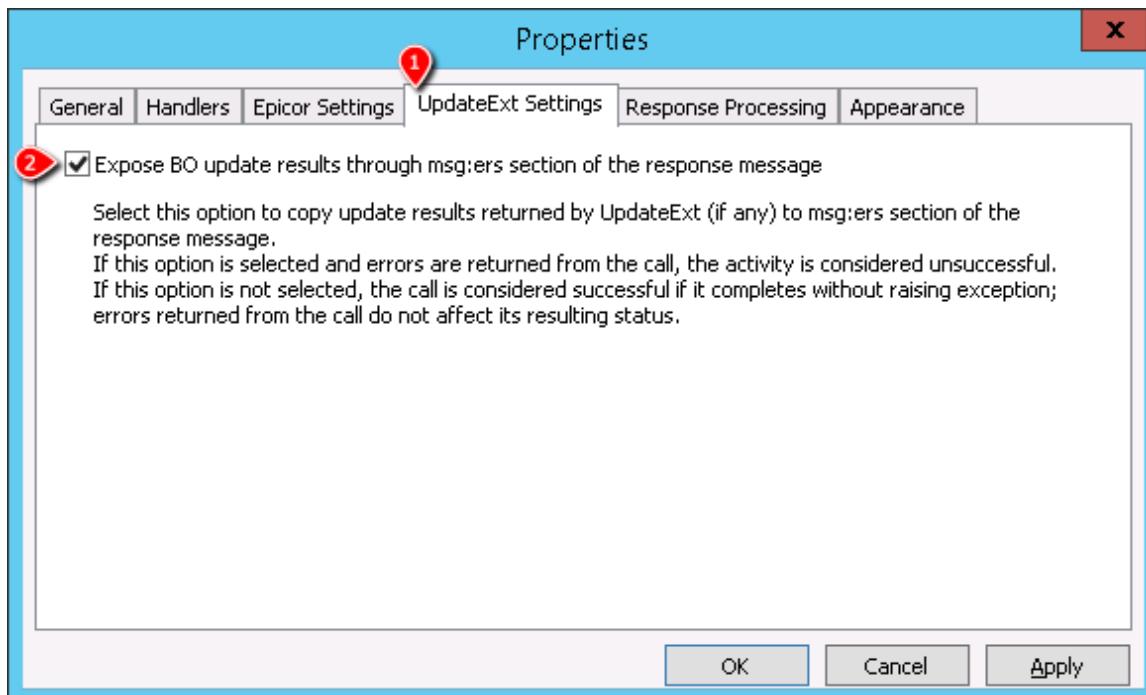


8. Click **OK**.

UpdateExt Setup

If you configure UpdateExt method, the **Properties** dialog displays an additional **UpdateExt Settings** tab.

1. In the Web Method **Properties** window, open the **UpdateExt Settings** tab.



2. On the **UpdateExt Settings** tab, select the **Expose BO update results through msg:ers section of the response message** check box to parse UpdateExt response to add information from BOUpdError as error in Service Connect Document Tracking.

When this check box is selected the update results returned by UpdateExt (if any) are copied to msg:ers section of the response message.

If this option is selected and errors are returned from the call, the activity is considered unsuccessful. If this option is not selected, the call is considered successful if it completes without raising exception; errors returned from the call do not affect its resulting status.

.NET Call



Use this workflow element to call methods exposed by a .NET Business Object (BO). You must import the BO 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.

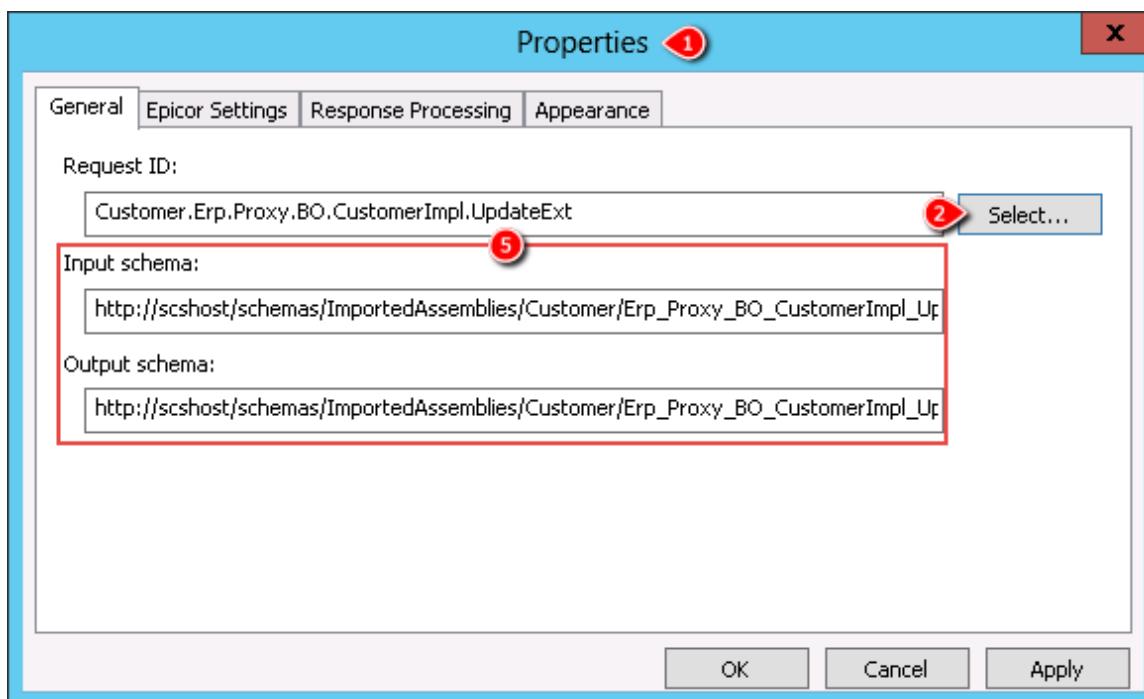
.NET Call Example

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

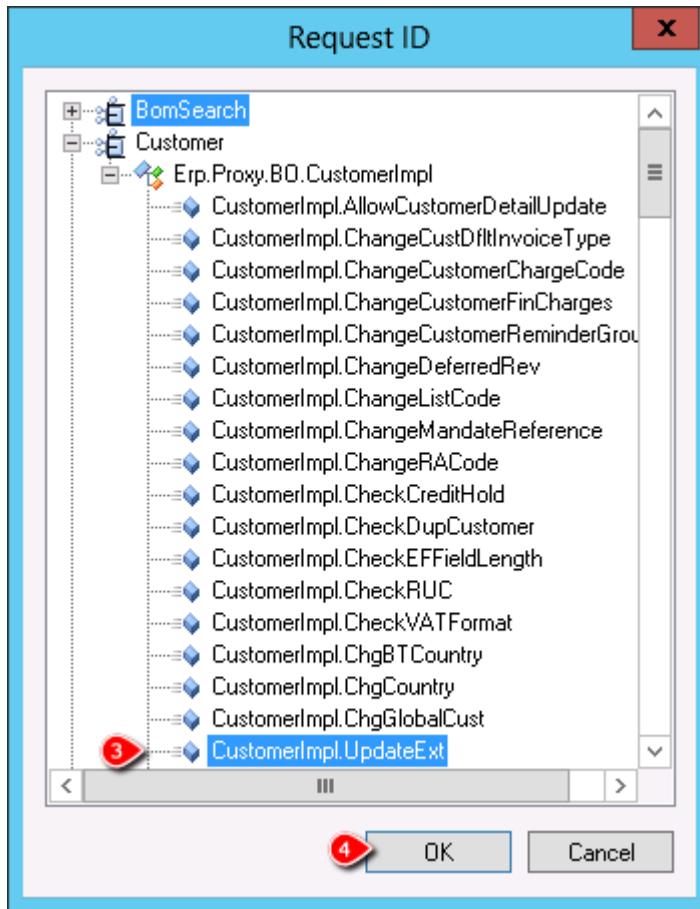
To set up a .NET Call:

- When you add a **.NET Call** to the workflow design area, the **Properties** window displays.



- On the **General** tab, click **Select** to choose the method you want to call in the Request ID field.

3. In the Request ID dialog, expand the corresponding assembly node and select the method.



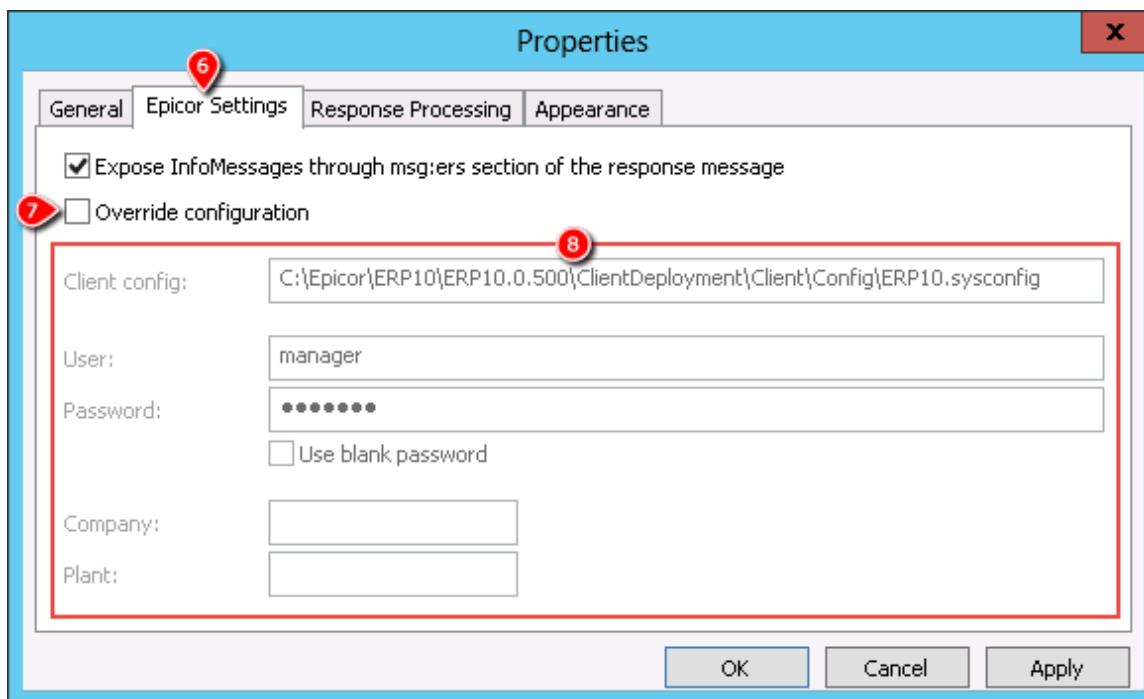
4. Click OK.

The Request ID field displays the method you selected.

You can also enter the value in the Request ID field directly.

5. 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.

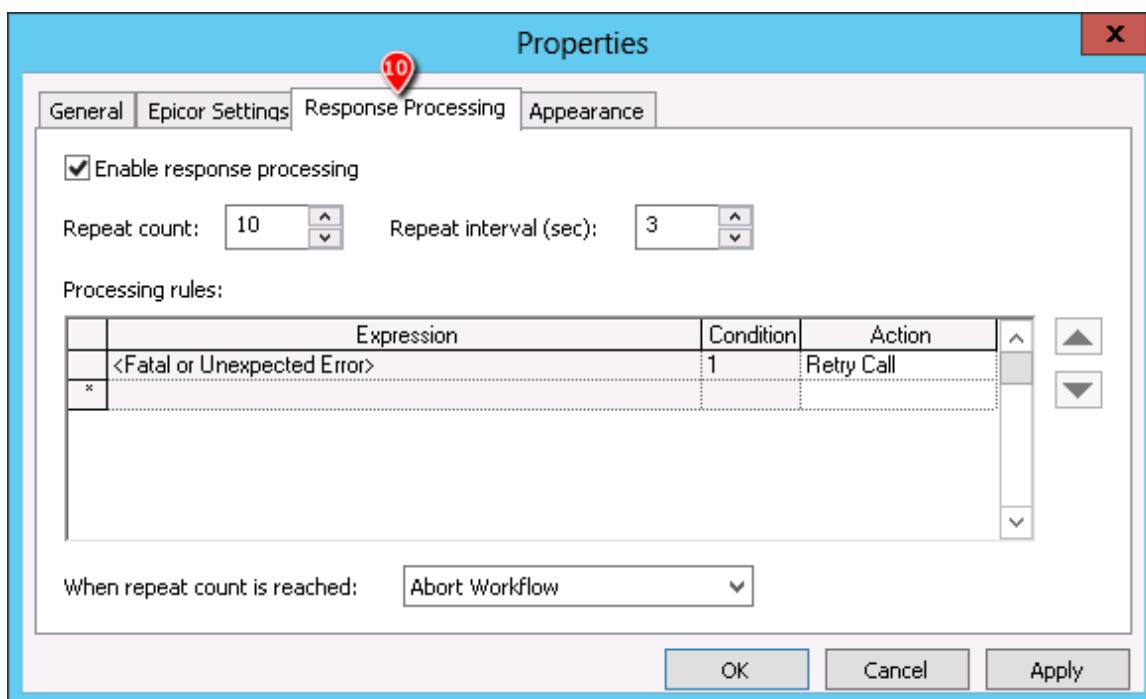
6. If the method used is an Epicor Assembly, on the **Epicor Settings** tab, the method default configuration from the Administration Console is shown as grayed text.



7. If you need to override a logon configuration entered during the .NET Reference import in the Epicor Service Connect Administration Console, select the **Override configuration** check box.
Review the .NET References section in Connectivity Components chapter for more information.
8. Enter your company-specific settings.
9. If the Epicor logon configuration was customized in the Workflow Designer, .NET Call icon changes to the following:



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



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

REST Operation



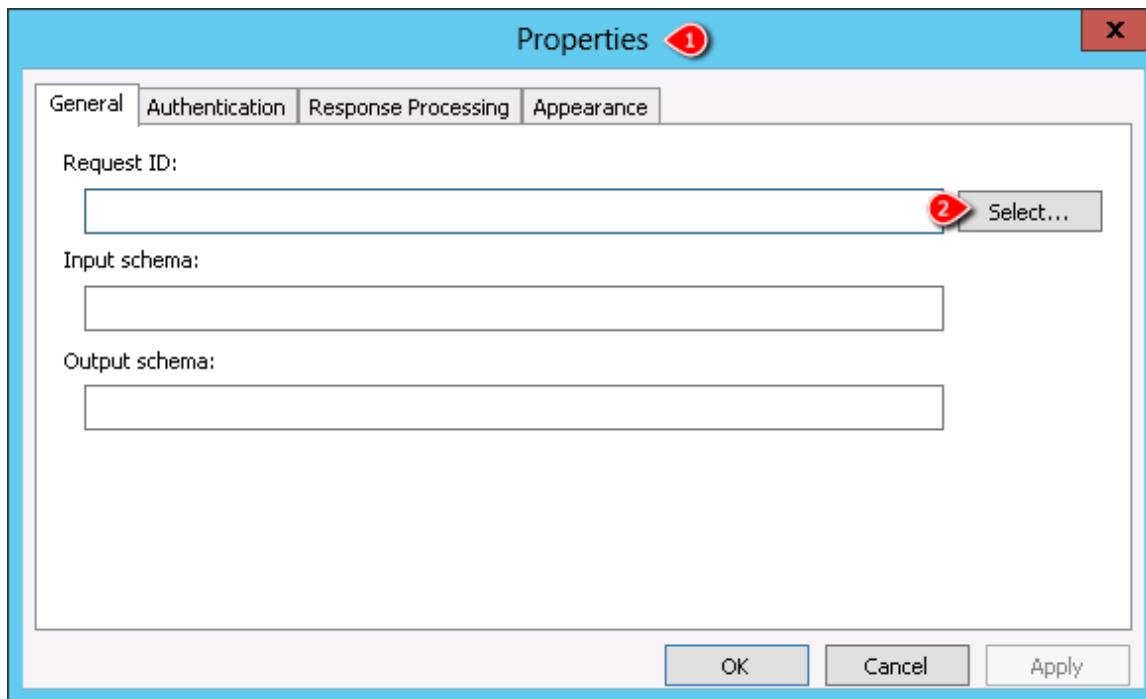
Use this workflow element to call methods exposed by a REST web service reference.

You must import the method as a RESTful Reference in the ESC Administration Console before you can use it in a REST Operation. To use the REST Operation workflow element, it must be licensed.

Set Up a REST Operation

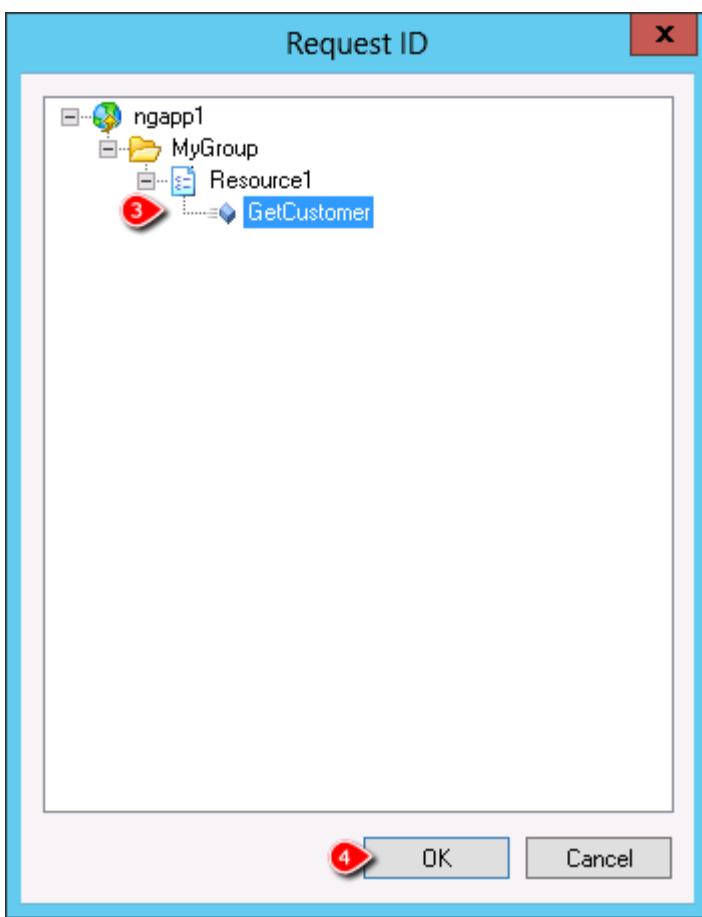
To set up a REST Operation:

- When you add a **REST Operation** to the **Workflow design area**, the **Properties** window displays.



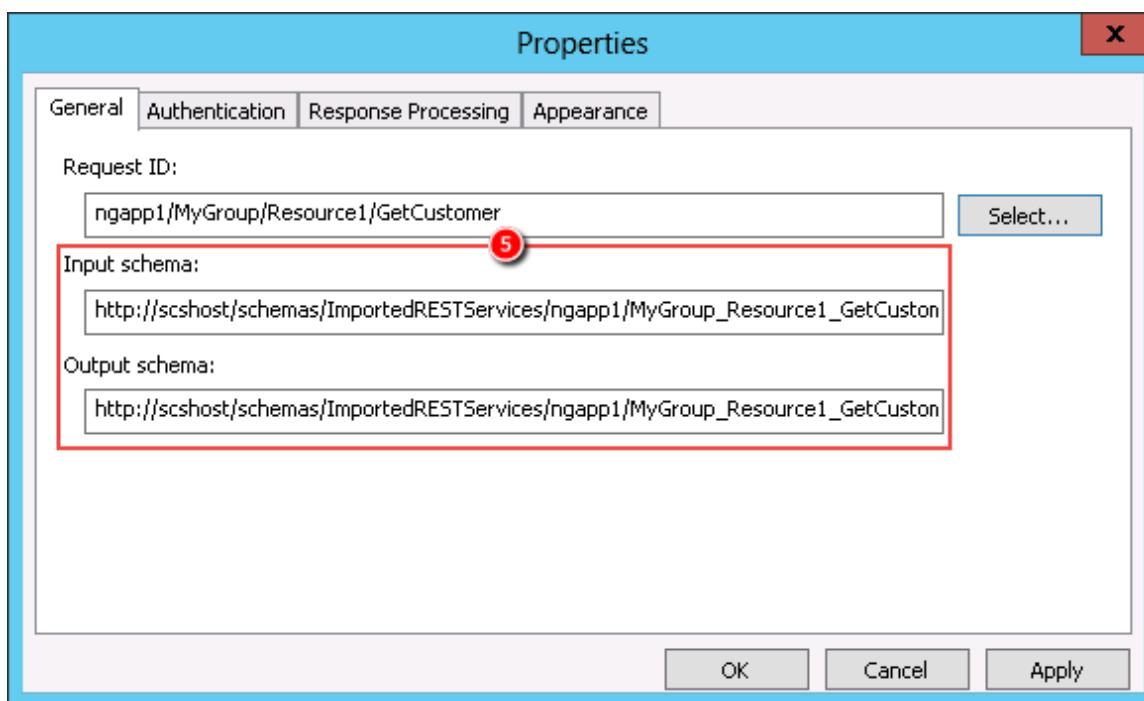
- Near the **Request ID** field, click **Select** to choose the method you want to call.
You can also enter the value directly in the **Request ID** field in the Properties window.

3. In the **Request ID** dialog, select the web service method.

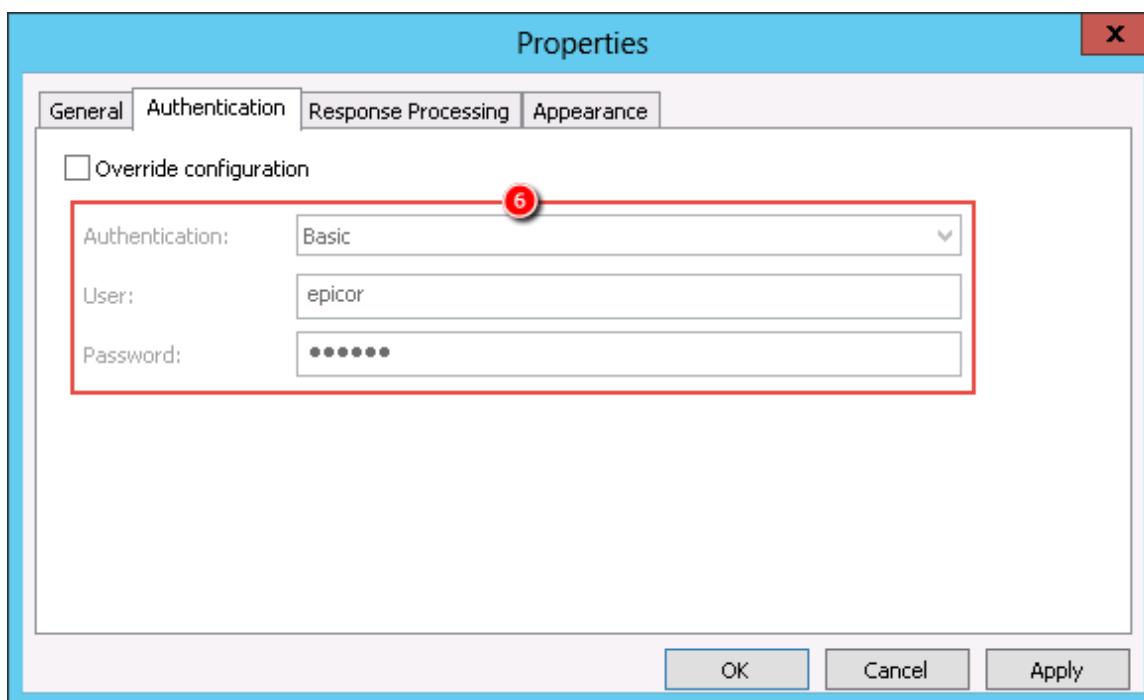


4. Click **OK**

5. The system automatically sets the **Input Schema** to the schema that corresponds to the REST Operation request. Likewise, the **Output Schema** is set to the schema that corresponds to the REST Operation response.

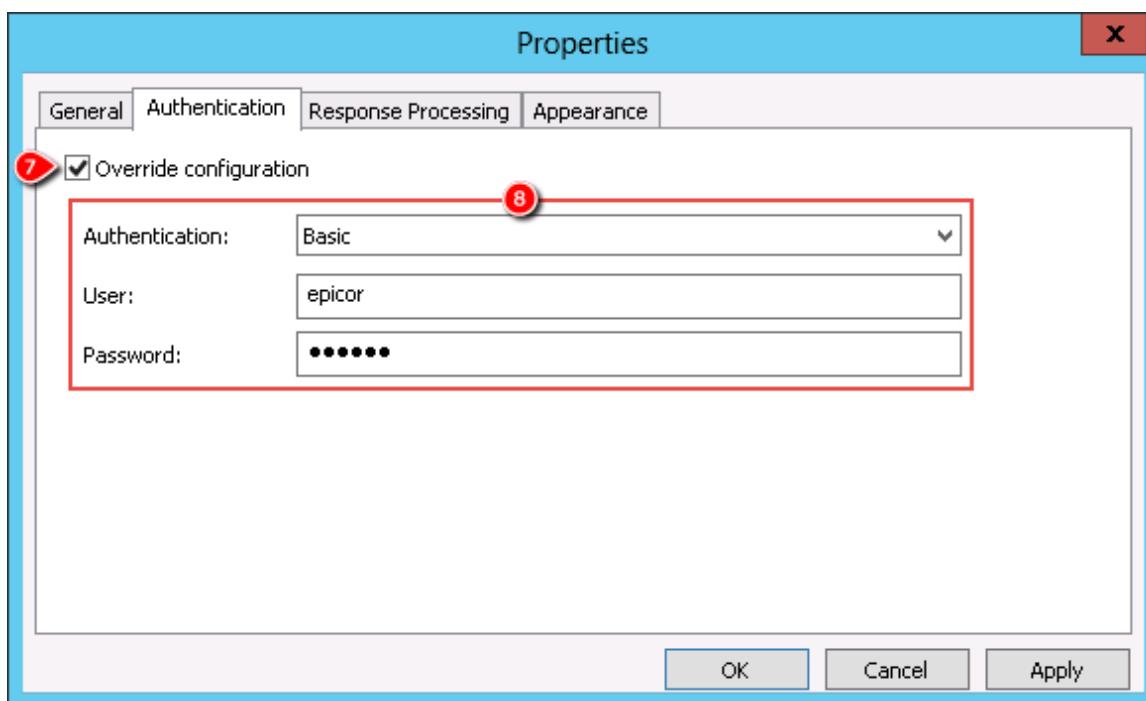


6. On the **Authentication** tab, the default RESTful Reference security settings specified during reference import in the Administration Console are displayed as grayed text.



7. To override the REST Operation logon details entered during the RESTful Reference import in the Epicor Service Connect Administration Console, select the **Override configuration** check box.

Review the RESTful References section in the Connectivity Components chapter for more information.



8. In the **Authentication** field, select one of the following authentication modes:

- **None**
- **Basic**

Enter the **User** and **Password**.

- **Windows**

Enter the **Domain**, **User** and **Password**.

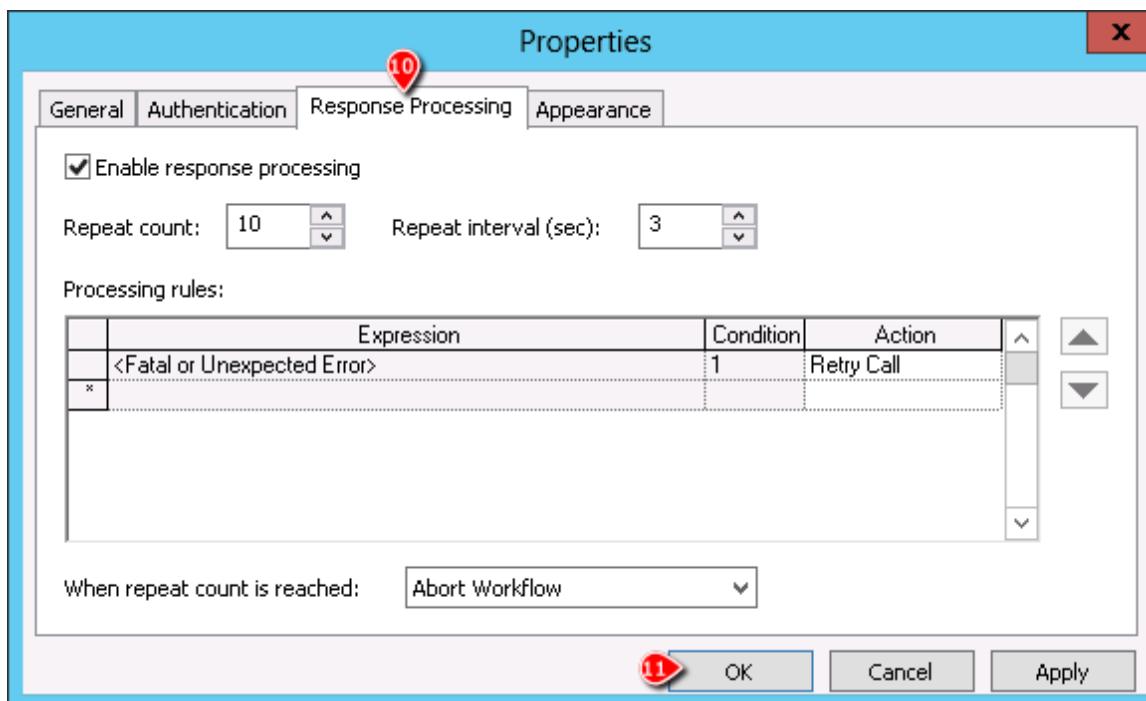
- **OAuth**

Enter the **Token server URL**, **User** and **Password**.

9. If you customize the Authentication configuration in the Workflow Designer, REST Operation icon in the workflow diagram changes.



10. On the **Response Processing** tab, you can set REST Operation element to re-call a REST web service method repeatedly in case of an error.



During calls to REST Operation from Service Connect workflows, Service Connect sometimes receives errors; if Service Connect calls the same REST Operation several seconds later, the call is successful. Reason of the errors depends on many factors outside Service Connect. On the **Response Processing** tab, you can set REST Operation element to re-call REST Operation repeatedly if the latter returns error.

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

11. Click **OK**.

12. If you enabled REST Operation response processing, its icon in the workflow diagram changes to the following:



DB Operation



Use this workflow element 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. Only the first data retrieval statement within the SQL commands batch marked as output returns data from a single DB Operation.

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.

You can change data parameters dynamically. To do it, add a link to your SQL statement. Links of the following three types are available:

- **Unsafe Link**

A links of this type is actually a placeholder. The link value is evaluated at runtime and inserted into SQL statement before a SQL statement execution against database. This type of link can be used anywhere inside SQL statement.

- **Safe Link**

You can use this type of link only to define SQL statement parameters values. You must also define the value type: String or Numeric.

- **SQL Name Link**

You can use this type of link only for column or table names.

Use the Safe Link to add typed link. Use the SQL Name Link to define table name or column name dynamically. Use these two link types to prevent from SQL injection.

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.

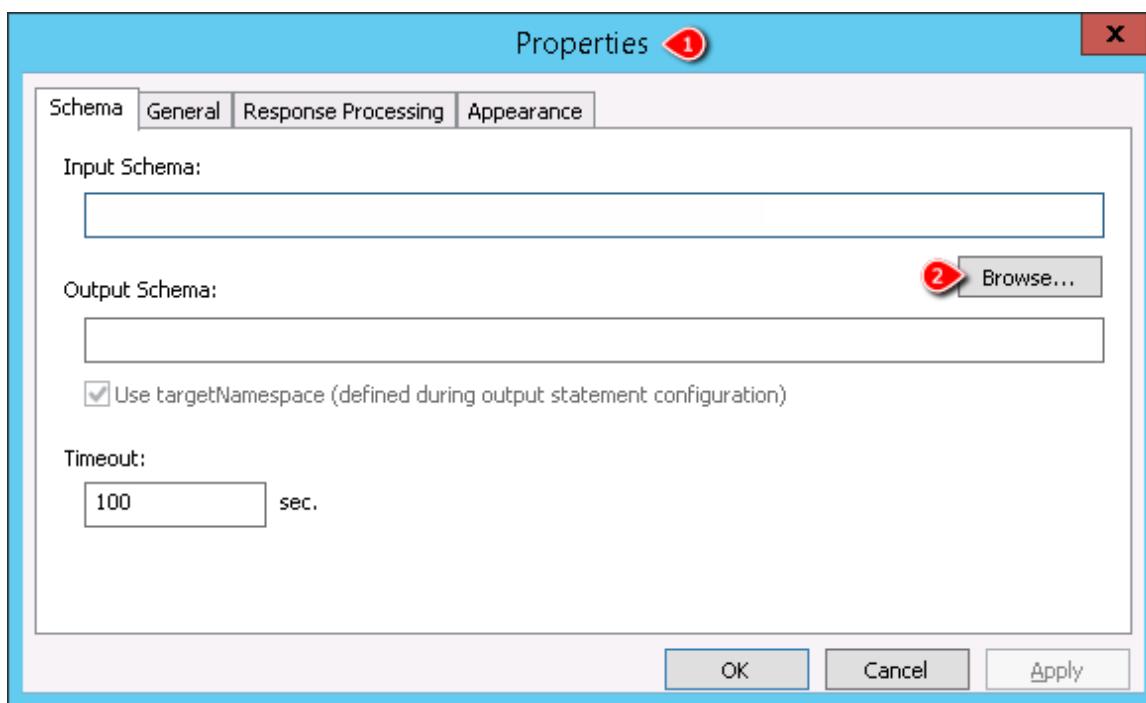
DB Operation Example

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 DBOperation

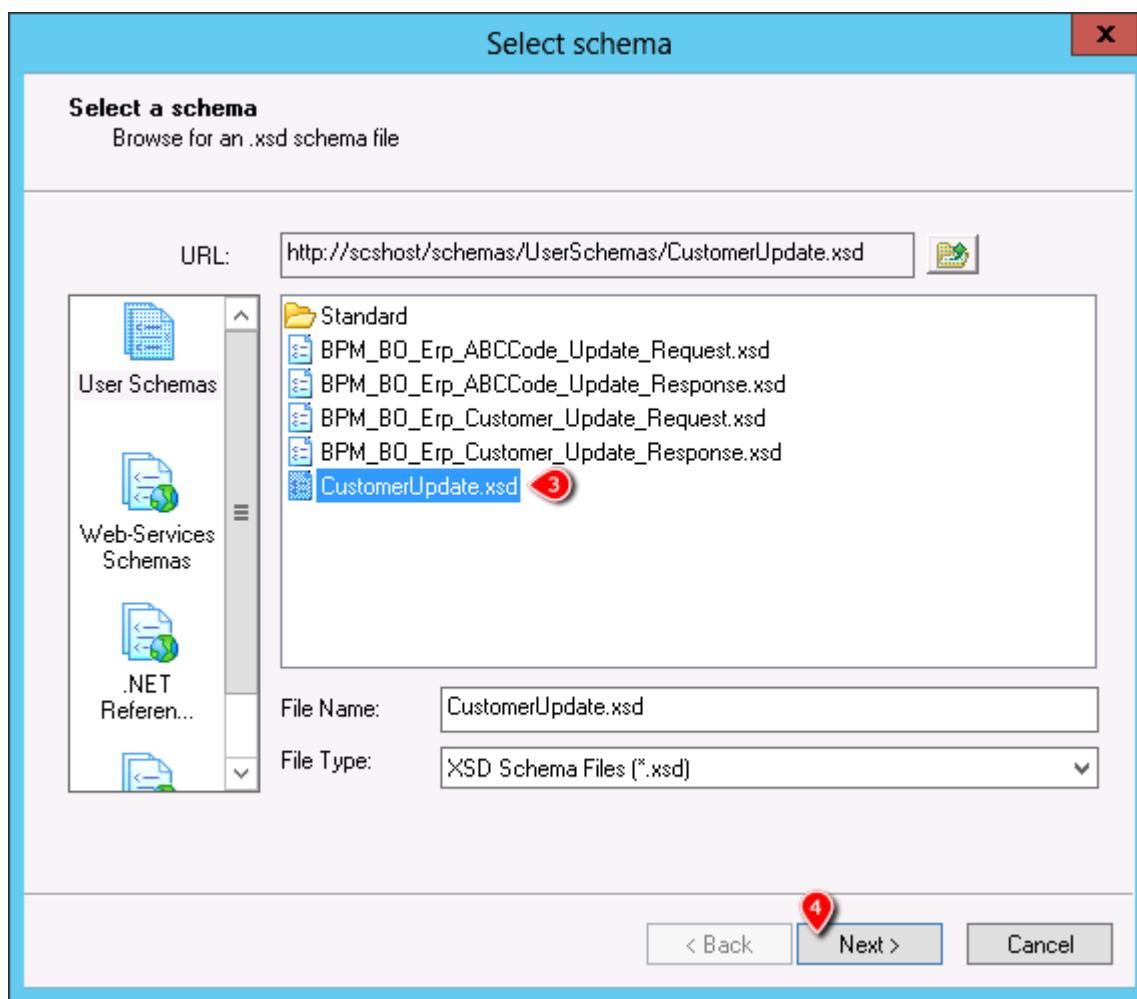
To set up a DBOperation:

- When you add a **DBOperation** in the Workflow design area, the **Properties** window displays.



- On the **Schema** tab, click the **Browse** button to find and select the **Input Schema** path to use, update, or select database data.

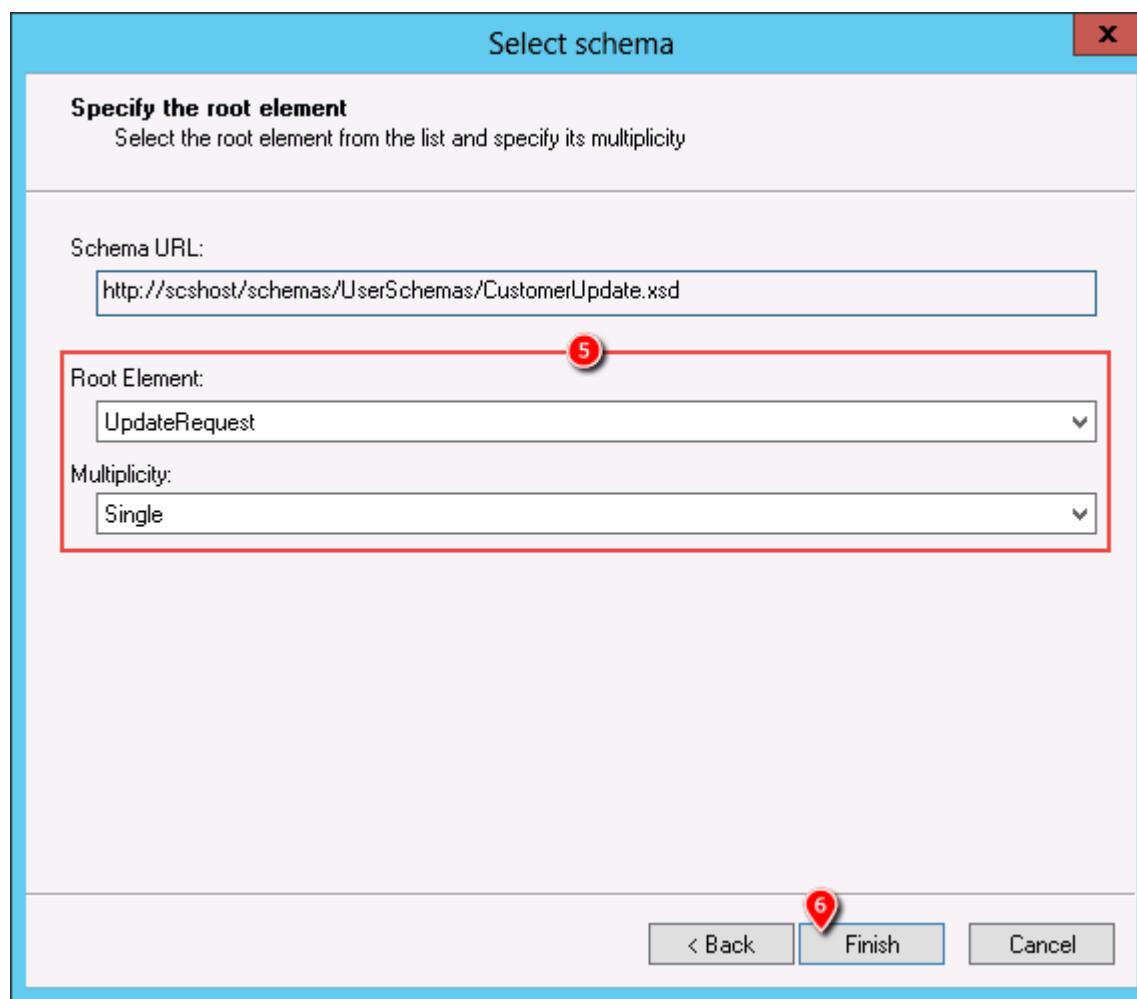
3. In the **Select Schema** window, navigate to and select an XSD schema to define the format of the document entering the DB Operation.



4. Click **Next**.

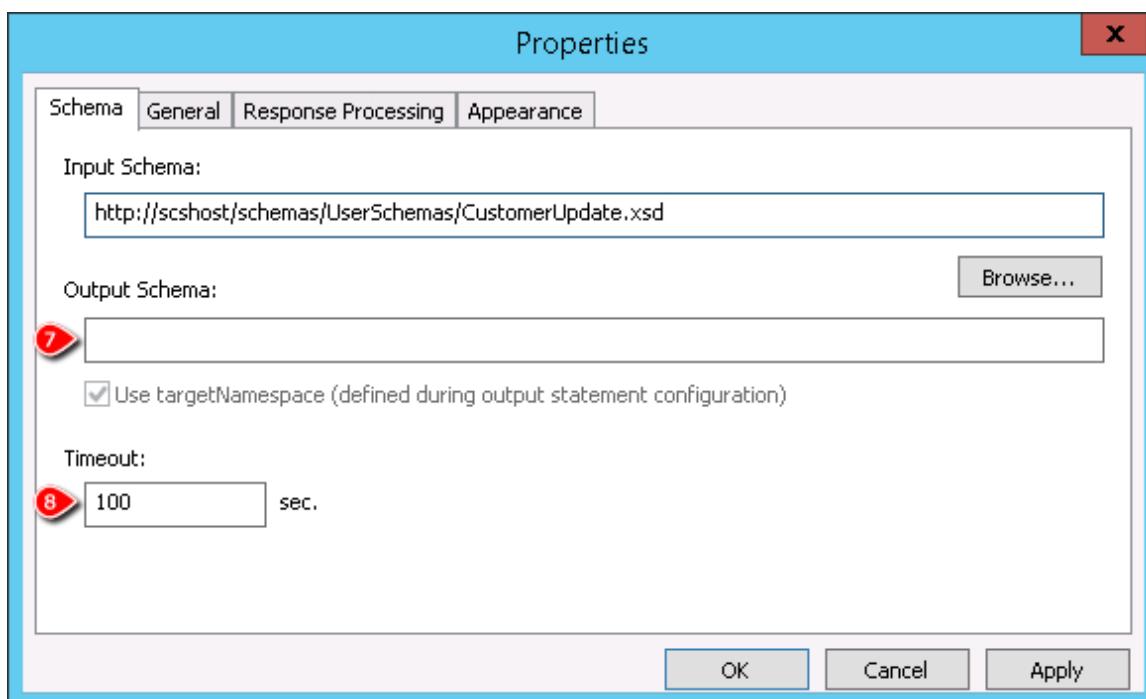
5. Select a Root Element and the Multiplicity for the selected schema.

Select Single to use just one element under the dta element, or select Multiple to utilize a collection under the dta element.



6. Click **Finish**.

7. The **Output Schema** field is read-only and is defined by the output statement.

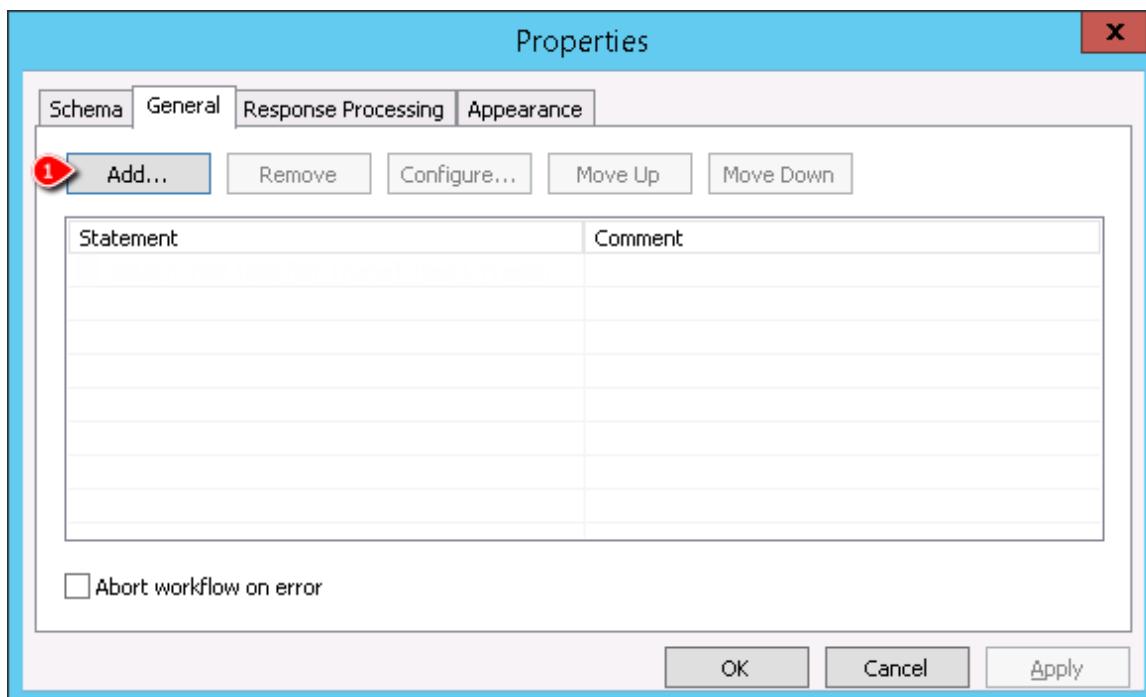


8. To set the time period the DB Operation waits for request to complete, use the **Timeout** field.

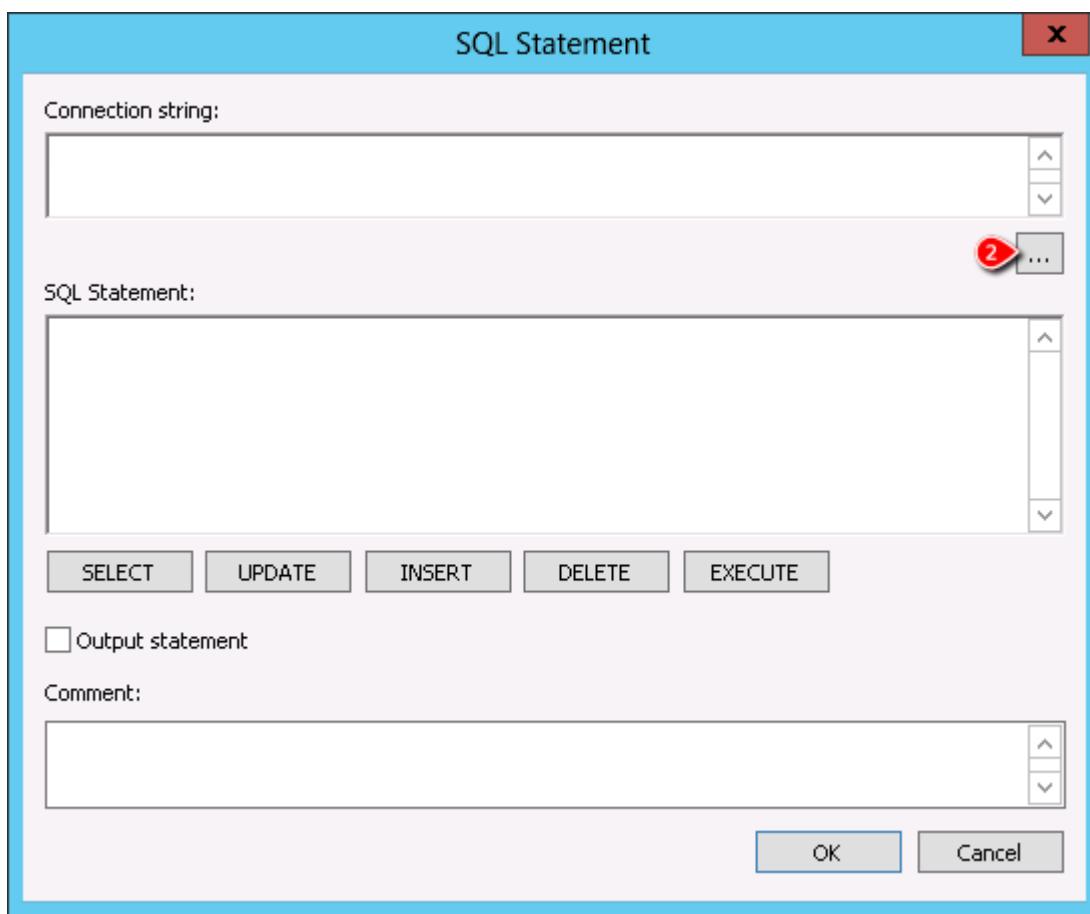
The default value is 100 seconds. If the field is empty, the default value is used. Setting the field value to -1 indicates that the request does not time out.

Build a SQL Statement

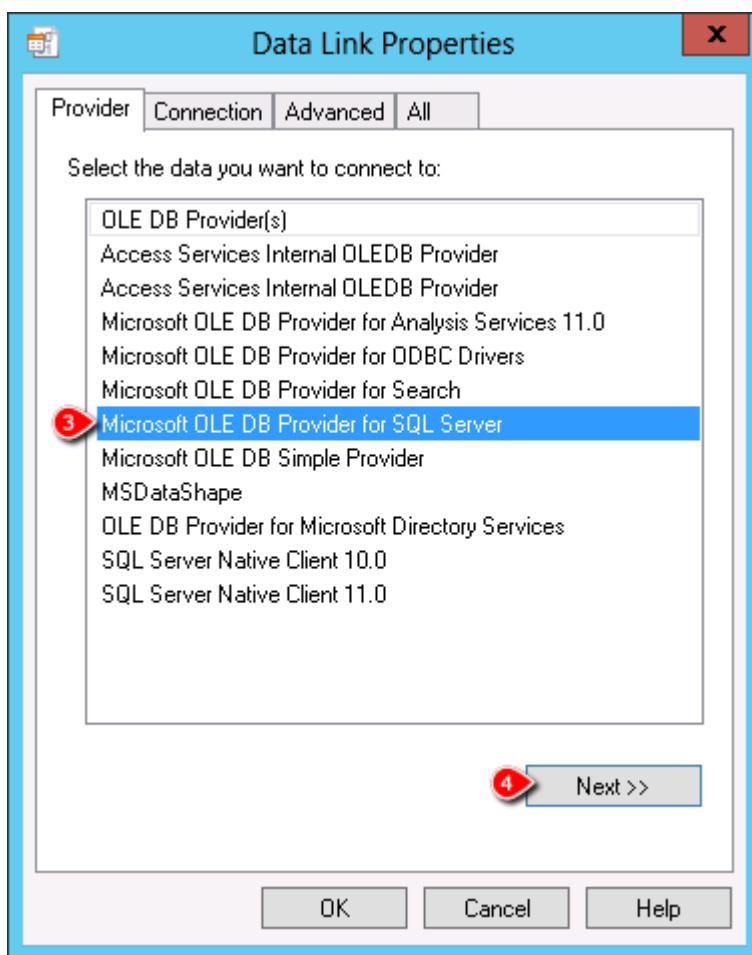
1. On the **General** tab, click the **Add** button to add a new statement.



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



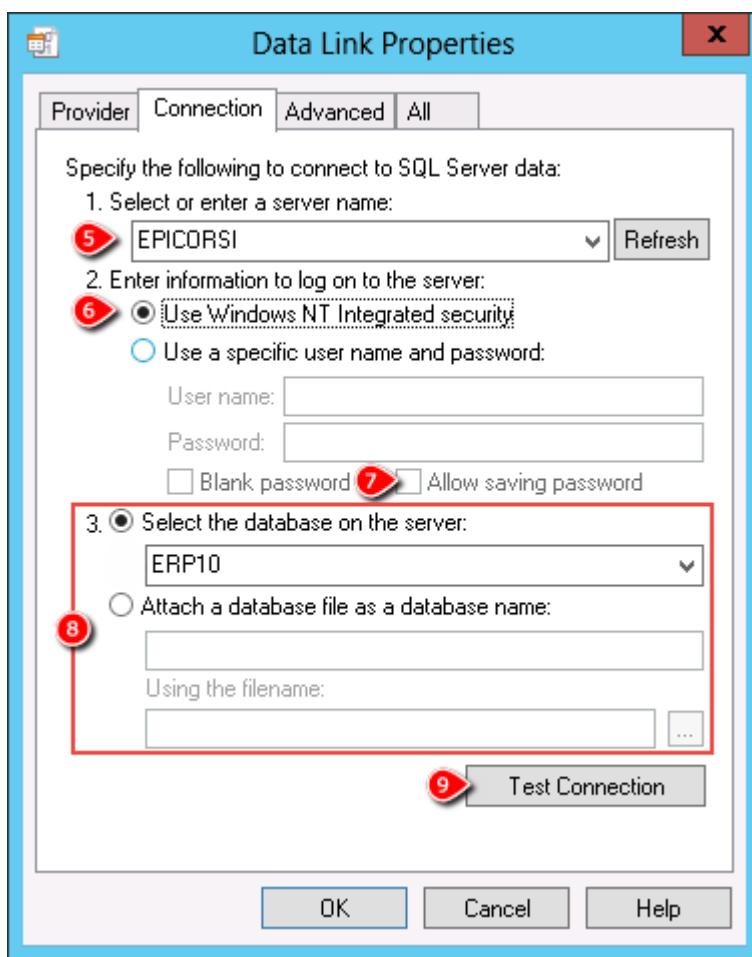
3. On the **Provider** tab, select an OLE DB Provider.



In this example, Microsoft OLE DB Provider for SQL Servers is selected.

4. Click **Next**.

5. On the **Connection** tab, select or enter SQL server name.



6. Select the authentication mode. The available option are:

- Windows NT Integrated security
- Specific **Username** and **Password**

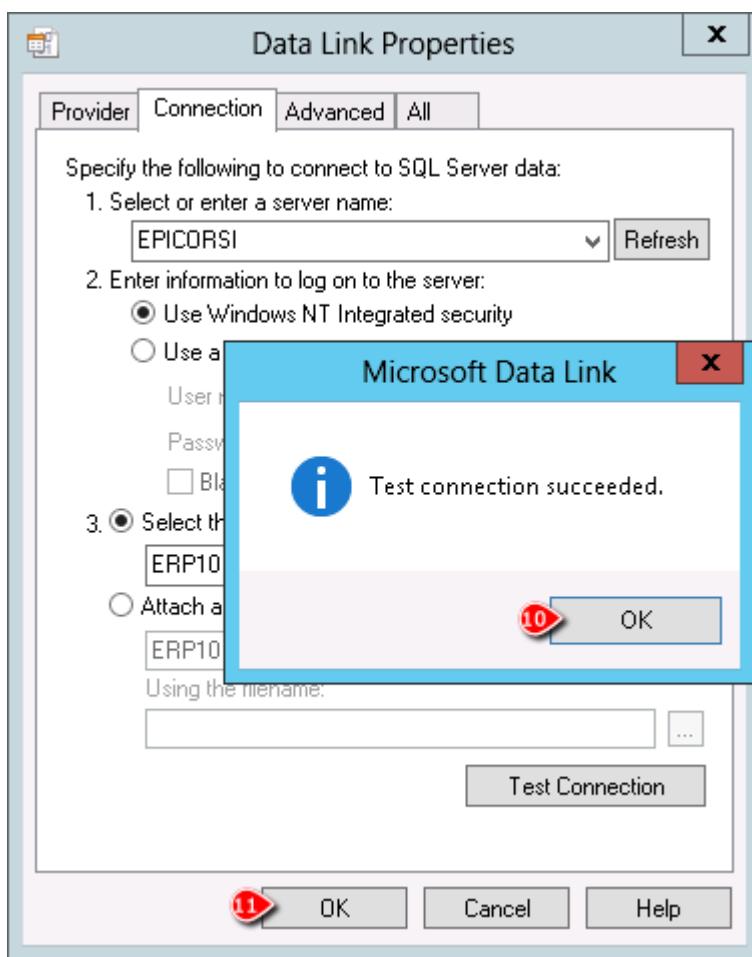
If you select this option, in the **Username** and **Password** fields, enter your credentials.

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

8. Select the SQL database you are going to use.

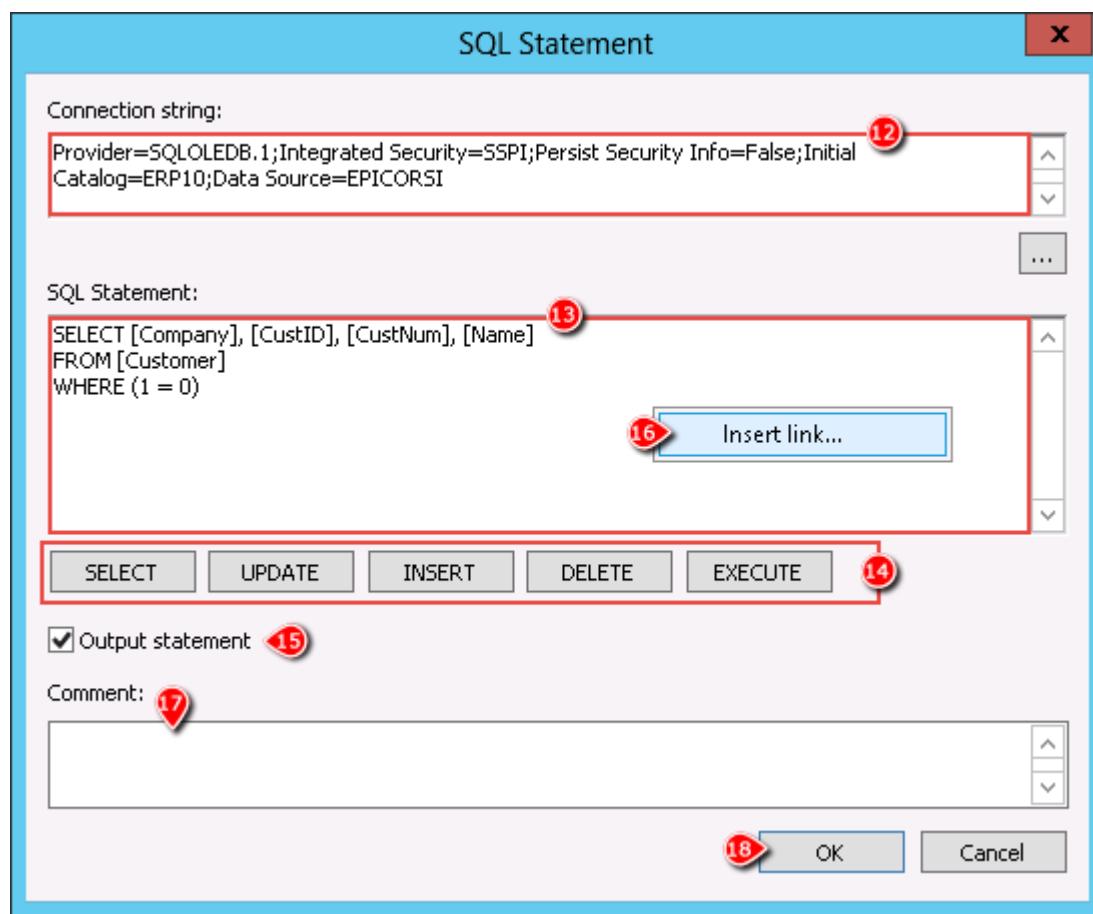
9. To test the established connection, click **Test Connection**.

10. To the connection confirmation message, click **OK**.



11. In the **Data Link Properties** window, click **OK**.
12. On the **SQL Statement** screen, 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.



13. Enter the **SQL Statement.**

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.

14. You can use the **SELECT, UPDATE, INSERT, DELETE, and **EXECUTE** buttons to build SQL Statement.**

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

15. 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.**

16. Optionally, if you want to change data parameters dynamically, you can add a link. To add a link, right-click anywhere on an empty space, select **Insert link and build an XPath rule in the Rule Assistant.**

You can right-click inside of the Connection string as well.

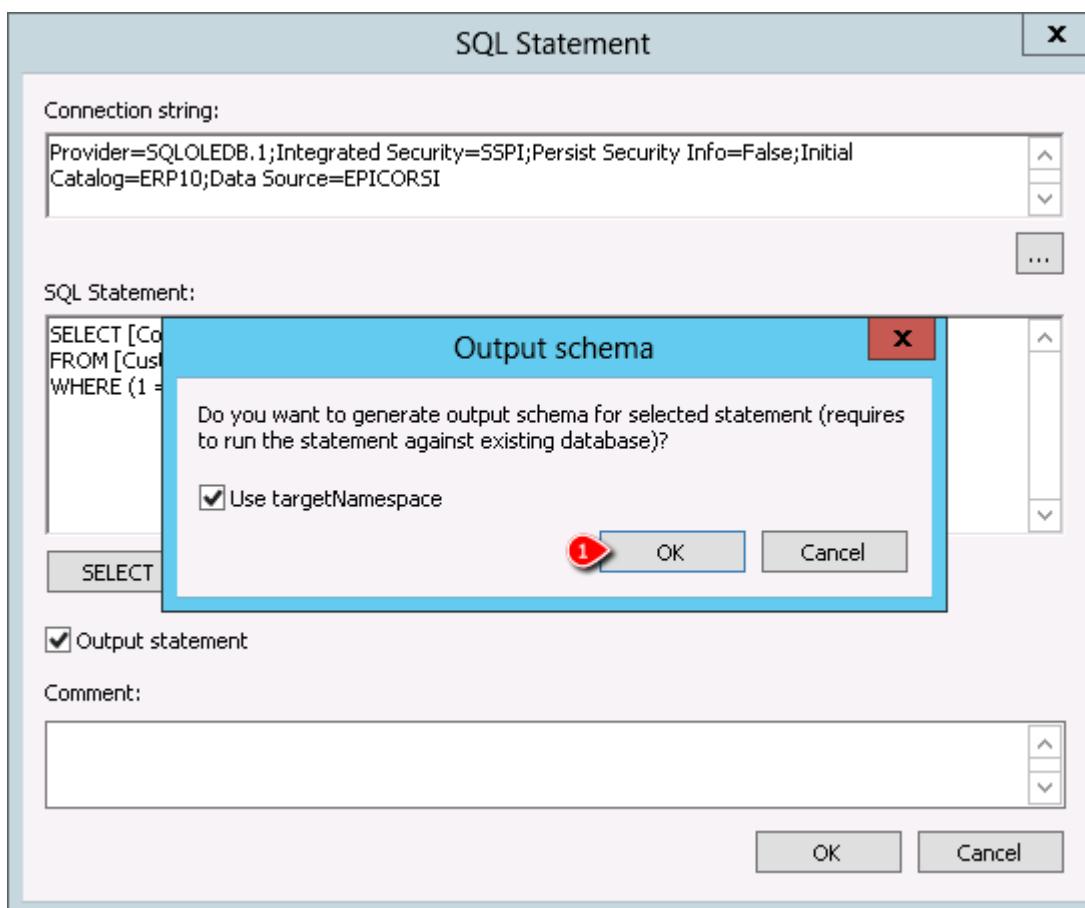
Each link is evaluated to string value applying XPath from the link. Each single quote in the string value is replaced by two single quotes to make processing quotes from the link value available and avoid unclosed quotes in the final SQL expression.

17. You can enter a comment in the **Comment field.**

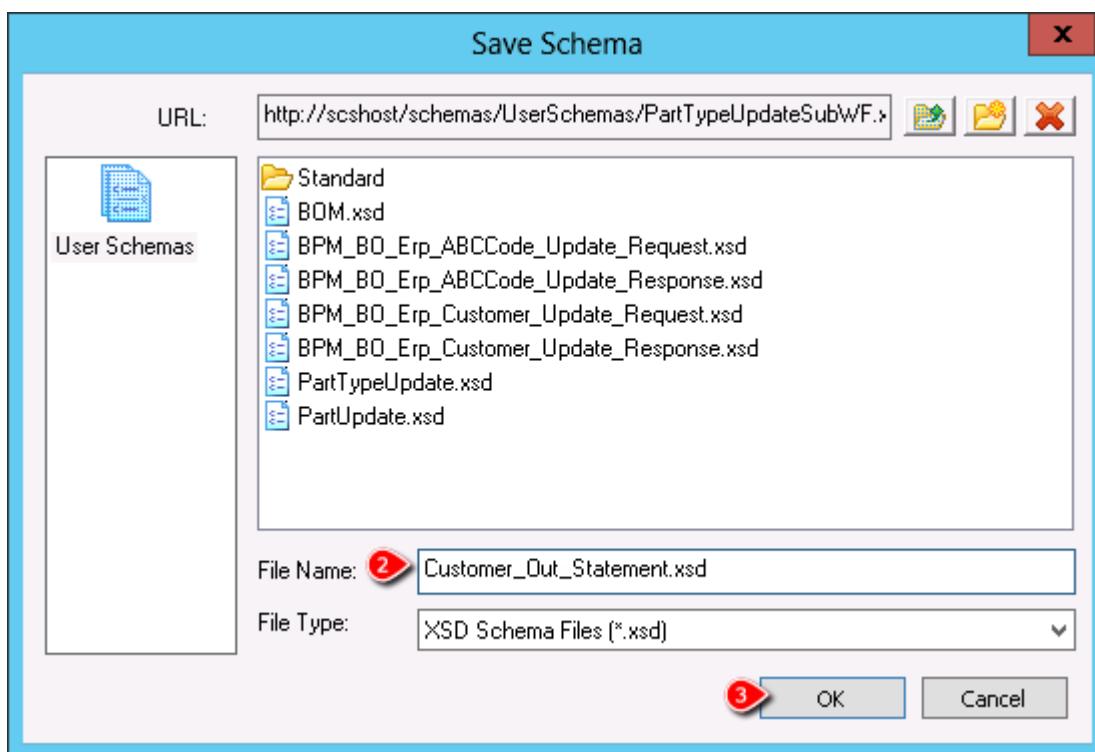
18. Click **OK.**

Finalize DBOperation Setup

1. If the **Output statement** check box was selected, to the **Output schema** message click **OK** to confirm output statement schema generation.



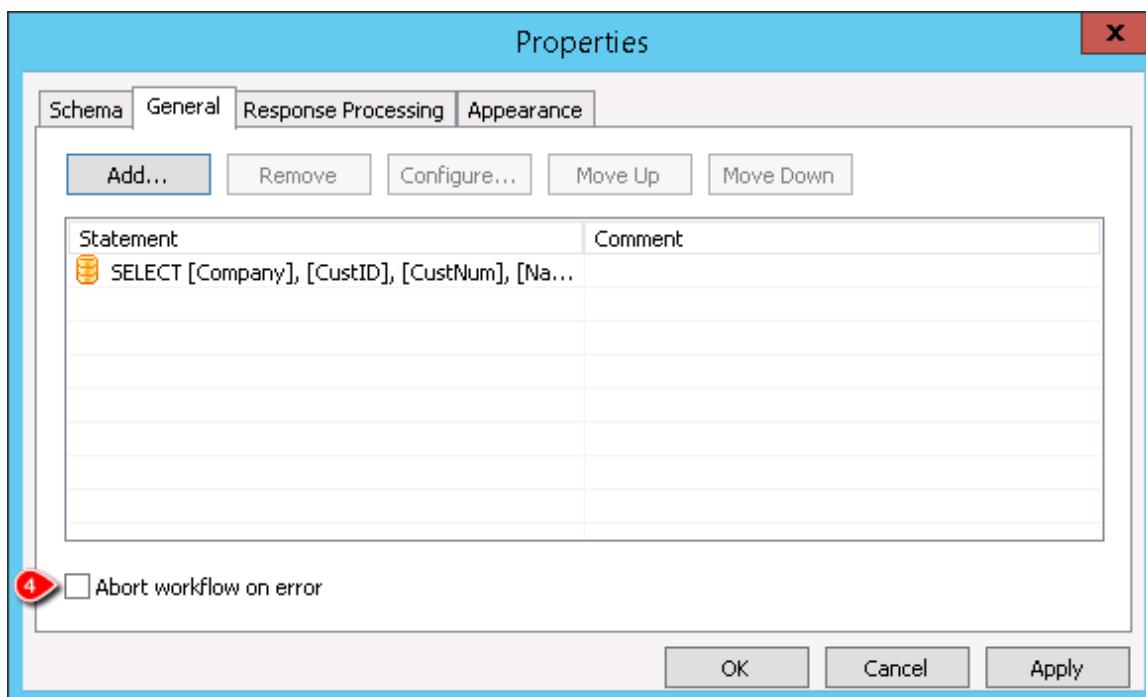
2. In the **Save Schema** window, enter a **File Name** for the new schema.



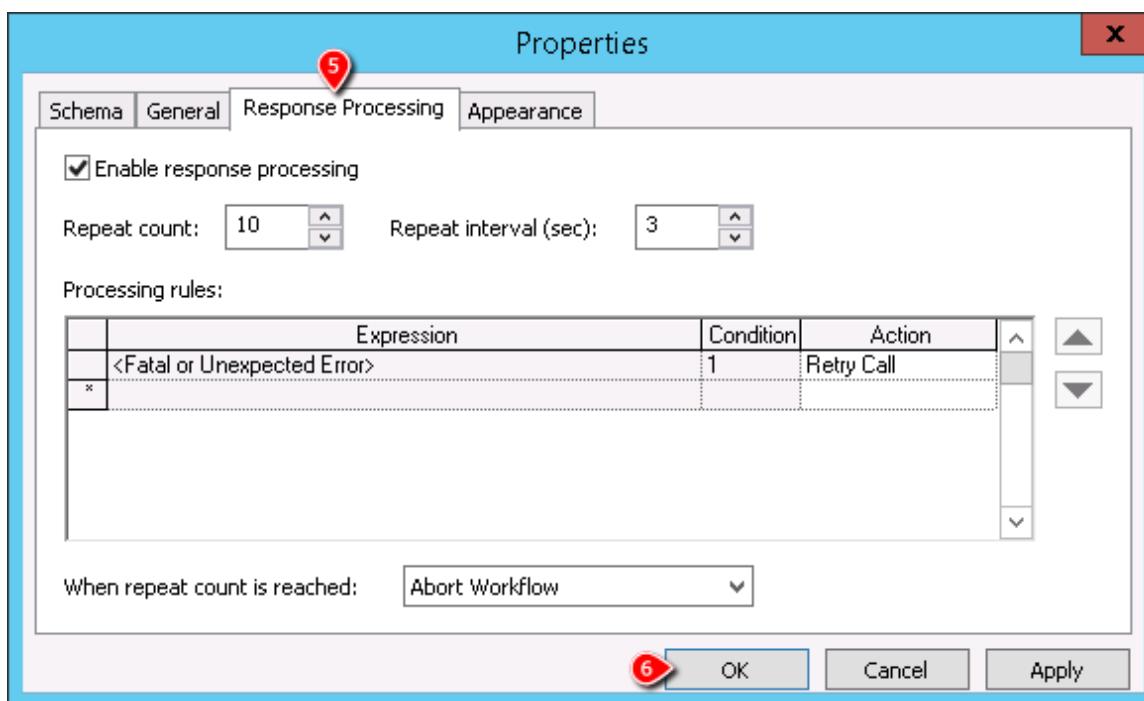
3. Click **OK**.

4. On the **General** tab, select the **Abort workflow on error** check box to stop the workflow execution in case the DBOperation throws an error.

This check box is cleared by default. For workflows created in ESC versions older than 10.1, this check box is selected.



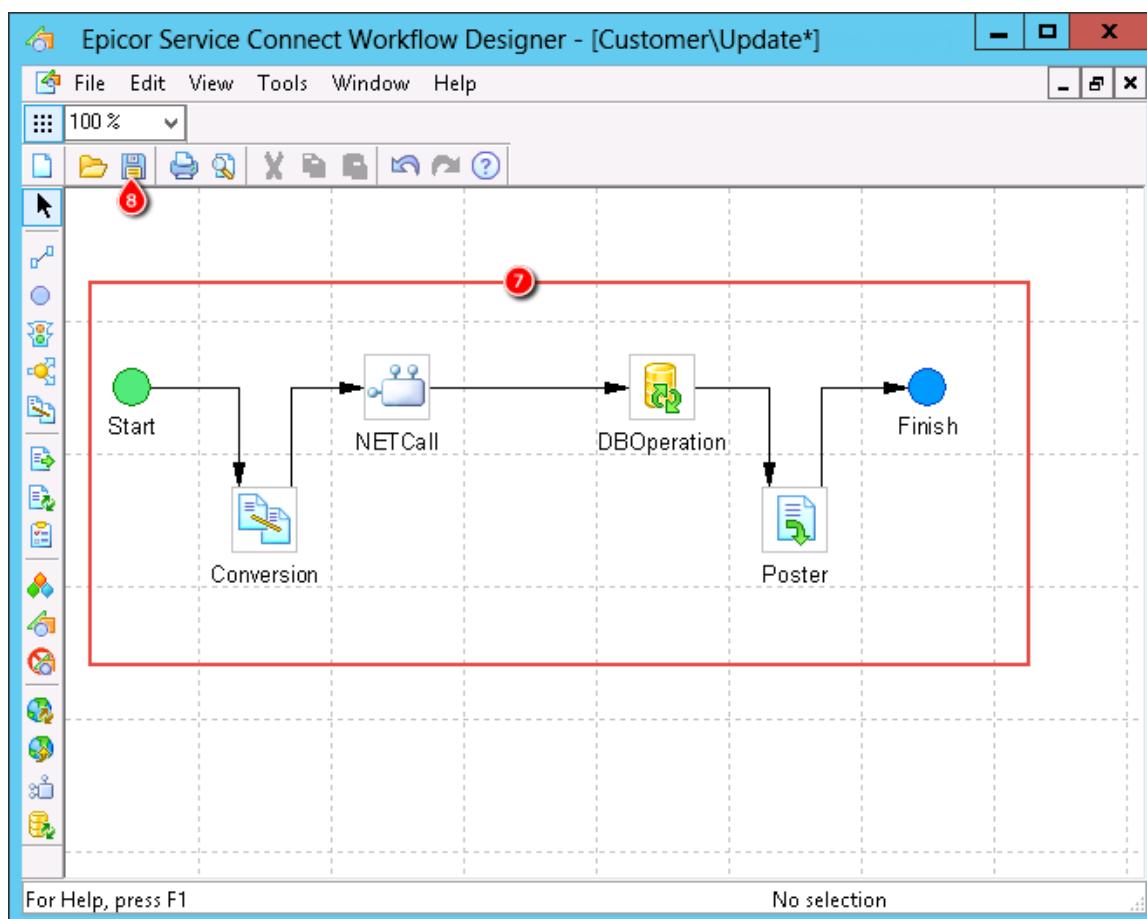
5. On the **Response Processing** tab, you can set the DBOperation to re-call the database in case the SQL Statements that the DBOperation run returned errors.



Review the Web Method section earlier in this chapter for more information on the Response Processing functionality.

6. In the **Properties** window, click **OK**.

7. Build the workflow with the **DBOperation** element.



8. If you validate the workflow or save it, and ESC finds an empty statement inside the DBOperation, the corresponding warning message is displayed in the validation result dialog.

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 workflow element 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 the way it calls a sub-workflow. The following options are available:

- Looping is interrupted and the trace status is set to **Complete**.
- 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 does 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.

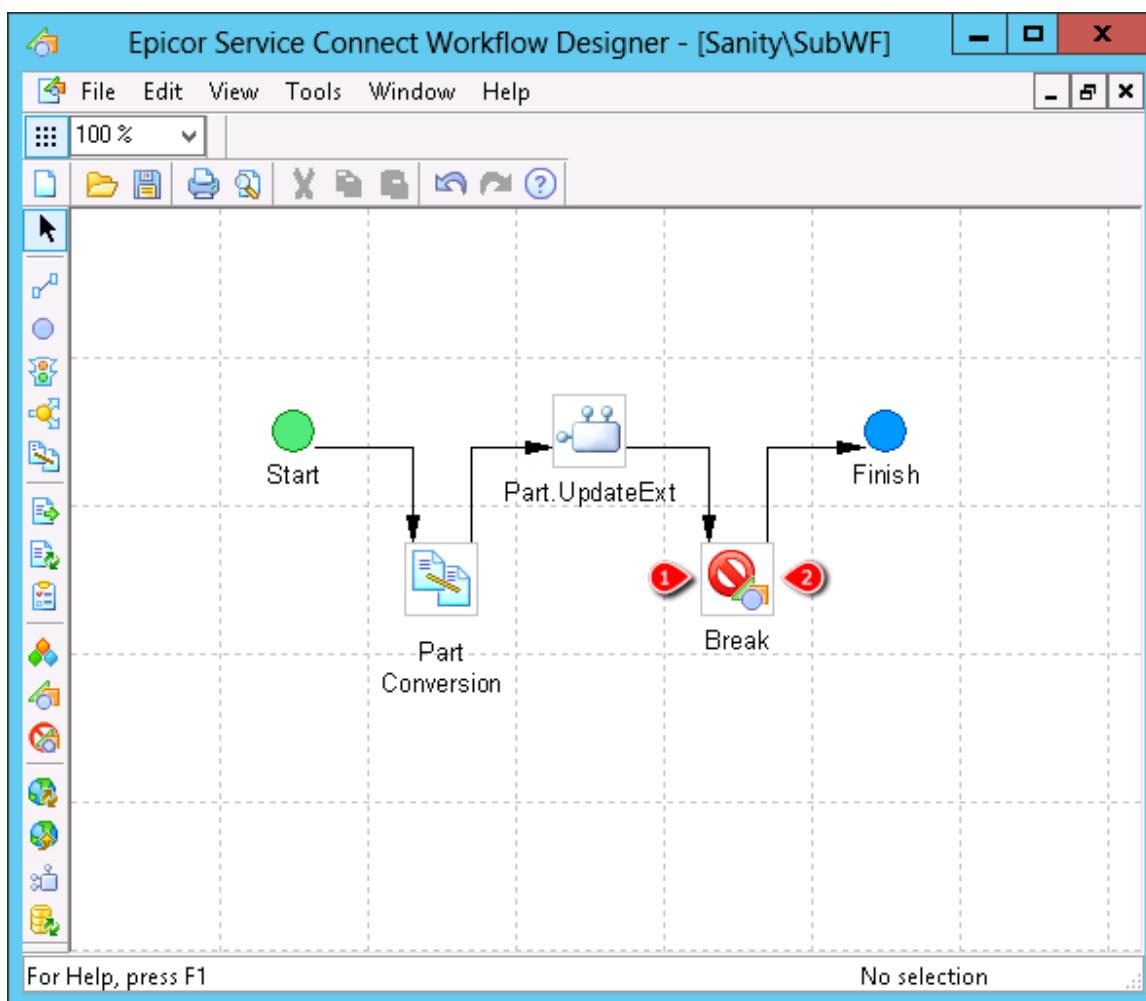
Break Example

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.

Set Up a Break

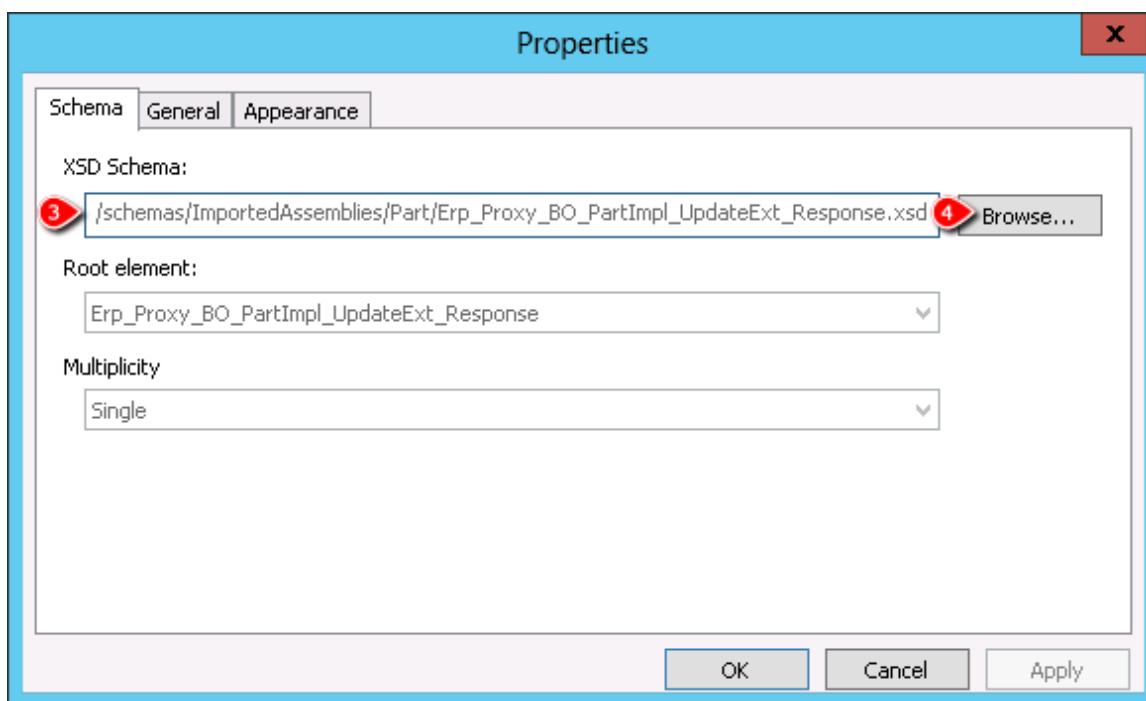
To set up a Break:

1. Depending on your business logic, place the **Break** element in your sub-workflow.



2. Double-click the **Break**.

3. The **Properties** window displays. If the previous workflow element already has the outbound schema set, this schema is automatically displayed on the **Schema** sheet in gray text as the Break **XSD Schema**.



4. Click the **Browse** button to find and select a different input schema for the element.

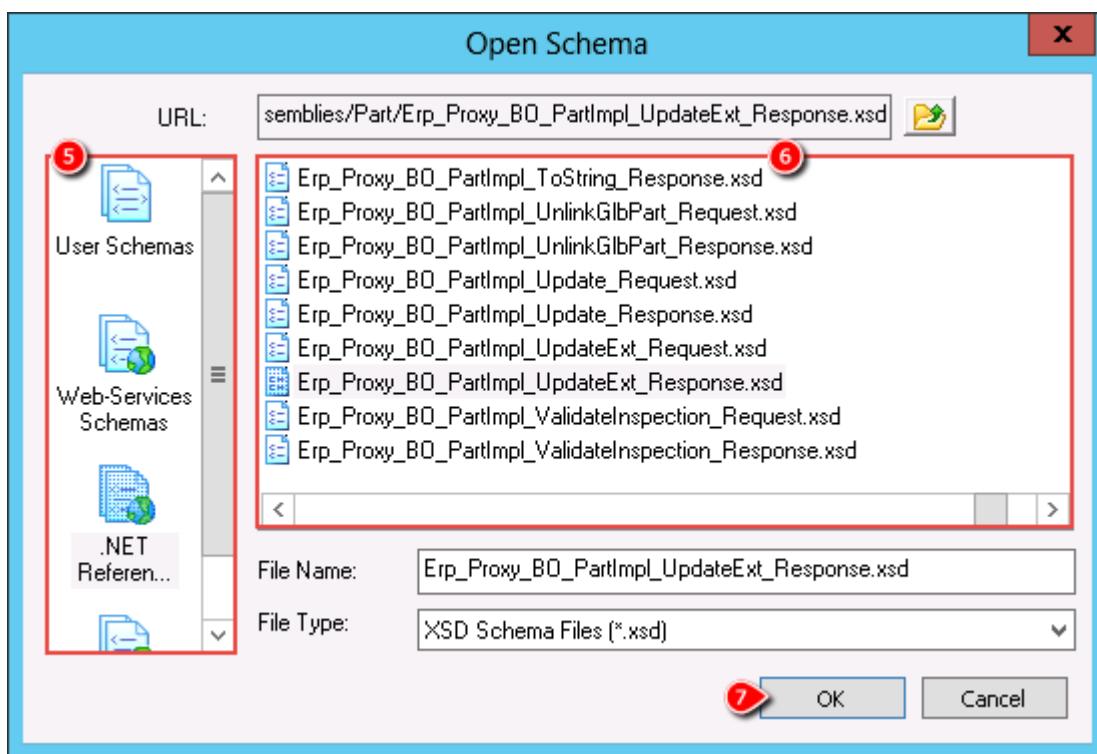
The Open Schema window displays.

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

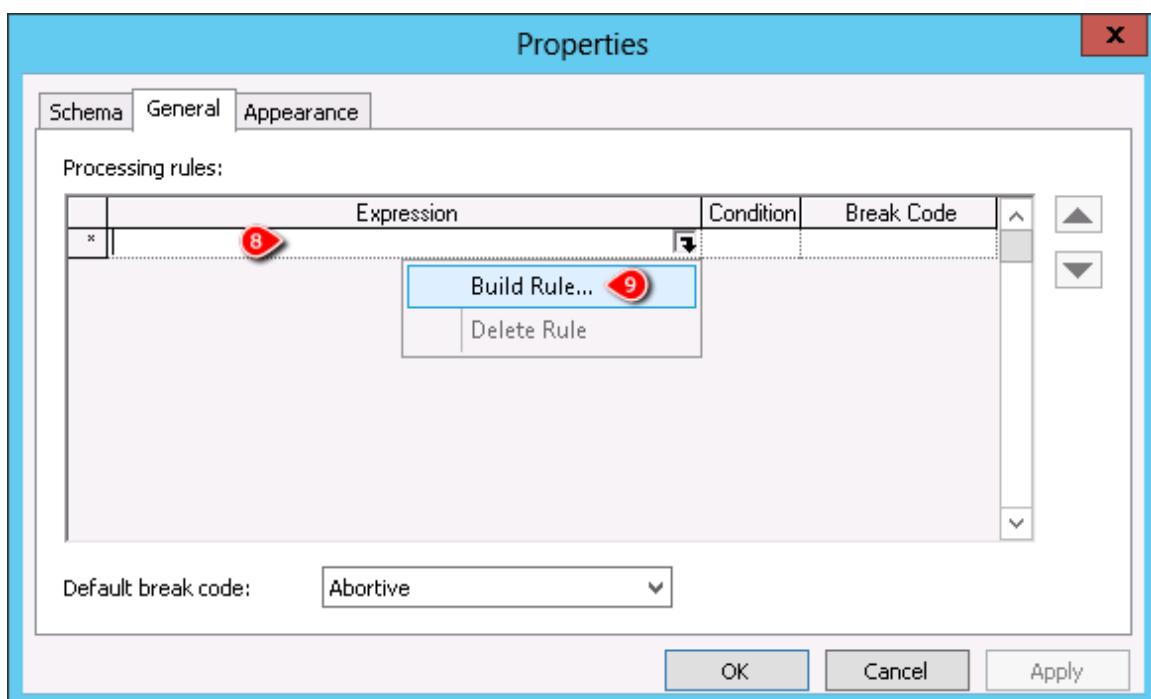
- Web-Services Schemas
- .NET Reference Schemas
- REST-Services Schemas
- User Schemas

These are the schemas generated in the Schemas node of Service Connect Administration Console, or in the Generate Schema from Sample Data tool in the Workflow Designer, or the schemas generated in

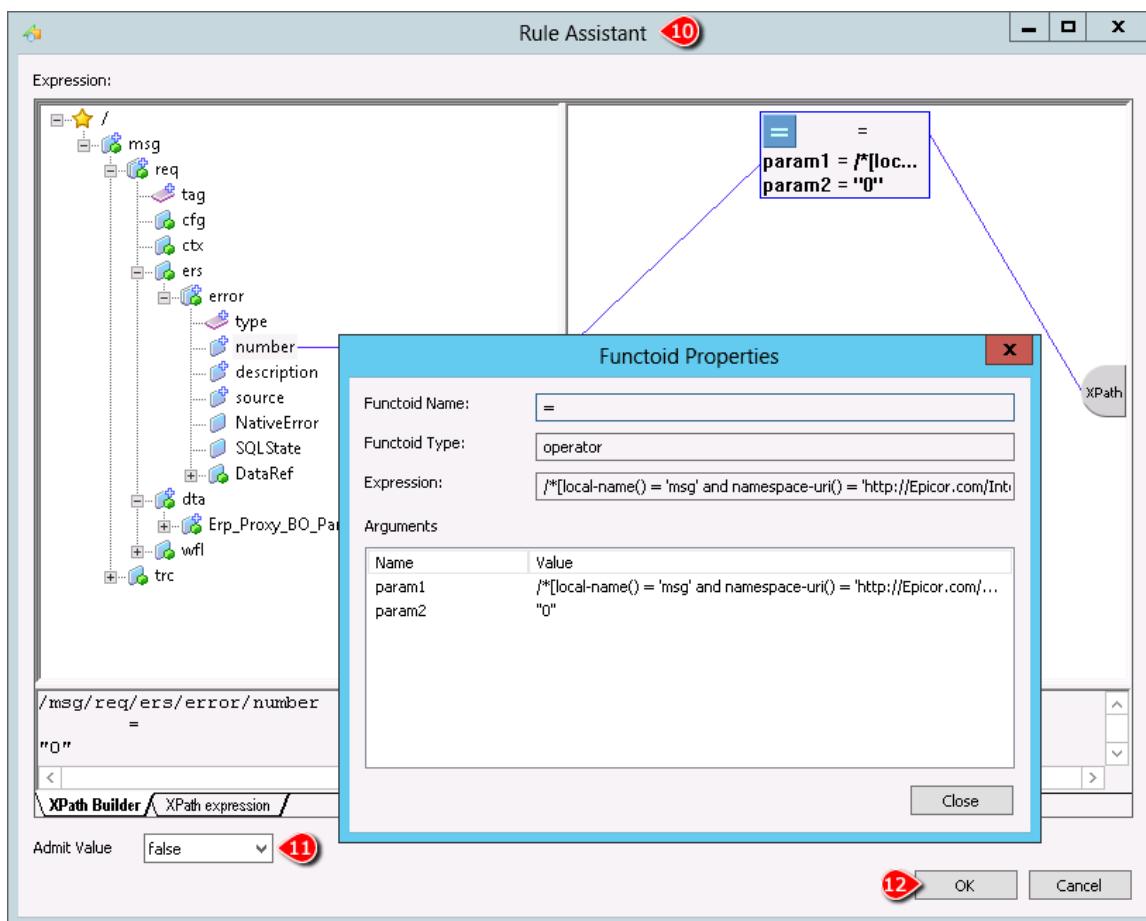
the Workflow Designer when you click the Create sub-workflow schema button on the Cycling tab of Sub-workflow properties dialog.



6. In the right pane, select a schema.
7. Click OK.
8. On the **General** sheet, in the **Processing rules** grid, click the **Expression** field.



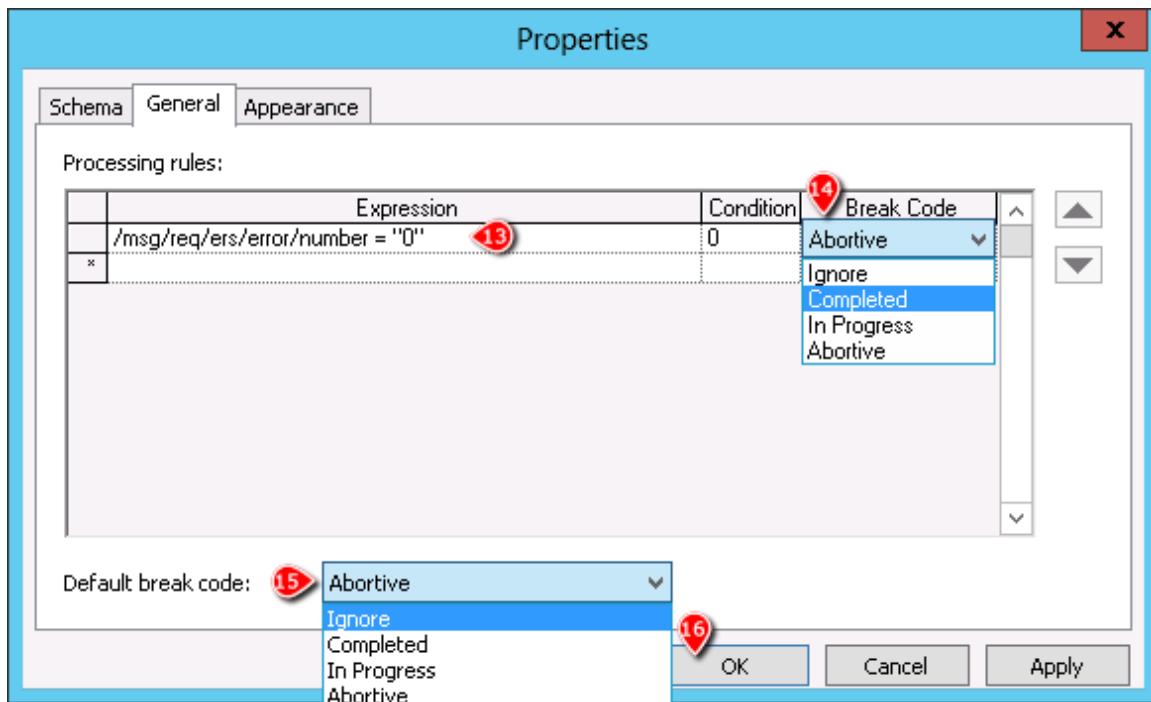
9. Click the **down arrow** button that appears in the field, and select **Build Rule**.
10. Use the **Rule Assistant** to compose an expression, which will be verified against incoming document.



In this example, the Break element looks for any error in the update response from .NET call.

11. Once the rule is created, at the bottom of the Rule Assistant window, in the **Admit Value** field, select True or False.
Select True, if you want to set the workflow behavior for the case when the message meets the condition you built.
12. In the Rule Assistant window, click **OK**.

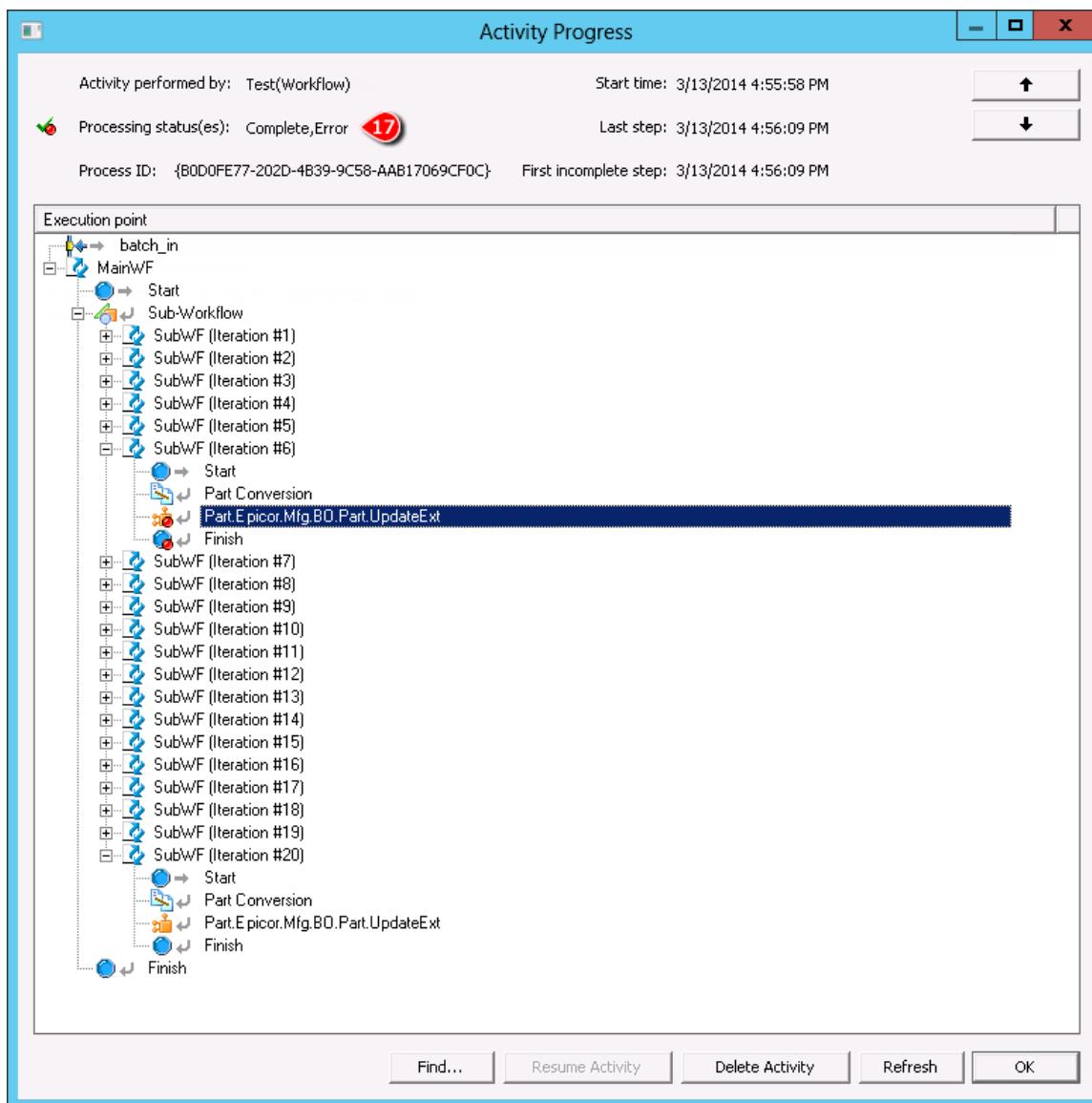
13. Notice your expression now displays in the **Processing rules** grid.



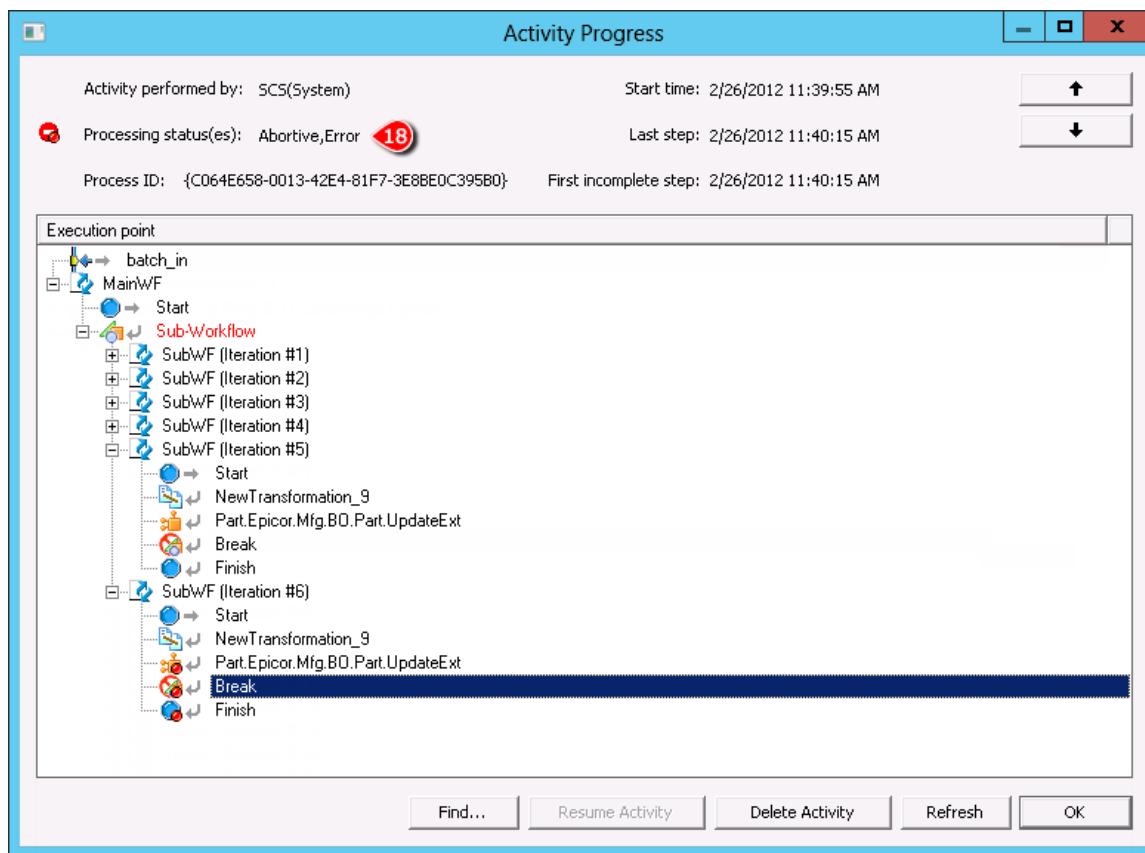
XPath statement is displayed in the simplified form: namespaces are omitted and only the xml elements names are displayed.

14. Click in the **Break Code** field and select the break code that should be passed to the main workflow.
15. In the **Default break code** field, select the break code that will be passed to the main workflow when none of the specified conditions is met.
16. Click **OK**.

17. In the ESC Administration Console, under the Document Tracking, 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.



- 18.** 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 state you defined — **Abortive** in this example.



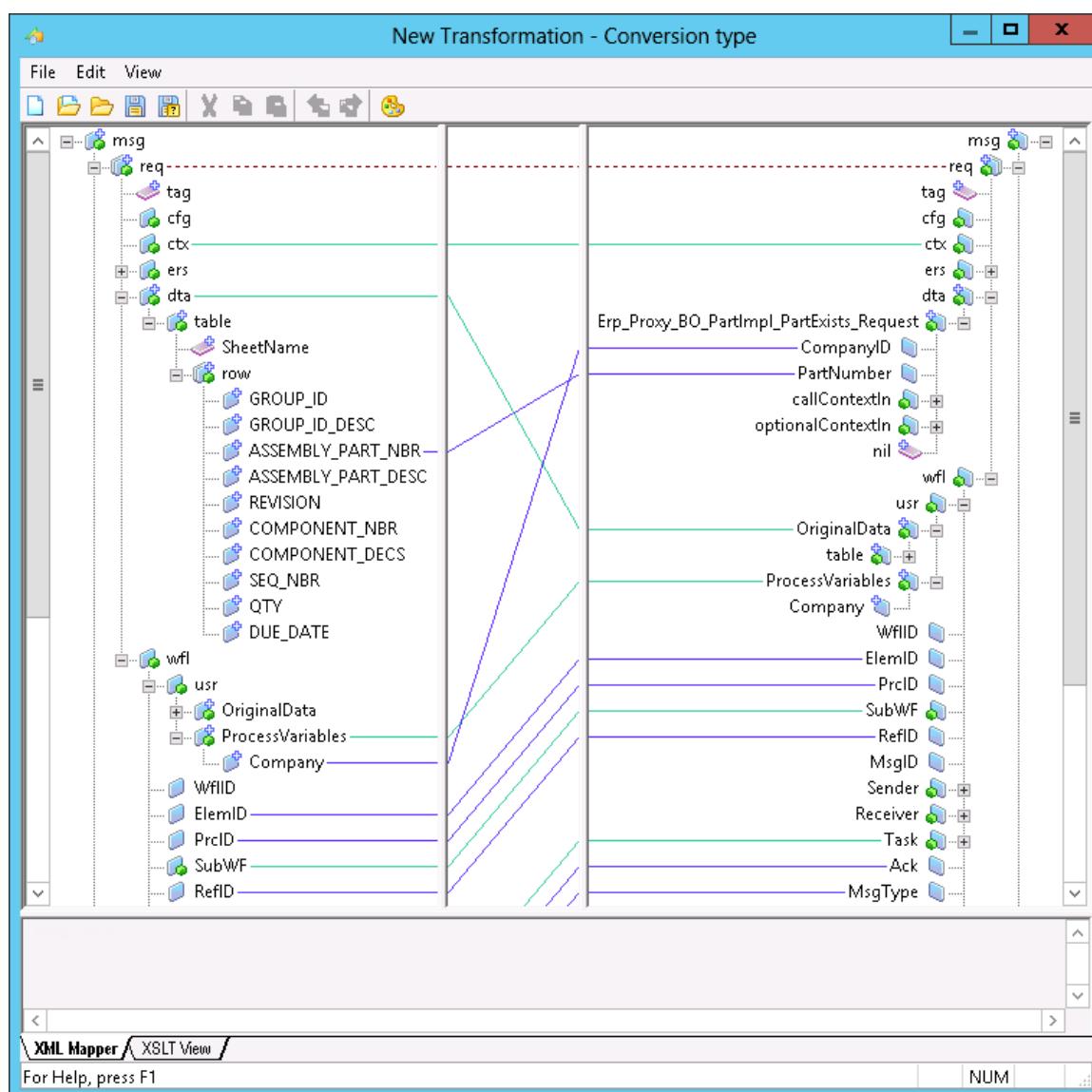
XML Mapper

The XML Mapper is a tool you can use from within a Conversion workflow element 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. 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.

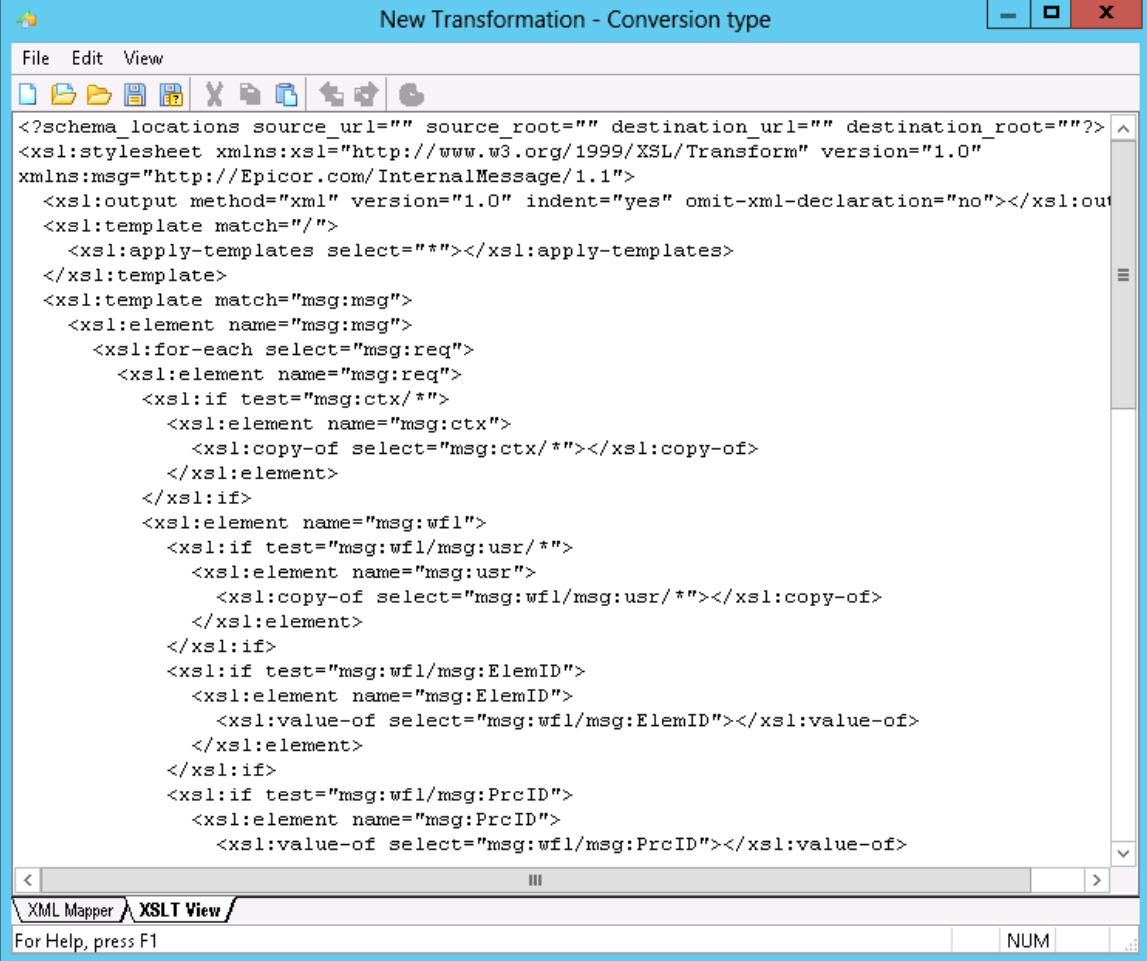
You can open the XML Mapper from the Configuration field of a Conversion Properties sheet by clicking Edit. This section of the document explains the XML Mapper interface and describes the various tools and techniques you can apply to a transformation.

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 .NET Call.



This graphic displays the XSLT View of the same transformation file.



The screenshot shows the 'New Transformation - Conversion type' window in the Epicor Service Connect Workflow Designer. The main pane displays the XSLT code for a transformation. The code defines an XSLT stylesheet with various templates and conditions to map XML elements from one source to another. The interface includes standard Windows-style menus (File, Edit, View) and toolbars with icons for opening files, saving, and navigating. At the bottom, there's a tab bar with 'XML Mapper' and 'XSLT View' selected, and a status bar at the very bottom.

```

<?schema_locations source_url="" source_root="" destination_url="" destination_root=""?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
xmlns:msg="http://Epicor.com/InternalMessage/1.1">
  <xsl:output method="xml" version="1.0" indent="yes" omit-xml-declaration="no"></xsl:ou
  <xsl:template match="/">
    <xsl:apply-templates select="*"/></xsl:apply-templates>
  </xsl:template>
  <xsl:template match="msg:msg">
    <xsl:element name="msg:msg">
      <xsl:for-each select="msg:req">
        <xsl:element name="msg:req">
          <xsl:if test="msg:ctx/*">
            <xsl:element name="msg:ctx">
              <xsl:copy-of select="msg:ctx/*"/></xsl:copy-of>
            </xsl:element>
          </xsl:if>
          <xsl:element name="msg:wfl">
            <xsl:if test="msg:wfl/msg:usr/*">
              <xsl:element name="msg:usr">
                <xsl:copy-of select="msg:wfl/msg:usr/*"/></xsl:copy-of>
              </xsl:element>
            </xsl:if>
            <xsl:if test="msg:wfl/msg:ElemID">
              <xsl:element name="msg:ElemID">
                <xsl:value-of select="msg:wfl/msg:ElemID"/></xsl:value-of>
              </xsl:element>
            </xsl:if>
            <xsl:if test="msg:wfl/msg:PrcID">
              <xsl:element name="msg:PrcID">
                <xsl:value-of select="msg:wfl/msg:PrcID"/></xsl:value-of>
              </xsl:element>
            </xsl:if>
          </xsl:element>
        </xsl:for-each>
      </xsl:element>
    </xsl:template>
</xsl:stylesheet>

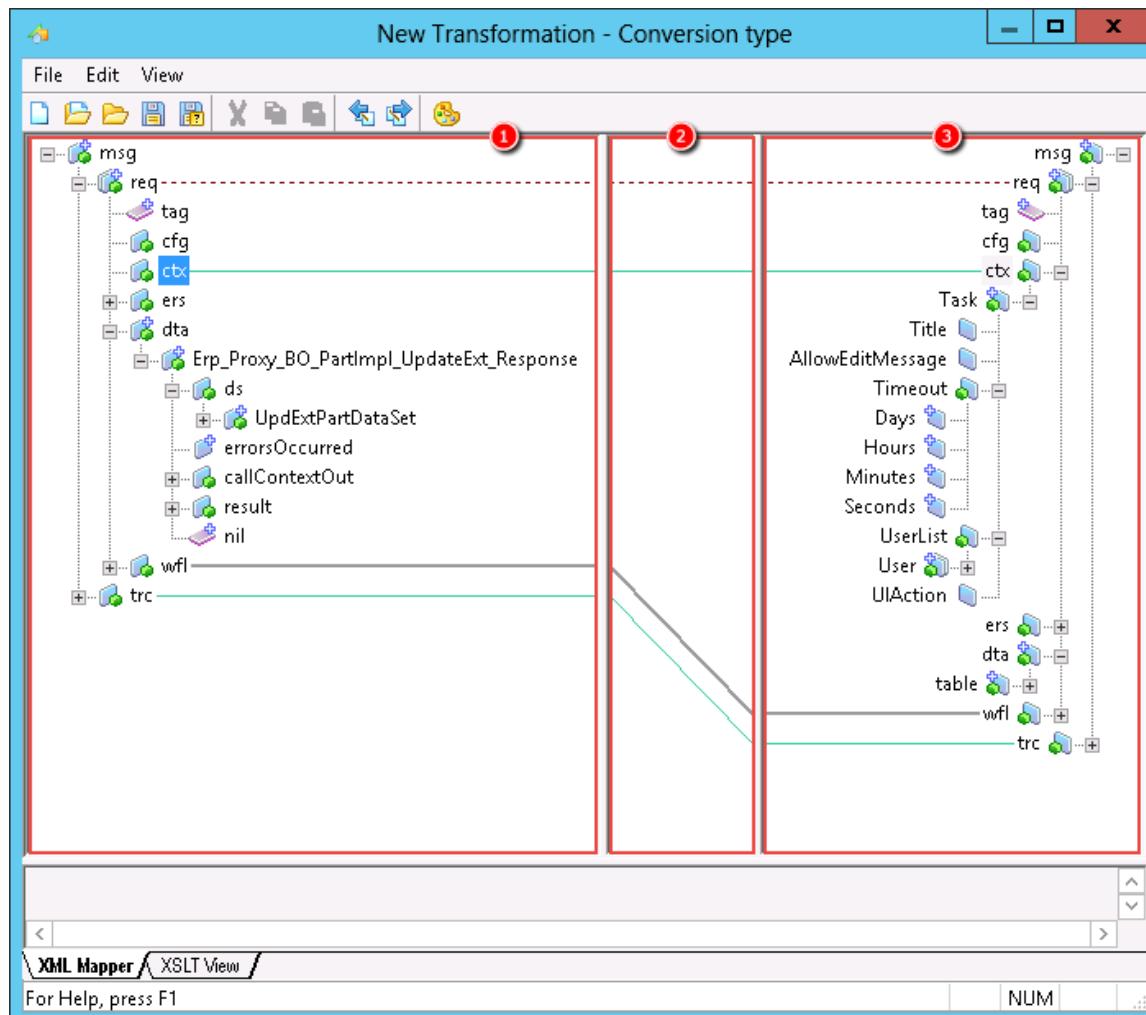
```

XML Mapper Graphical Interface

The XML Mapper is divided into three main areas:

1. The incoming document pane on the left
2. The center pane, which is used for Functoids

3. 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 the Epicor Service Connect Overview chapter. Typically, transformations will use nodes located within the business data node (**dta**) and user data node (**usr**, a child node of the **wfl** node).

XML Mapper features the following view modes you can use to easily find particular nodes within the conversion:

- Show All Nodes
- Show Mapped Nodes Only
- Show Issues Only

To view only those document structure nodes that are mapped, from the **View** menu, select **Show Mapped Nodes Only**. Nodes with Literal values defined in XML Mapper also display in this mode.

To view only the fake nodes, broken links and links with incorrect conditions, from the **View** menu, select **Show Issues Only**.

To return to the default view, in the **View** menu, select **Show All Nodes**.

XML Mapper preserves your view mode selection when you move forward and backward through different conversions.

Document Nodes

In XML Mapper, there are four basic types of document 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. <p>Example The business data node (dta) is an example. Only one dta node is allowed in the Service</p>

Node Icon	Description
	<p>Connect internal message envelope, and the node can contain a complex hierarchy of child nodes.</p>
	<p>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.</p> <p>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.</p> <p>If you just drag a link between two collections this will not copy all child fields automatically. To copy all child fields, select the Deep Copy check box in the Link Properties. You can also generate links for all child fields automatically if you hold the Ctrl key while dragging the line.</p> <p>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:</p> <ul style="list-style-type: none"> • 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 schema and can be rejected. <p>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.</p>

When you drag a link between two nodes, the XML Mapper tries to create the most appropriate link or links for these nodes. To use the Ctrl key is the most universal way to create valid working conversions. Disadvantage of dragging link with Ctrl key pressed is significant size of xslt file. The size of such xslt file can be up to 2..3 MB for a big BO. Processing of Conversions with such big xslt file requires additional 1..2 seconds. To show a large xslt conversion XML Mapper also requires additional 5..10 seconds, there may be thousands links.

To create links quickly and to keep the conversion xslt size relatively small, you can mix both ways: create a link using Ctrl, then select and remove links for some child nodes to re-map them without Ctrl key. Especially this way should be used if you want to map from a different node.

Attribute Nodes



Attribute nodes, which provide additional information about an element node, appear as child nodes to their respective elements.

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.

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.

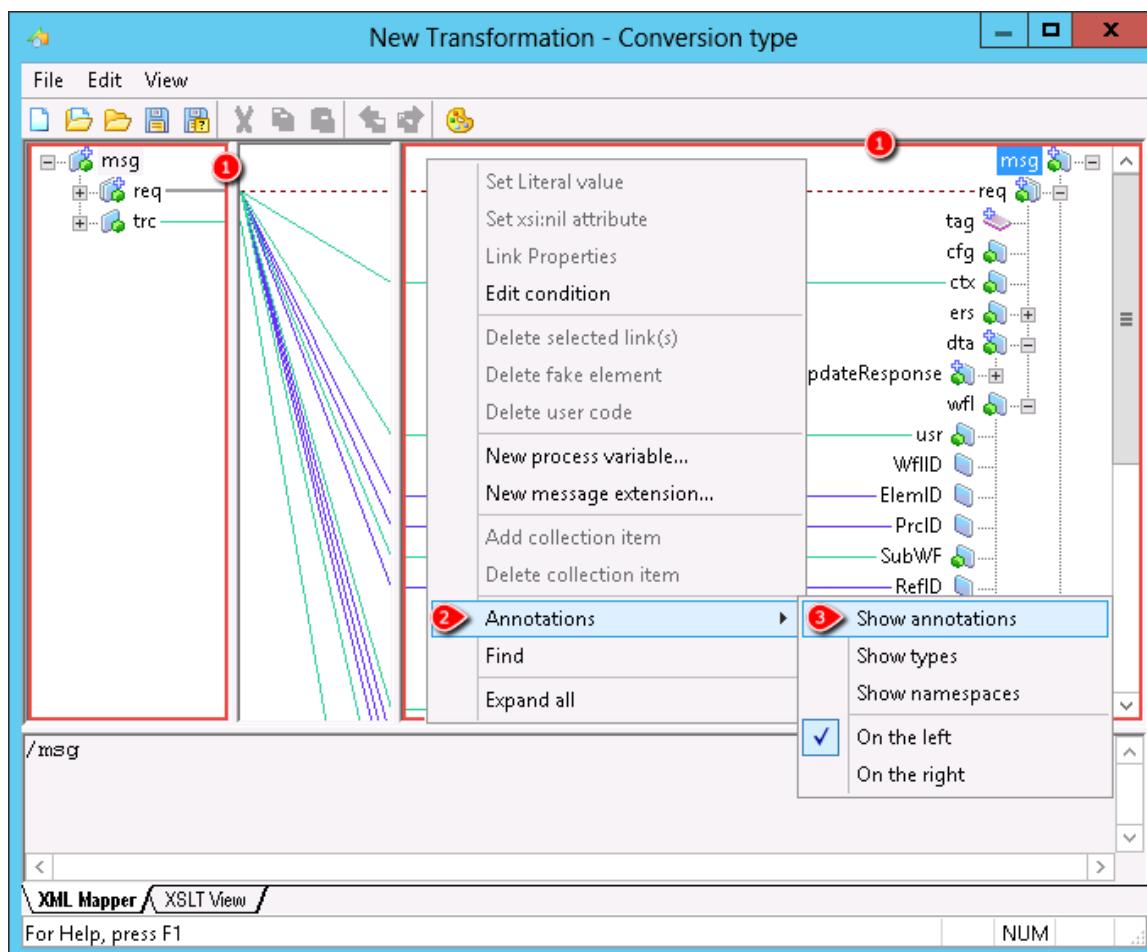
Node Annotations and Data Types

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.

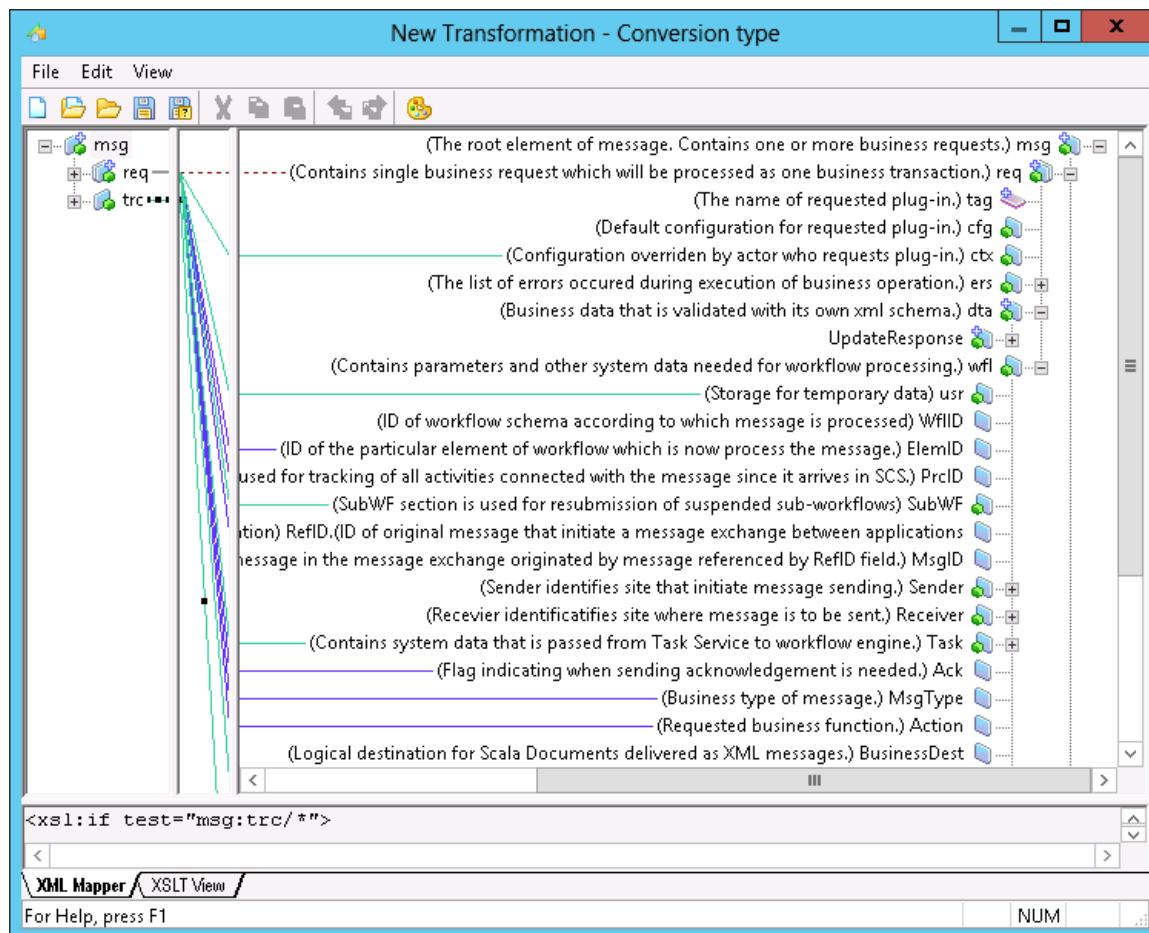
To show the node annotations:

1. Right-click anywhere in the incoming or target document pane.



2. Highlight **Annotations**.
3. Select **Show annotations**.

The following graphic shows the annotations for a target document.

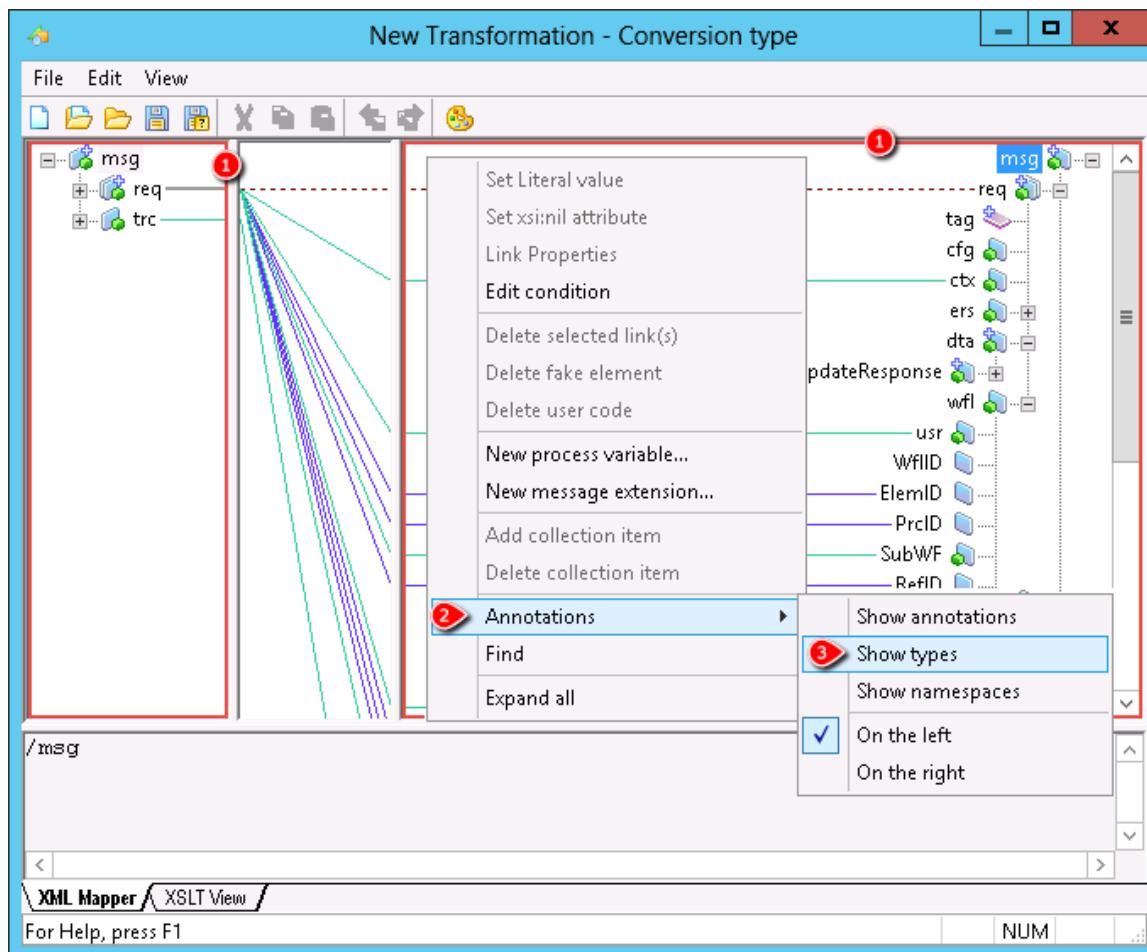


Notice some nodes do not have annotations. It is possible to add annotations to user-defined schemas. For an example of how to add annotations to a schema, see the ScalInternalMsg.xsd file, located in the SCS\ScalInternalMsg folder where Service Connect is installed.

Data types indicate the type of data required by the XSD schema for the node.

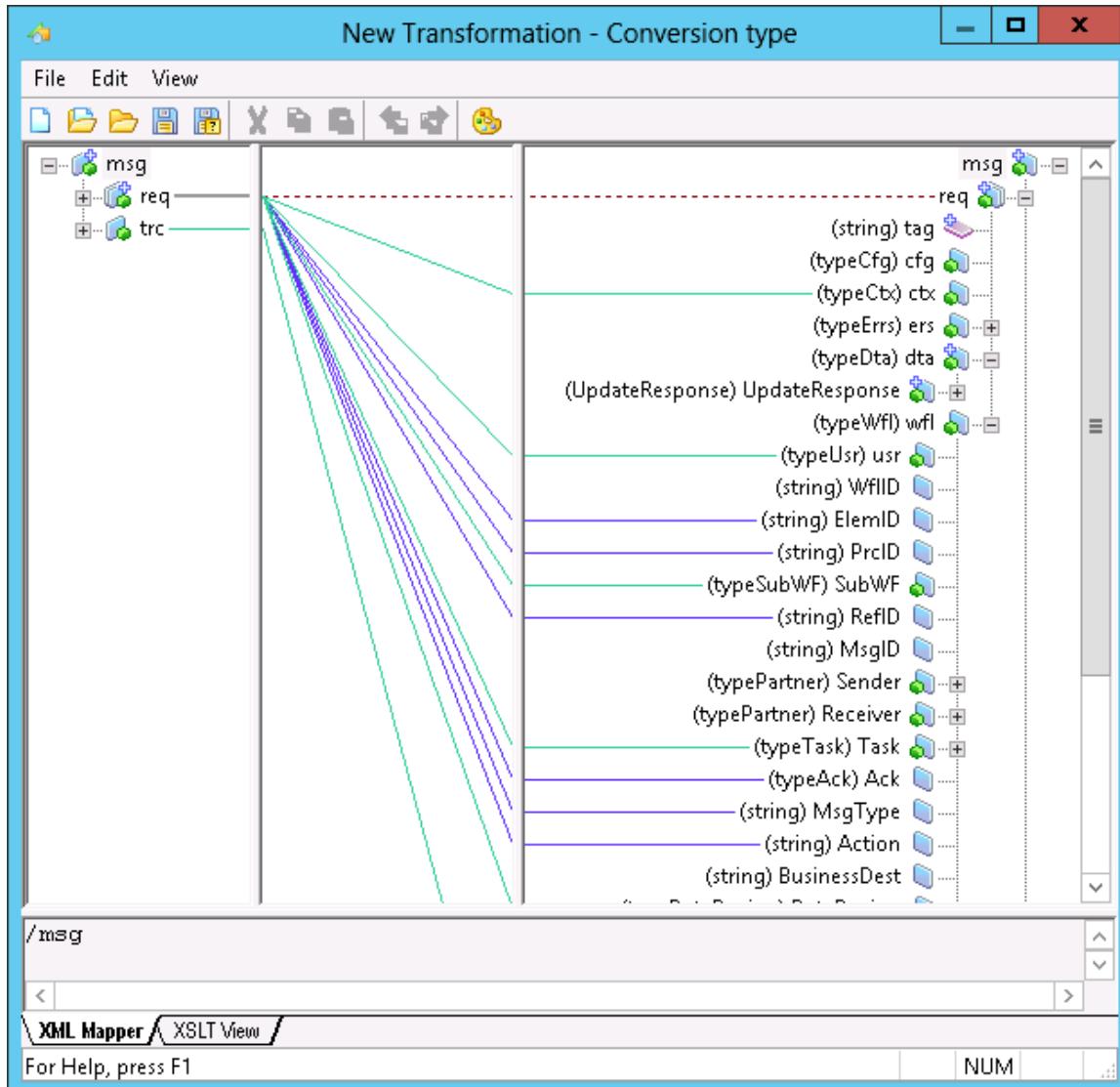
To show the data types:

1. Right-click anywhere in the incoming or target document pane.



2. Select **Annotations**.
3. Select **Show types**.

The following graphic shows the data types for the target document.

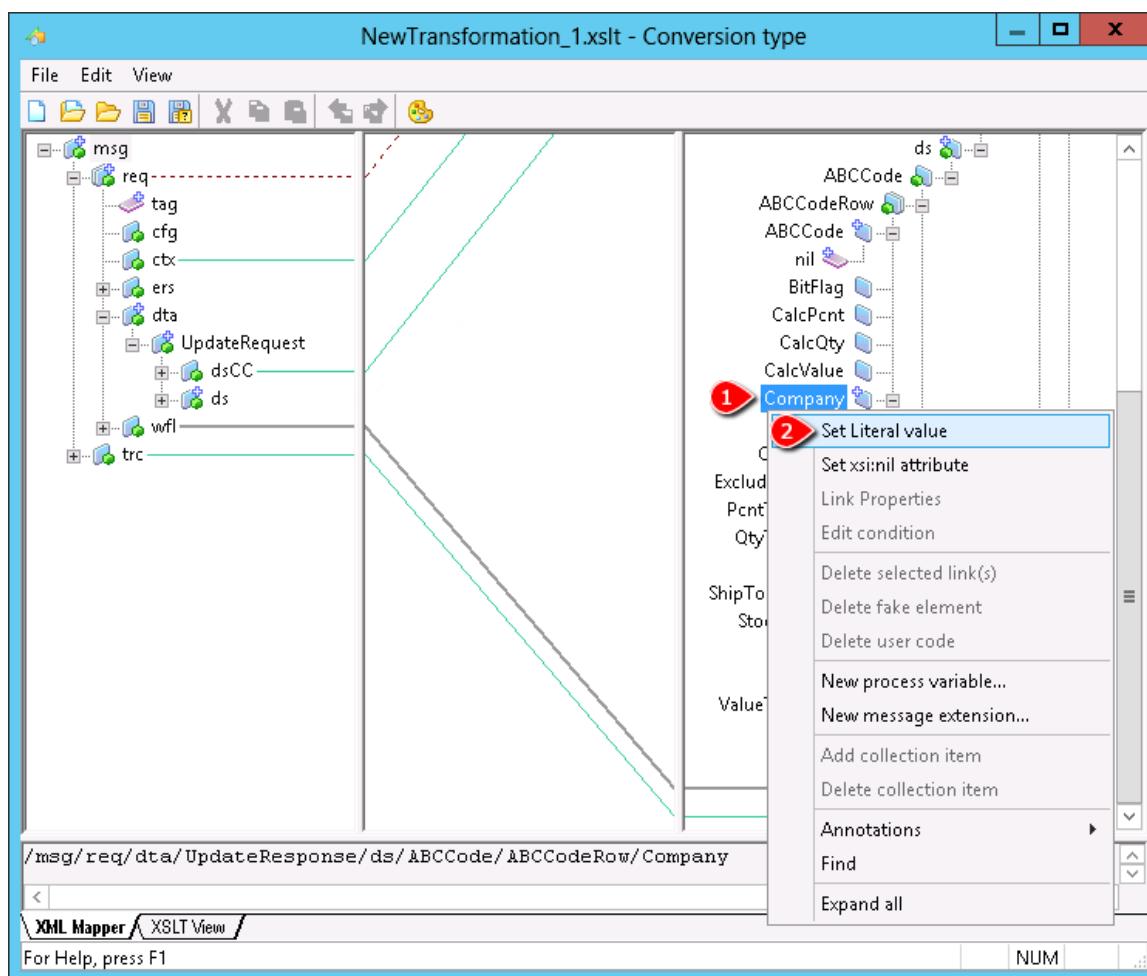


Apply a Literal Value

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.

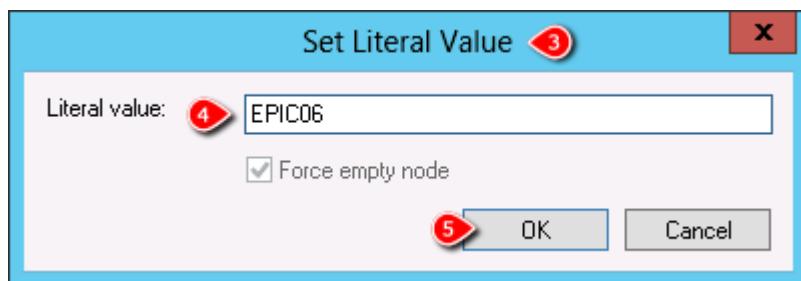
To apply a literal value to a node:

- Right-click a node in the target document.



- Select **Set Literal value**.

- The **Set Literal Value** window displays.

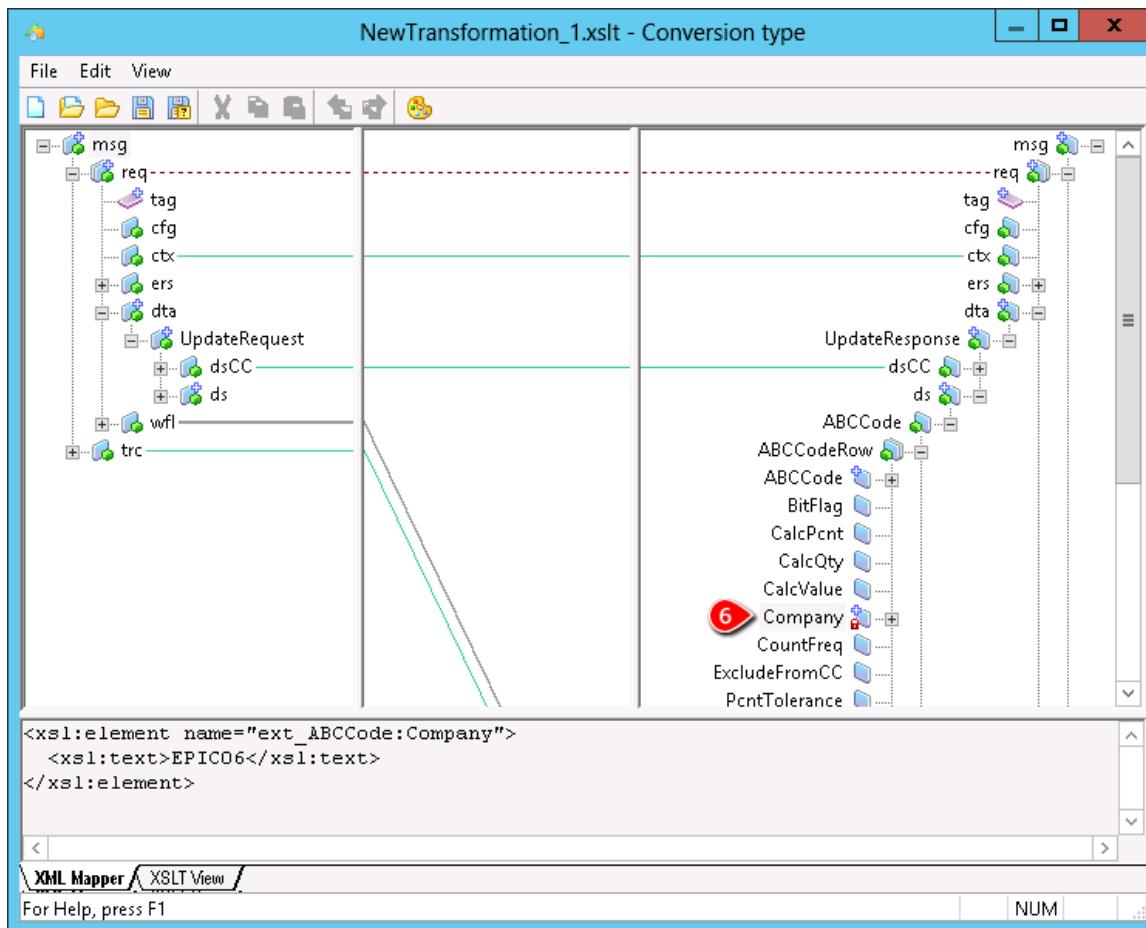


- Enter a **Literal value** or select the **Force empty node** check box to indicate the node should display in the target document but will not have a value.

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.

- Click **OK**.

6. Notice the node now has a lock sign on its icon.



In this example, the value of the Company node is set to the string value EPIC06.

Set Up a Node Condition

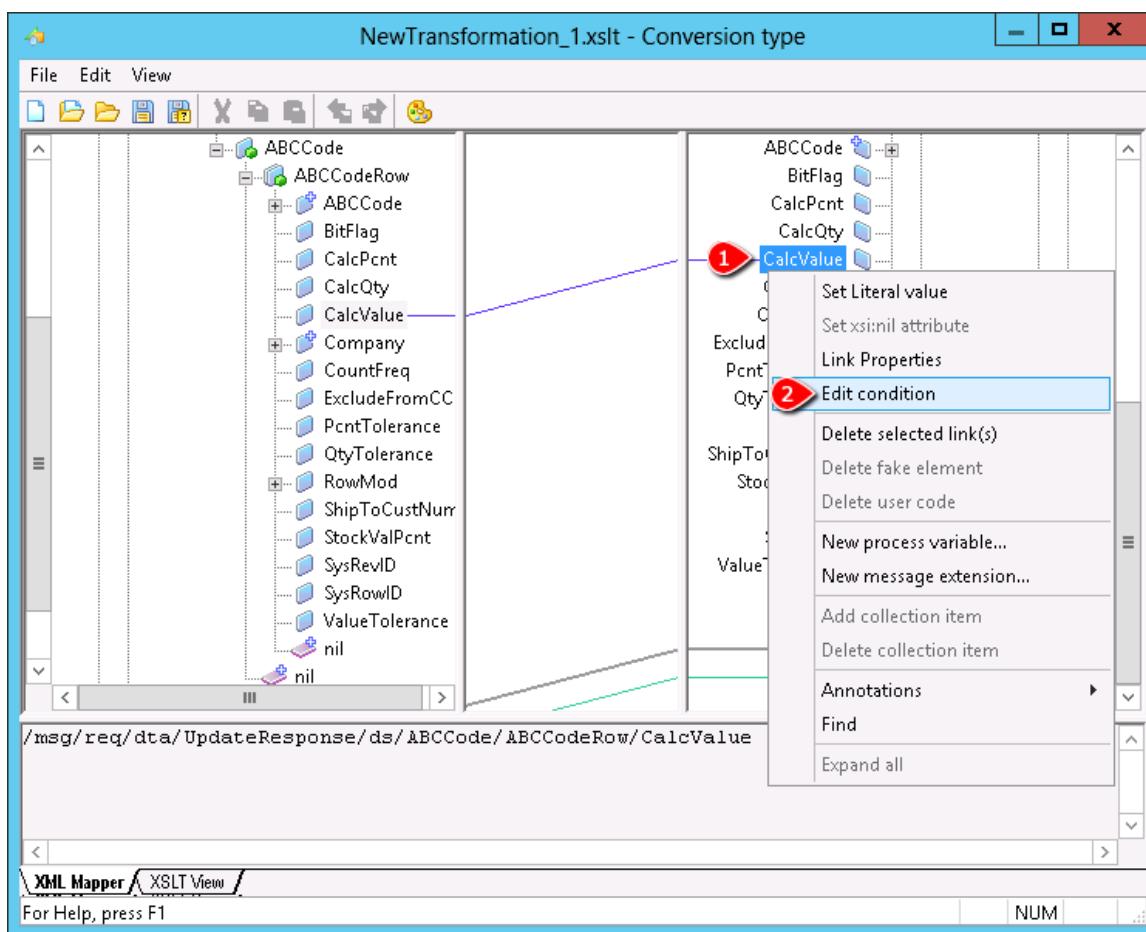
A node condition acts as criteria that specifies whether the node in the target document is created. If the condition is satisfied, the node in the target document is created. If the condition is not satisfied, the node is omitted.

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.

The target document node for which you want to create a condition must be linked to a node in the incoming document or have a literal value set.

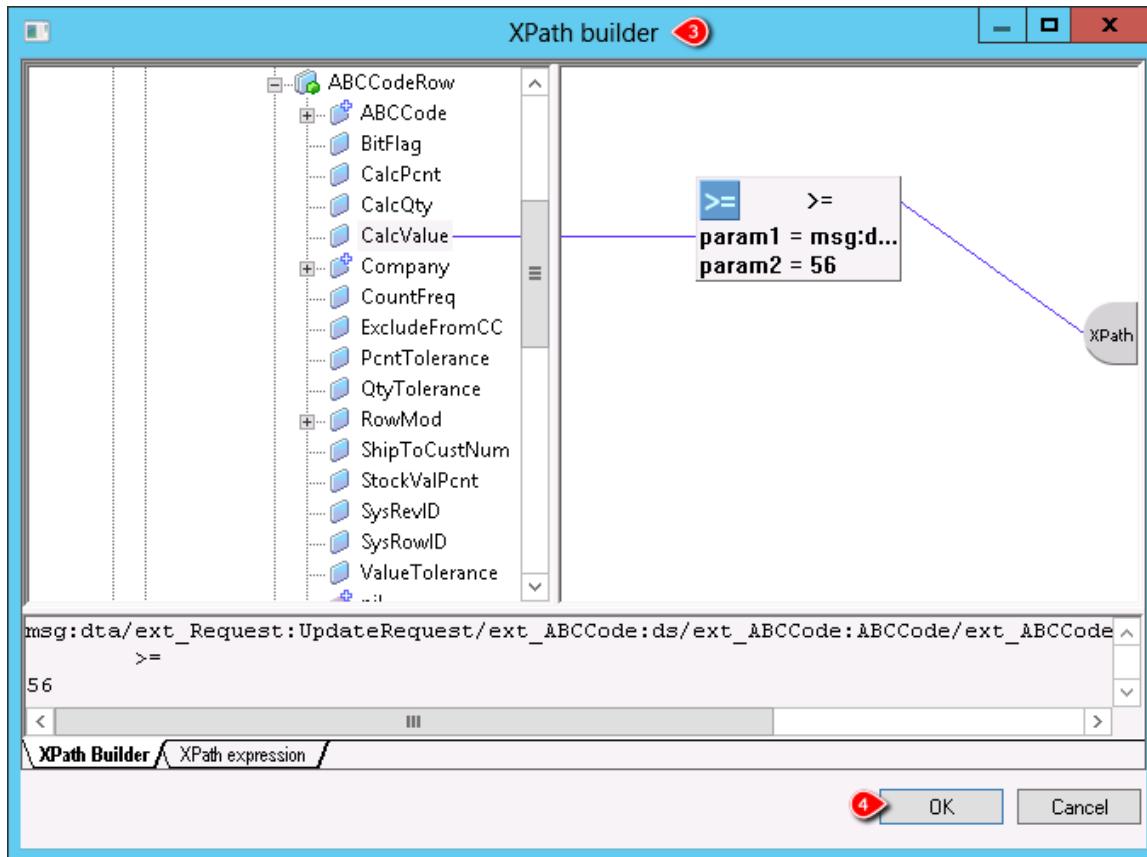
To set up a node condition:

1. Right-click a node in the target document.



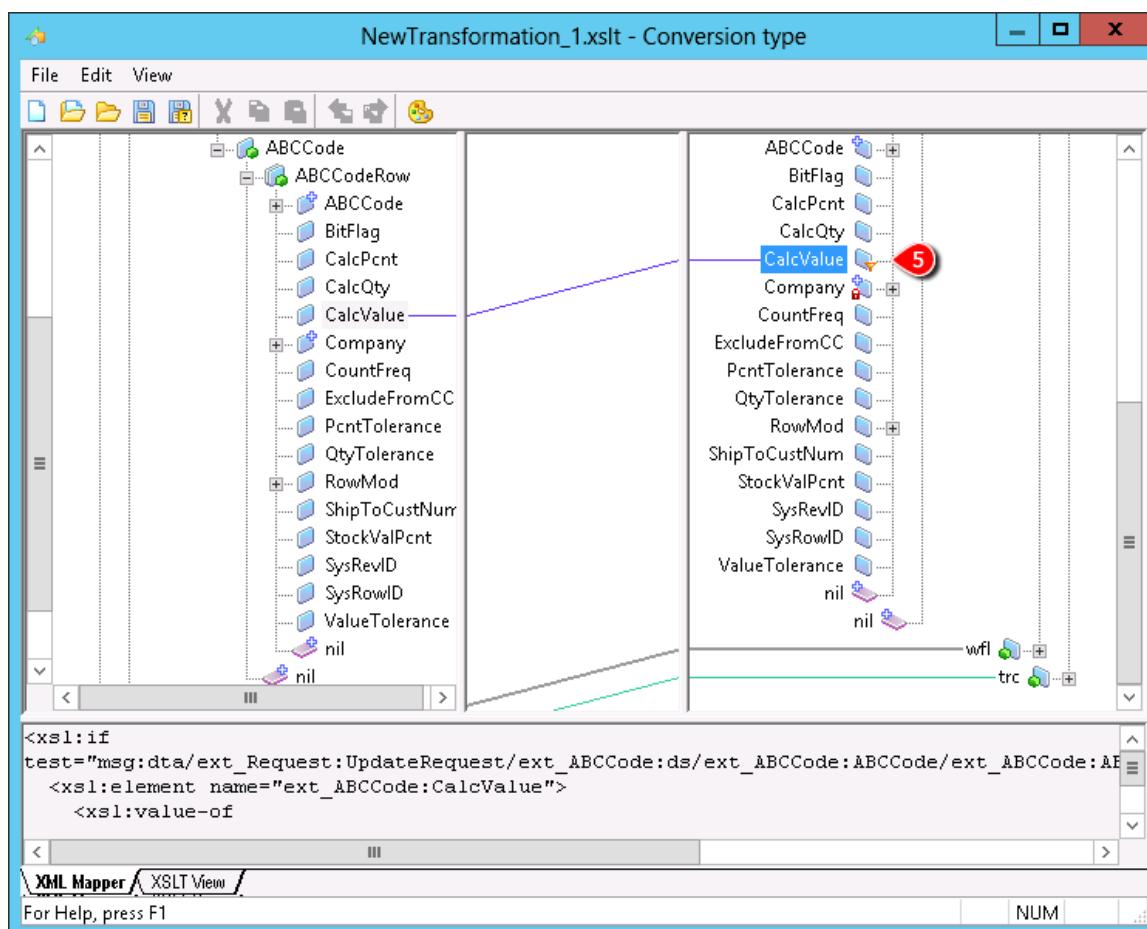
2. Select **Edit condition**.

3. 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.



4. In the **XPath builder** window, click **OK**.

5. Notice the node now has a filter sign on its icon.



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.

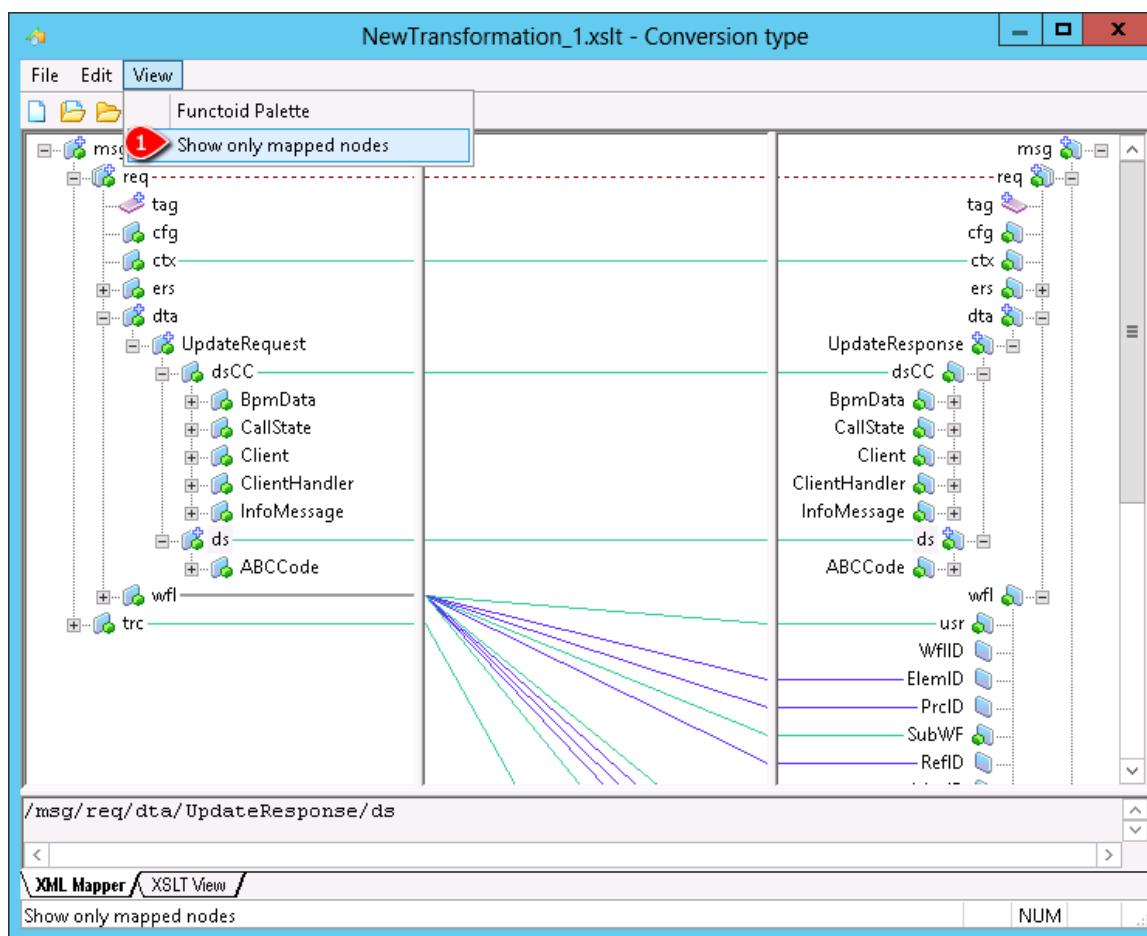
Mapping	Description
	This mapping is represented as a solid, green line.
	A mapping between two simple collection nodes. For every XML element represented by the incoming collection node, an identical XML element will be created in the target collection node. This mapping is represented as a dashed, 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. Forbidden mappings that are created against the schema rules are also displayed in red.

View Mapped Nodes Only

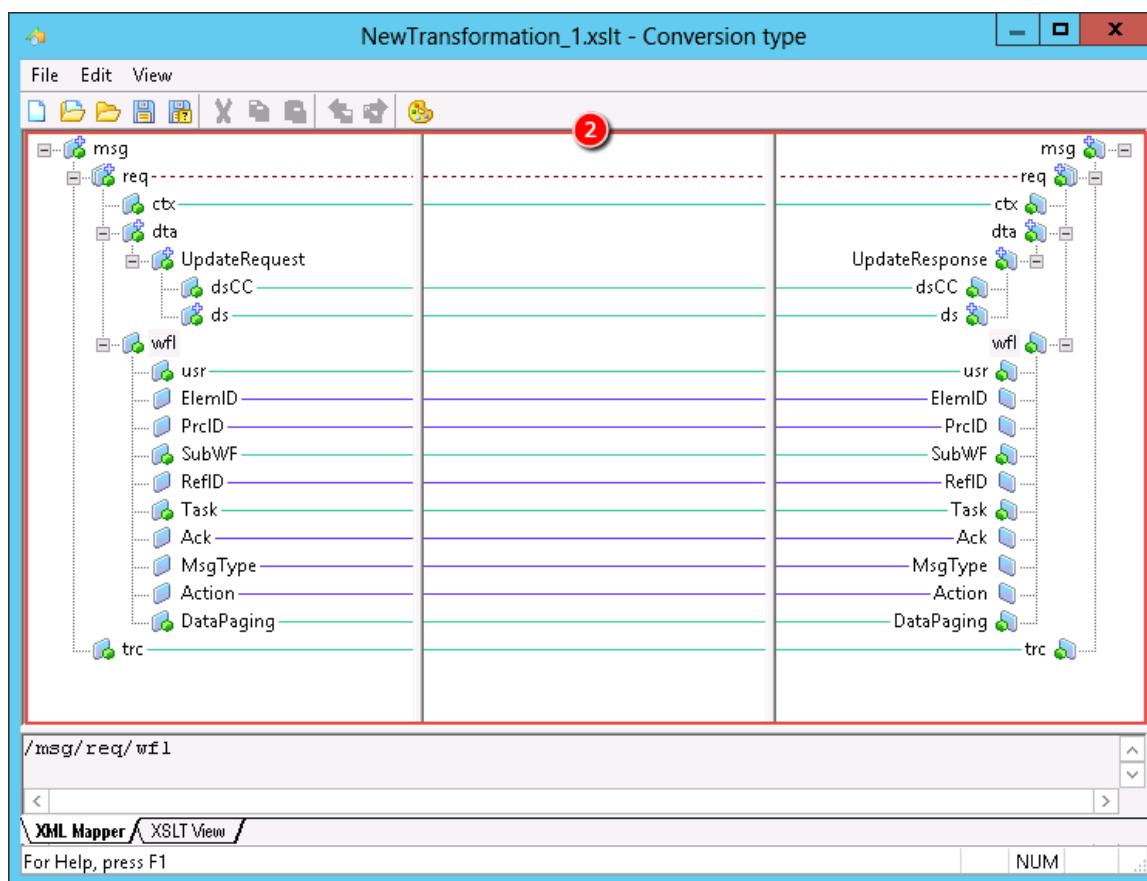
It is possible to only display nodes mapped between the incoming and target document.

To show only mapped nodes in XML Mapper:

- From the **View** menu, select **Show only mapped nodes**.



2. View the simplified graphical interface that only includes mapped nodes and literal values.



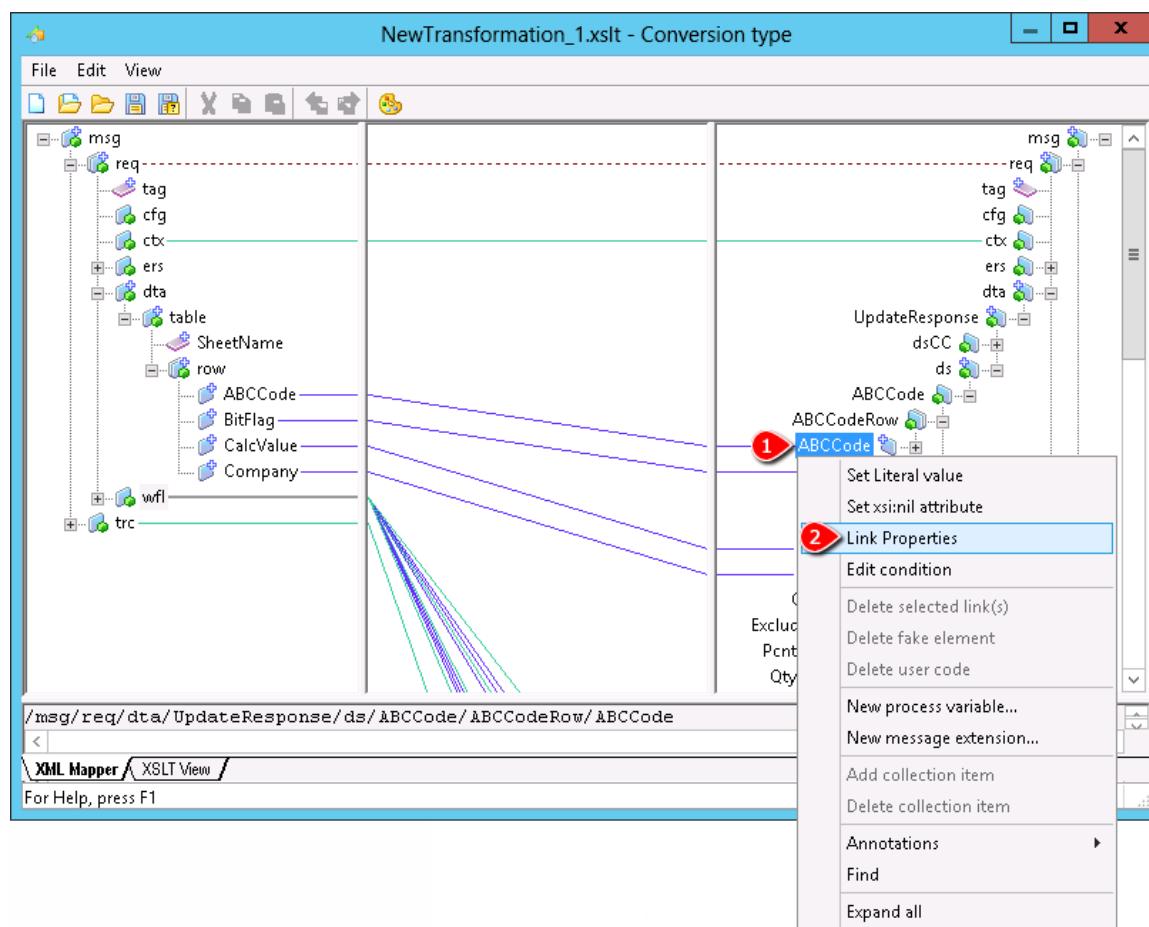
Set Up a Link Condition

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:

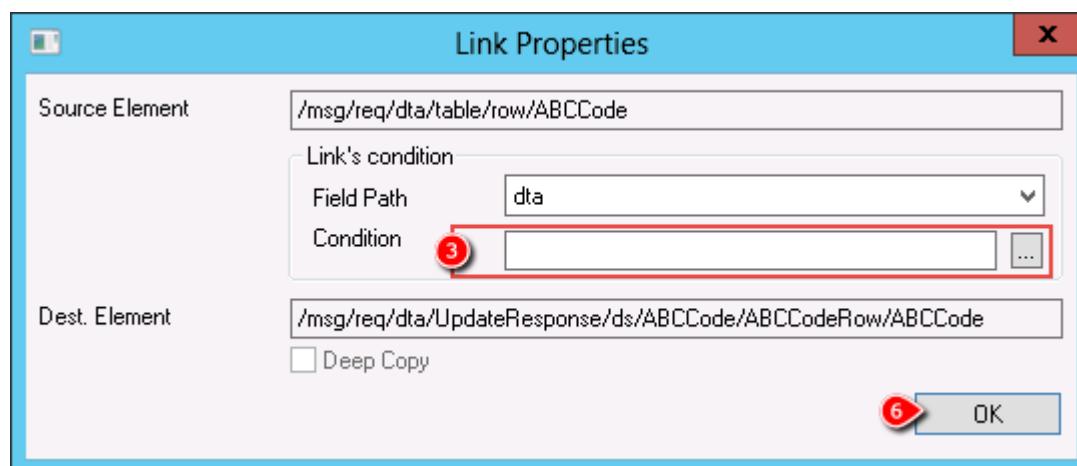
1. Right-click a node in the target document.

The node must be linked to a node in the incoming document.



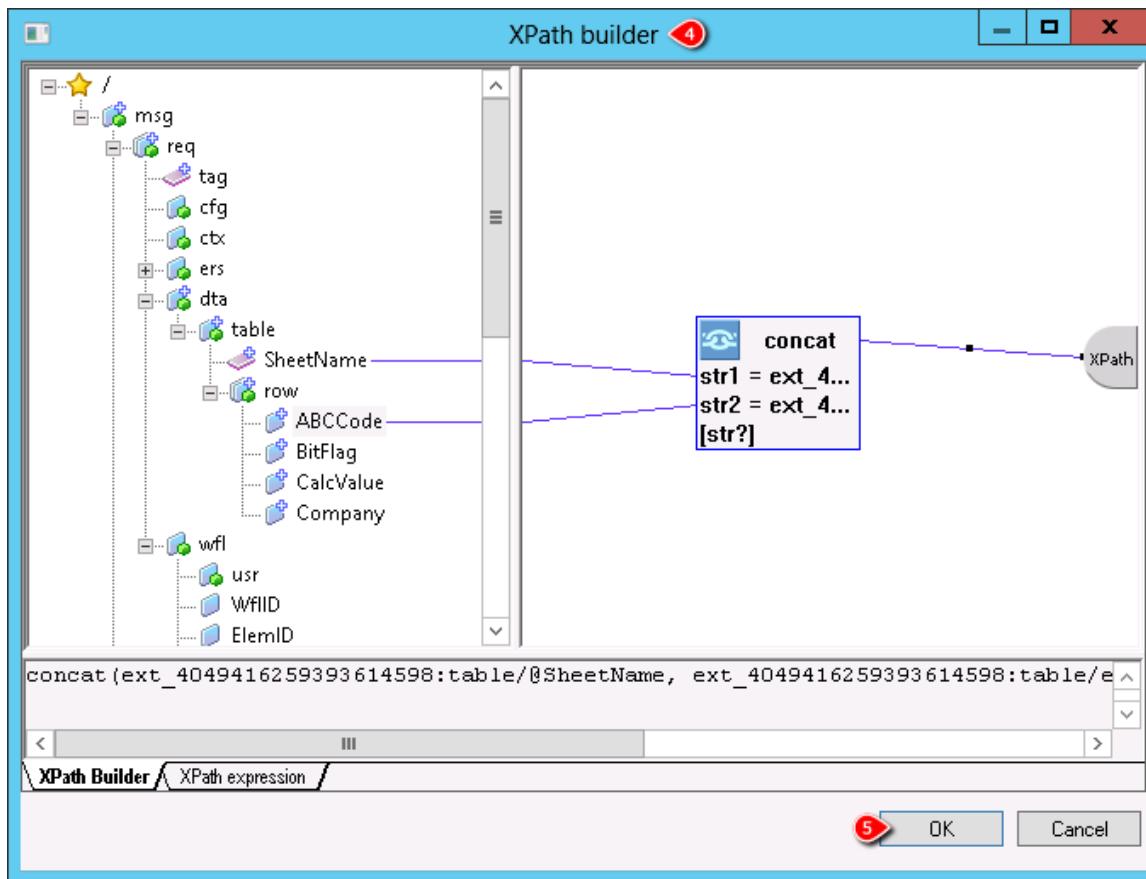
2. Select Link Properties.

3. 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 build the XPath expression in the XPath builder.



4. The **XPath builder** displays. Create a rule that will act as the selection criteria.

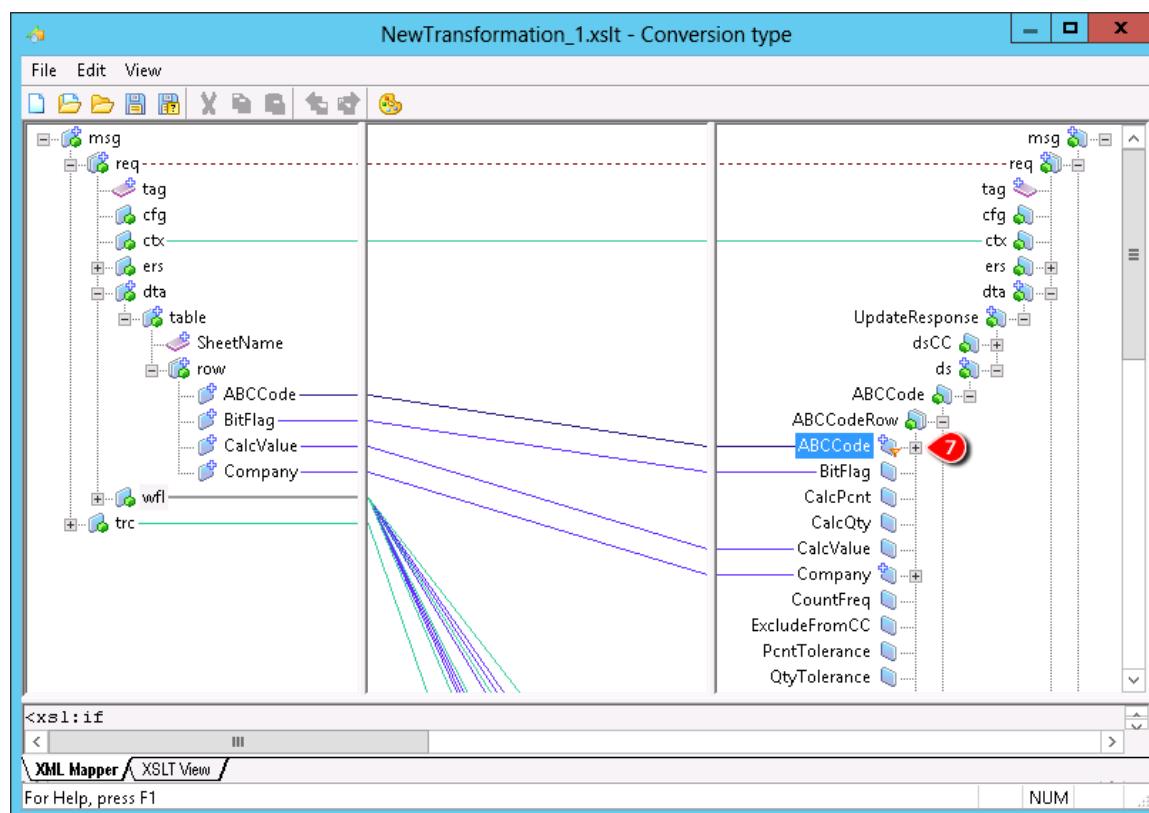
You can use Functoids to create the rule.



In this example, the ABCCode node in the target document is a concatenation of a Excel sheet name from the incoming document and an incoming document node (ABCCode).

5. In the **XPath builder** window, click **OK**.
6. In the **Link Properties** window, click **OK**.

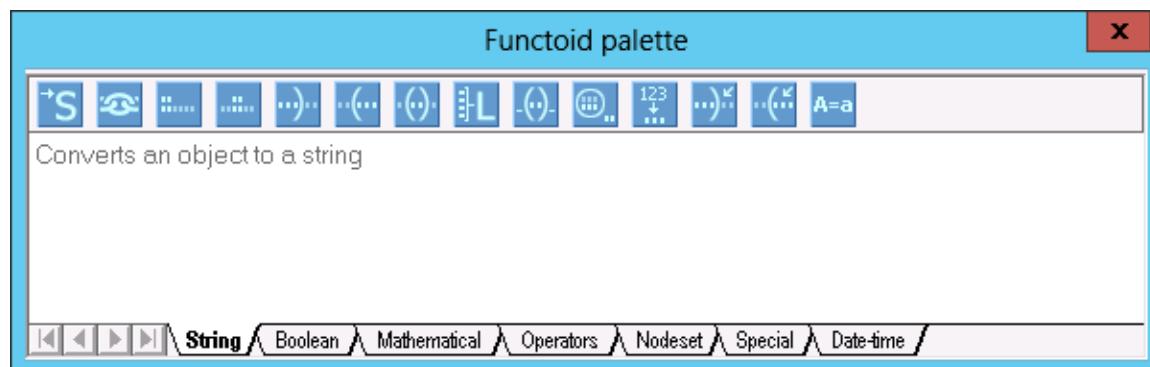
7. Notice, in the target document, the node icon changes to display a filter sign.



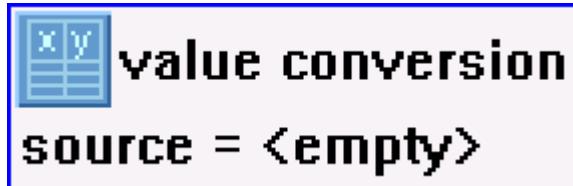
Functoids

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.

The following graphic displays the Functoid Palette.

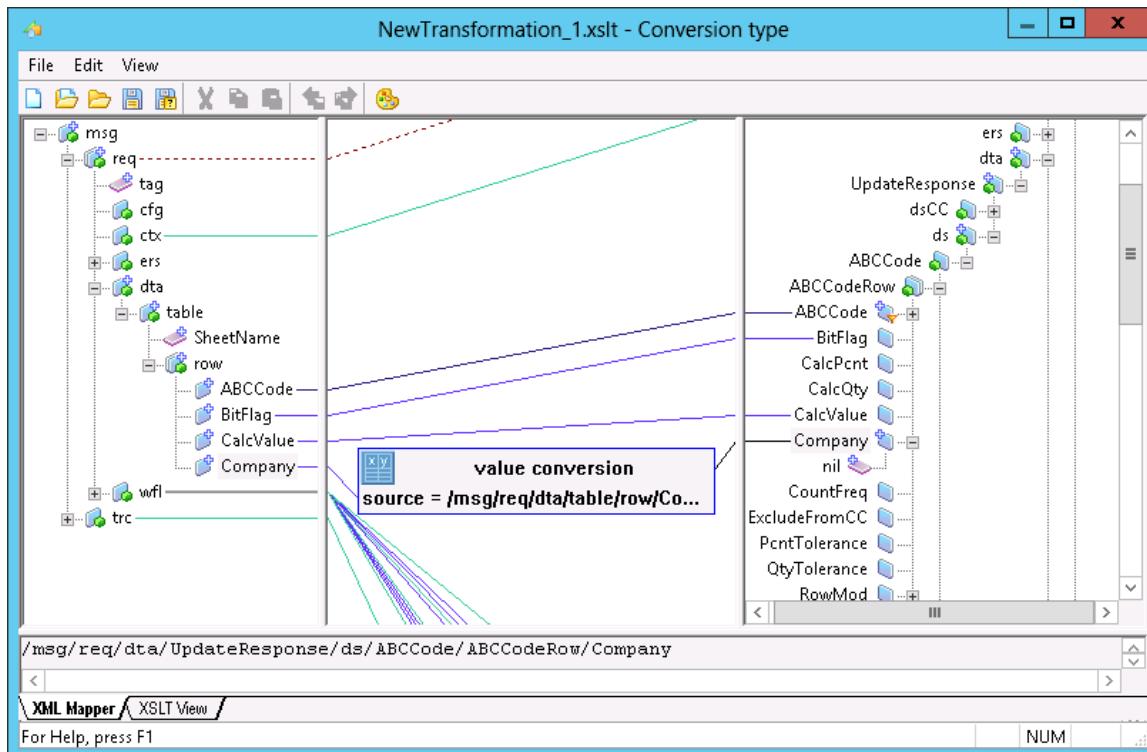


The following graphic shows how a functoid appears when it is added to the center pane of the XML Mapper.



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.

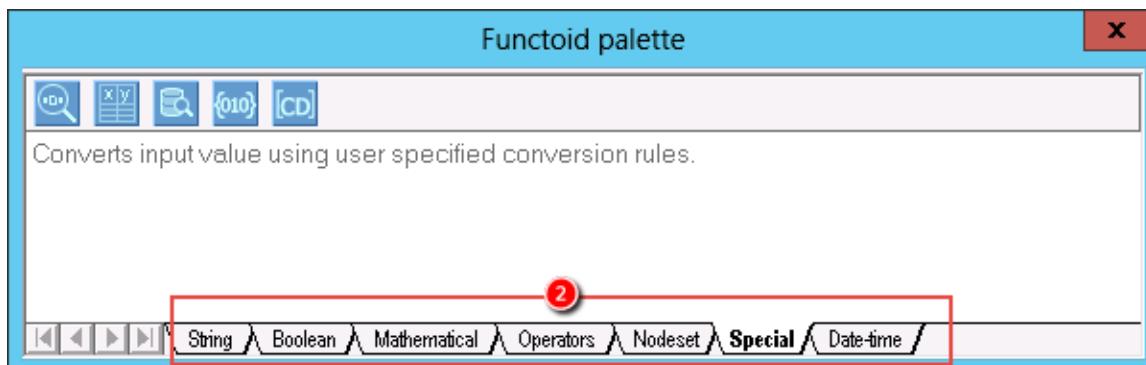
Add a Functoid to a Conversion

To add a functoid to a Conversion:

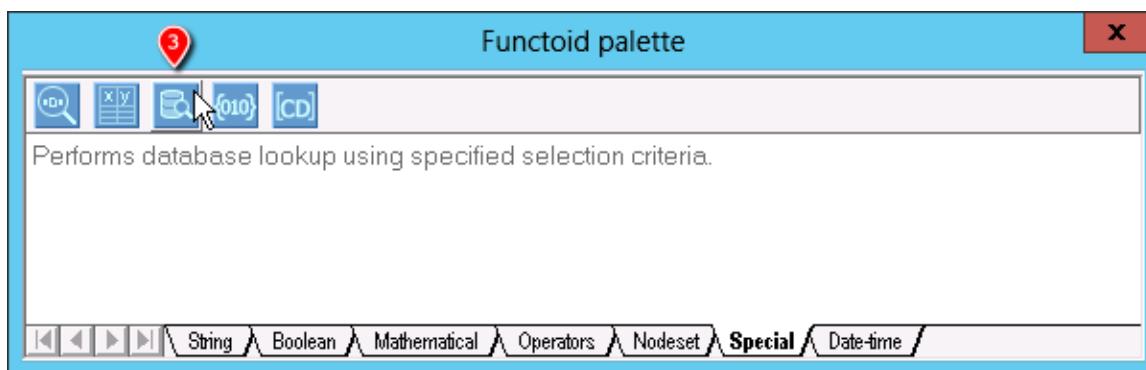
1. In the XML Mapper, click the **Functoid Palette** button on the **Standard** toolbar.



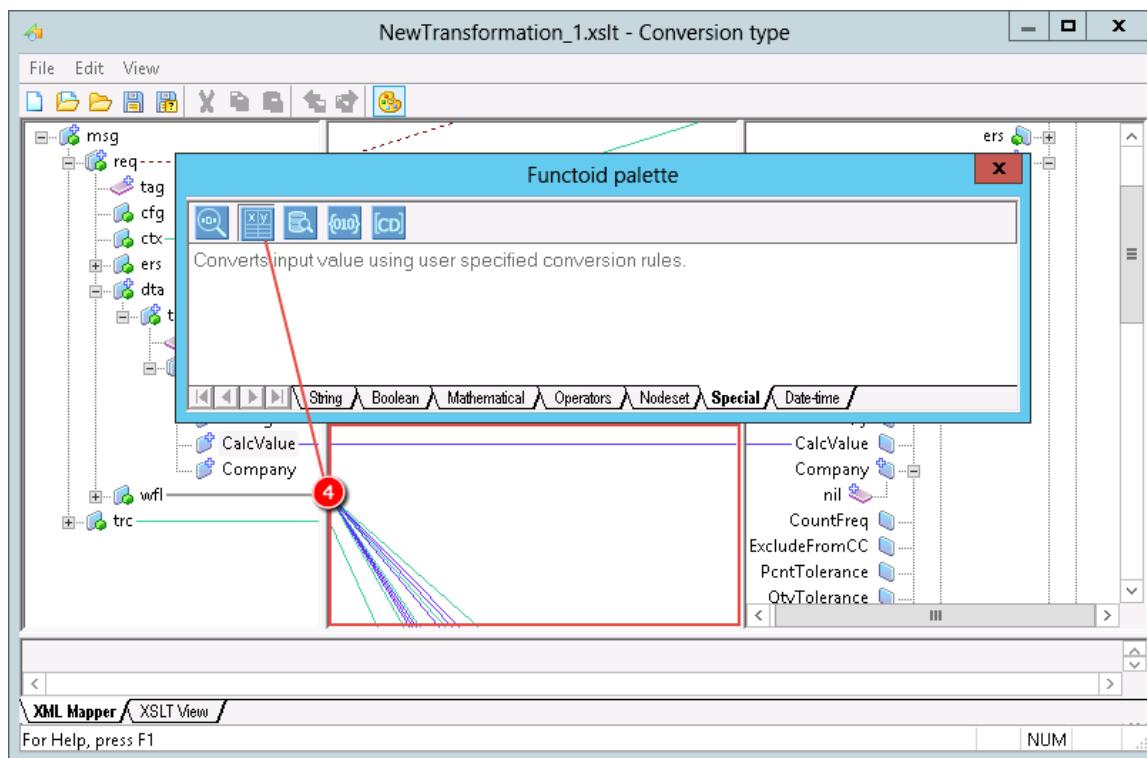
2. Select the appropriate tab at the bottom of the Functoid Palette.



3. Point to a functoid button in the palette to see its description.

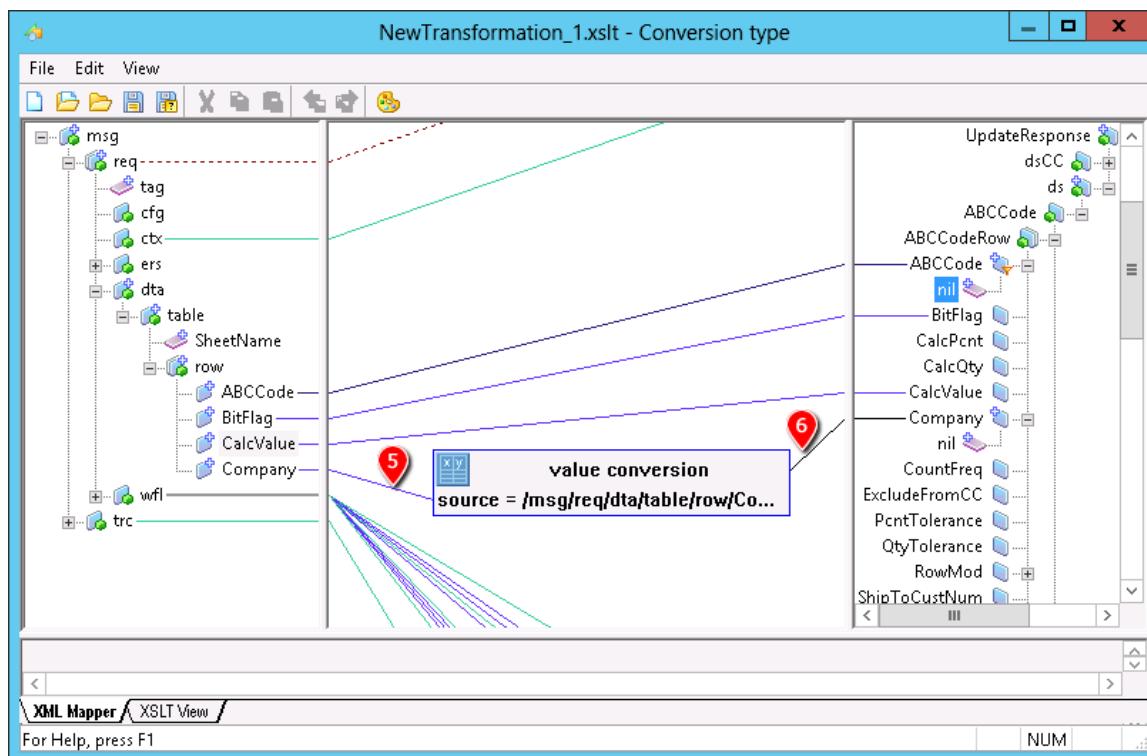


4. Click a functoid button and drag the functoid to the center pane of the XML Mapper.



5. 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.



6. Map the functoid to another functoid or map the functoid to a node in the target document.

String Functoids

The following table lists the String Functoids, sorted by type.

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. For example, if you provide the following arguments: substring("12345",2,3), the result is 234. substring("12345", 0, 3) results in 12.
	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

Functoid	Name	Description
		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

The following table lists the Boolean Functoids, sorted by type.

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

The following table lists the Mathematical Functoids, sorted by type.

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.
	min	Returns the minimal value of all the nodes in the node-set. Each node is first

Functoid	Name	Description
		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

The following table lists the Operators Functoids, sorted by type.

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

The following table lists the Nodeset Functoids, sorted by type.

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.
	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

The following table lists the Special Functoids, sorted by type.

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

Functoid	Name	Description
		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, {xxxxxxxx-xxx ... xxxx}.
	toCDATA	Wraps the node to the CDATA section.

Date-time Functoids

The following table lists the Date-time Functoids, sorted by type.

Functoid	Name	Description
	toDateTime	Converts an object to the dateTime data type.
	currentDateTime	Returns the current date. currentDateTime returns a date-time value with a precision up to seconds.
	currentDateTimeEx	Returns date-time value with a precision up to milliseconds. This functoid is similar to currentDateTime functoid, but currentDateTime returns a date-time value in format "yyy'-'MM'-dd'T'HH':'mm':'ssK"

Functoid	Name	Description
		whereas currentDateTimeEx returns a date-time value in format "yyyy'-'MM'-'dd'T'HH':'mm':'ss'.fffK".
	localDateTime	Returns local date and time of the ESC server with fractional seconds (accuracy up to milliseconds).
	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.
	formatTimeEx	<p>This functoid is similar to format-time functoid, but contrary to format-time formatTimeEx supports fractional seconds and is able to take a value and format string with fractional seconds.</p> <p>For example, a value "15:56:42.894" and a format string "hh-mm-ss.fff tt" passed to formatTimeEx produce an output string "03-56-42.894 PM". Notice a fractional seconds format character 'f' exists in the format string.</p>

Functoid	Name	Description
		Similarly to dateTimeFromStringEx functoid the formatTimeEx has an additional lang parameter that takes locales like "en-US" or "ru-ru" as format-date.
	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.
	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.
	addDurationToDateTimes	Returns the dateTime computed by adding the duration specified in the second argument to the dateTime specified in the first argument.
	subtractDurationFromDateTimes	Returns the dateTime computed by subtracting the duration specified in the second argument from the dateTime specified in the first argument.
	secondsAfterMidnight	Returns a numeric value that represents the number of seconds after midnight in the dateTime value.
	dateTimeFromString	Converts the specified string to dateTime value using the specified format. The functoid receives the date string as the first argument and the format string as the second argument.
	dateTimeFromStringEx	Similar to dateTimeFromString, dateTimeFromStringEx functoid has

Functoid	Name	Description
		an additional lang parameter that takes locales like "en-US" or "ru-ru" as format-date.

Date-time Functoids Format String

The following format string elements are accepted in the Value field of Date-time functoids.

dateTimeFromString, format-date functoids

Format String Element	Description
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.
99	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.

dateTimeFromString, format-time functoids

Format String Element	Description
h	Hours as 0-12.
hh	Hours as 00-12.
H	Hours as 0-12.
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.

dateTimeFromString, dateTimeFromStringEx, localDateTime, currentDateTimeEx, formatTimeEx functoids

Format String Element	Description
f	Fractional part of a second in datetime value.
F	A format string for the dateTimeFromString and formatTimeEx functoids can have up to 7 consecutive "f" or "F" symbols. localDateTime and currentDateTimeEx work with format strings that contain up to three "f" characters.

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.

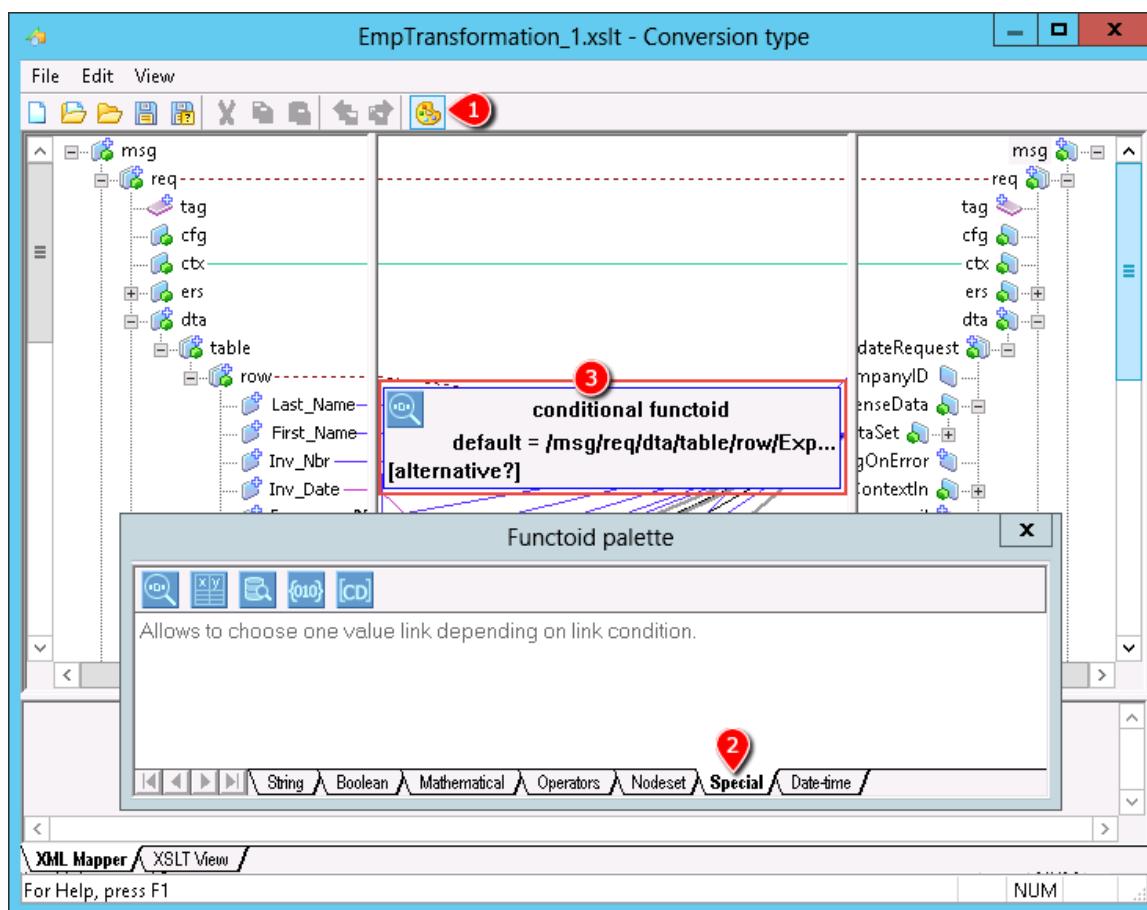
Conditional Functoid Example

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.

Set Up a Conditional Functoid

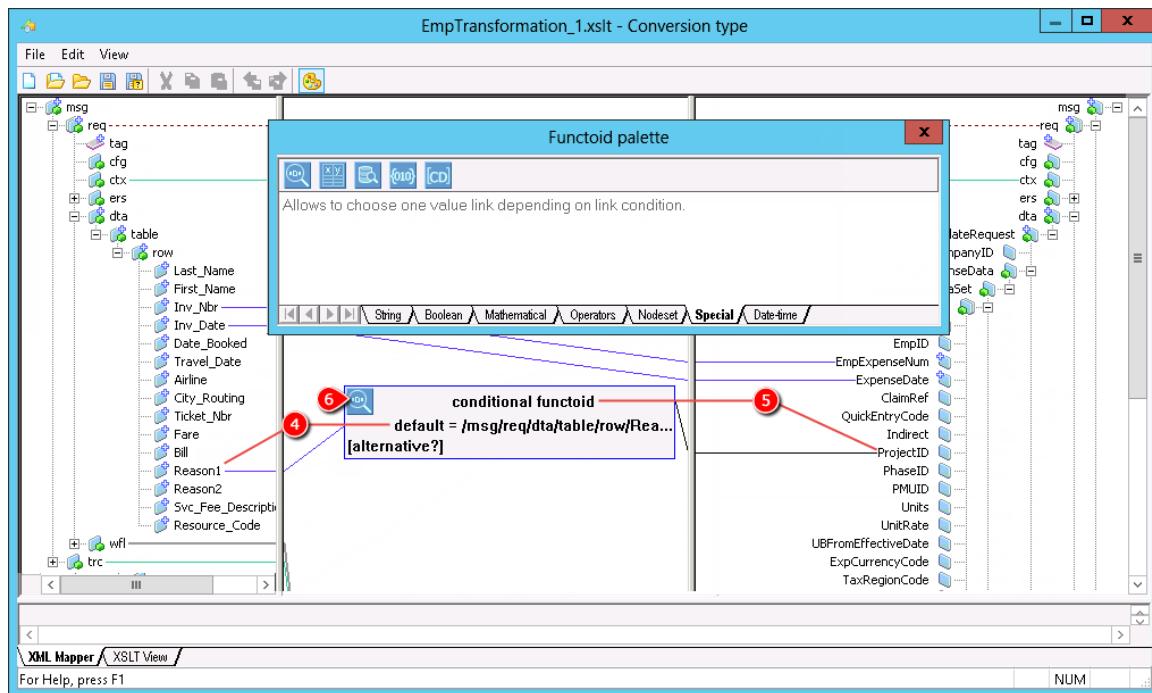
To set up a conditional functoid:

1. On the XML Mapper standard toolbar, click the **Functoid Palette** button.

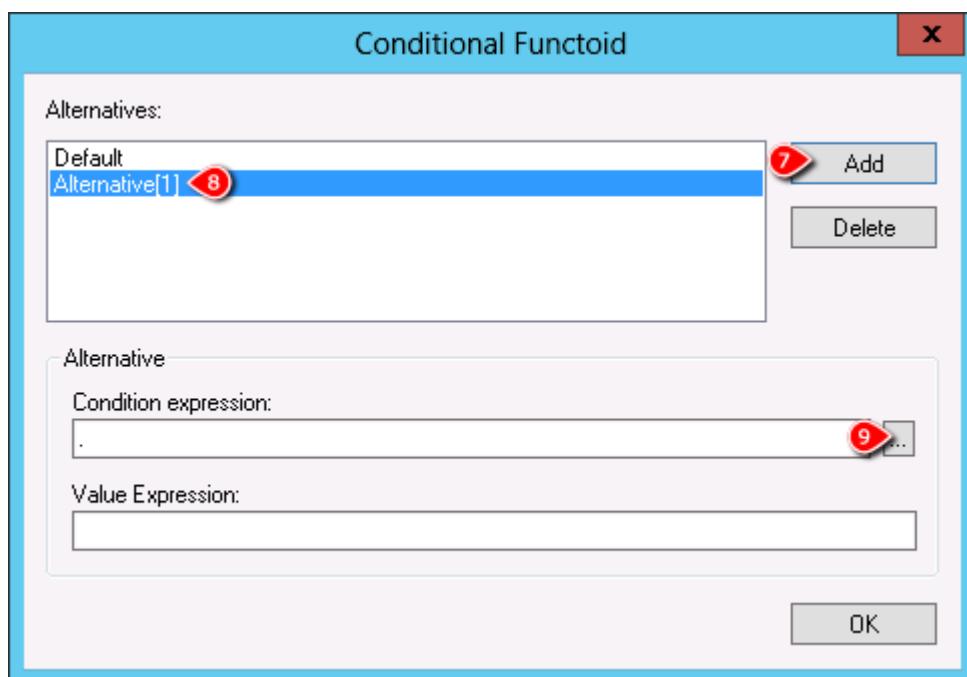


2. Open the **Special** tab of the Functoid palette.
3. Add the **conditional** functoid to the XML Mapper middle pane.

4. Map the **Reason1** node in the incoming document to **default**.



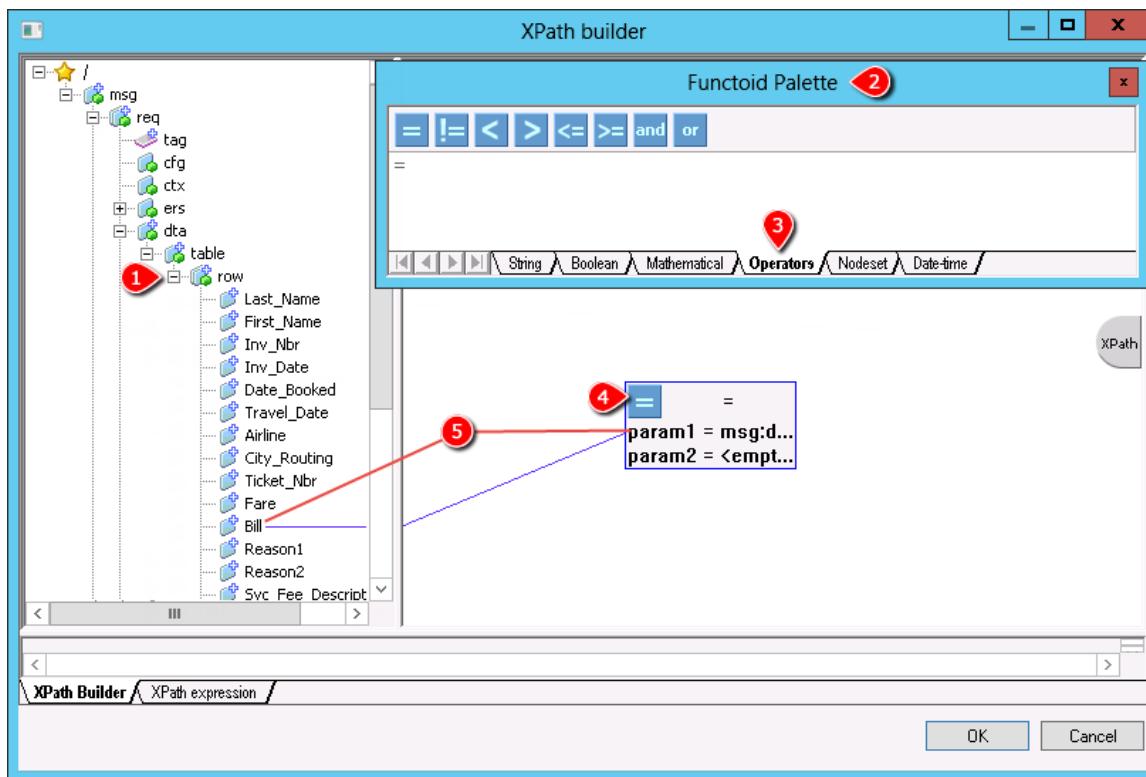
5. Map the conditional functoid to the **ProjectID** node in the target document.
 6. Double-click the functoid to open the **Conditional Functoid** window.
 7. Click **Add**.



8. From the **Alternatives** list, select **Alternative[1]**.
 9. Click the ... (Ellipse) button next to the **Condition expression** field.

Build XPath

1. In the XPath Builder **tree view**, expand the **req > dta > table > row** node.

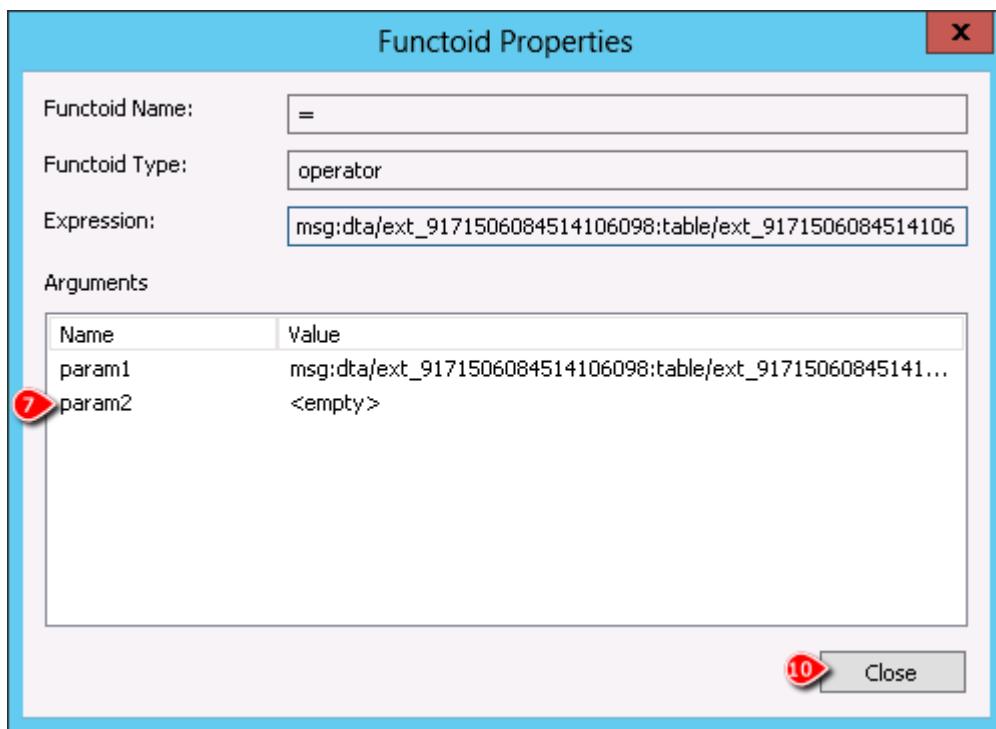


2. Right-click in the right pane and select **Show Functoid Palette**. **Functoid Palette** displays.

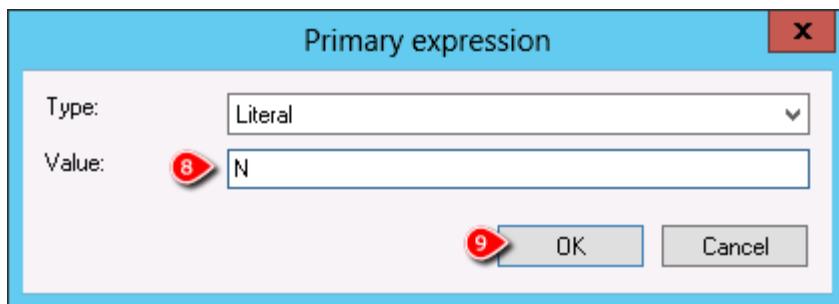
Not all the functoids that display in the XML Mapper also display in the XPath Builder. The list of functoids available in the XPath Builder depends on the context from which the XPath Builder is called. Conditional, Value Conversion, Union, DBOperation, ToCDATA, CreateGUID functoids cannot be used in XPath Builder because they are predefined XSL constructions (<xsl:choose>, <xsl:otherwise>, <xsl:variable>) and are not suitable for XPath.

3. In the **Functoid Palette**, open the **Operators** tab.
4. Drag and drop the **equals** (=) functoid to the right pane of the XPath Builder.
5. Map the **Bill** node to **param1**.
6. Double-click the functoid to open the **Functoid Properties**.

7. In the **Functoid Properties** window, double-click **param2** where the Value is <empty>.



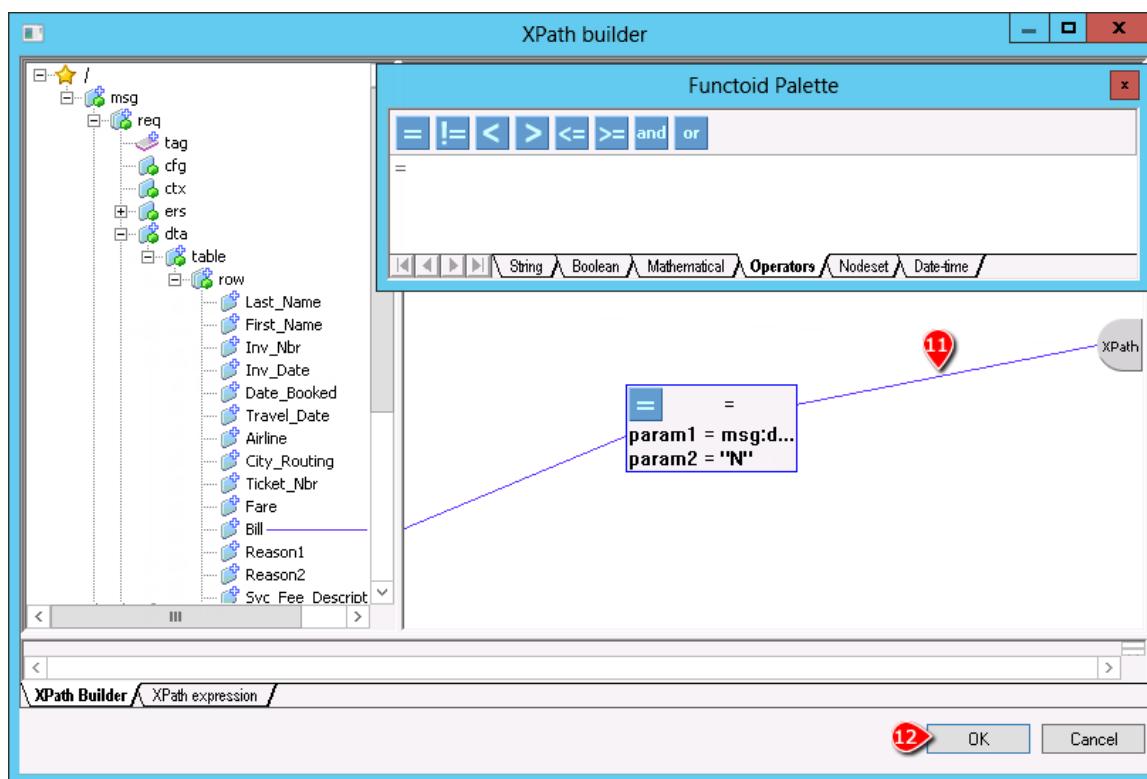
8. The **Primary expression** window displays. Enter **N** as the **Value**.



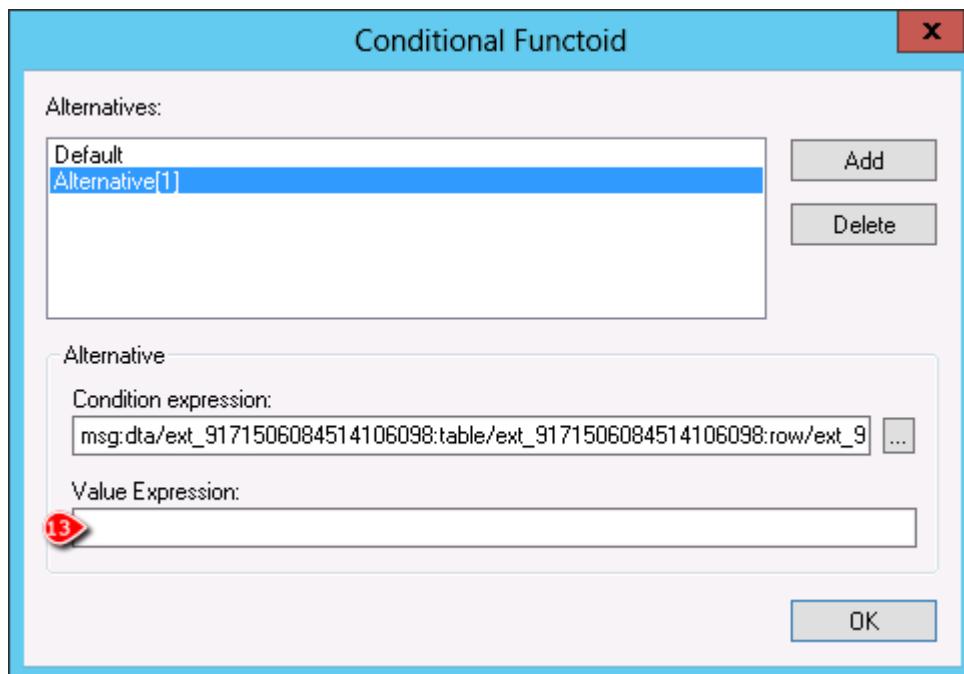
9. Click **OK**.

10. In the **Functoid Properties** window, click **Close**.

11. Map the functoid to the **XPath** marker.



12. Click **OK** again to return to the Conditional Functoid dialog box.
13. 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:

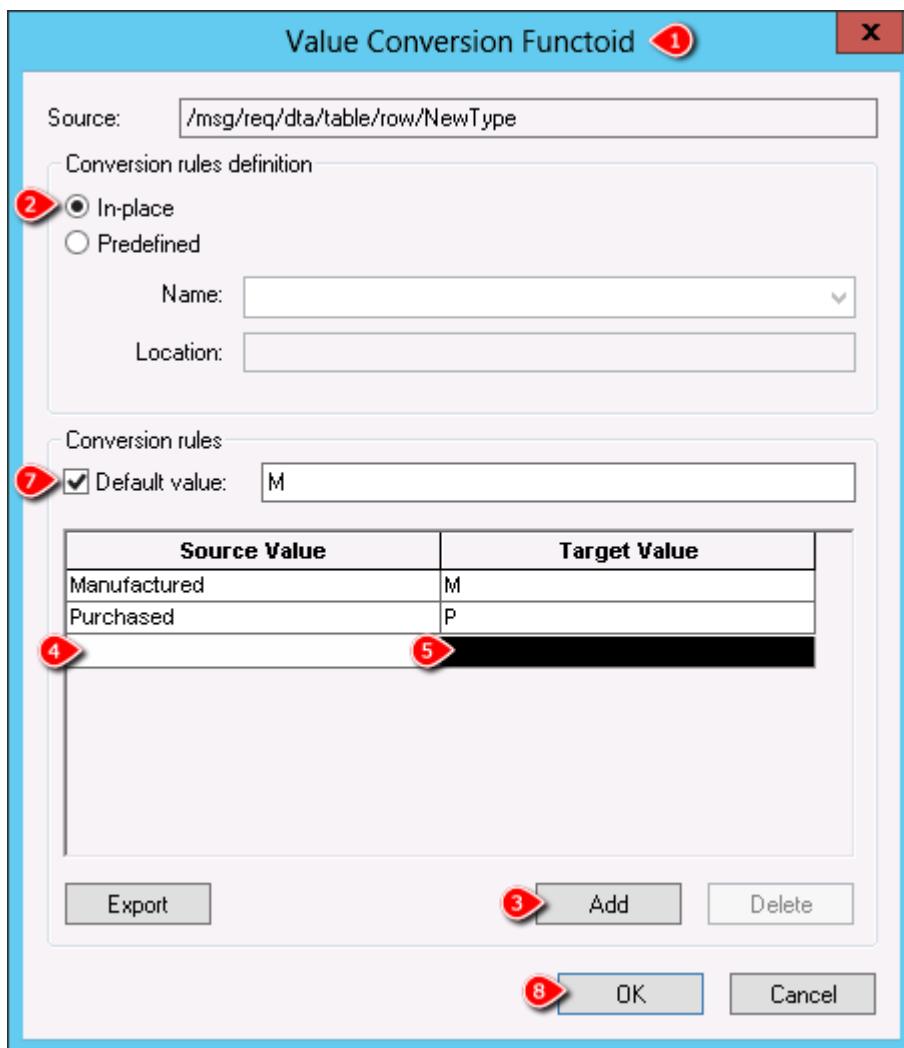
- **In place** - Rules are defined in the Value Conversion Functoid dialog box.
- **Predefined** - Rules are defined in an external XSLT transform file. You can import the XSLT file by reference into the current transform which allows the rules to be applied to the functoid.

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** for the conversion rule.
5. Enter the **Target Value** for the conversion rule.
6. Repeat these steps to add additional conversion rules.
7. 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.
8. 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:

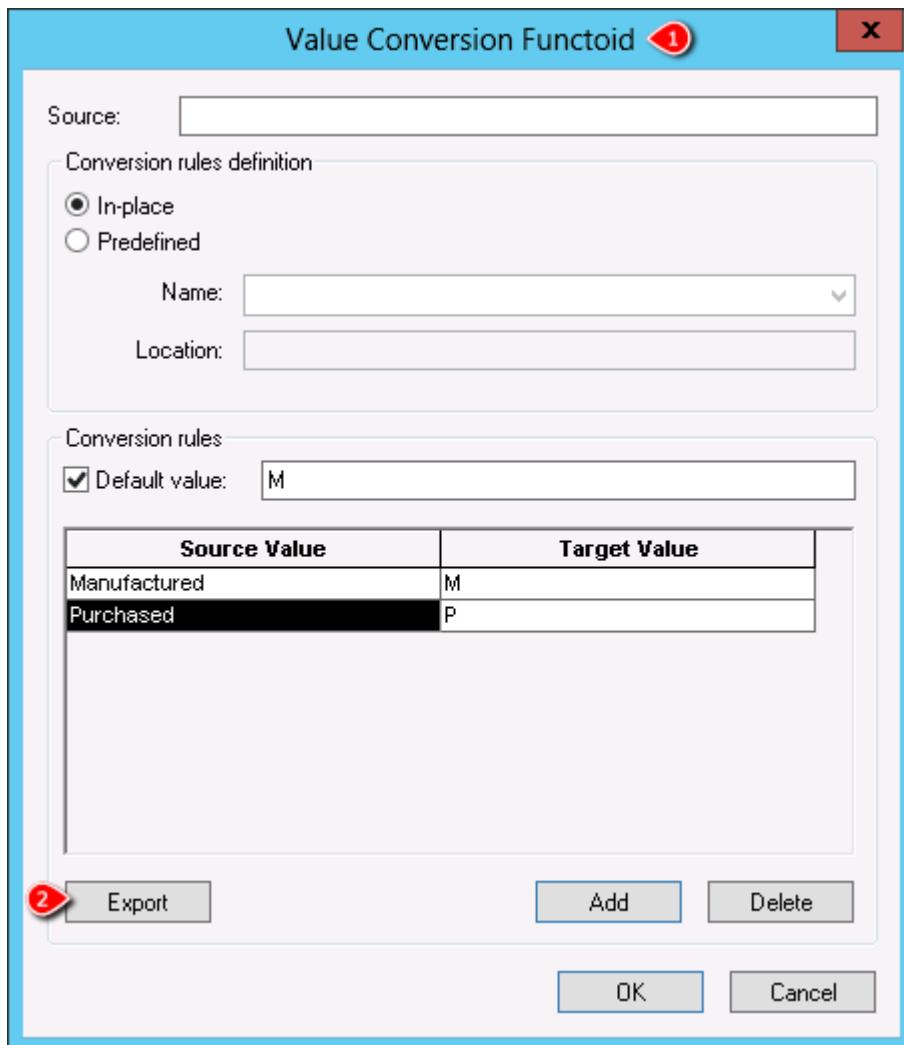
- Create the rules. You can write the rules to an XSLT file; or, you can create the rules in the Value Conversion Functoid dialog box and export them.
- Add a reference to the rules file in the XML Mapper.
- Use the rules file for the conversion functoid.

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.

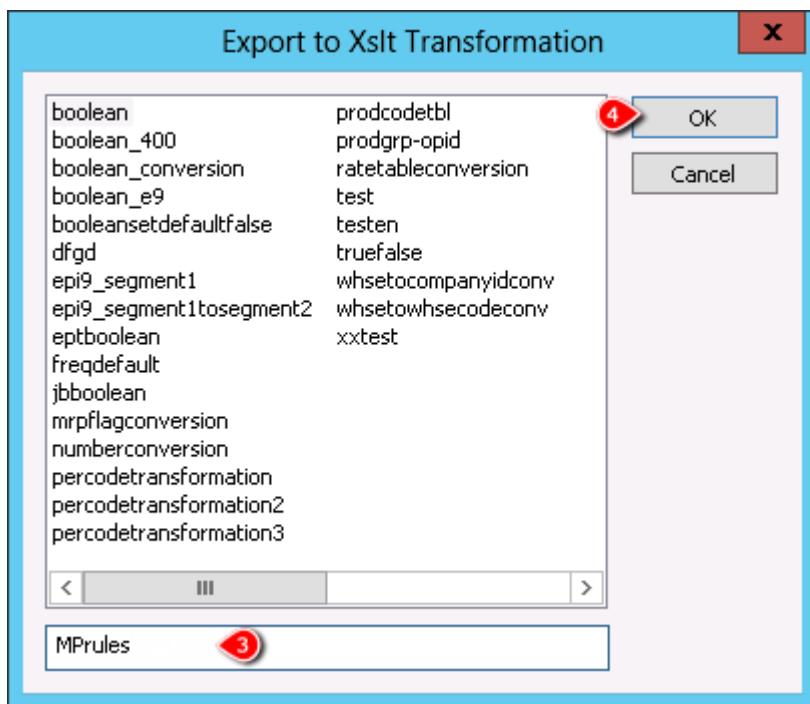
To create pre-defined conversion rules:

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**.

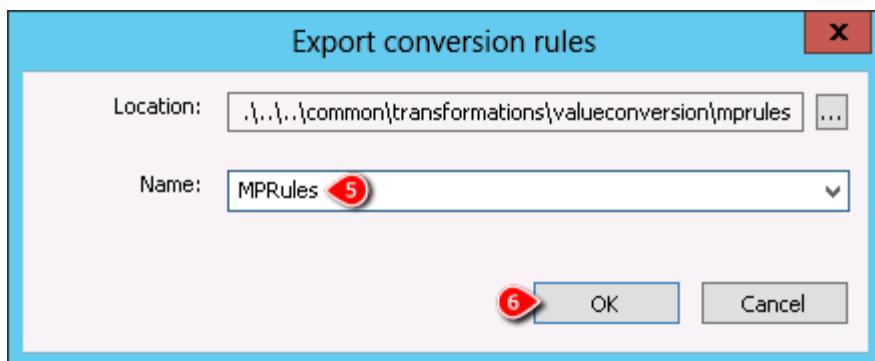
3. The **Export to Xslt Transformation** window displays. Enter a name for the transform file.



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.

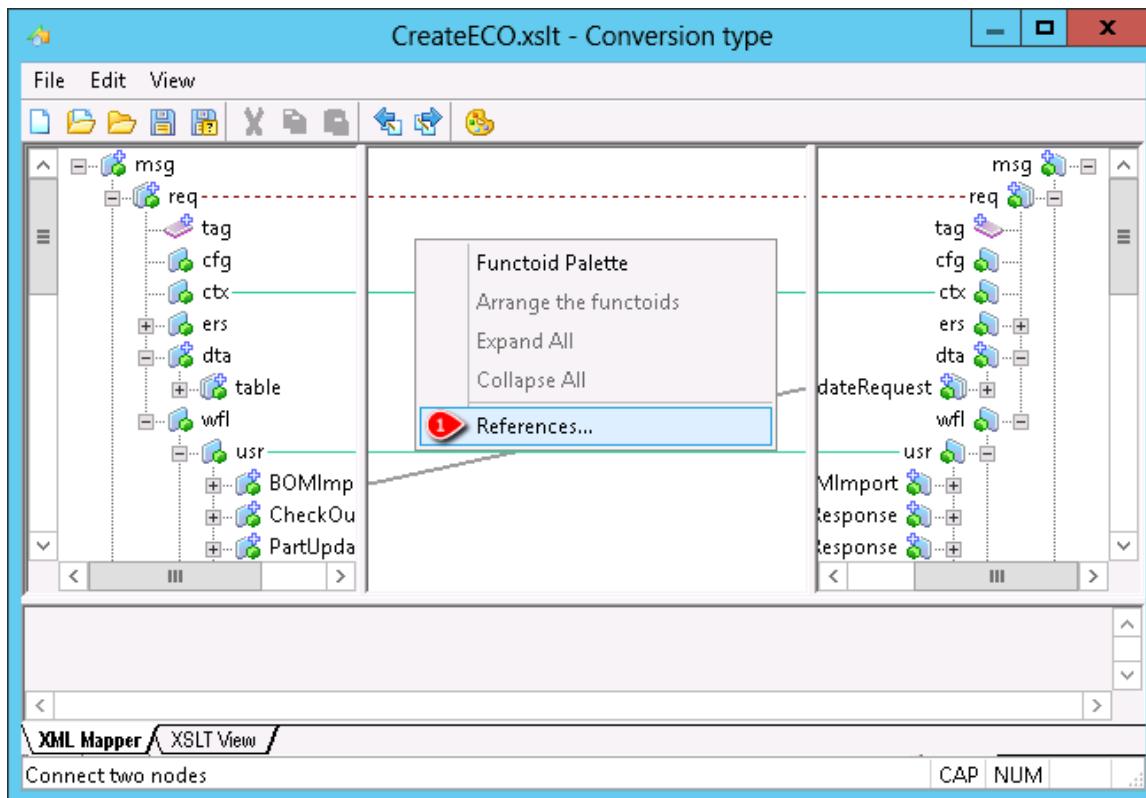
You can save more than one set of rules to the same file. You can then access all the rules by referencing just one file in the XML Mapper. Select an already existing rules file during this step to add a set of rules to it.

Add a Reference to the Rules

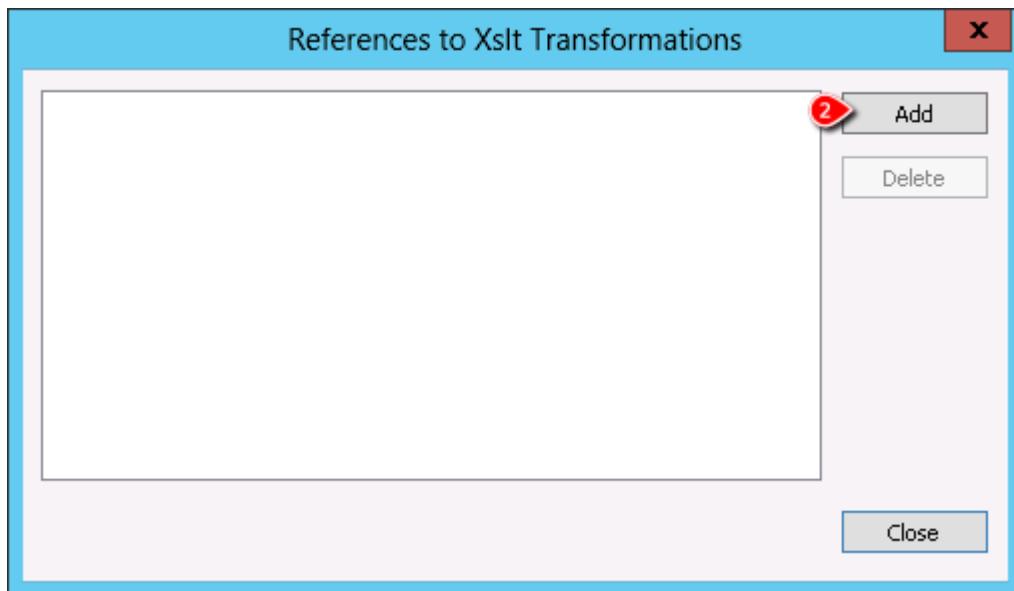
Adding a reference makes the rule sets in the rules file available for use in any conversion functoid.

To add a reference:

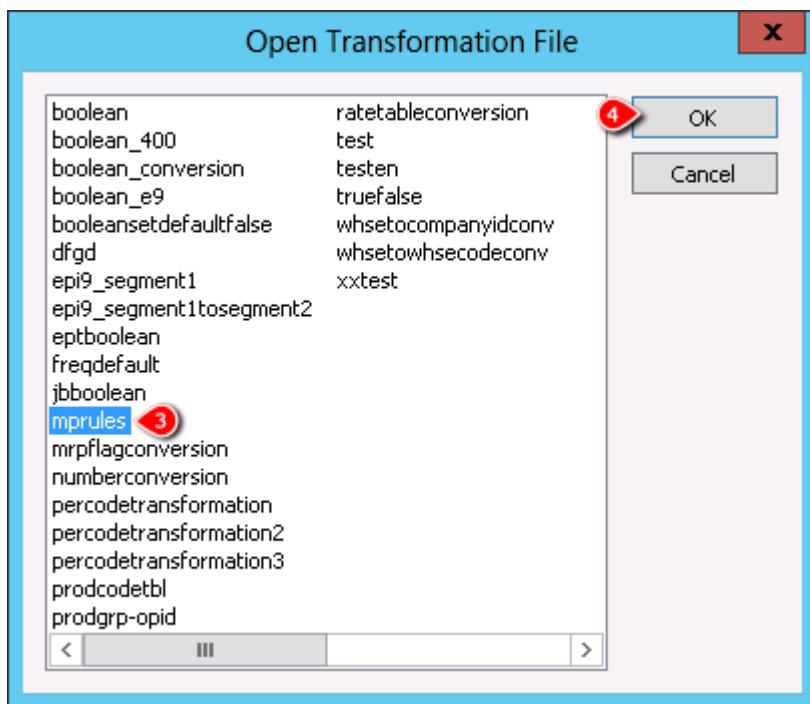
1. Right-click the center pane of the XML Mapper and select **References**.



2. The **References to Xslt Transformations** window displays. Click **Add**.

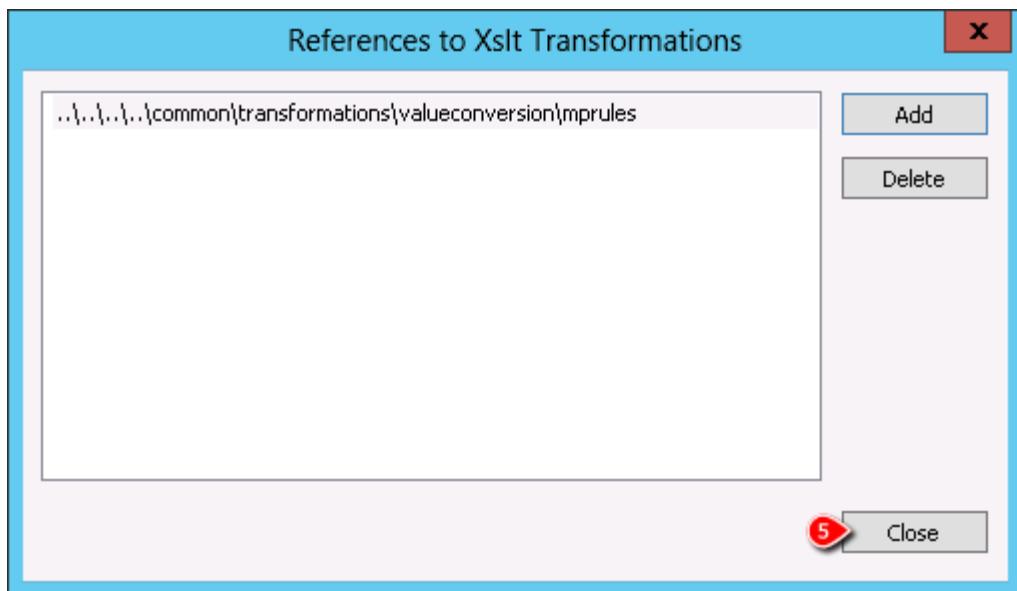


3. The **Open Transformation File** window displays. Select the conversion rules file.



4. Click **OK**.

5. The **References to Xslt Transformations** window displays. Click **Close**.

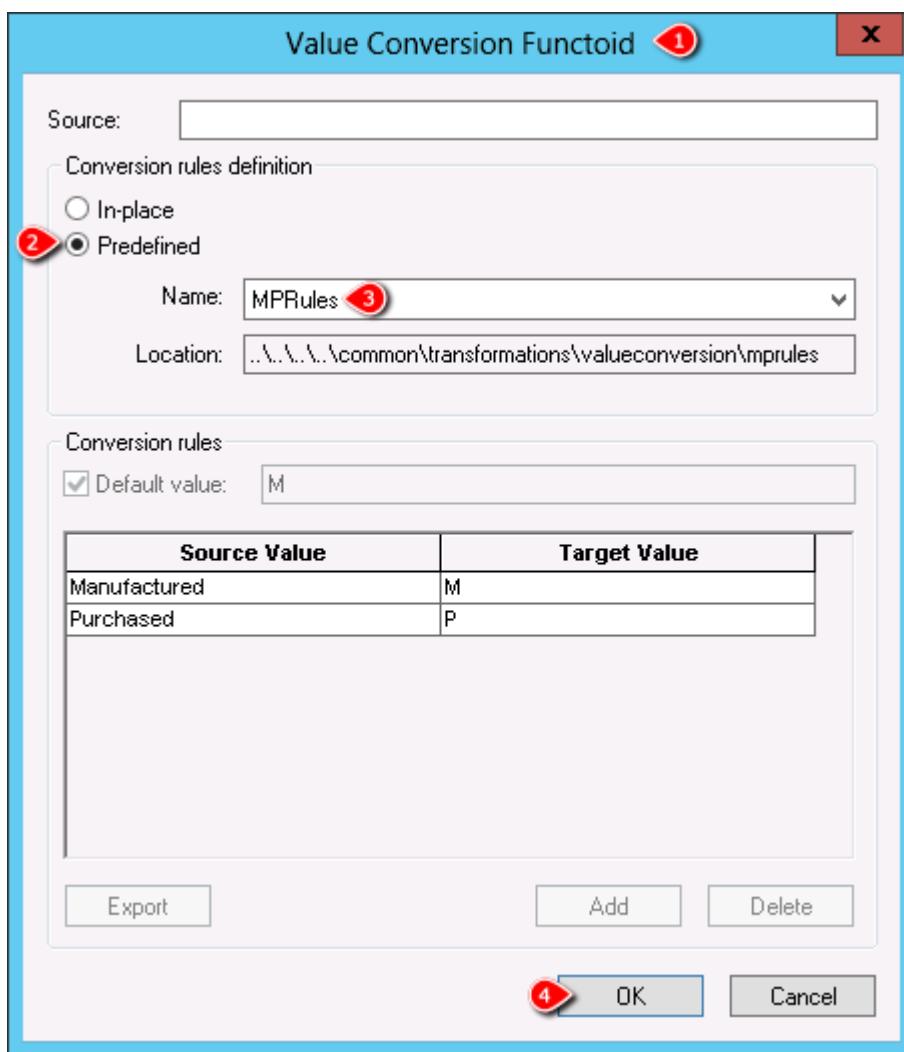


Use the Rules File for the Conversion Functoid

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:

- Double-click the functoid to open the **Value Conversion Functoid** window.



- In the **Conversion rules definition** section, select **Predefined**.
- Select the **Name** of the rule set you want to use.
- 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.

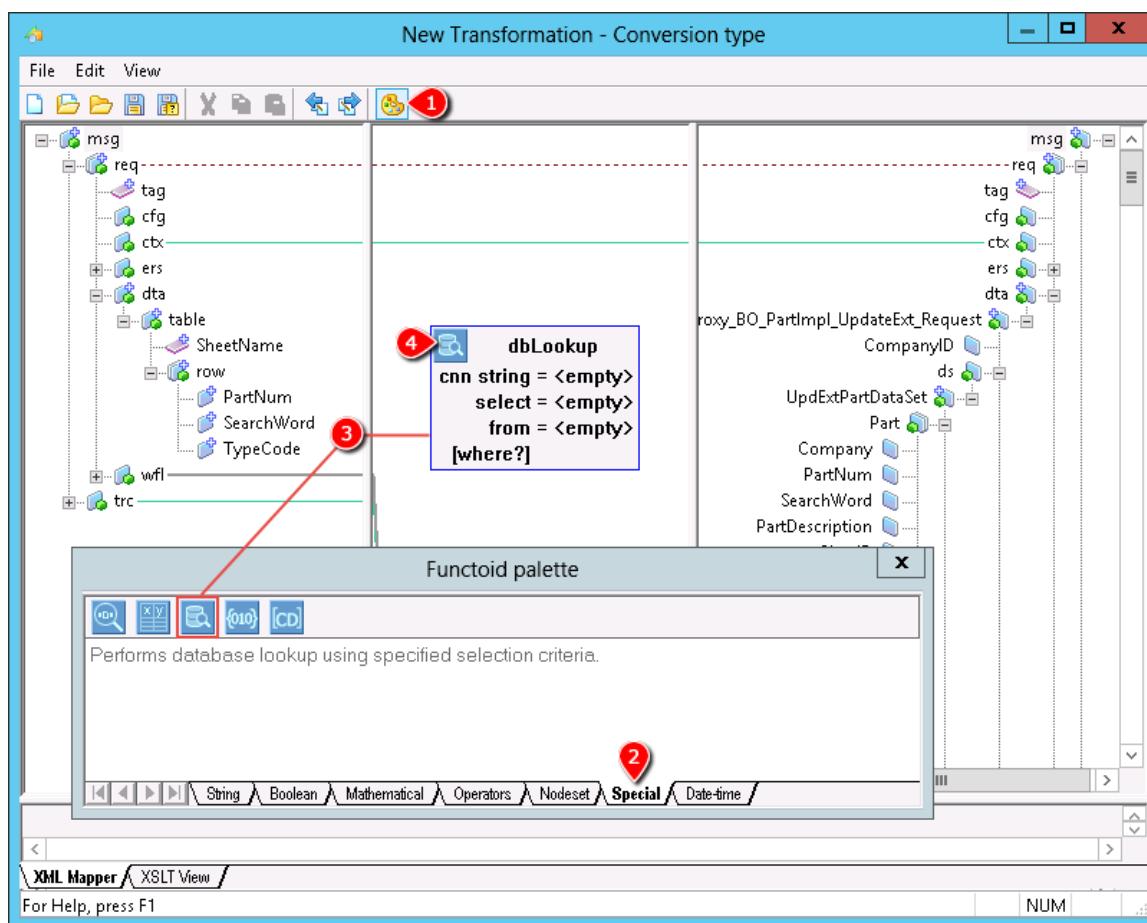
DB Lookup Functoid Example

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.

Use DB Lookup Functoid

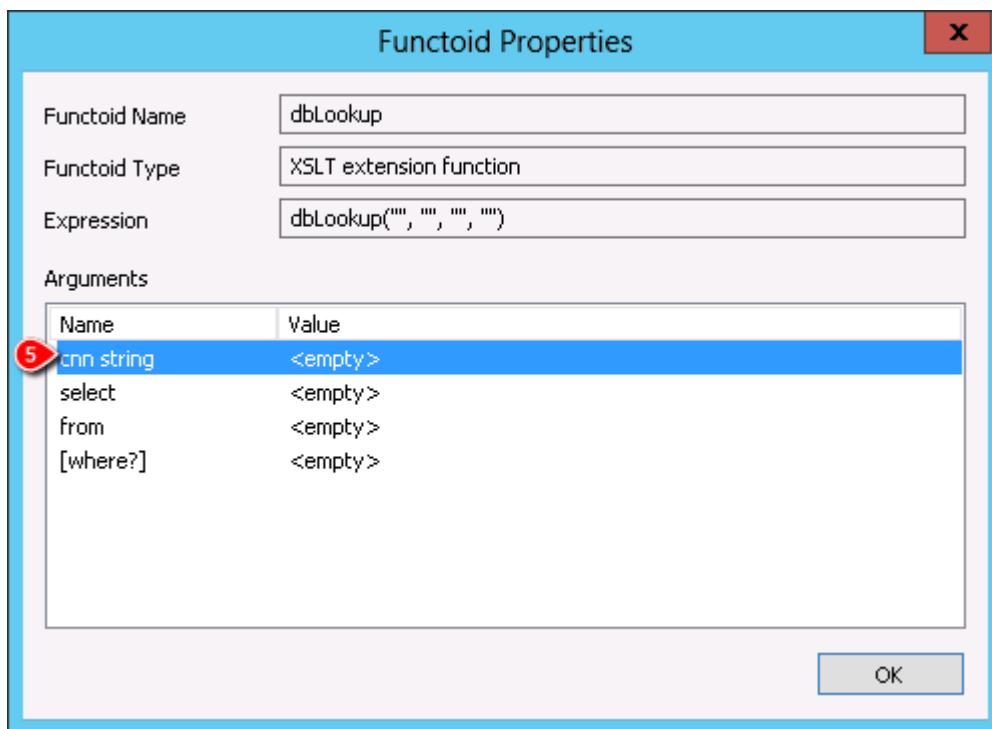
To add and configure the DB Lookup functoid:

1. On the XML Mapper standard toolbar, click the **Functoid Palette** button.

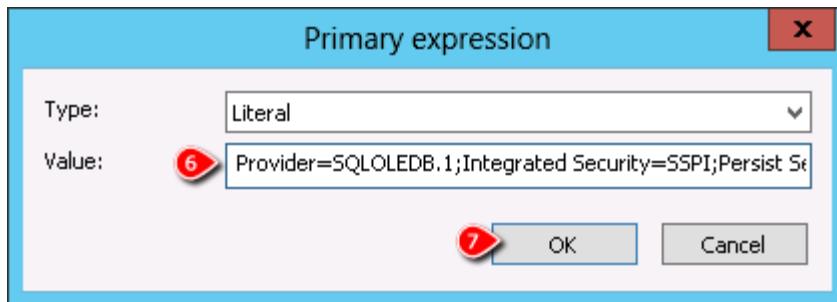


2. Open the **Special** tab.
3. Add the **dbLookup** functoid to the XML Mapper middle pane.
4. Double-click the functoid to open the **Functoid Properties** window.

5. In the **Arguments** list, double-click **cnn string** to establish the connection to the database.



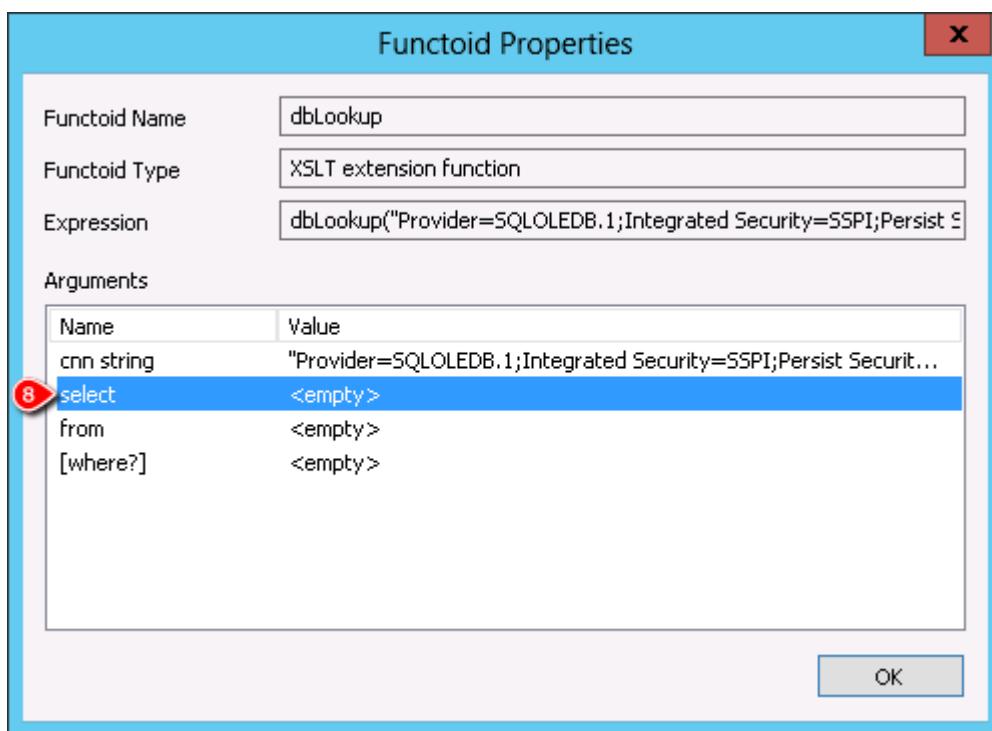
6. In the **Value** field, enter the connection string.



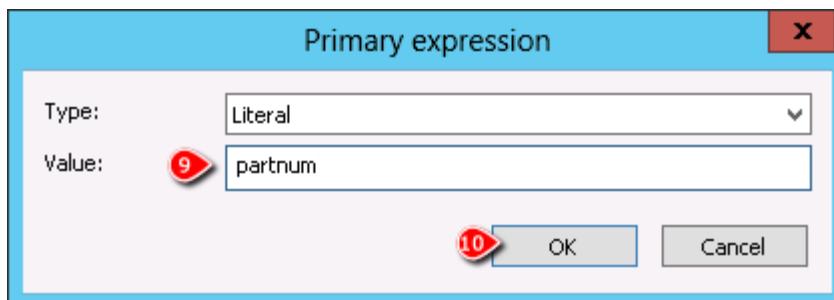
7. Click **OK**.

For more information on how to build a connection string, review the DB Operation section earlier in this chapter.

8. In the **Arguments** list, double-click **select**.



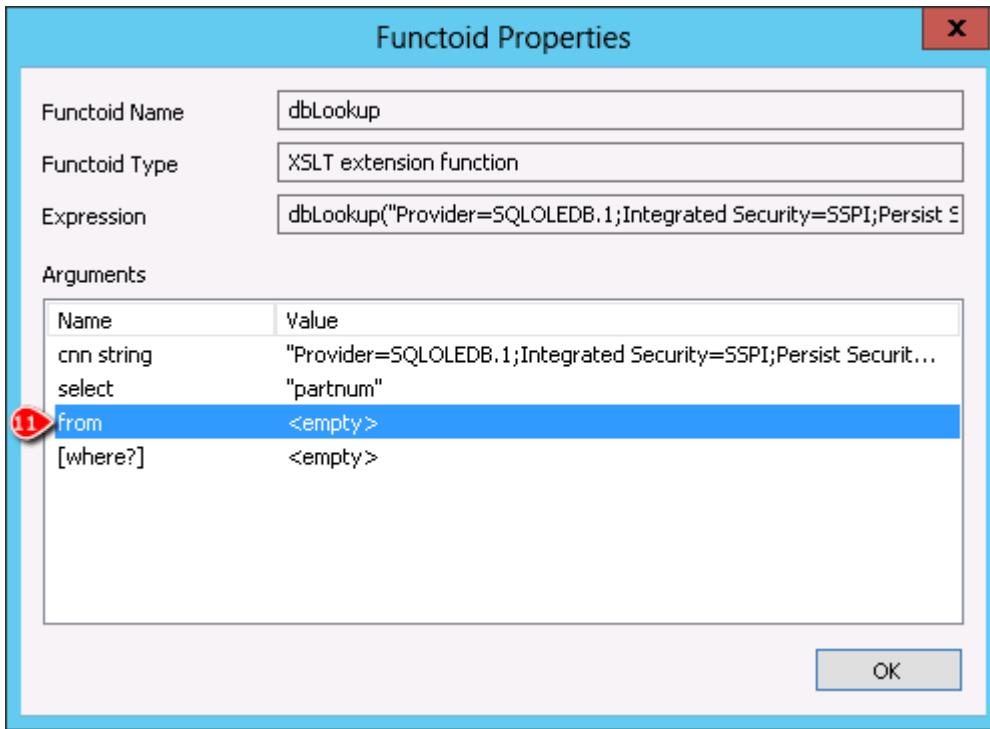
9. Specify the **Value** you want to select from the database.



In this example, the partnum column is selected.

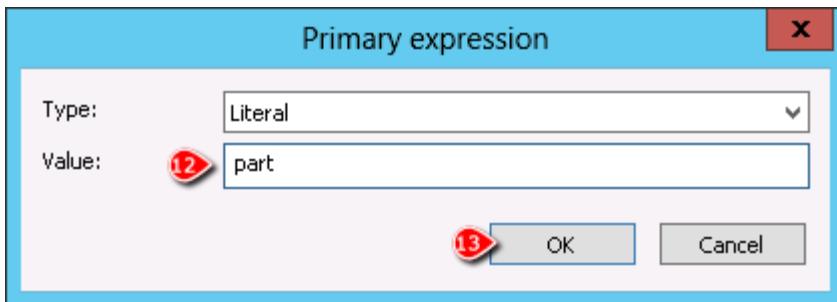
10. Click **OK**.

11. In the Arguments list, double-click **from**.



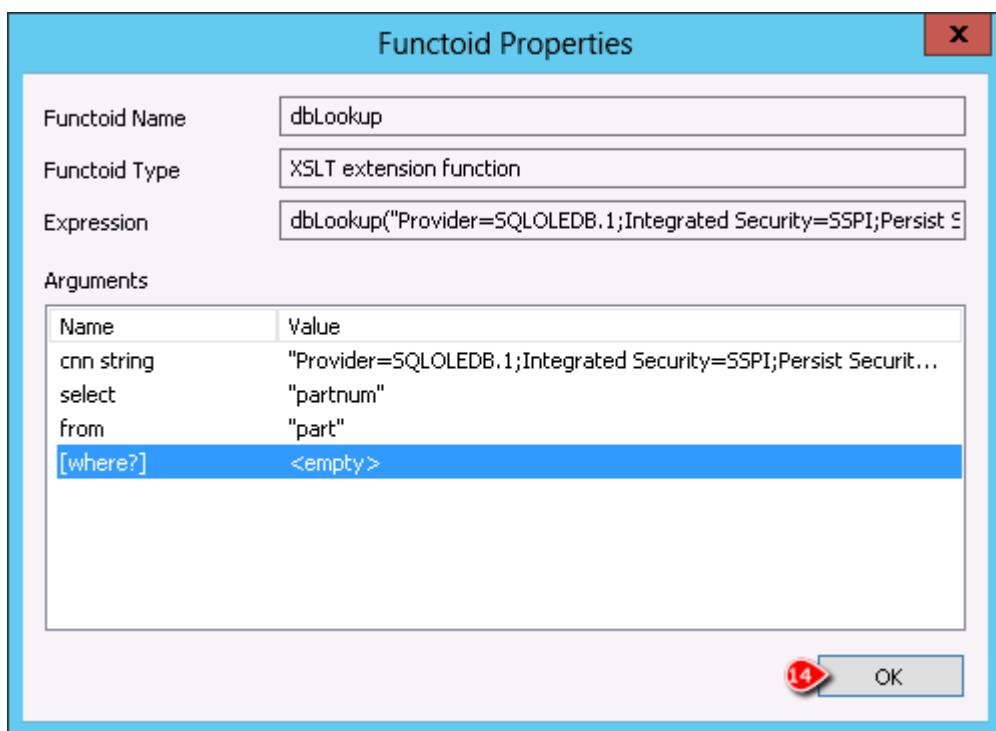
12. In the **Value** field, enter the table from which you want to select data.

In this example, the part table is selected.



13. Click **OK**.

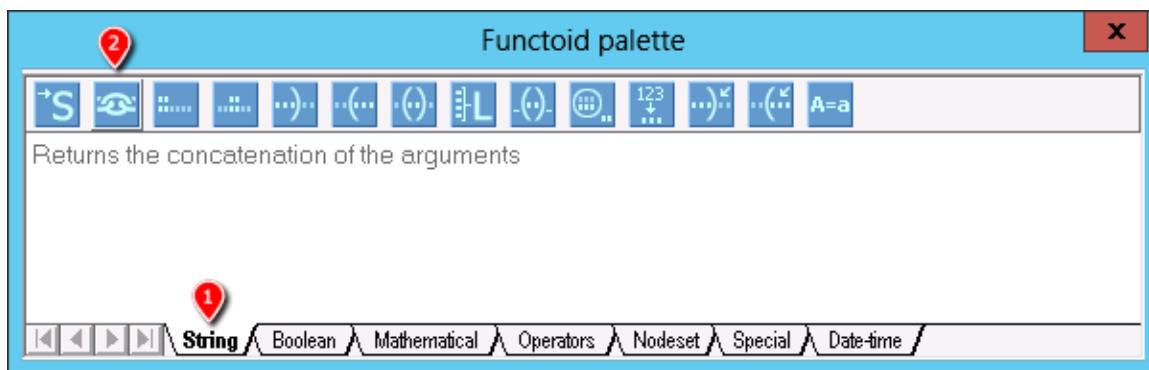
14. Click **OK** to close the Functoid Properties window.



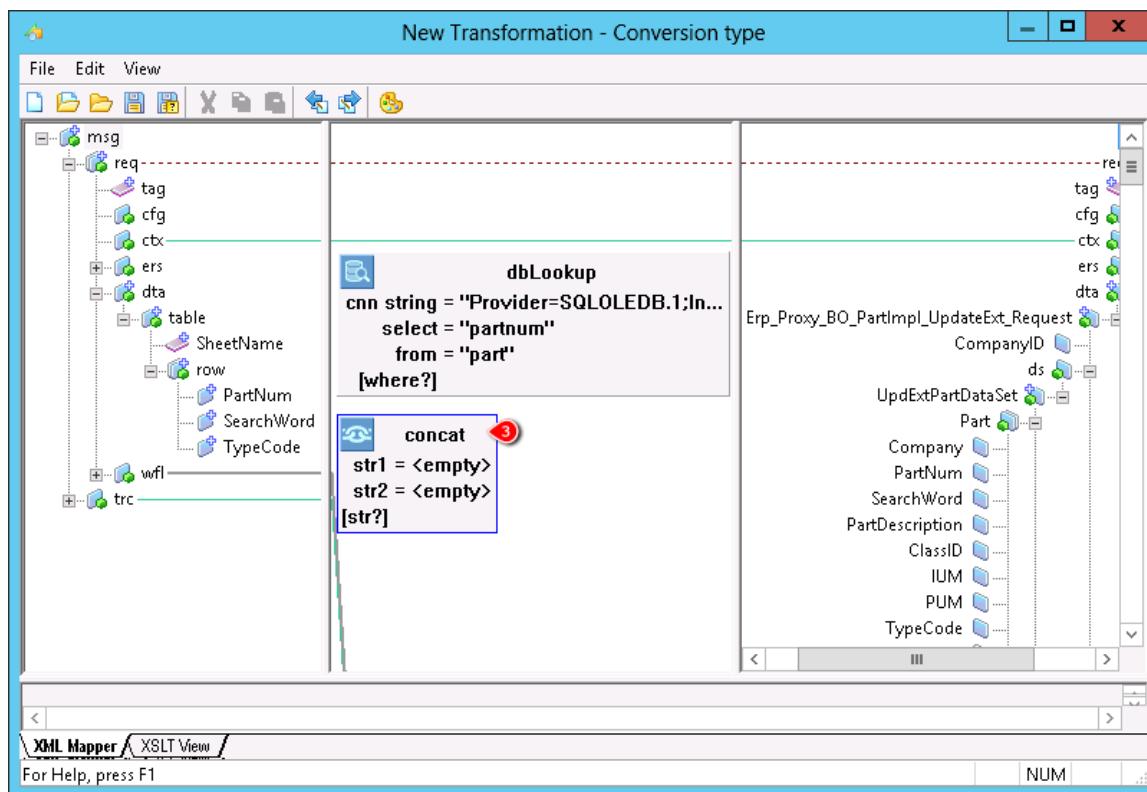
Next, use another functoid to generate the where expression.

Generate the WHERE Expression

1. Open the **String** tab of the Functoid Palette.

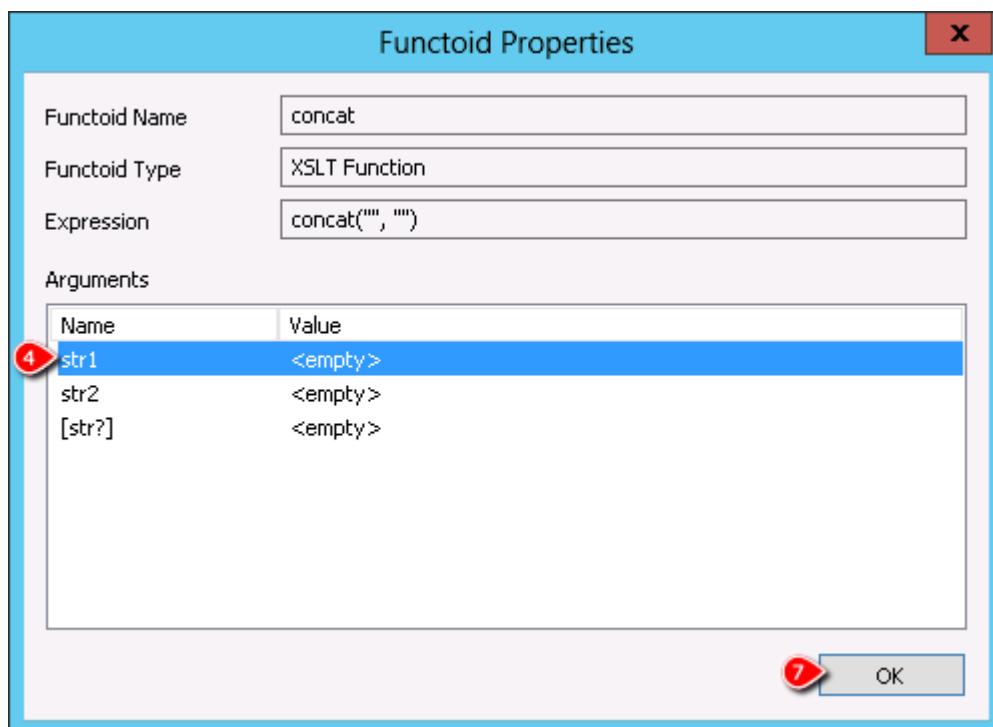


2. Add the **concat functoid** to the XML Mapper middle pane.

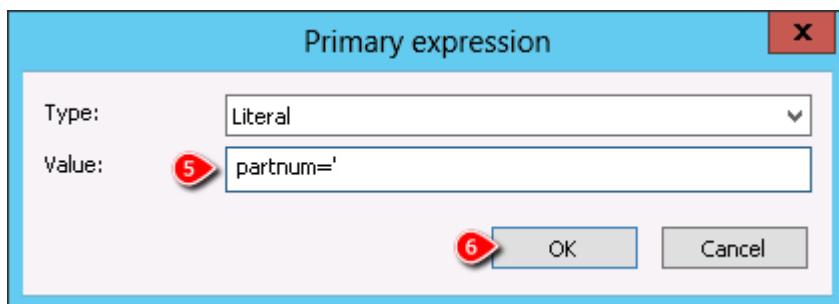


3. Double-click the functoid to open the **Functoid Properties** window.

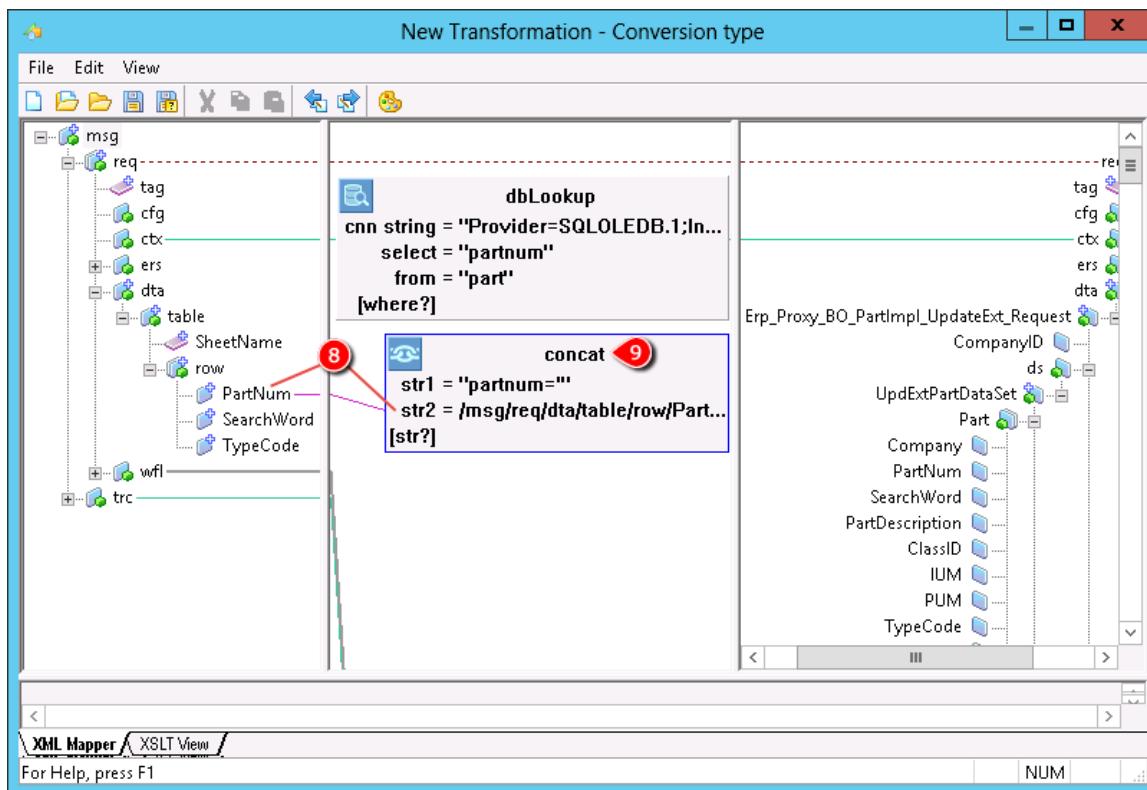
4. In the **Arguments** list, double-click **str1**.



5. In the **Value** field, enter the first argument.

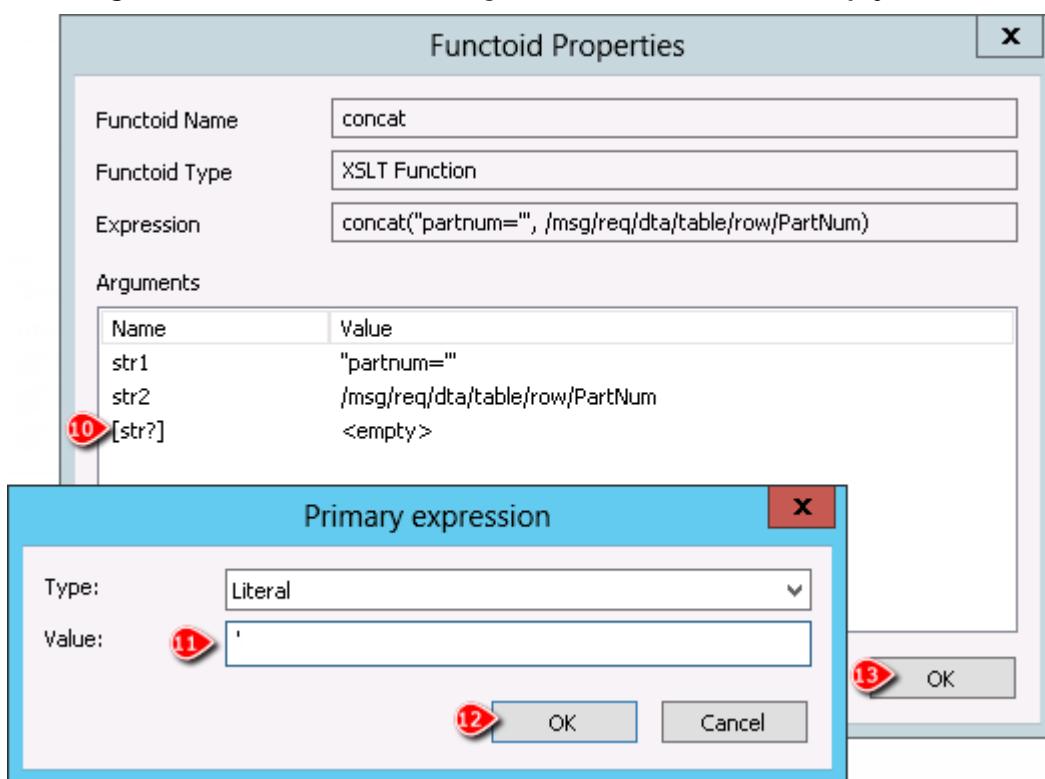


6. Click **OK** to close the Primary expression window.
 7. Click **OK** to close the Functoid Properties window.
 8. Map the **PartNum** node in the incoming document to the **str2** value in the functoid.



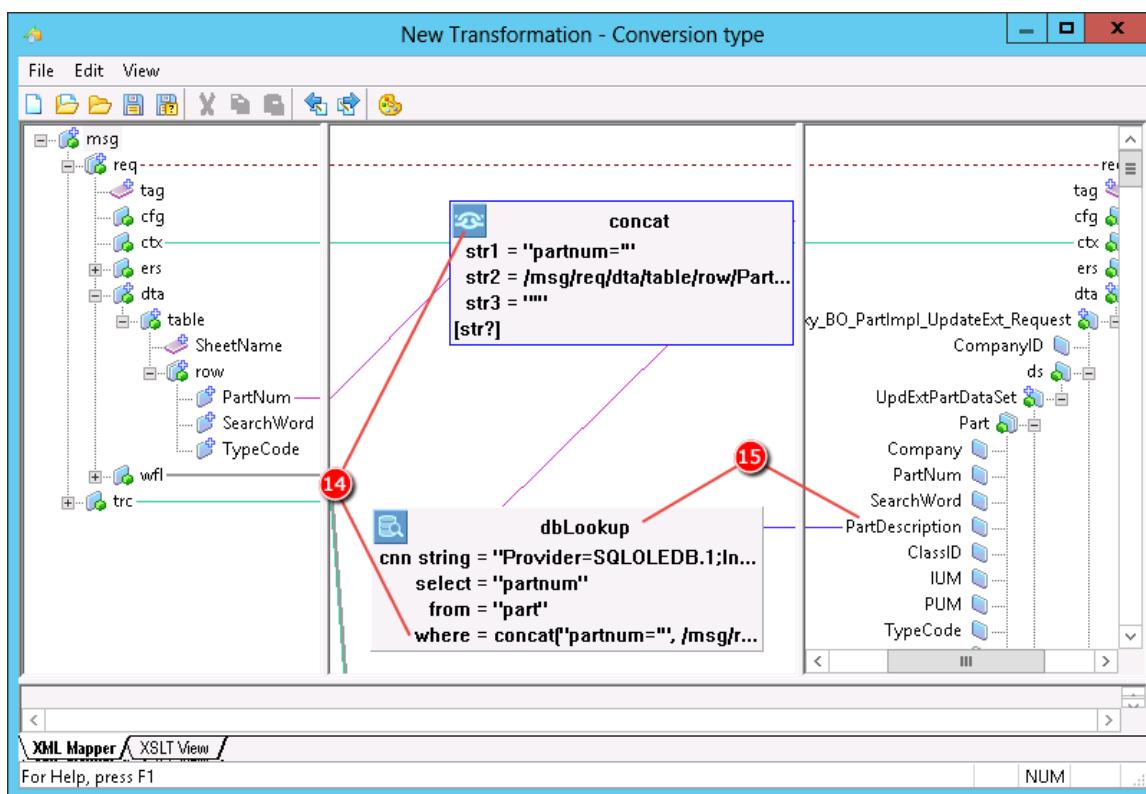
9. Double-click the functoid to open the **Functoid Properties** window.

10. In the **Arguments** list, double-click the argument where the value is <empty>.



11. Finish the statement.
12. Click **OK** to close the Primary expression window.
13. Click **OK** to close the Functoid Properties window.

14. Map the **concat** functoid to the **where** argument of the **dbLookup** functoid.



15. 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.

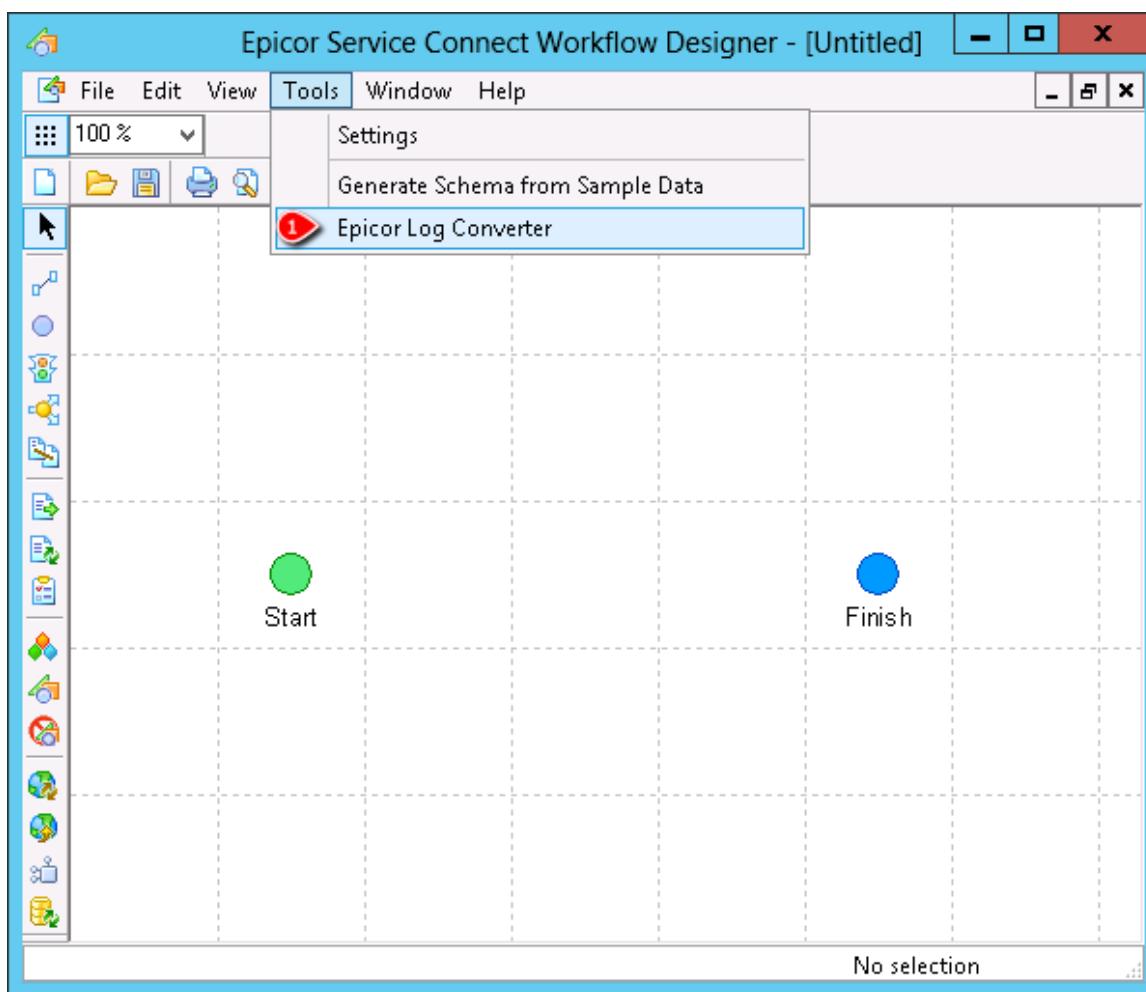
The parser searches for the following sections in the Epicor log file:

- <businessObject>BOName</businessObject>
- <methodName>MethodName</methodName>

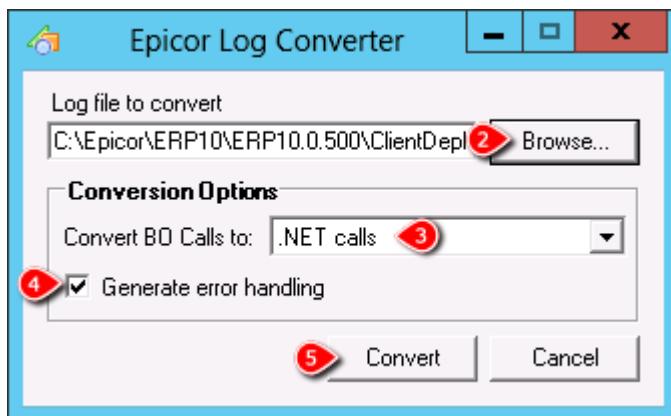
The parser then creates a sample workflow where the found methods are transformed sequentially into the Web Service or .NET call components with or without error handling and Conversions in between. The Web Service Request ID is formed as BOName.MethodName.

Convert an Epicor Log File

- From the **Tools** menu, select **Epicor Log Converter**.



- Click the **Browse** button to find and select the Epicor log file you want to convert.



- Click the **Convert BO Calls** drop-down list to select if you want to convert BO calls to **Web method calls** or **.NET calls**.

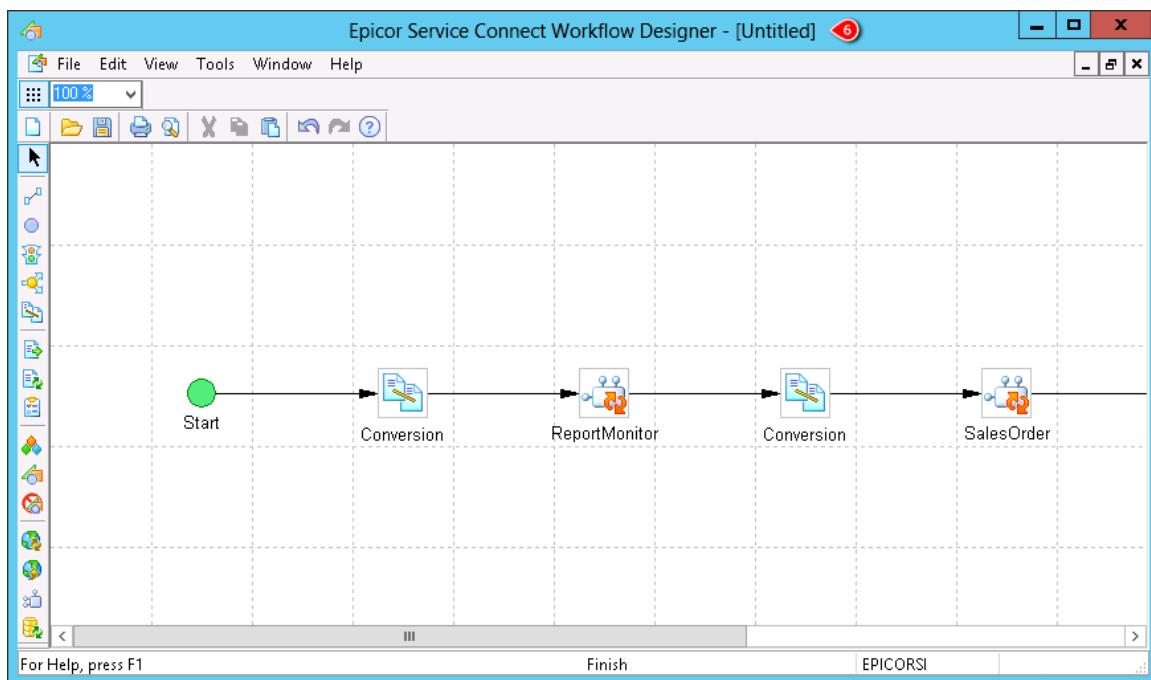
4. If you want errors to be handled, select the **Generate error handling** check box.

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**.

6. 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 width 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 elements.

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.

Service Connect only processes Excel files without protection.

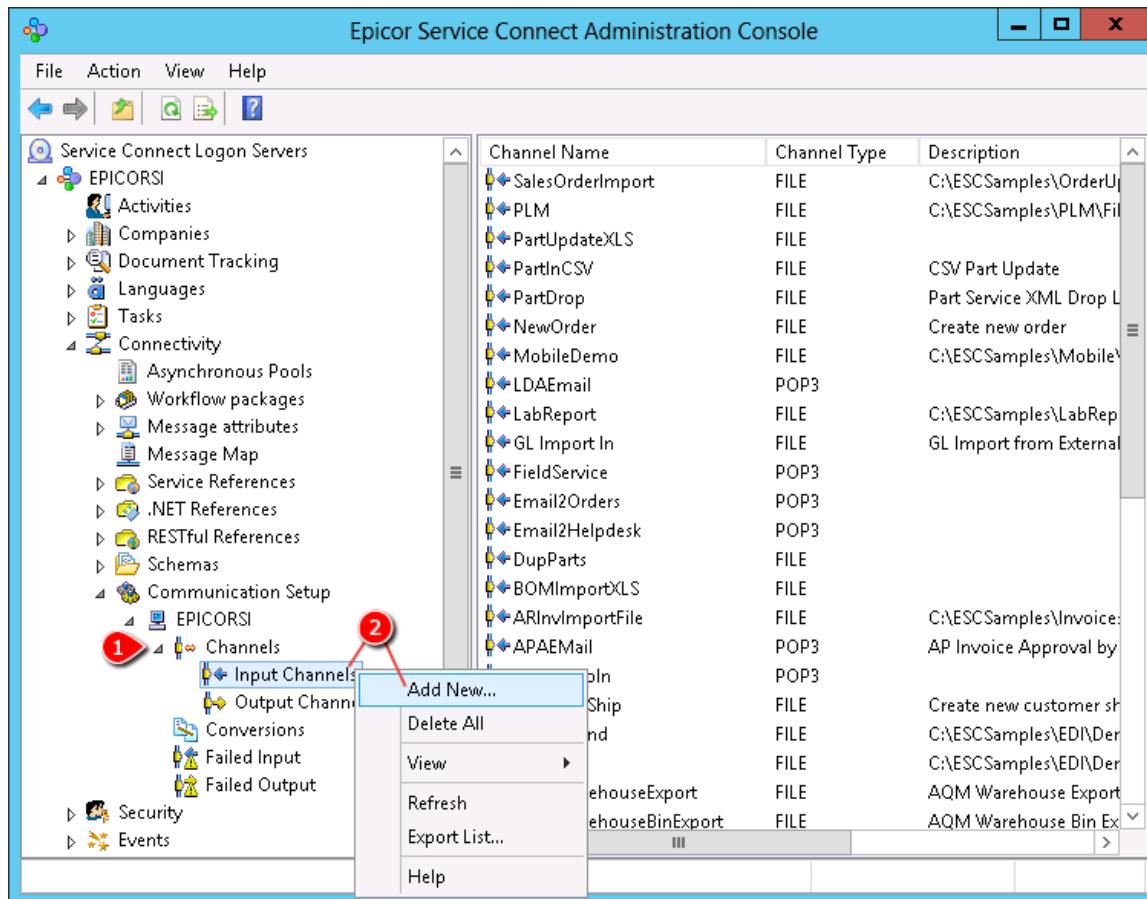
Sample Data Example

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

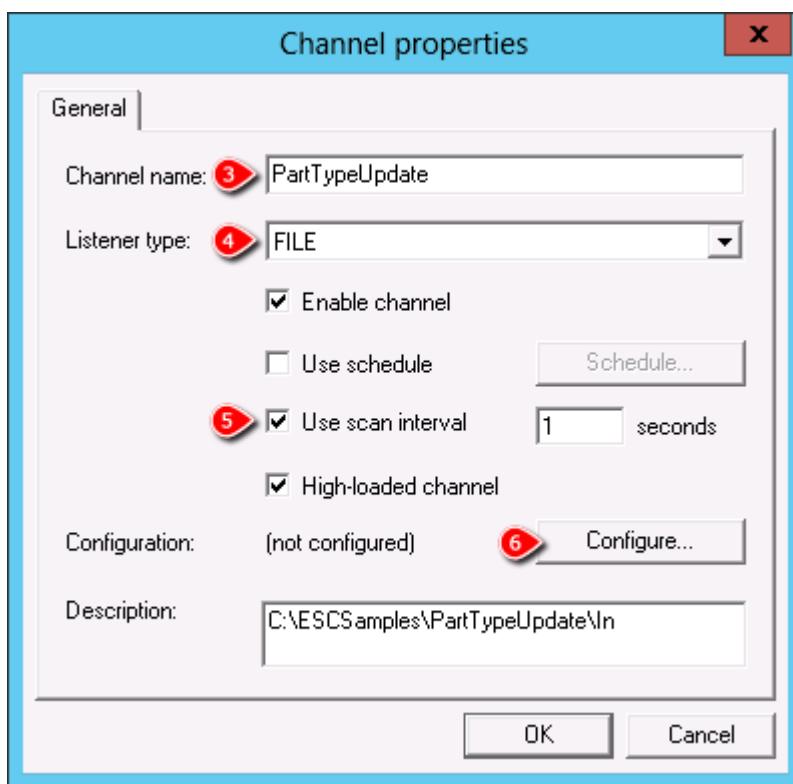
To configure the Input Channel:

1. In the Epicor Service Connect Administration Console **tree view**, expand the **Connectivity > Communication Setup > <Server Name> > Channels** nodes.

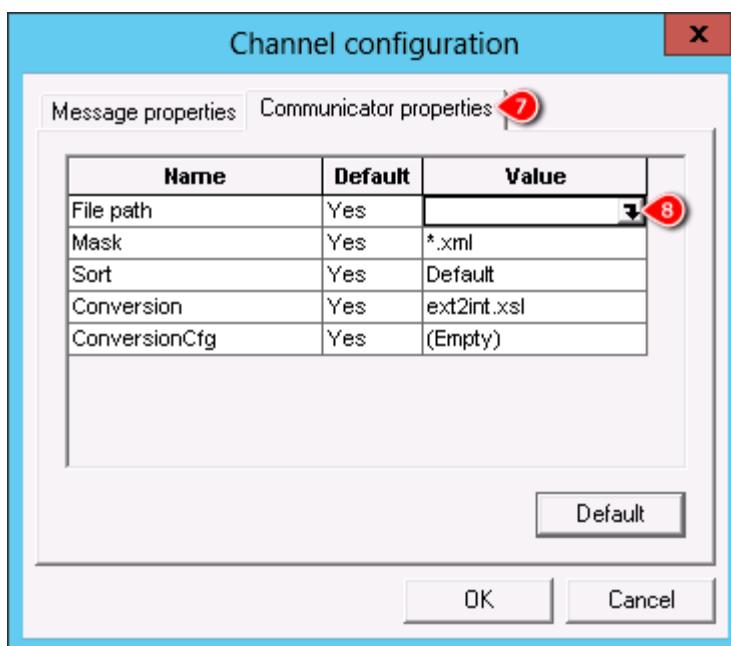


2. Right-click **Input Channels**, and select **Add New...**.

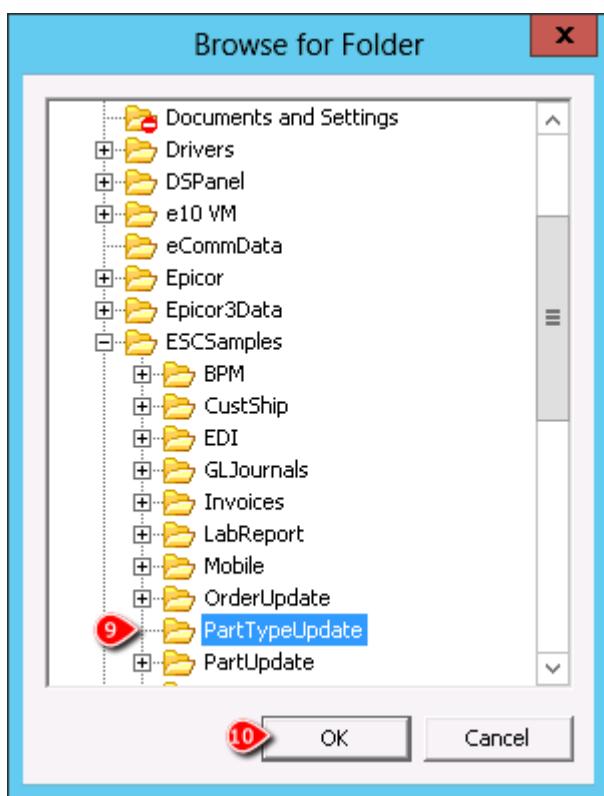
3. The **Channel properties** window displays. In the **Channel name** field, enter **PartTypeUpdate**.



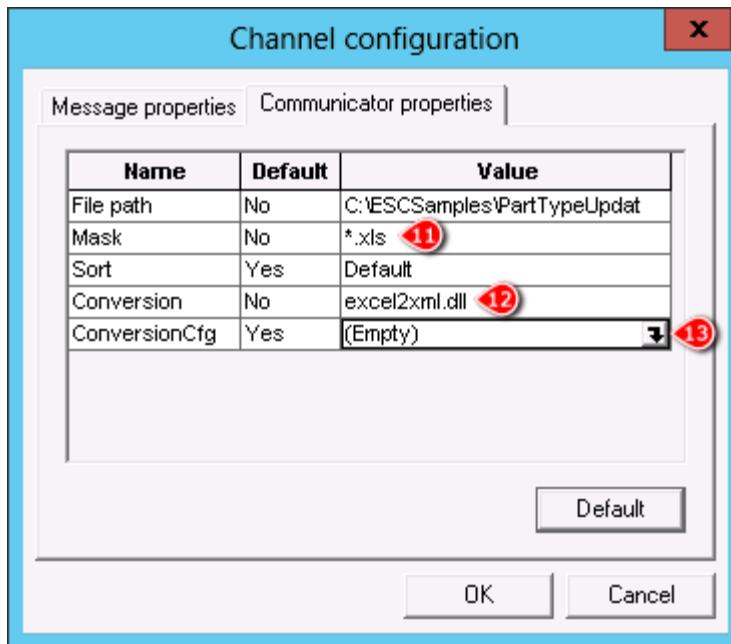
4. In the **Listener type**, select **FILE**.
5. Select the **Use scan interval** check box.
6. Click **Configure**.
7. The **Channel configuration** window displays. Open the **Communicator properties** tab.



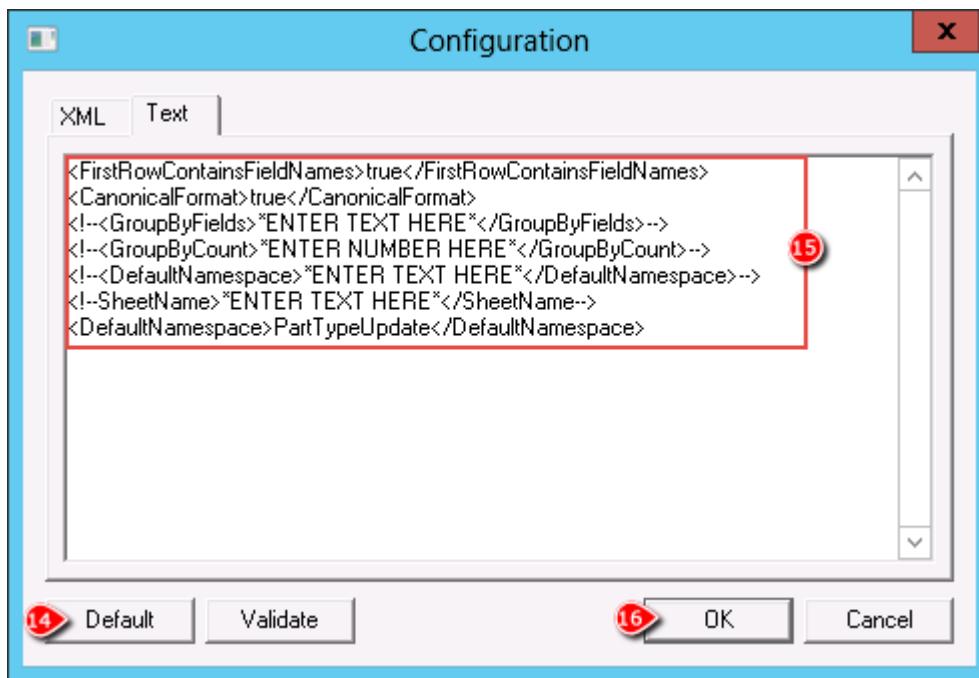
8. In the **File path** row, click the **Value** column, then click the **down arrow** button.
9. In the **Browse for Folder** window, navigate to the folder where the spreadsheet is stored and from which the channel will consume the files.



10. Click **OK**.
11. Verify the **Mask** field displays ***.xls**.



12. Verify the **Conversion** field displays **excel2xml.dll**.
13. In the **ConversionCfg** row, click the **Value** column, then click the **down arrow** button.
14. The **Configuration** window displays. On the **Text** tab, click the **Default** button.



15. After the default configuration, add the following line:

<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.

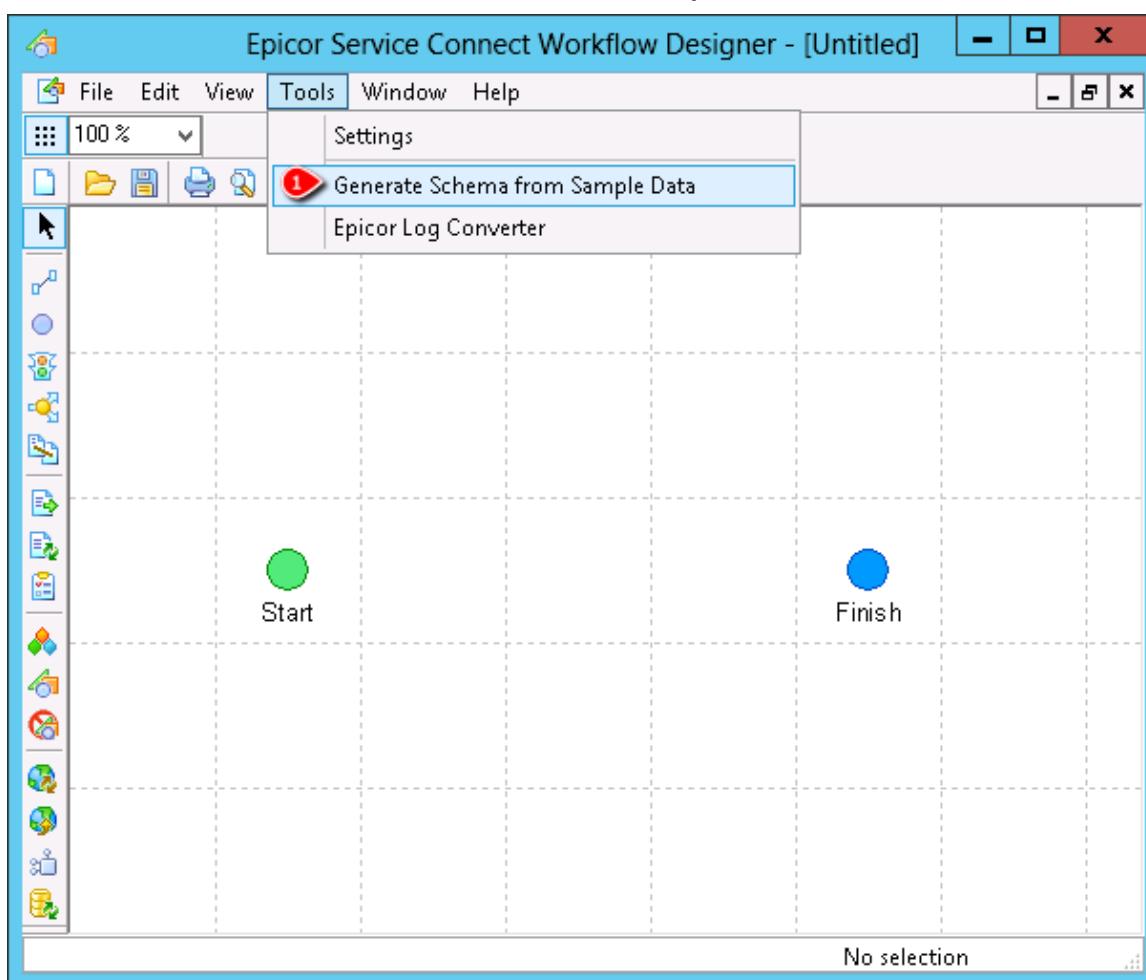
Each column header in the spreadsheet becomes an element beneath the row element. The maxOccurs attribute is set to unbounded for the row element. Thus, Service Connect expects the incoming spreadsheet to have one or more rows of data. Also, the TargetNamespace attribute contains the namespace entered during the configuration step.

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

Generate a Schema

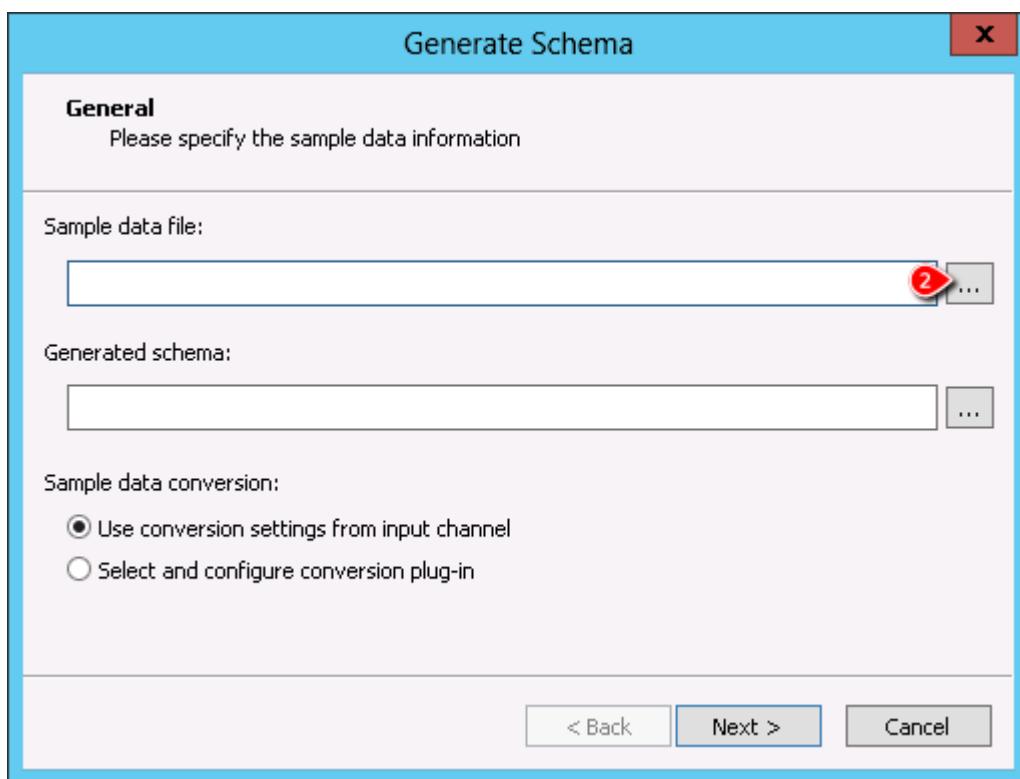
To generate a schema for the sample Excel file:

- From the **Tools** menu, select **Generate Schema from Sample Data**.

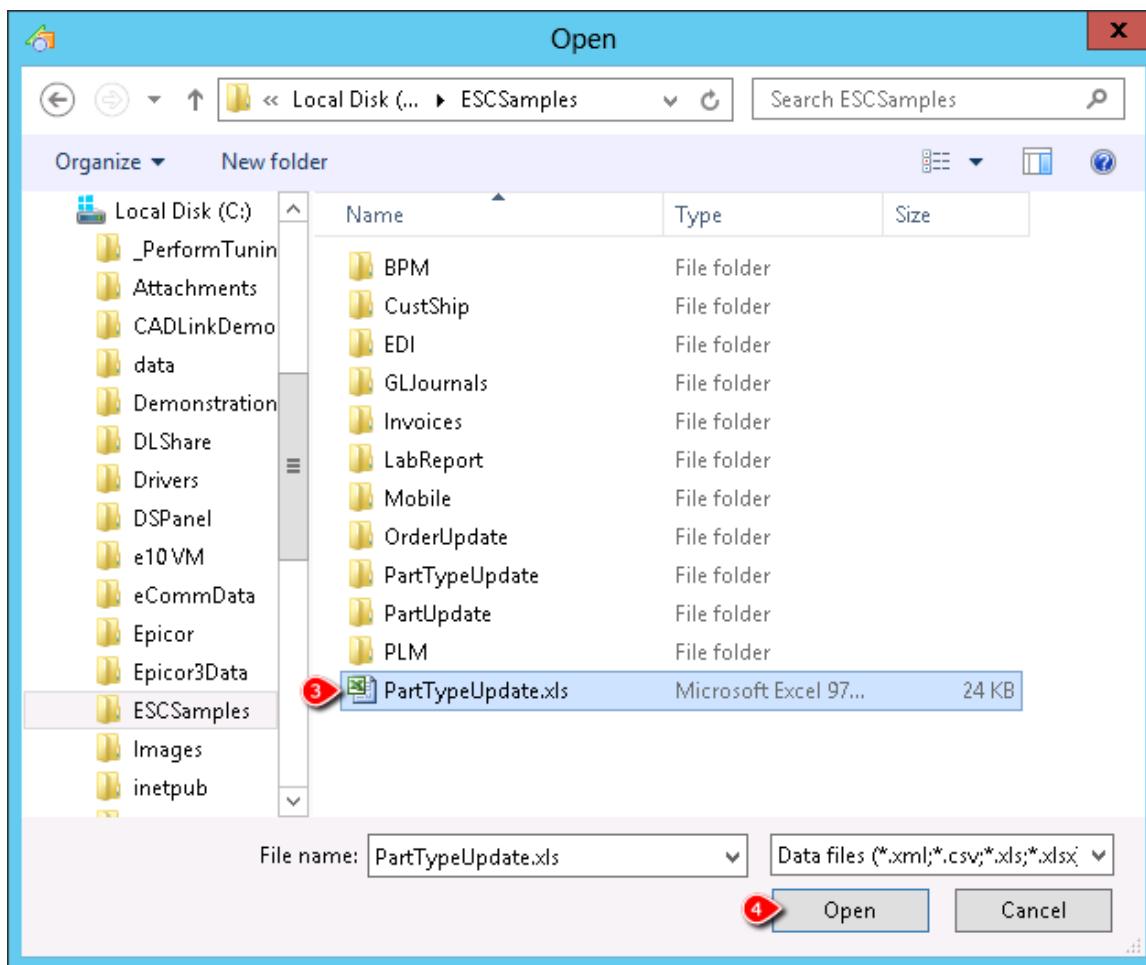


The **Generate Schema** window displays.

2. Next to the **Sample data file** field, click the (Ellipse) button.

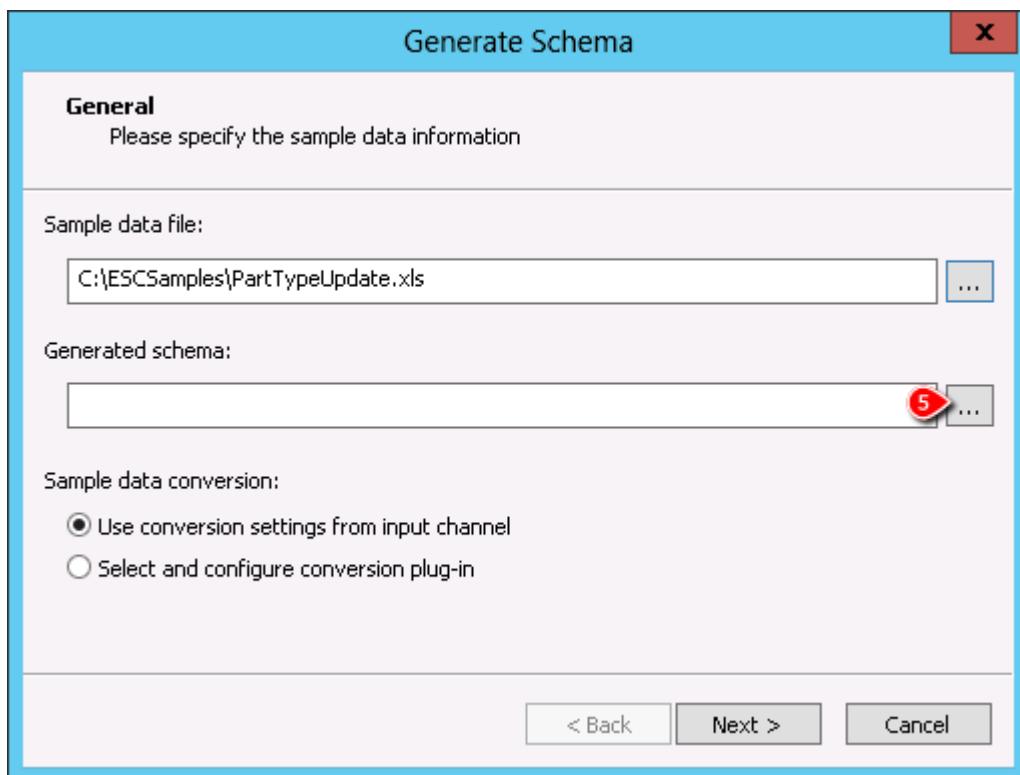


3. The **Load Sample File** window displays. Find and select the Excel file.

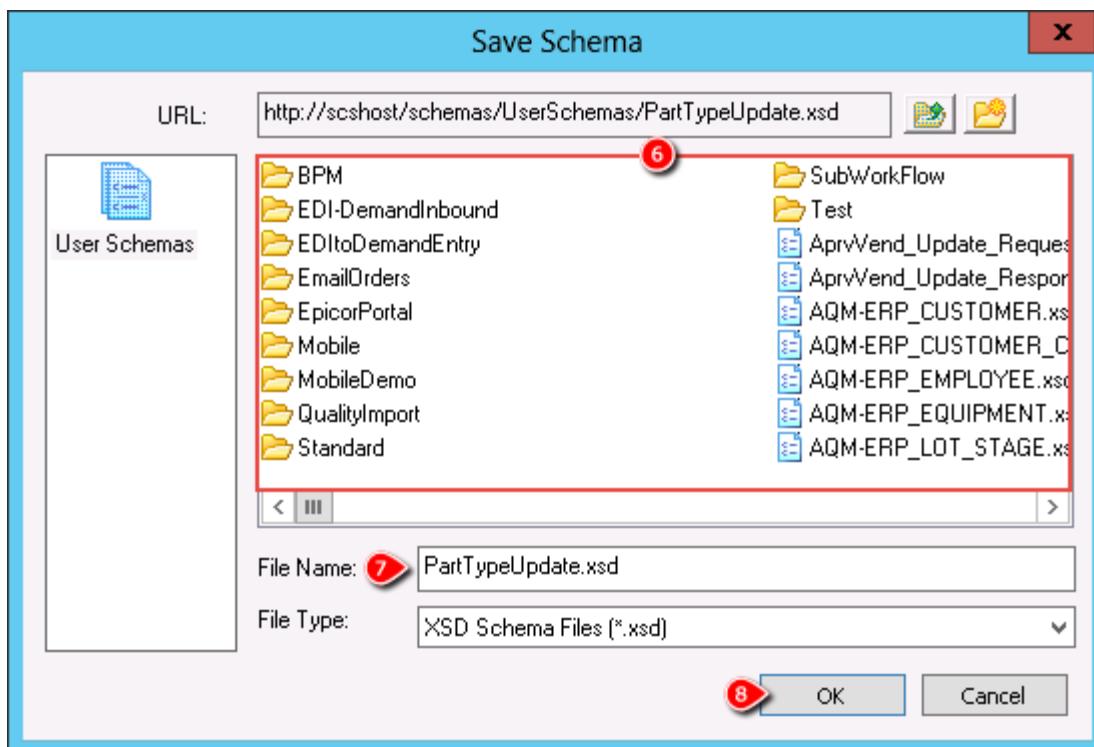


4. Click **Open**.

5. On the **Generate Schema** window, next to the Generated schema field, click the **Browse** button.

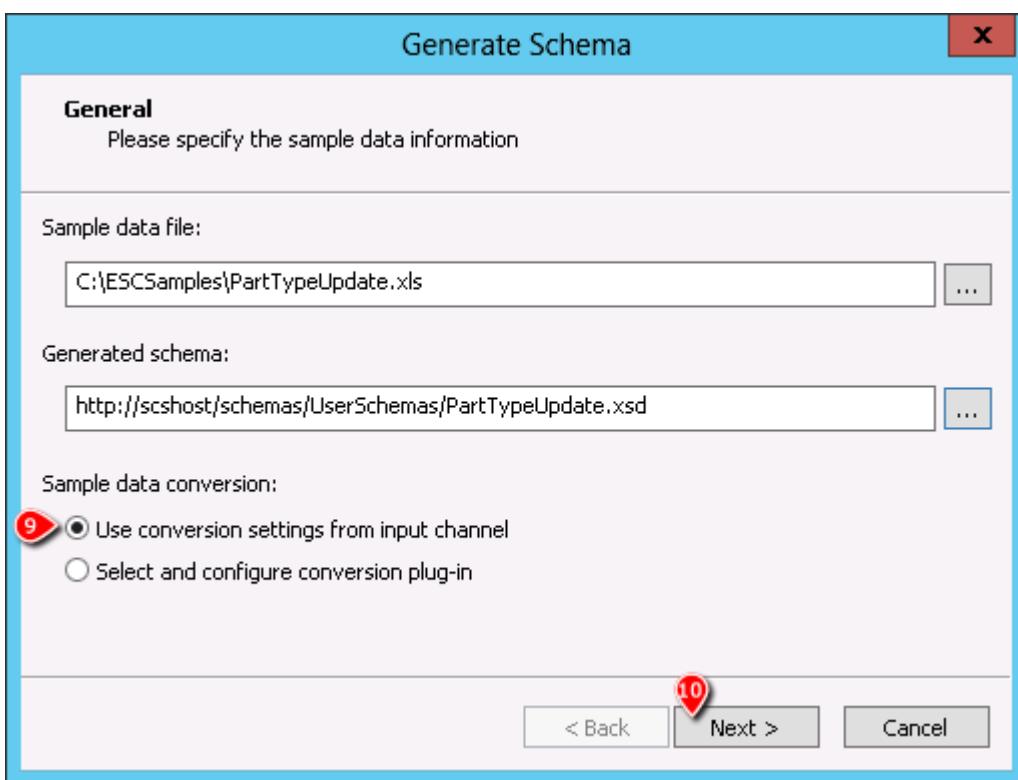


6. In the **Save Schema** window, navigate to the folder where to save the schema.



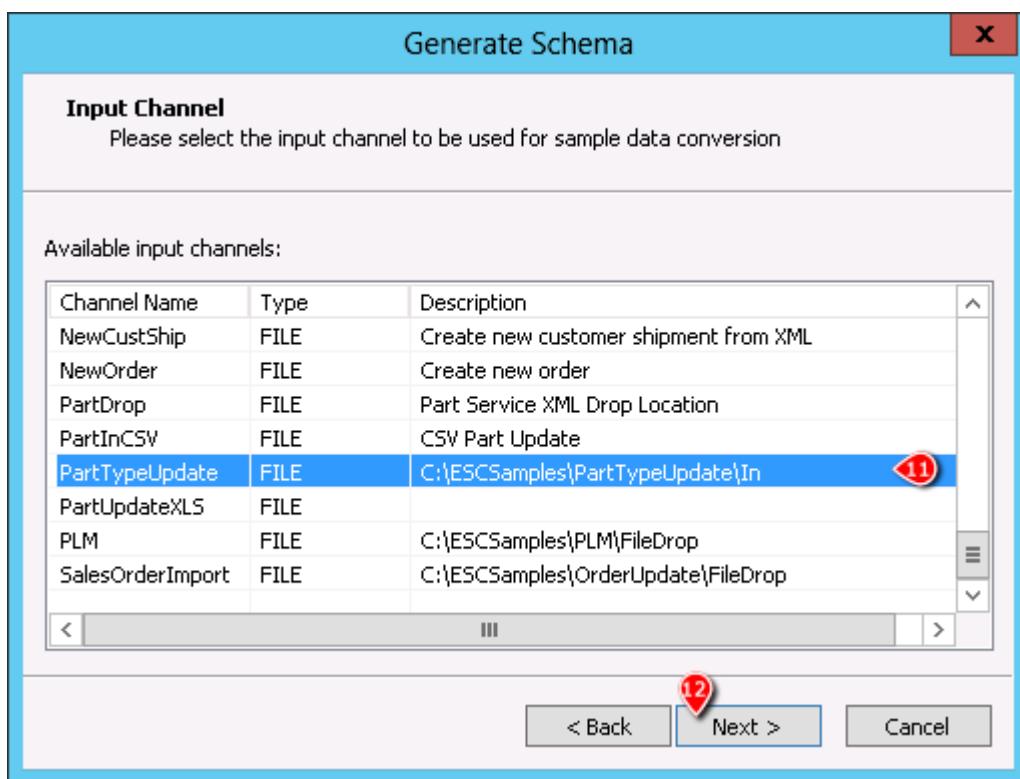
7. In the **File Name** field, enter the schema file name.

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.



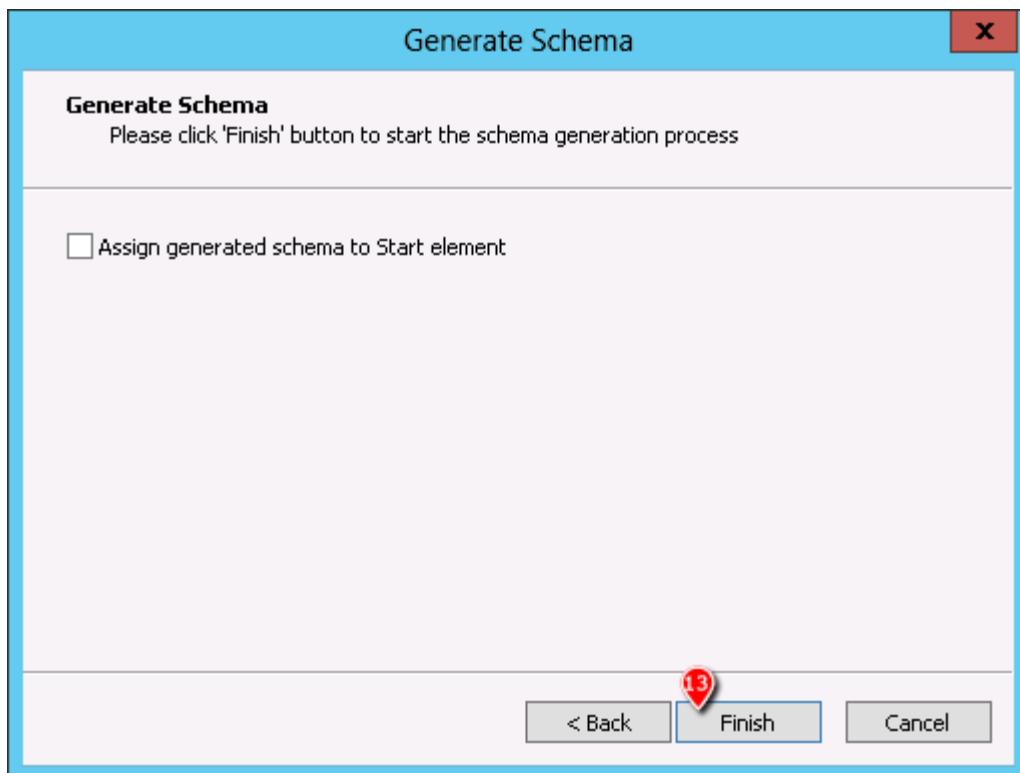
10. Click **Next**.

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

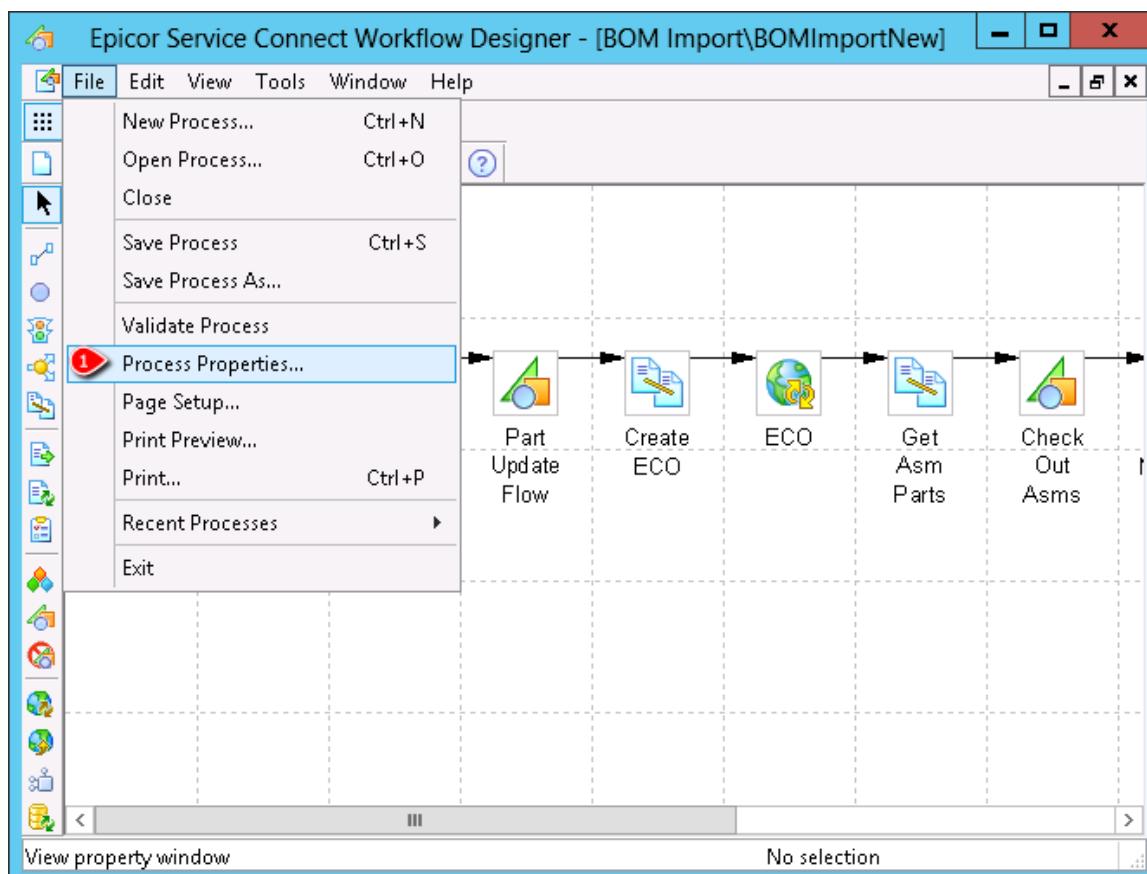
Use the Process Properties window to set up process parameters such as document version, default asynchronous pool of the process and document tracking options. Use this tool to configure message extensions or process variables specific for the particular workflow and define company and Site information.

General Process Settings

Use the General tab to define version, asynchronous pool or workflow description of the current process.

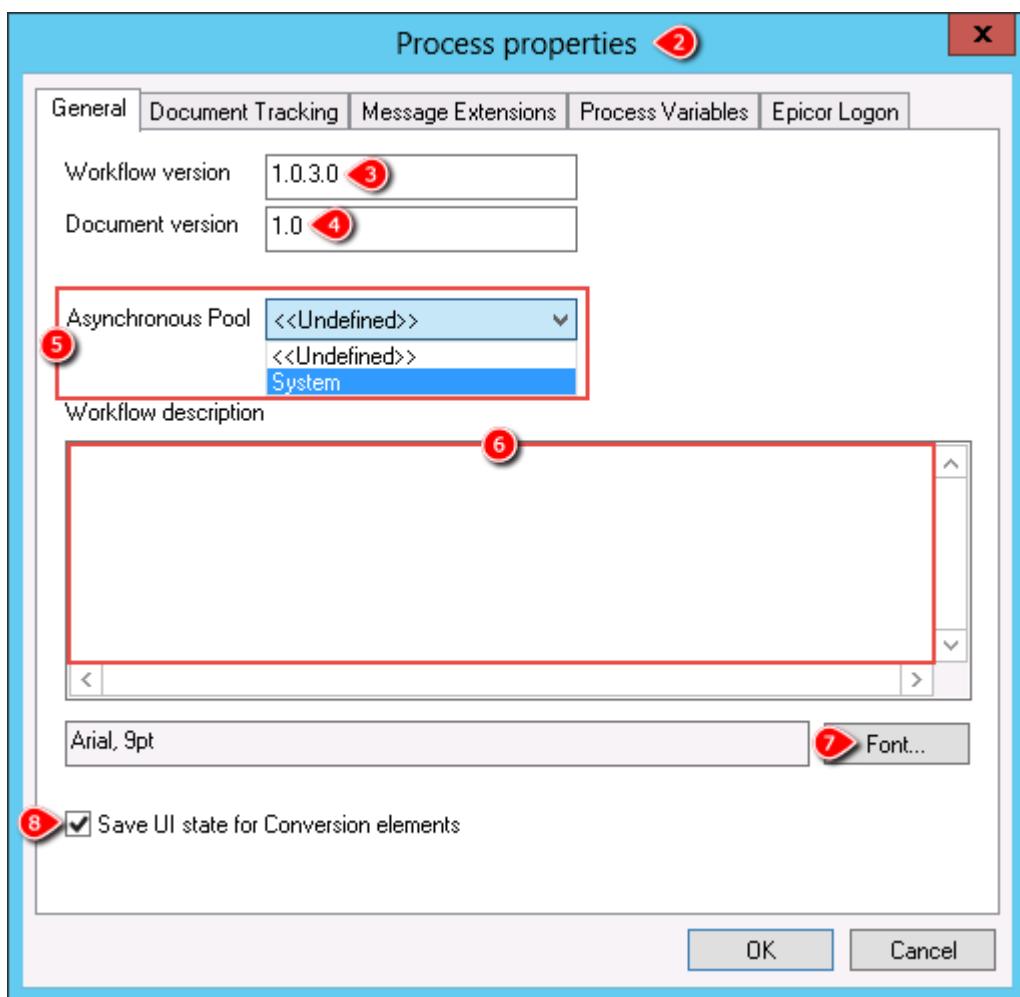
To define general Process Properties of the workflow:

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 context menu.

2. The **Process properties** window displays.



3. View the **Workflow version**.

4. 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.

5. 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.

6. 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.

7. To select the font and color for the comment, click the **Font** button.

8. 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.

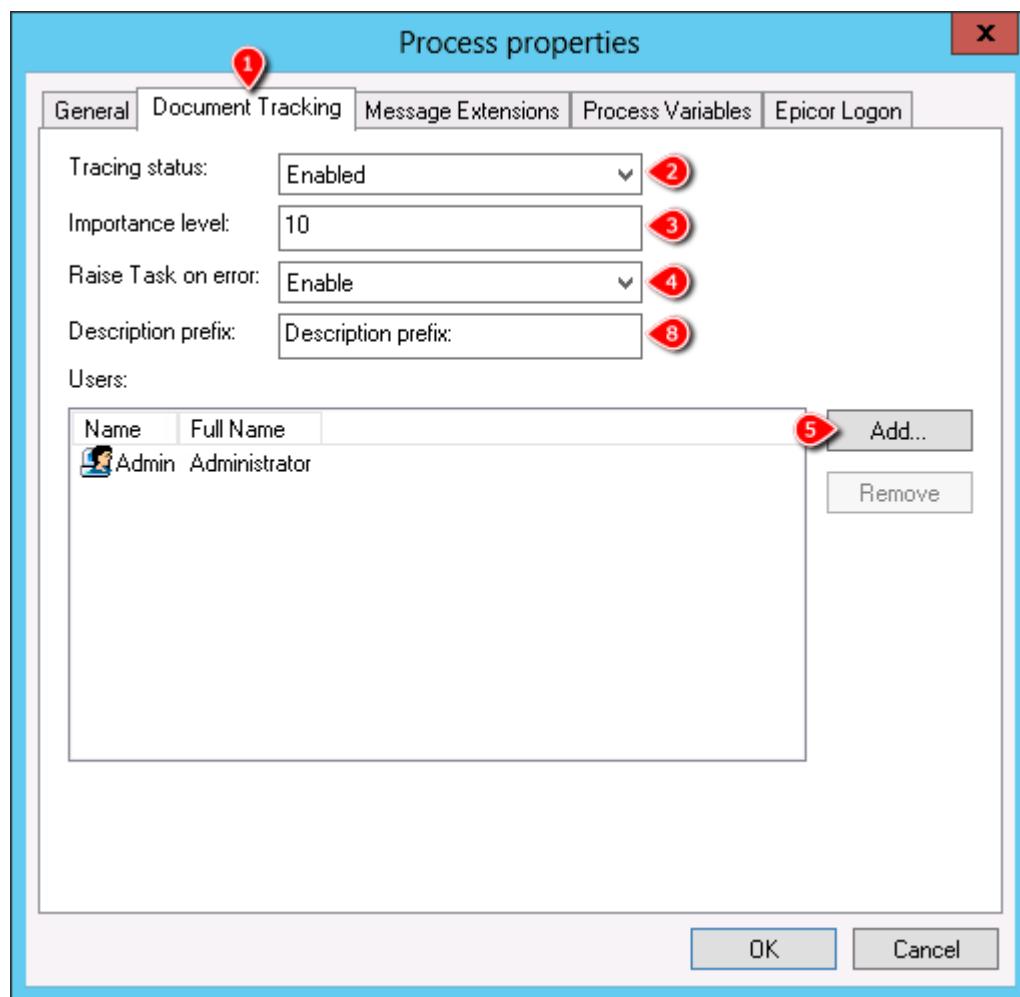
Remain in the **Process properties** window.

Document Tracking Process Settings

Use the Document Tracking tab to control the way messages are processed by the workflow.

To set up the Document Tracking properties for the workflow:

1. In the **Process properties** window, open 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.

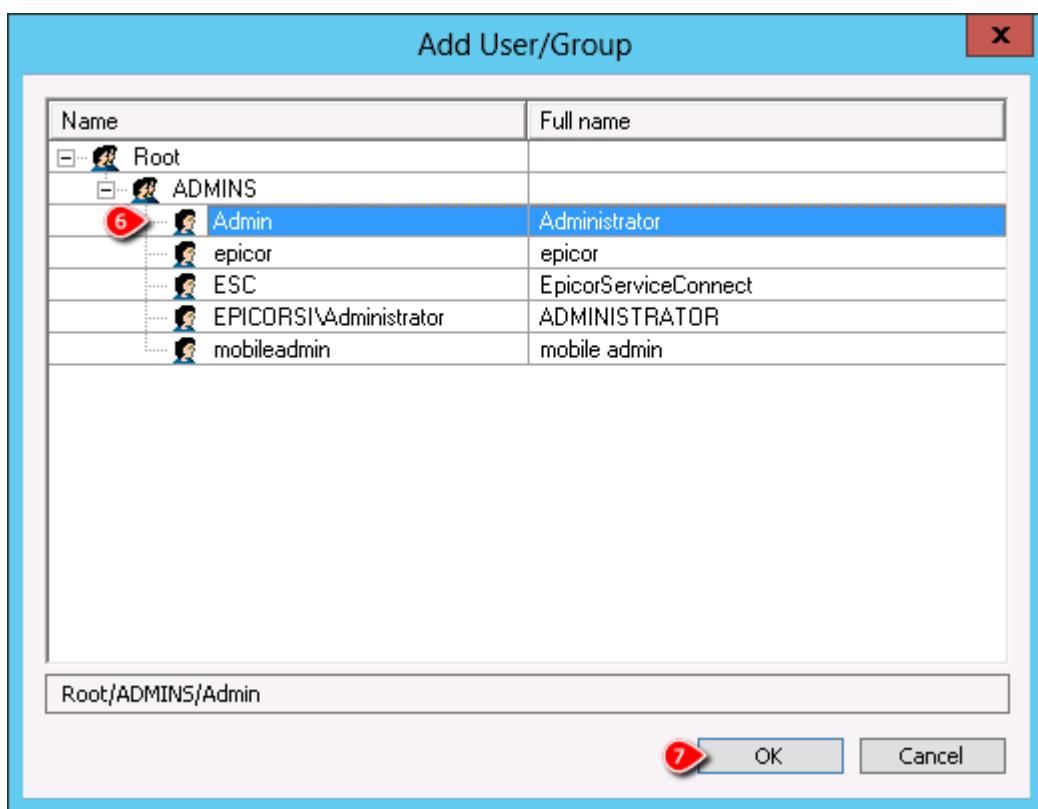
By default, 10 is used as the value of this parameter. This value is compared to the threshold value, specified in the ESC Administration Console, and the traces with importance level less than the threshold value are deleted from the database. If the value of the Importance Level parameter for a process equals 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 finishes with an error. To do this, click the **Raise Task on error** drop-down list and select **Enable**.

5. If the **Raise Task on error** is enabled, click the **Add** button to define to whom the Task is assigned.

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 **Add User/Group** window, select the user or user group to whom the automatic Task should be assigned in case a Web method or .Net method throws an error.



7. Click **OK**.

8. In the **Description prefix** field, define the Task description that will display in the Task Monitor.
Remain in the **Process properties** window.

Message Extensions

Message extensions, also known as containers, are used to store data so it is available to workflow elements after a Web Method or a .NET Call. The data can be information passed into the workflow or derived from other workflow elements, 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.

Message Extensions Example

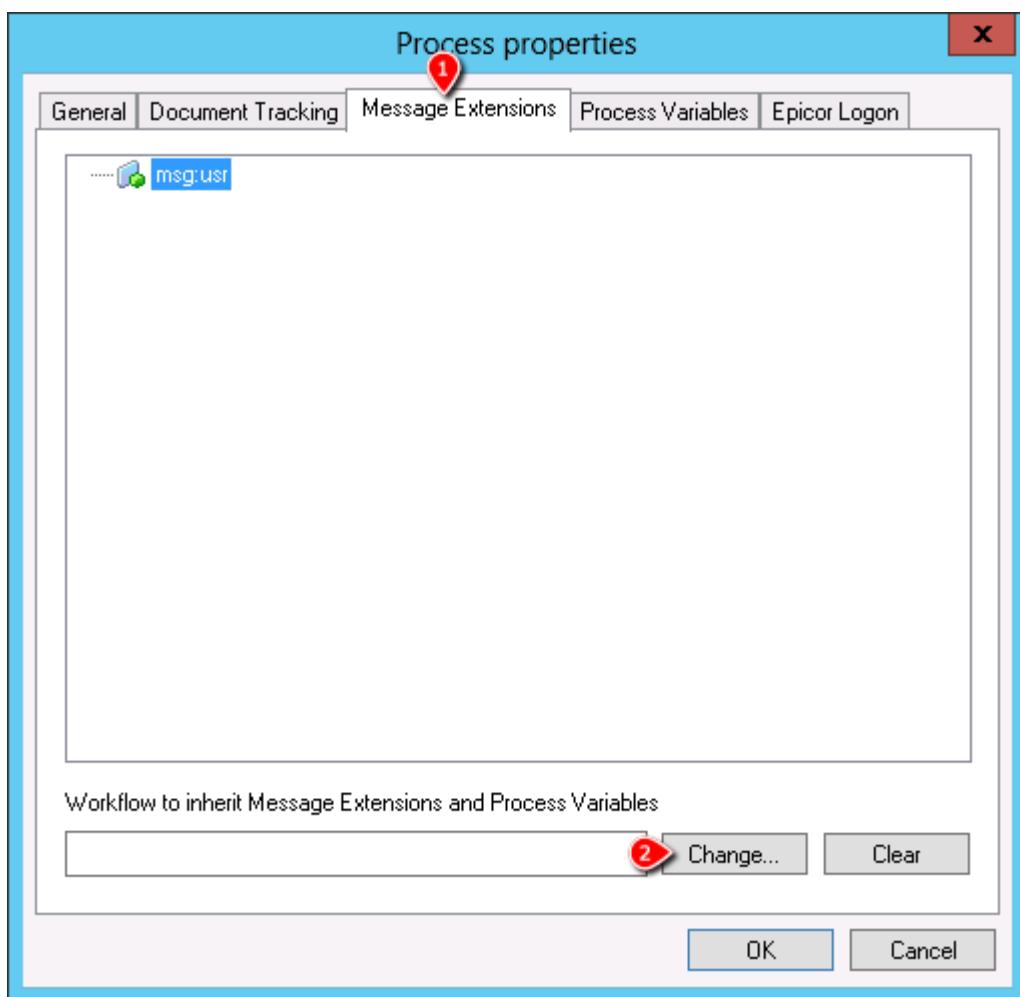
In the following example workflow, the incoming data is an Excel spreadsheet that contains part update information.

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 elements.

Create a Message Extension

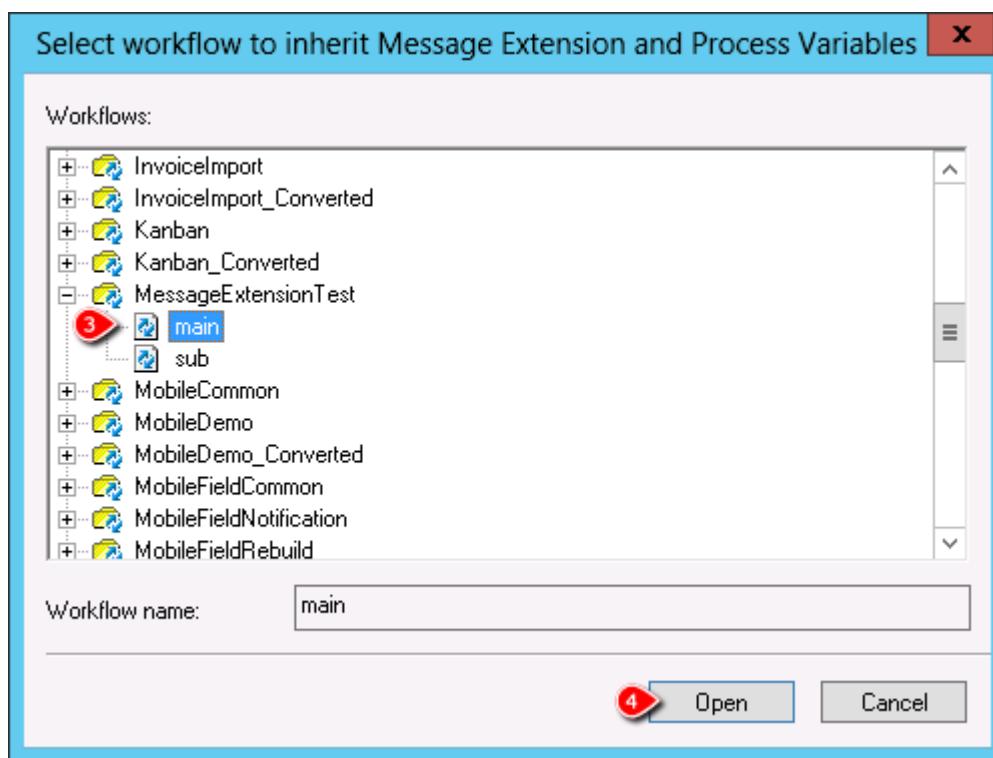
To create an extension:

1. Open the **Message Extensions** tab.



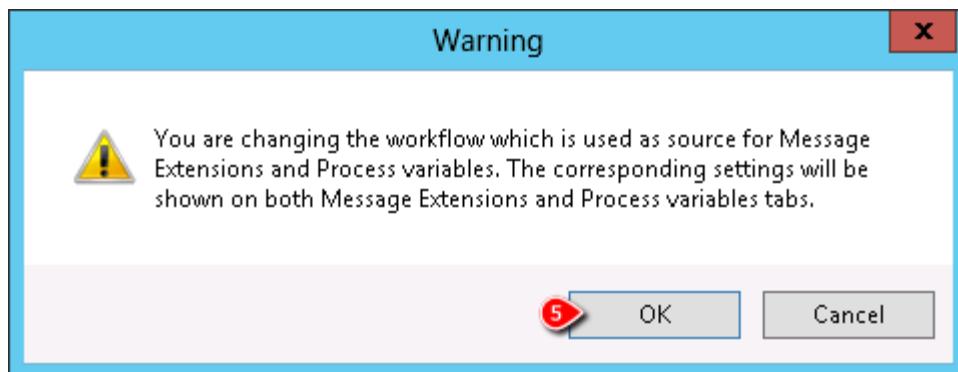
2. If you want to select a workflow as a source for Message Extensions and Process Variables, click **Change**.

3. In the **Select workflow to inherit Message Extensions and Process Variables** window, select the workflow.

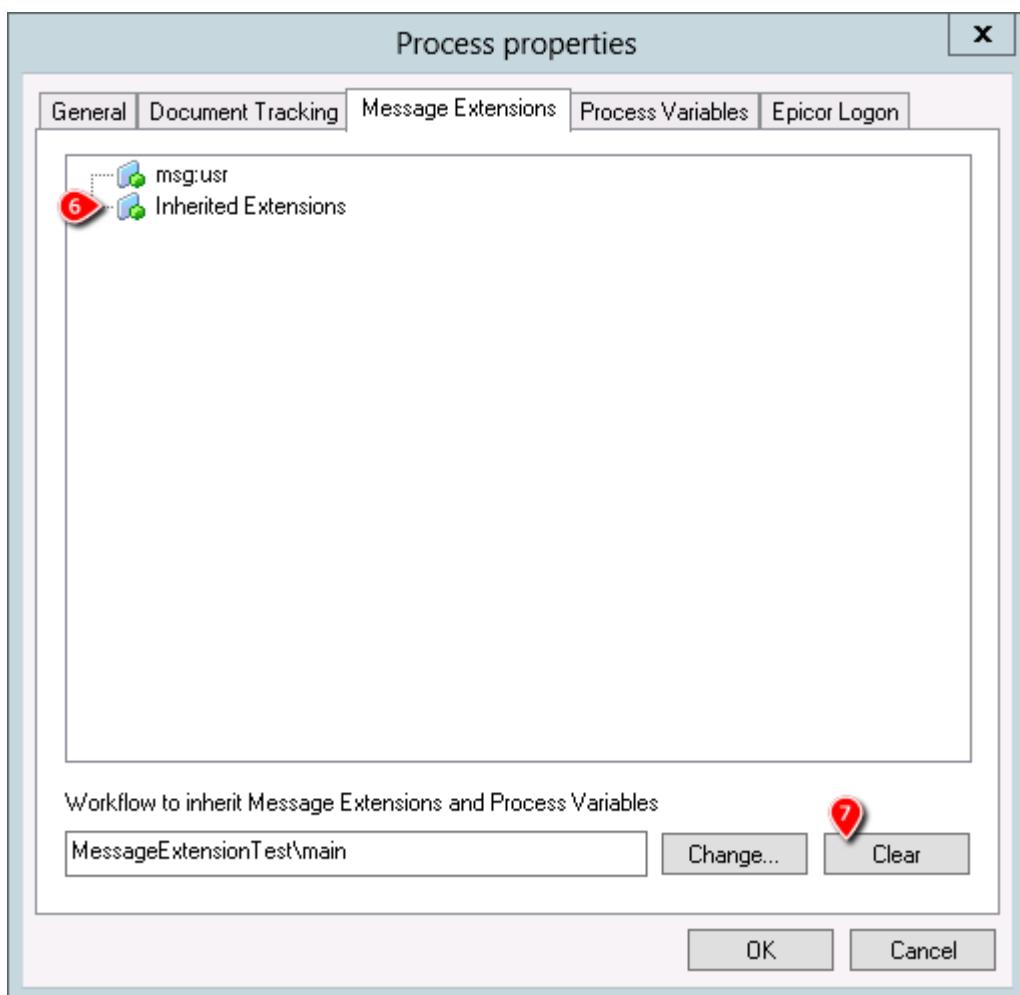


4. Click **Open**

5. To the **Warning** message, click **OK**.



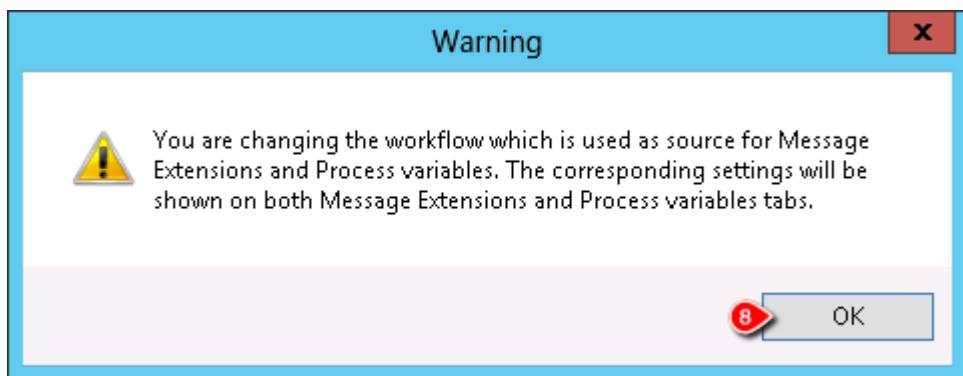
6. Notice the tree displays the **Inherited Extensions** node.



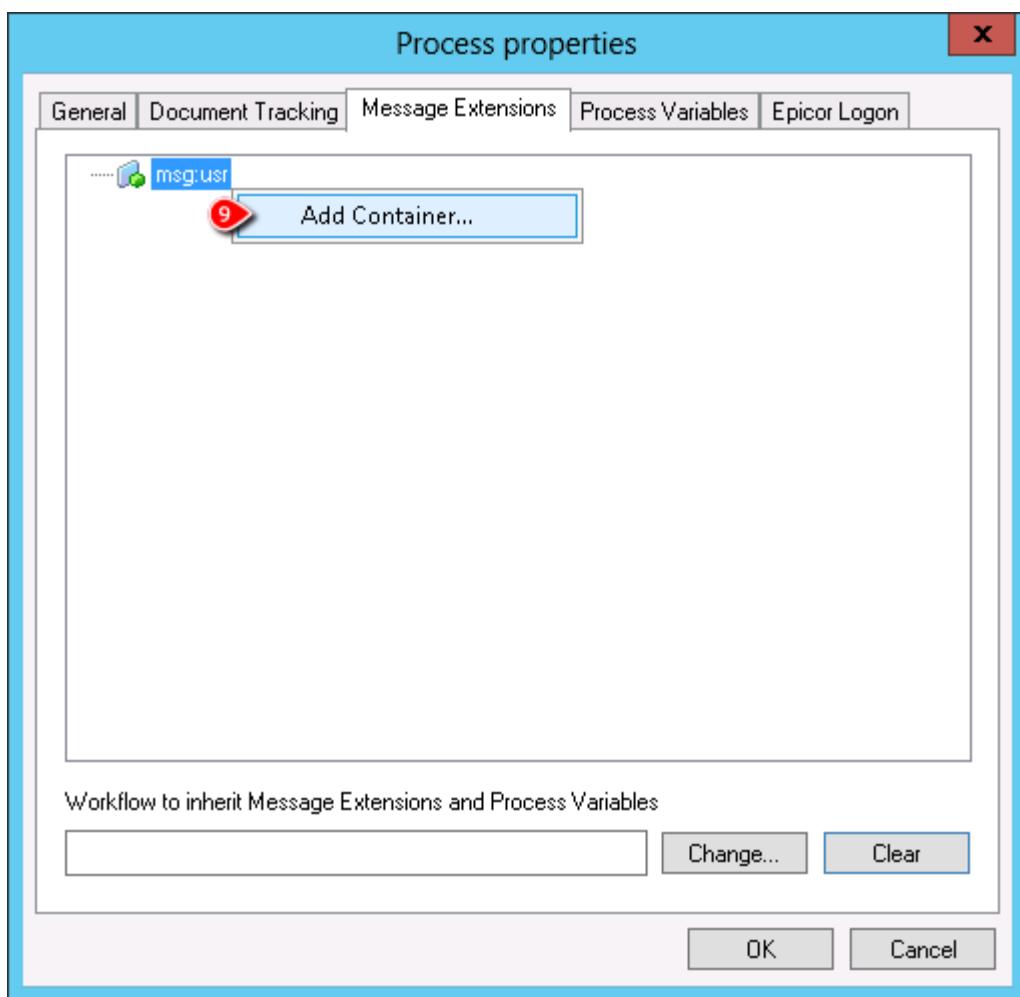
The Validate workflow functionality checks for name conflicts between inherited Extensions/Variables and local ones specified in this workflow. In case such conflicts are found, a warning displays. If an inherited variable or extension has the same name as a local one, the local variable or extension will be seen in XML Mapper, XPath builder and used in runtime.

7. To cancel the Message Extensions inheritance, click the **Clear** button.

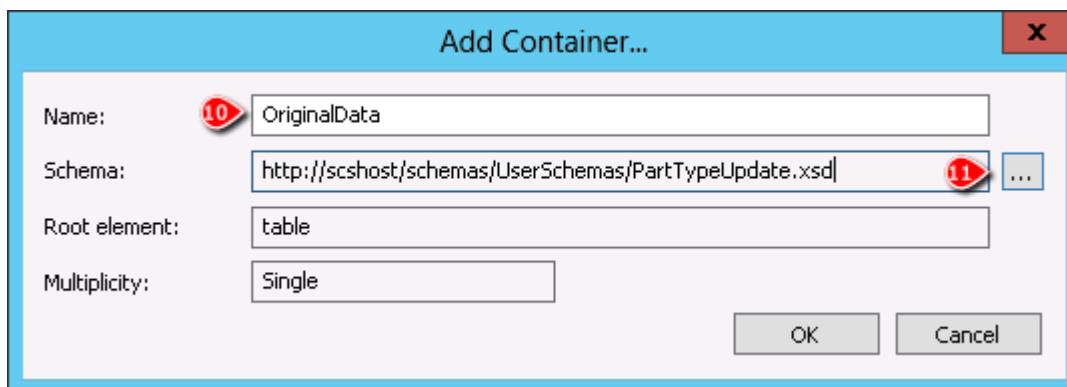
8. To the **Warning** message, click **OK**.



9. Right-click **msg:usr** and select **Add Container**.

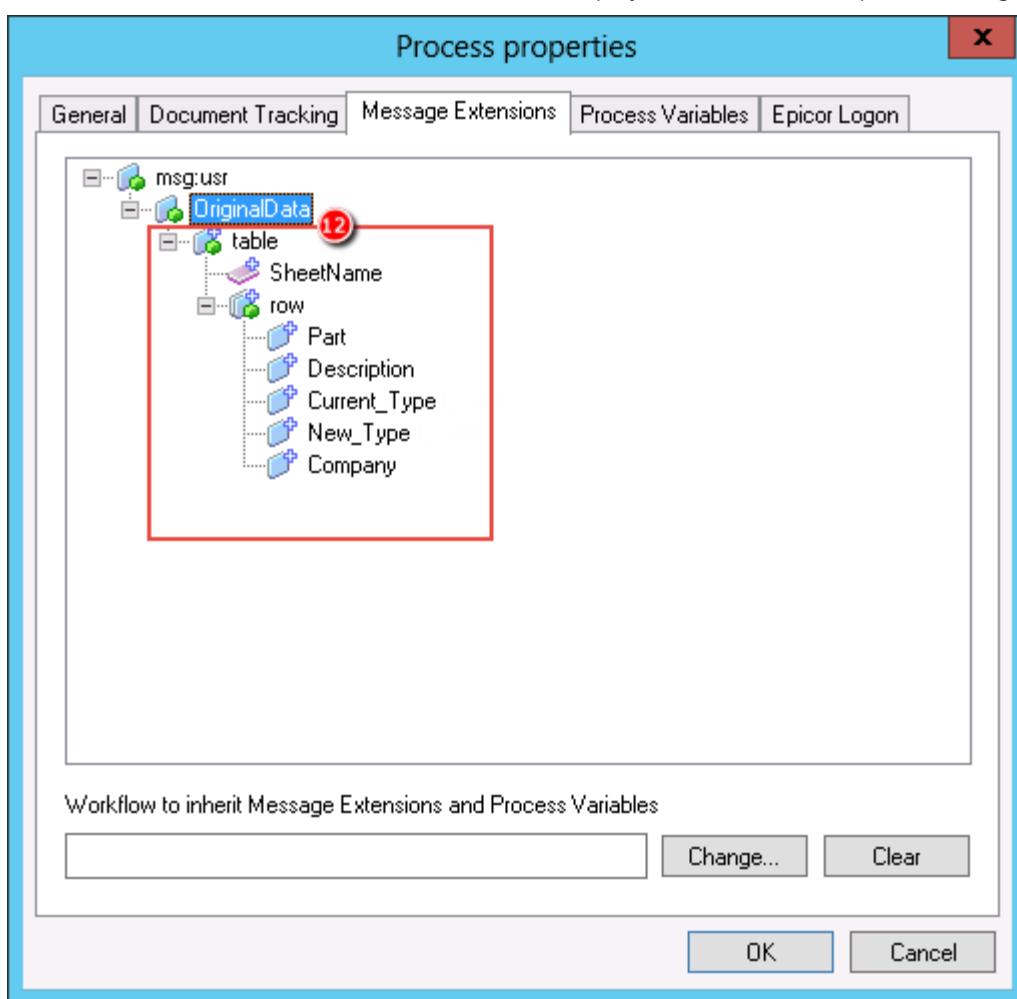


10. The **Add Container** window displays. Enter a **Name** for the container.



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.

11. 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.
12. After the container is created, the data structure displays in the Process Properties dialog box.

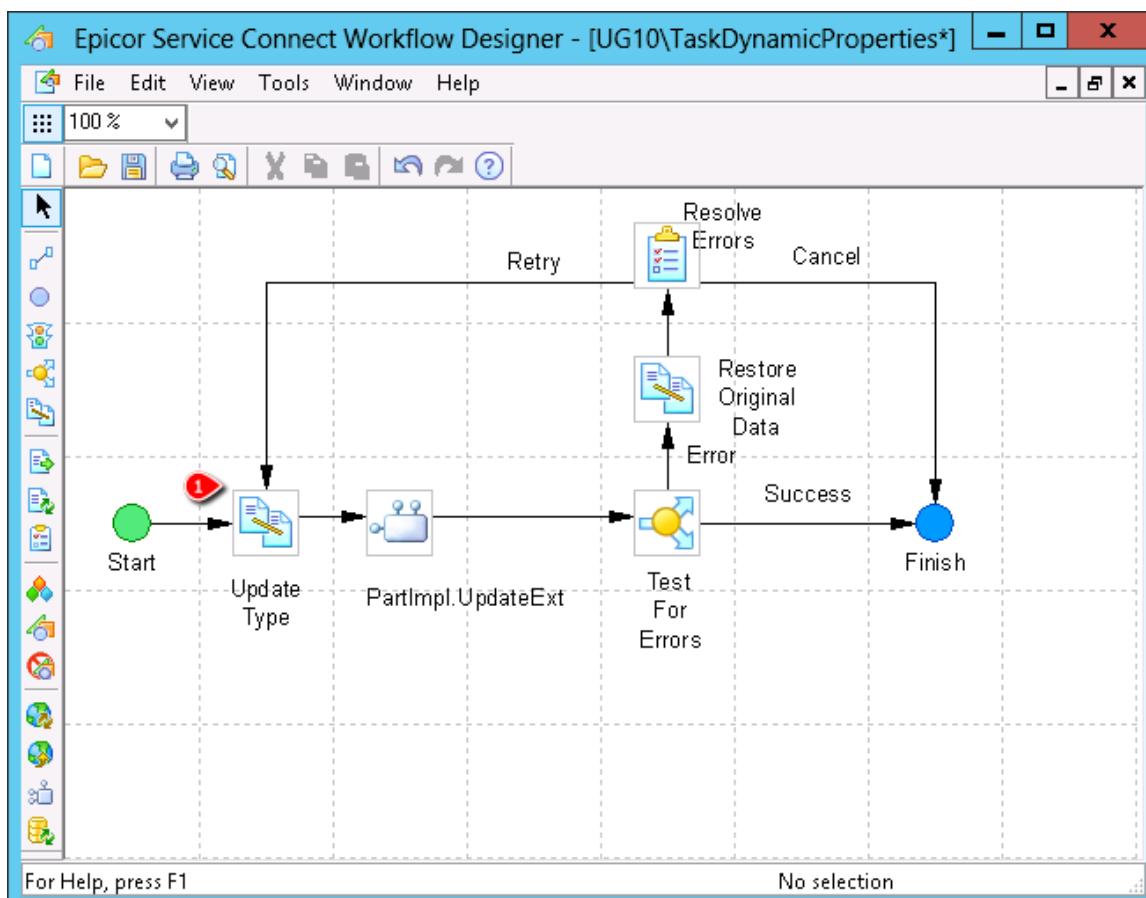


You can now use the container to restore the original data.

Use the Container

To use the container:

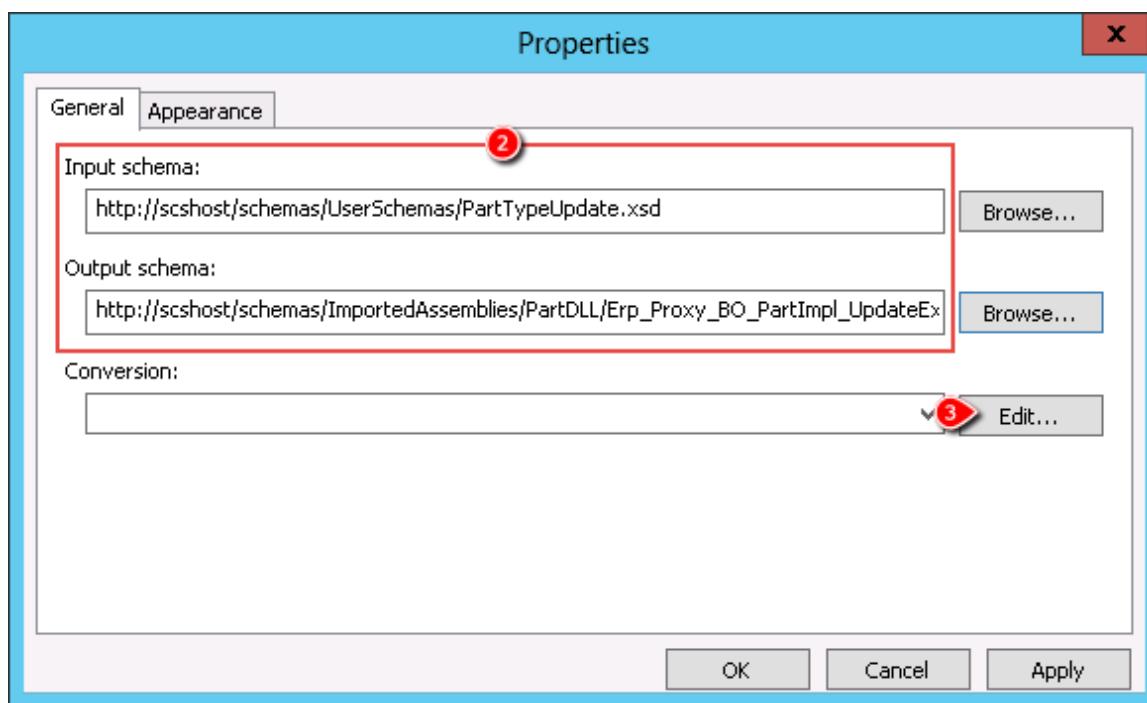
1. On the workflow design area, double-click the **Conversion** workflow element where you want to use the container to open its **Properties**.



2. Select the **Input schema** and **Output schema**.

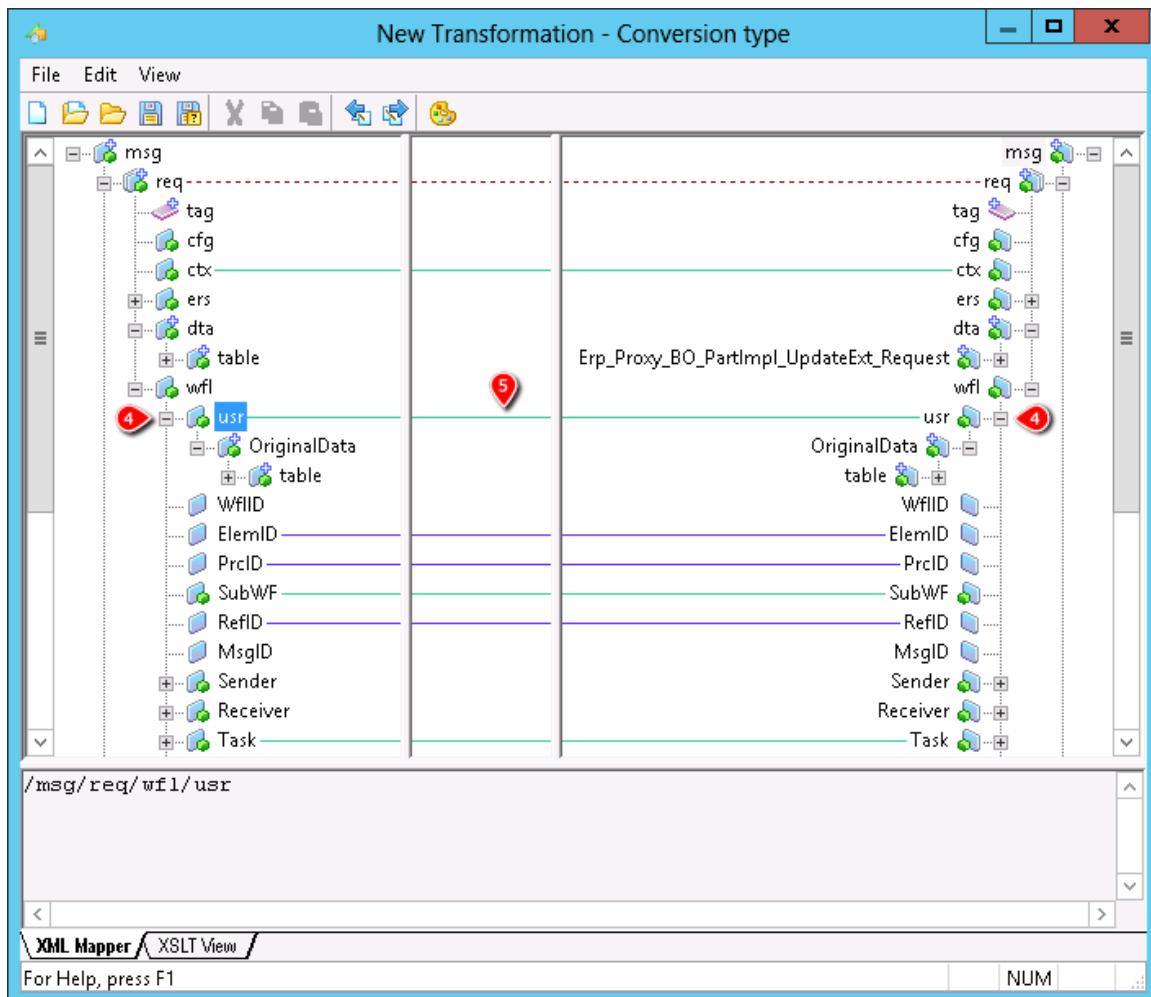
In this example, the Input Schema for the Conversion is the schema used to import part information into Service Connect. The Input Schema name is PartTypeUpdate.xsd. The Output Schema is the .NET BO method

request schema used for the .NET Call, which is the next element in the workflow. The Output Schema name is Erp_Proxy_BO_PartImpl_UpdateExt_Request.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.

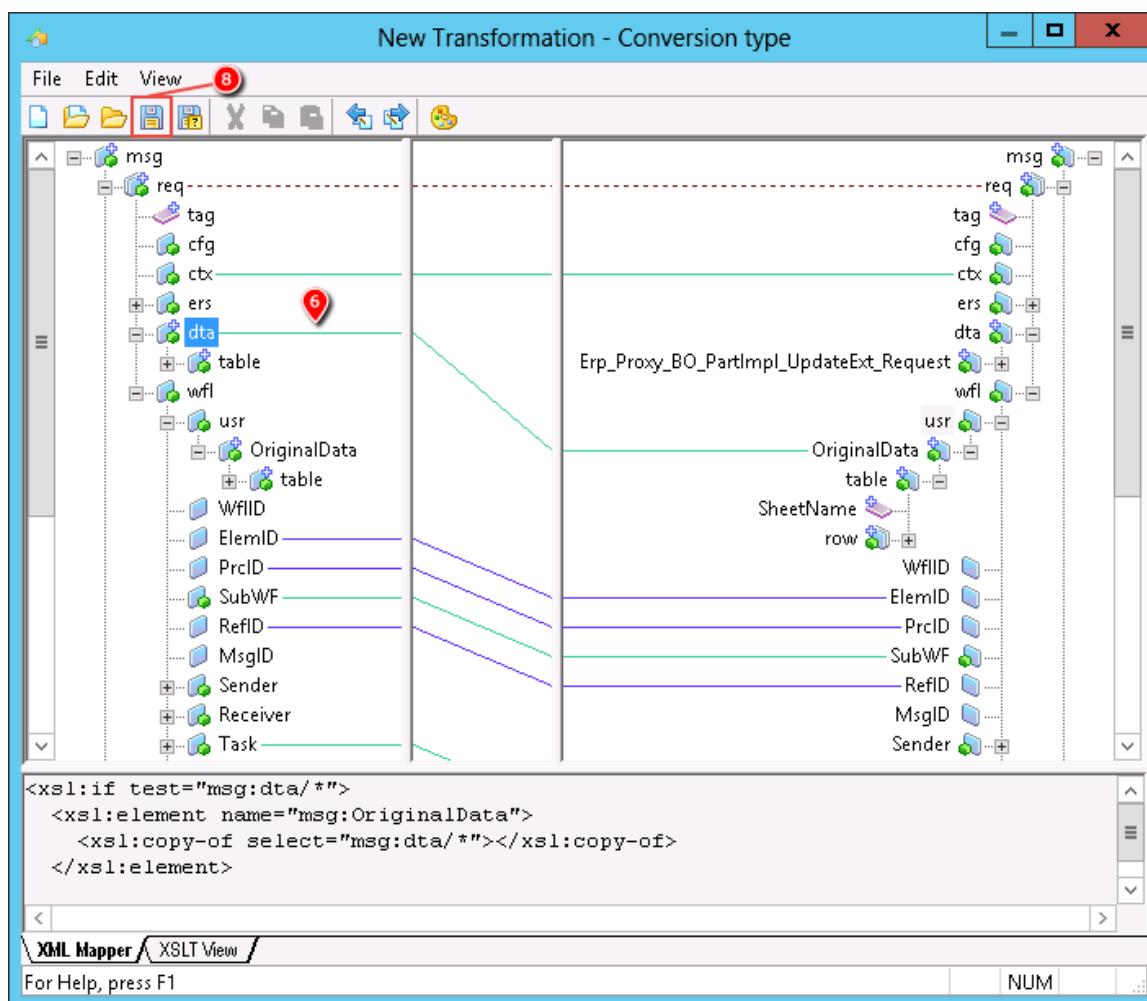


The OriginalData container appears beneath the **usr** node on both sides.

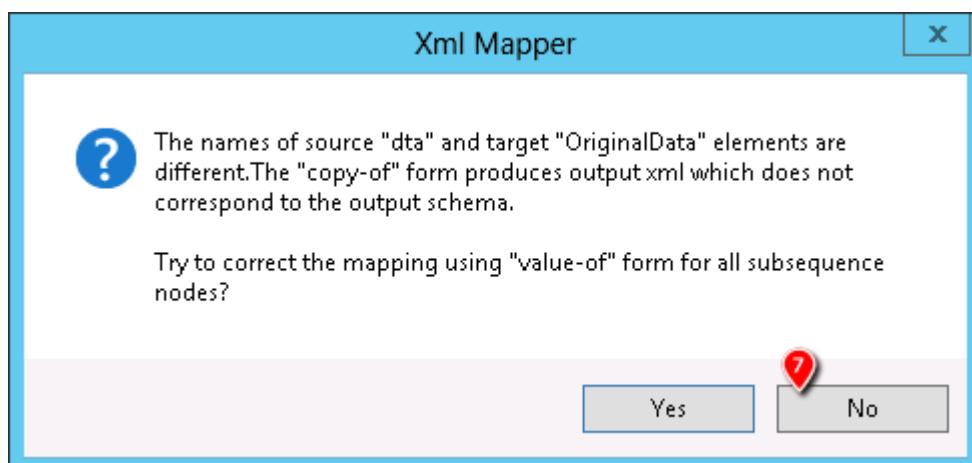
5. Delete the mapping between the **usr** nodes.

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.



7. To the XML Mapper message, click **No**.



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.

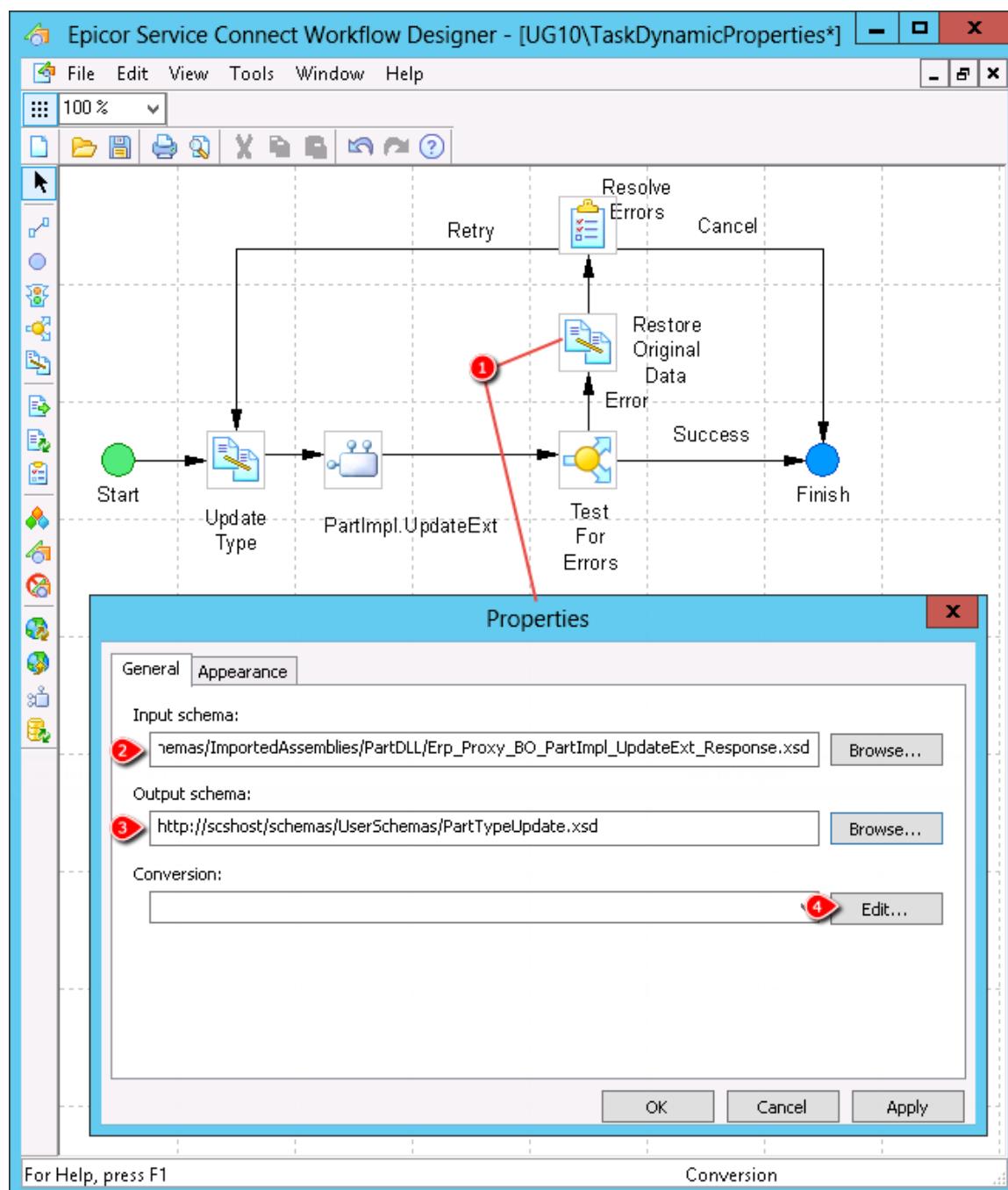
After the .NET Call, you can find the information in the container under the wfl>usr node where it is available for further processing. Data returned from the .NET Call will appear in the dta node.

- Save the conversion and close the XML Mapper window.

Combine the Data

To combine the data from the container and the .NET Call:

- Open the **Properties** of the **Conversion** that follows the .NET Call.



In this example, the **Restore Original Data** conversion is used. To make the data available to edit in the Task workflow element, restore the original data to the **dta** node of the internal envelope.

2. In the **Properties** window, select the **Input schema**.

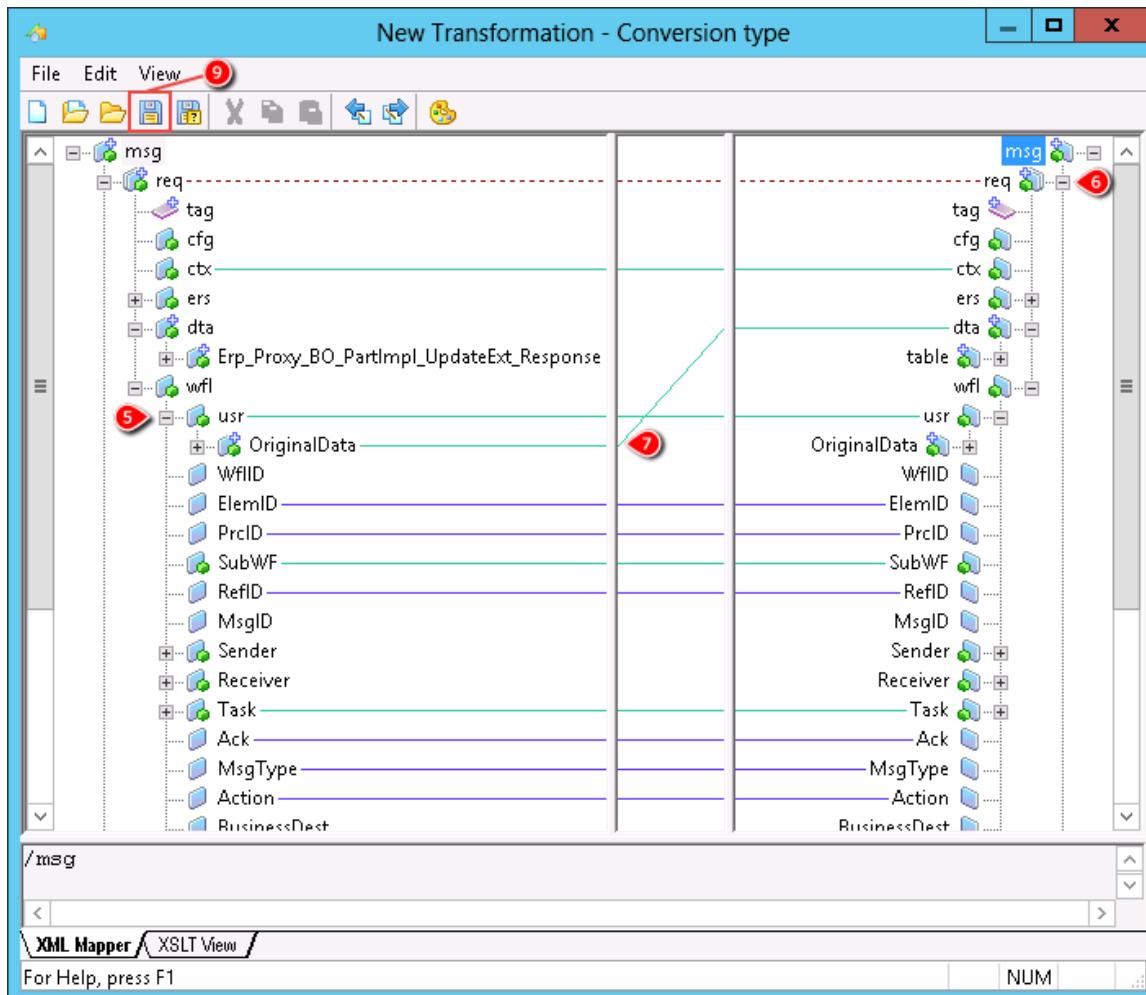
In this example, the Input Schema for the Conversion is the Part BO method response schema.

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**.

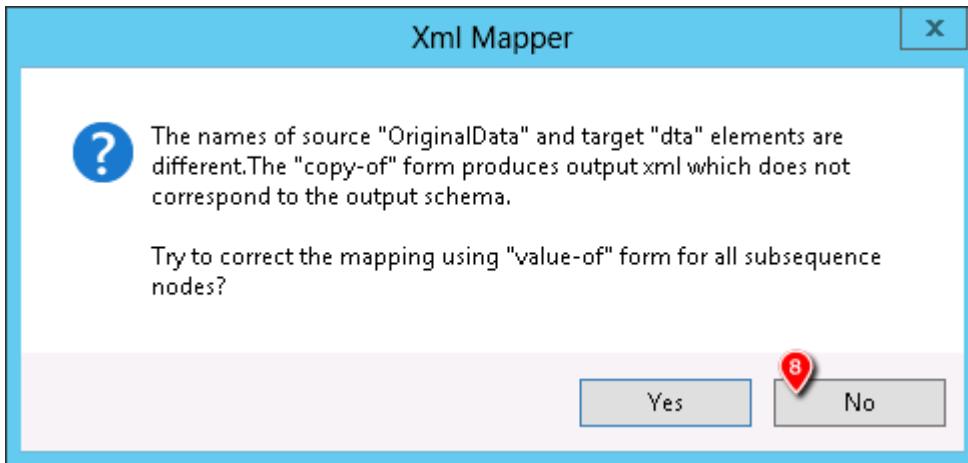


The **OriginalData** node contains the restored original data.

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.

8. To the XML Mapper message, click **No**.



9. Save the conversion and close the XML Mapper.

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.

You can pass process variables from the main workflow to a Sub-workflow. Process variables from a Sub-workflow are not passed back to the main workflow, they can only be passed in one direction.

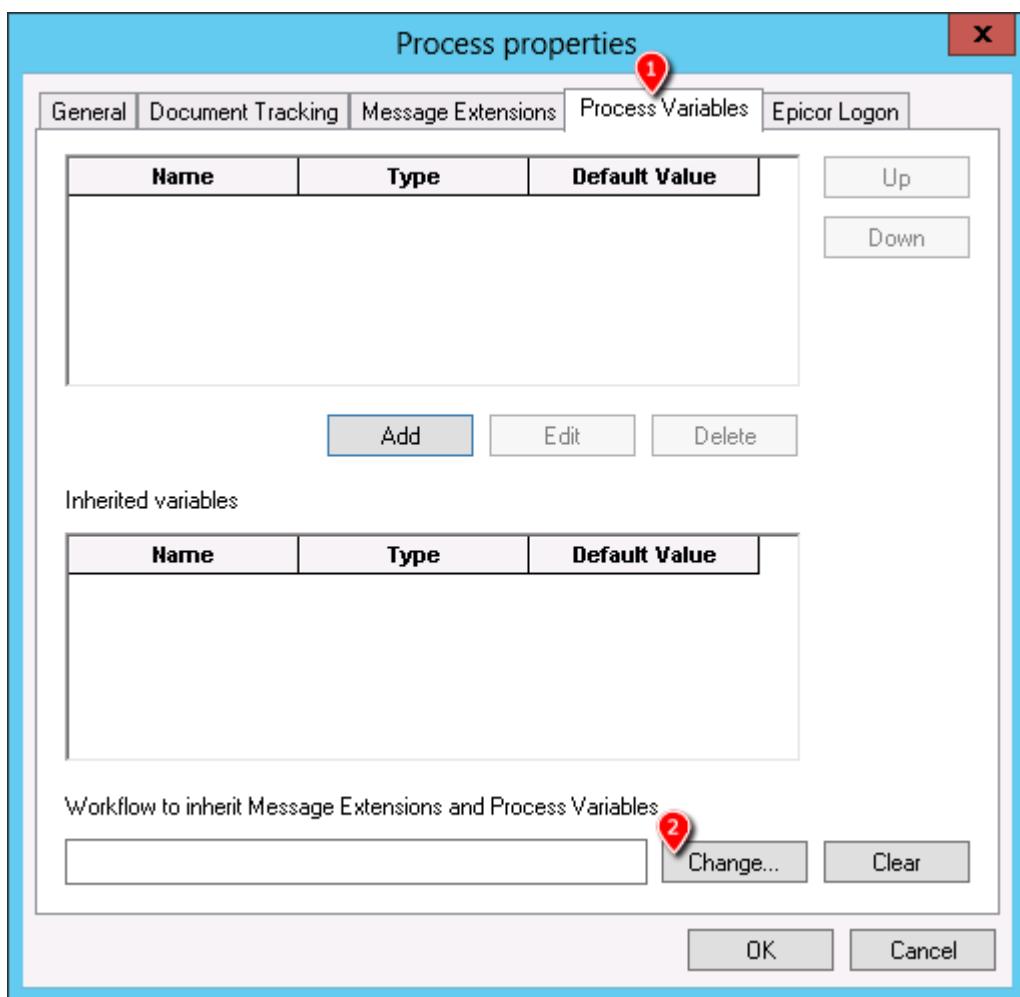
Process Variables Example

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

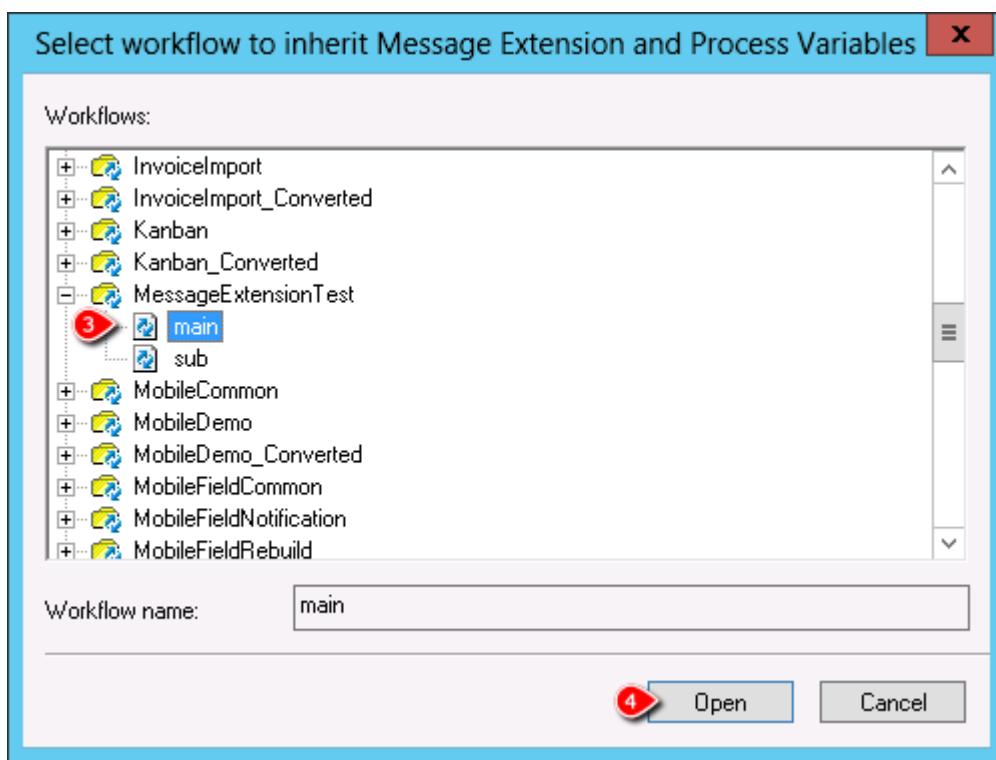
To set up a process variable:

1. In the **Process Properties** window, open the **Process Variables** tab.



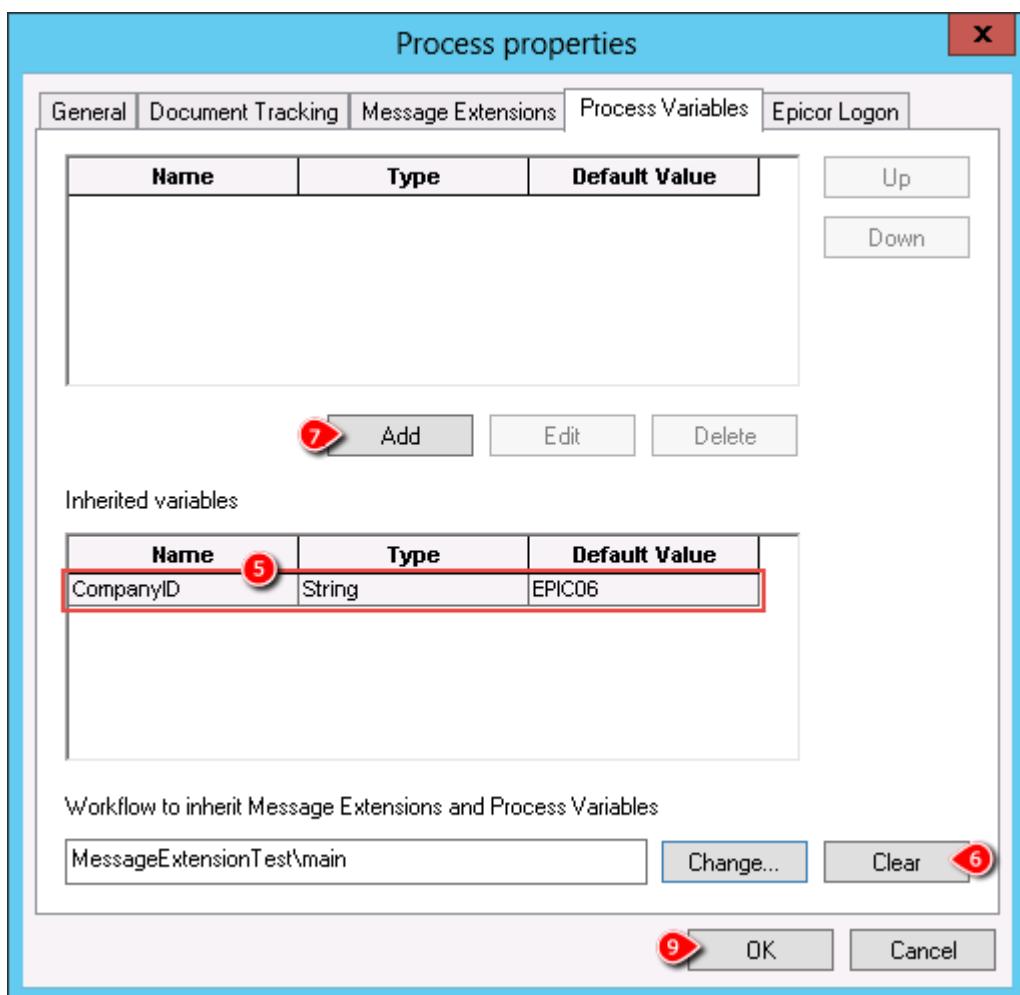
2. To inherit Process Variables and Message extensions from another workflow, click the **Change** button to find and select a workflow.

3. In the **Select workflow...** window, select the workflow.



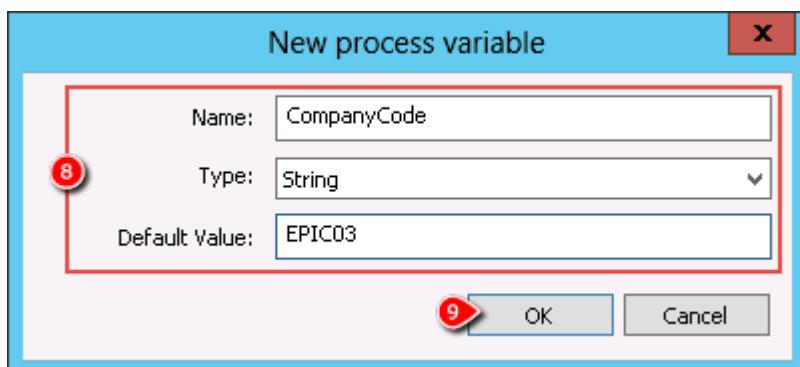
4. Click **Open**.

5. Once a workflow is selected, variables display in the **Inherited variables** grid.



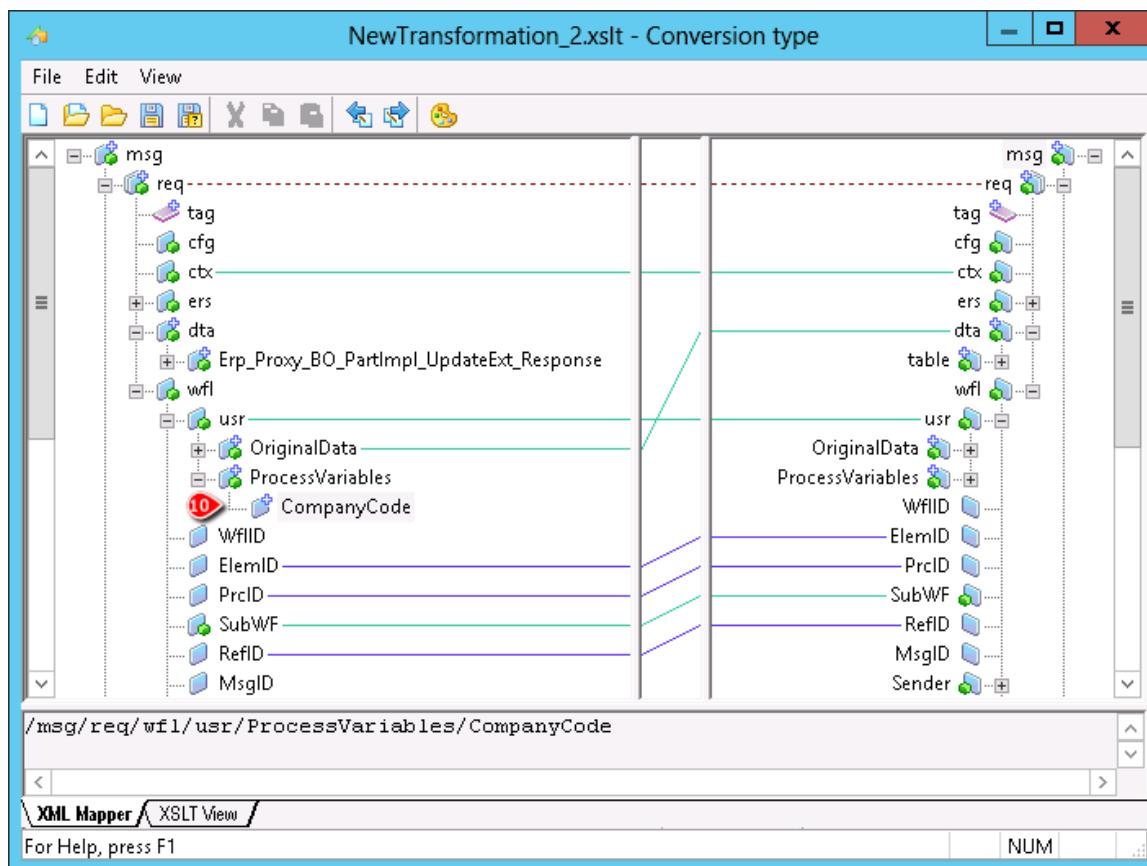
For more information, view the **Message Extensions** topic above.

6. To remove inherited process variables, click **Clear**.
7. Click **Add**.
8. Define the variable **Name**, **Type** and **Default Value**.



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

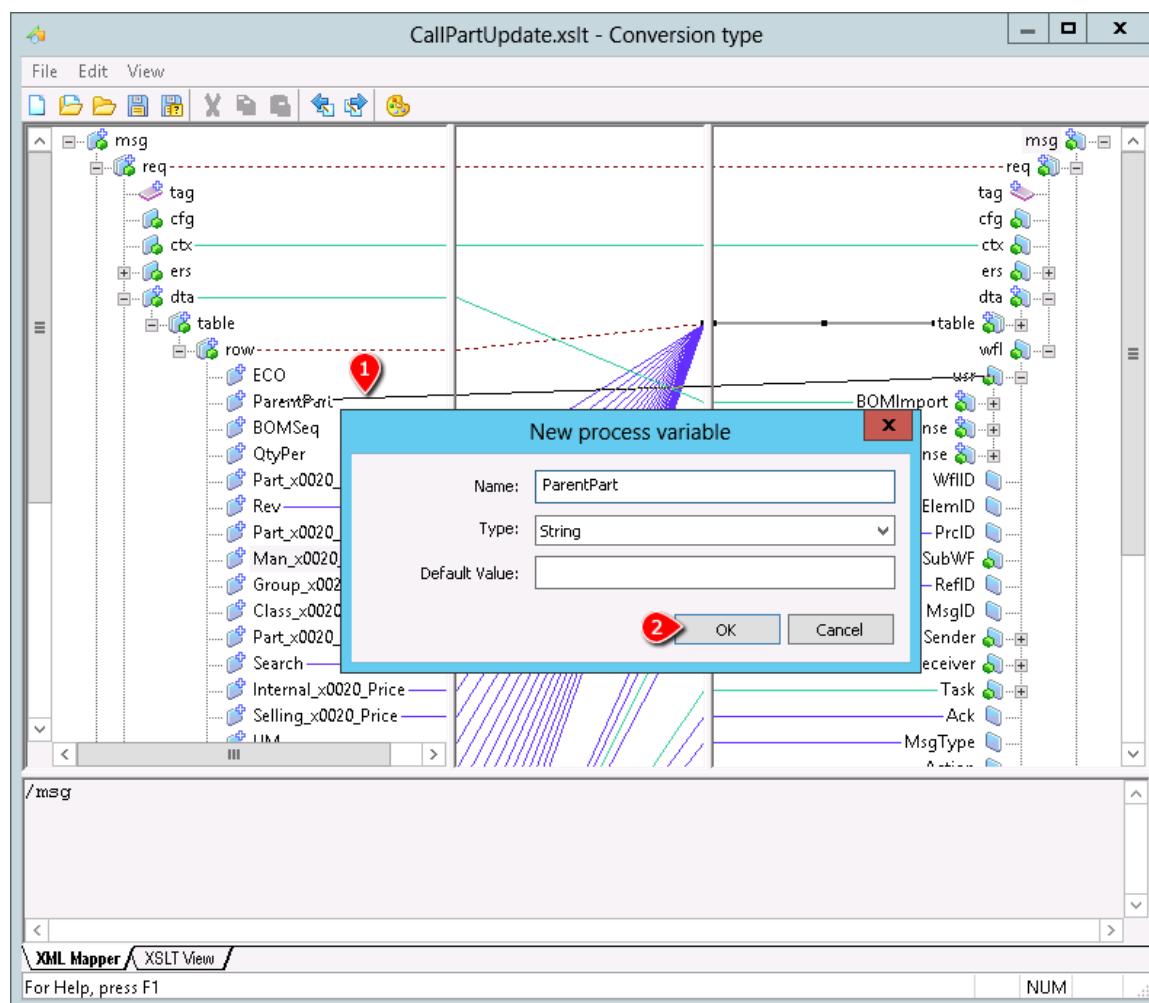
10. The variable is available for use from within the usr node under ProcessVariables.



Drop Nodes to the USR Section

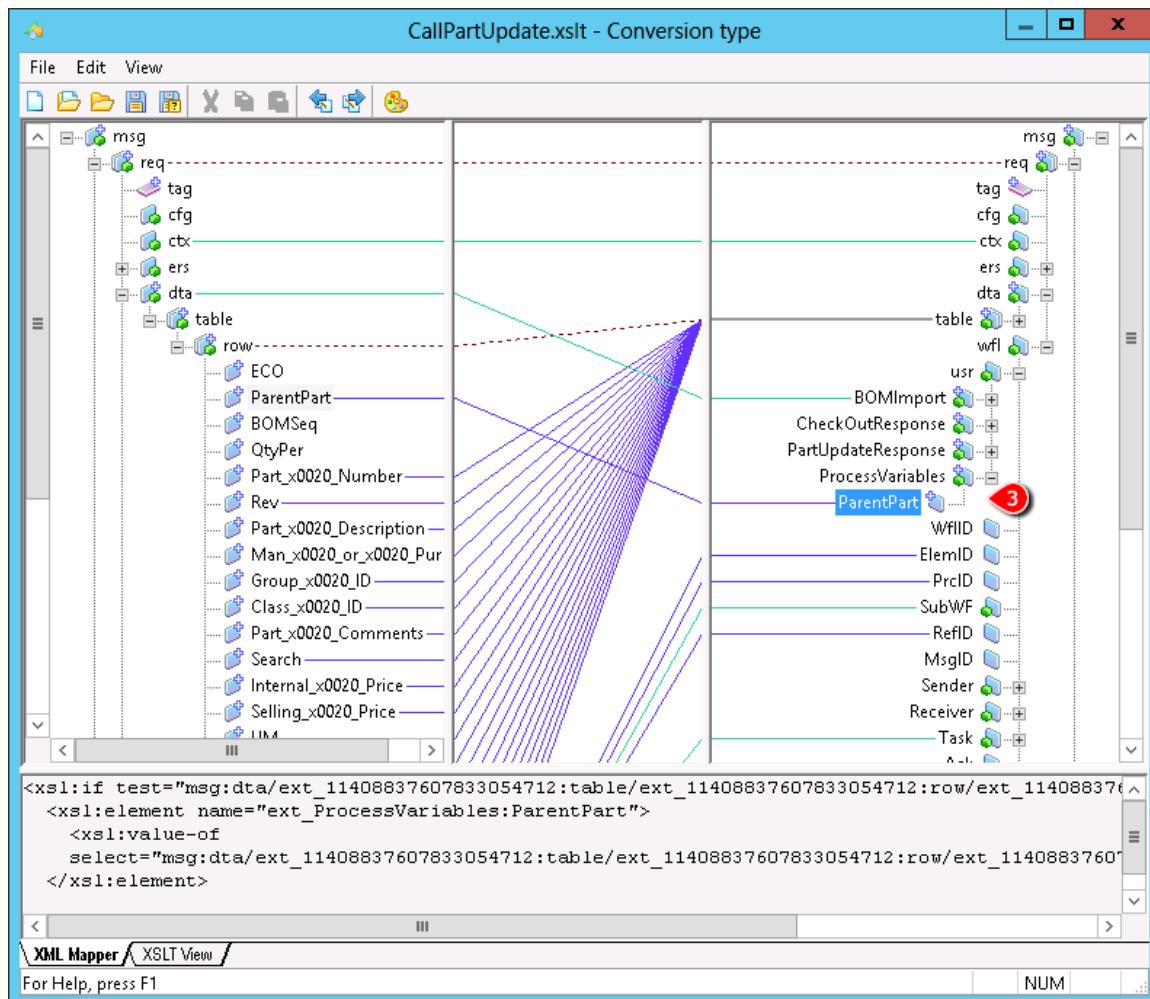
When you configure a Conversion element and drop nodes to the <usr> section, XML Mapper creates Process variables and Message extensions.

1. In XML Mapper, drop a node from the source document to the <usr> section of the target document.

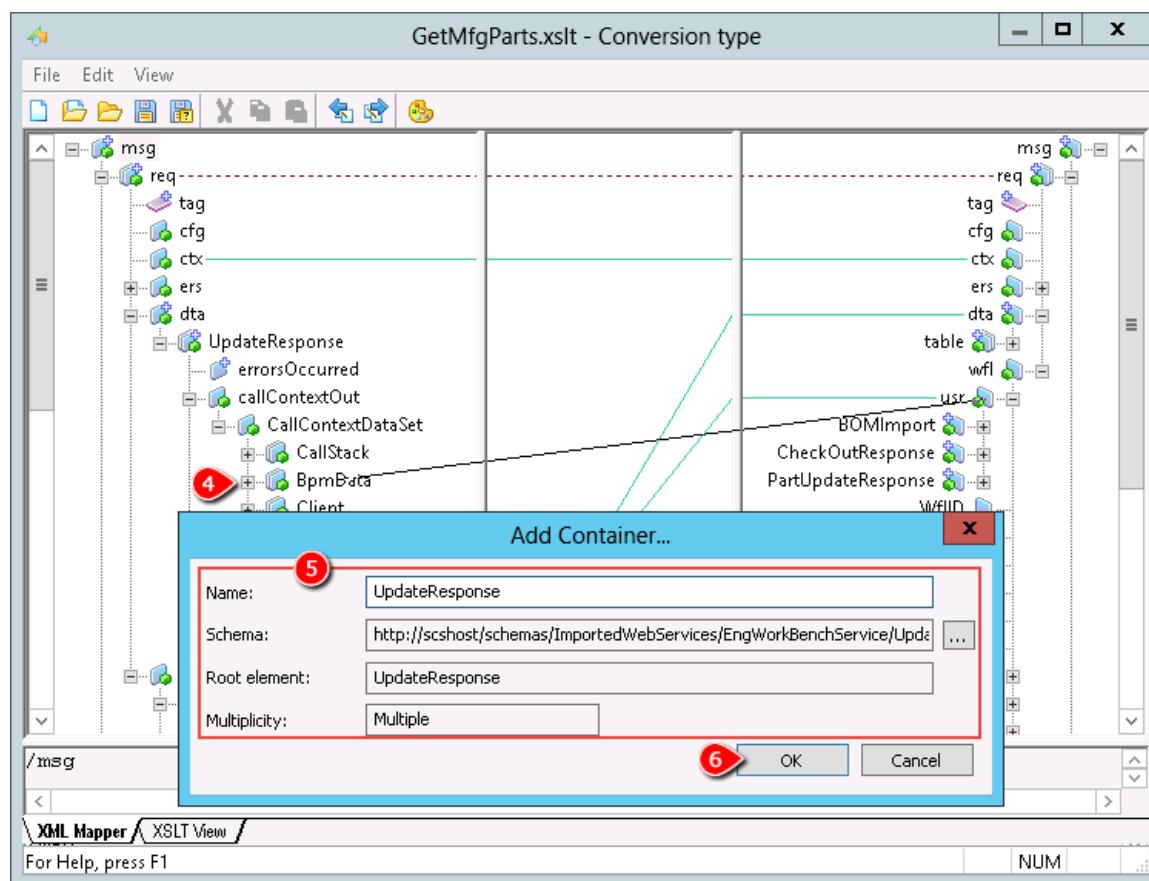


2. In the **New process variable** window, edit the variable properties and click **OK**.

3. When you map a simple type node to the <usr> section, a new Process variable is created and mapped.

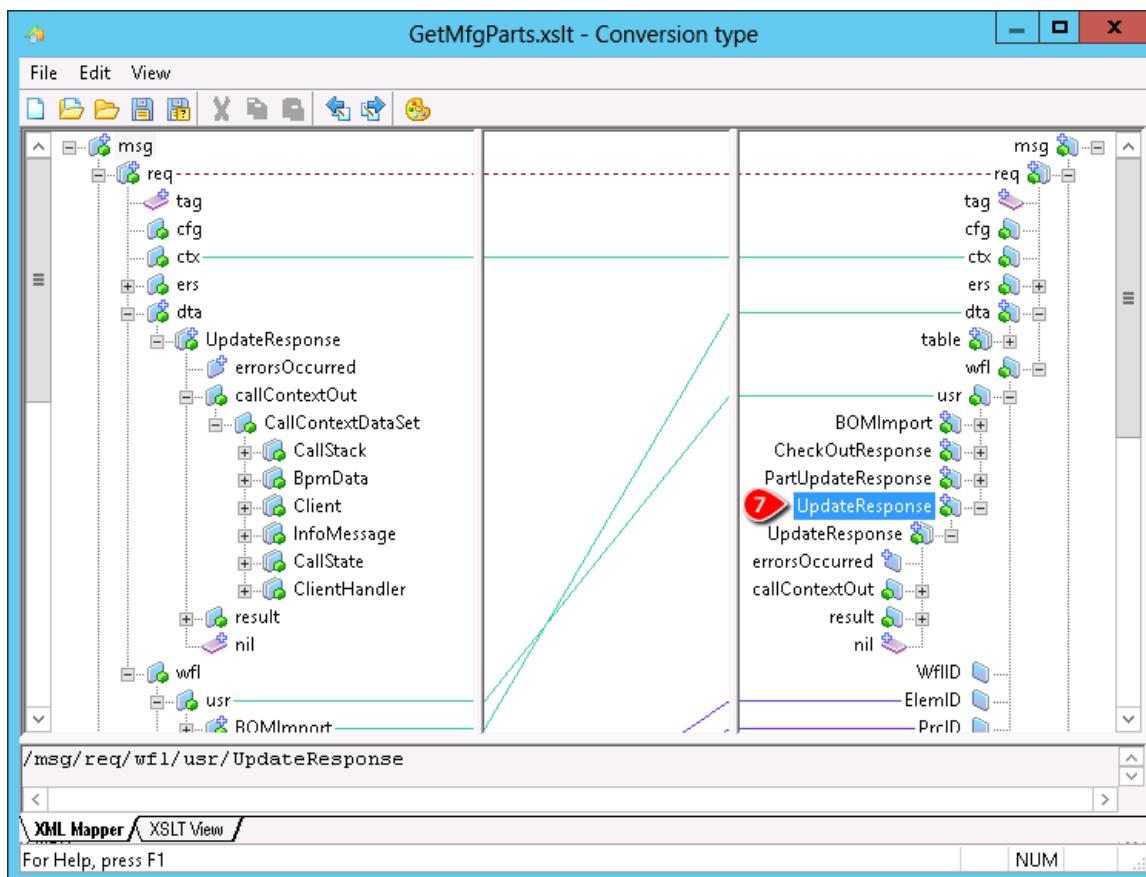


4. When you map a complex type node under the source <dta> section to the target <usr> section, a new message extension is created in target schema.



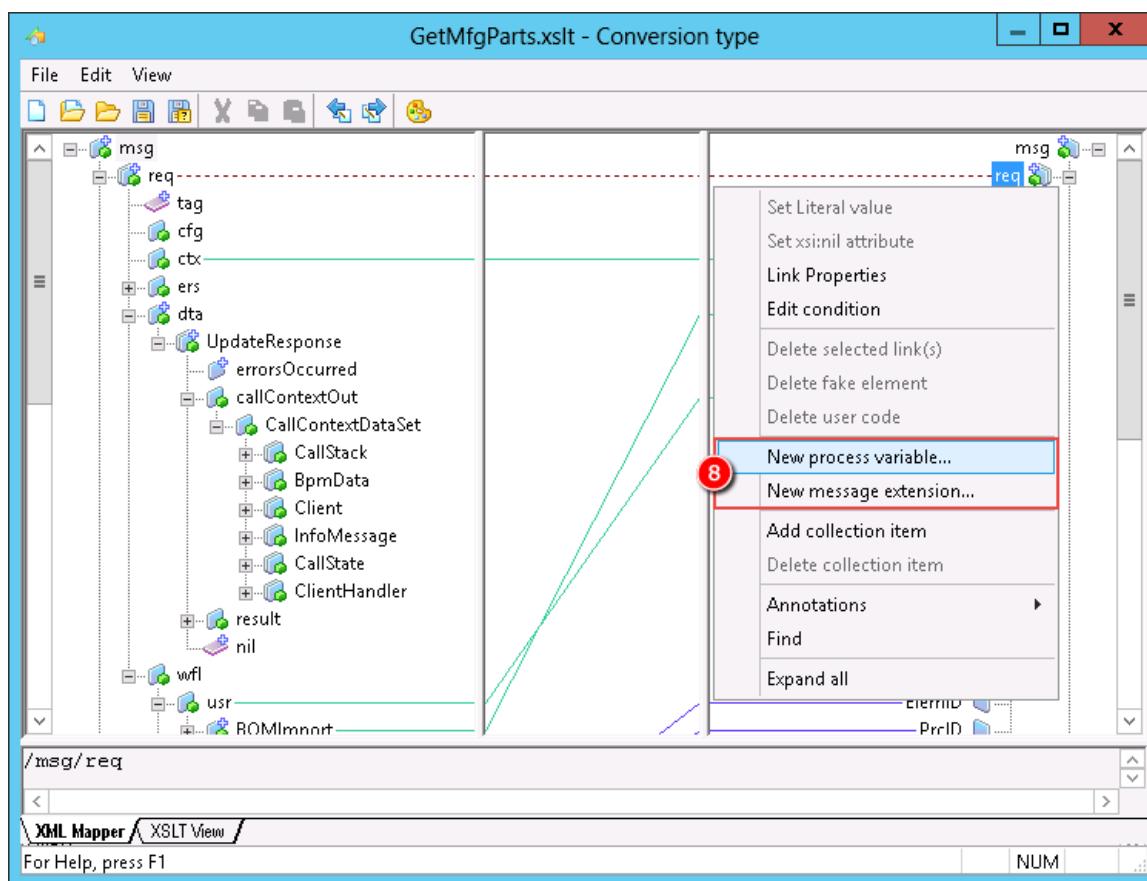
5. In the **Add Container** window, edit the message extension properties.
6. Click **OK**.

7. View the new message extension in the target schema.



For complex nodes, no automatic mapping is created.

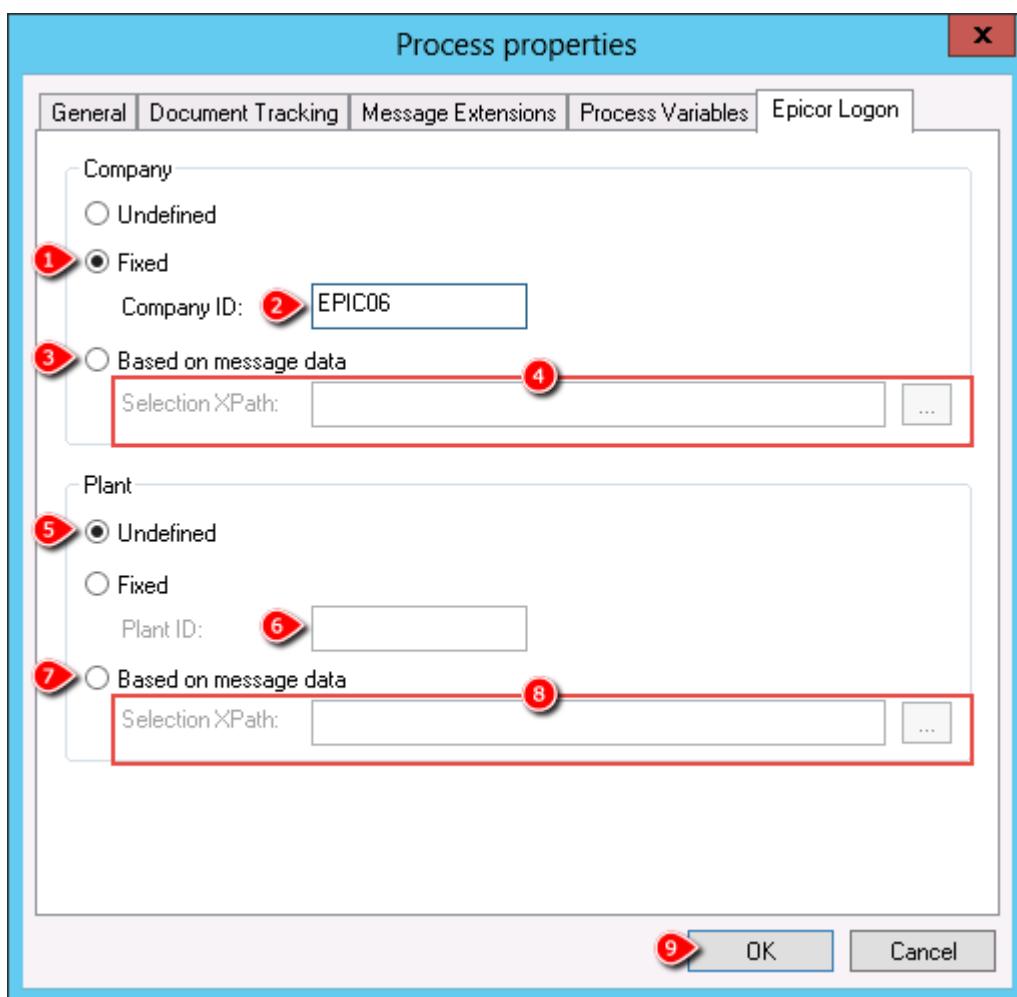
8. You can alternatively right-click the right pane and select **New process variable** or **New message extension** to create variables or containers.



Epicor Logon

Use the Epicor Logon tab to set up Company and Site information.

- To set a particular Epicor Company, on the **Epicor Logon** tab, in the Company pane, select the **Fixed** option.



- In the **Company ID** field, enter a company code.
- 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**.
- In the **Selection XPath** field, enter the XPath, or click the ... (Ellipse) button to build the XPath in XPath Builder.
Use the simplified form of XPath statement: omit namespaces and use only the xml elements names.
- To set a particular Epicor Site, in the Plant pane, select the **Fixed** radio-button.
- In the **Plant ID** field, enter a Site code.
- If you want the Site 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.
Use the simplified form of XPath statement: omit namespaces and use only the xml elements names.
9. Click **OK**.

Process Validation

Use the Validate Process functionality to check the workflow you created for consistency.

The validation functionality checks that all the following conditions are met:

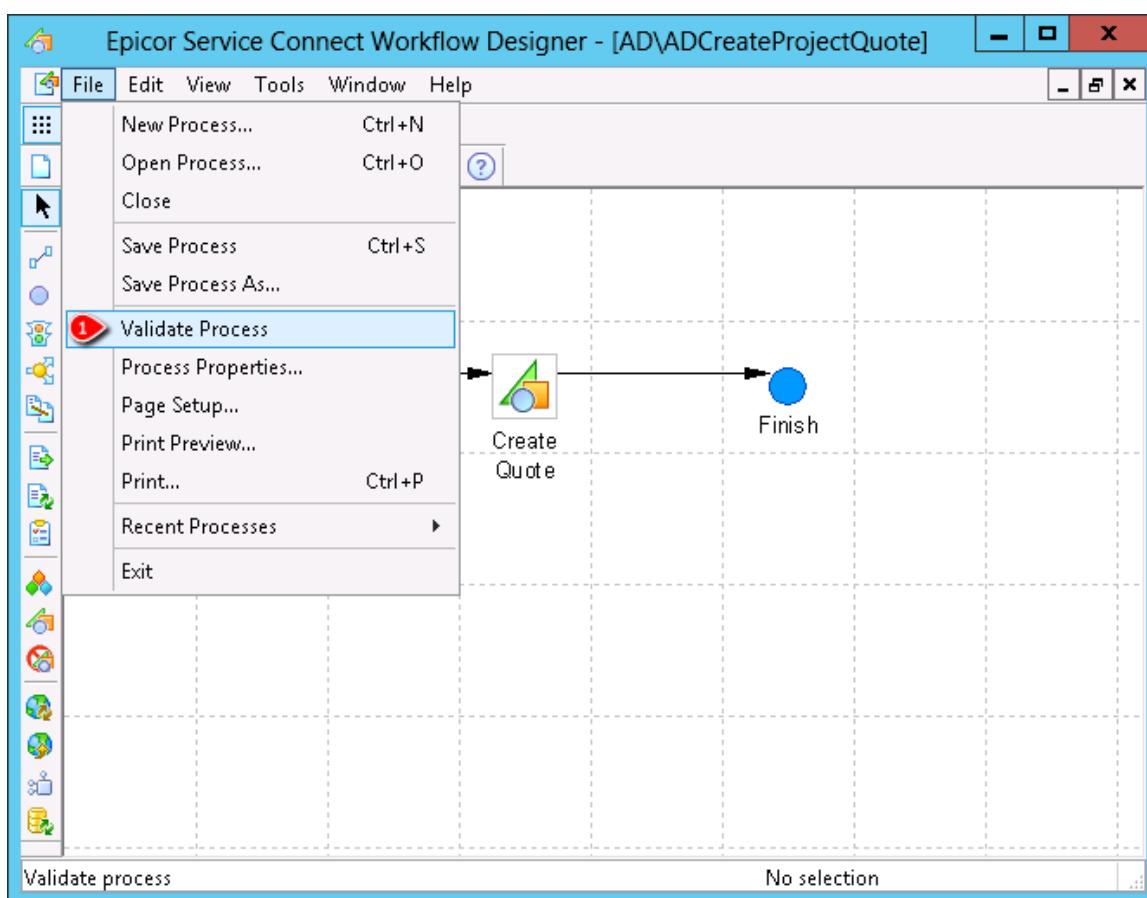
- The process has one and only Start element.
- The process has one and only Finish element.
- There is one and only link from the Start element and no links to it.
- There is at least one link to the Finish element and no links from it.
- Each element has at least one link to it and one link from it.
- Each Sub-Workflow element references an existing workflow.
- Each Conversion element contains wfl/SubWF mapping in order to continue workflow execution after Task.
- Inherited message extensions and process variables are not overwritten by the local ones specified in this workflow.
- DBOperation element has no empty statements.

If any of the conditions is not met, a warning message is displayed. Each warning displays the type and name of the workflow element that is not set up properly.

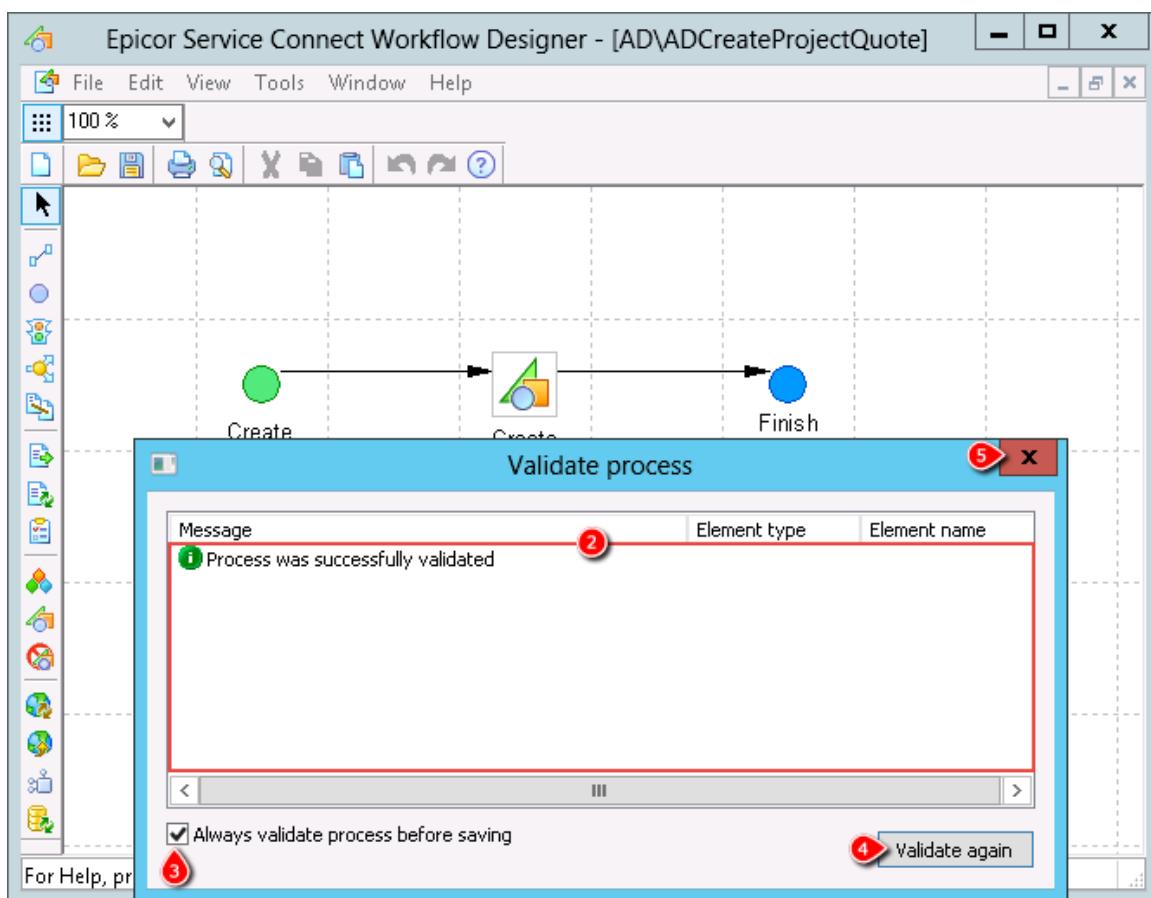
By default, every time you save a workflow, it is validated automatically.

To validate a workflow

- From the **File** menu, select **Validate Process**.



2. In the **Validate process** window, view the validation result.

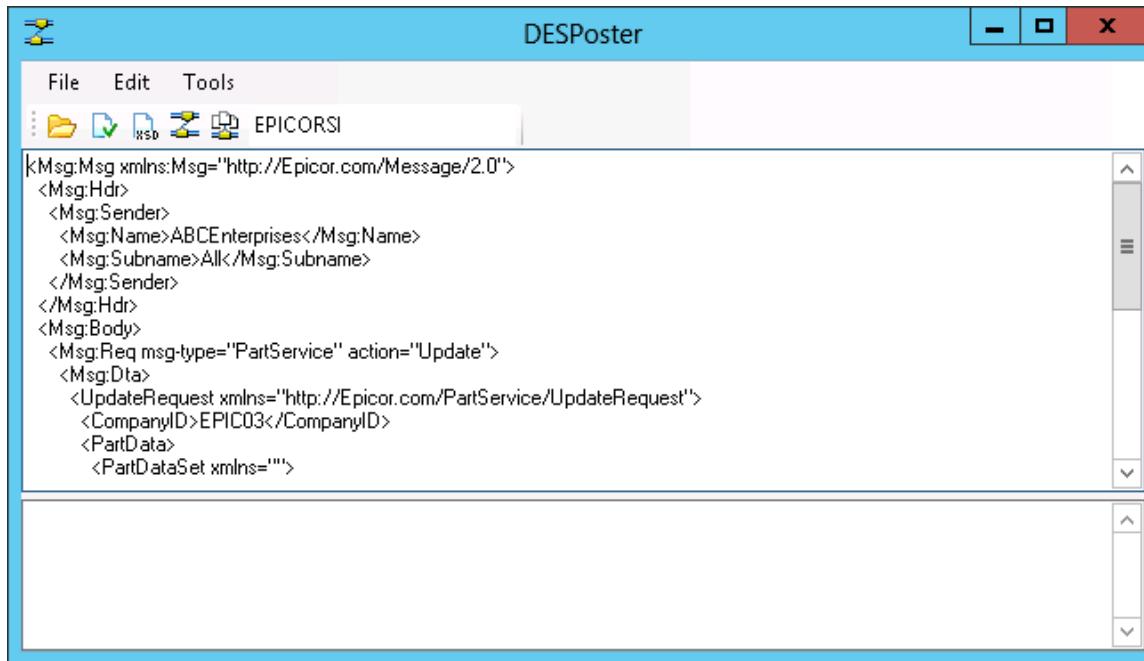


3. If you want to validate your workflows every time you save them, verify the **Always validate process before saving** check box is selected.
4. To validate the workflow again, click the **Validate again** button.
5. Close the **Validate process** window.

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 customize 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 ERP Business Process Management
- 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 the Workflow Designer chapter 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.

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 Service Connect workflows that interact with Epicor ERP.

- To make Epicor ERP .NET Business Objects (BOs) available to Service Connect, import the .NET BOs as Service Connect .NET References.
- Create document schemas.
- Create message attributes.
- Create input and output channels.

Epicor ERP .NET Business Objects Setup

No additional installation steps are required to use Epicor 10 .NET Business Objects (BOs). You import them as .NET References in the ESC Administration Console. You can then use these references from within a Service Connect workflow.

A business object contains the code that runs a business process. There are hundreds of .NET BOs that control the Epicor 10 application functionality.

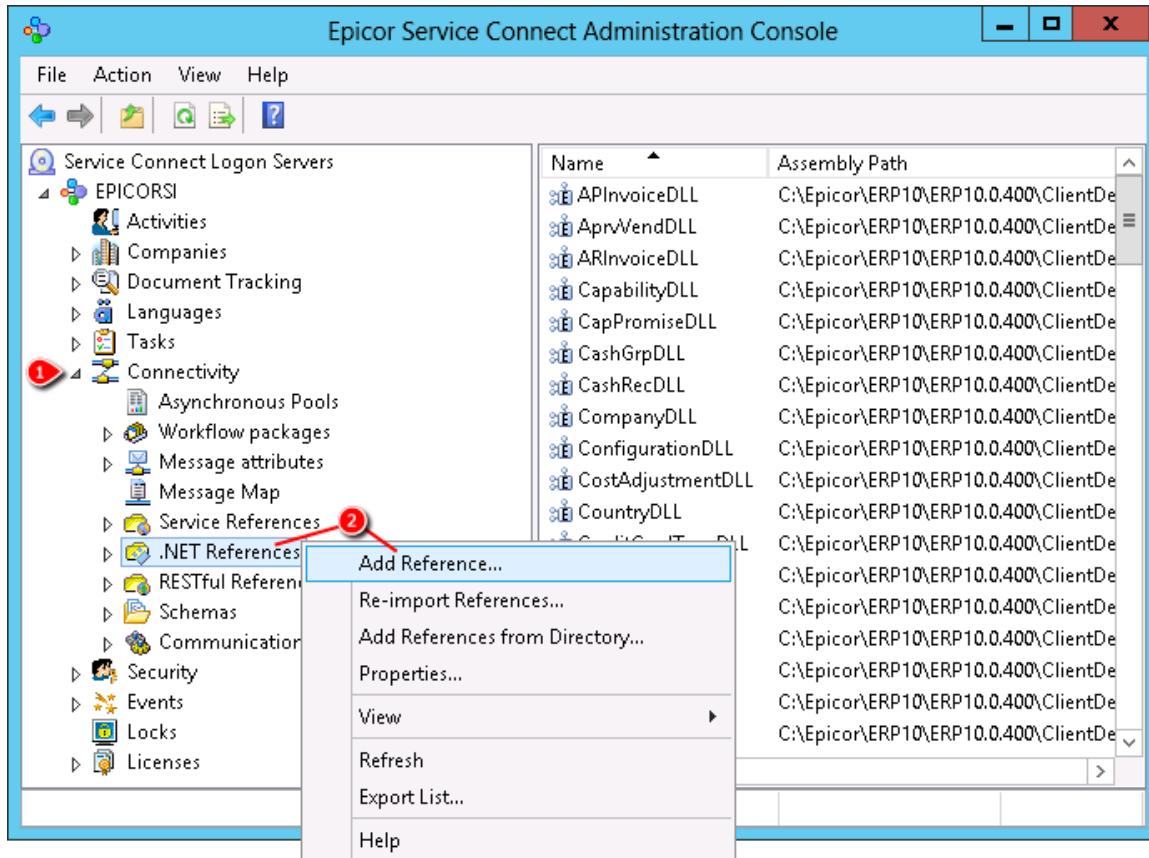
You can call Epicor 9.05 and Epicor 10 .NET assemblies from Service Connect. Calling .NET assemblies improves performance because in this case you call Epicor .NET BOs directly.

.NET References are configured links to specific .NET assemblies. After you add a .NET Reference in the ESC Administration Console, you can call .NET object methods within the Workflow Designer.

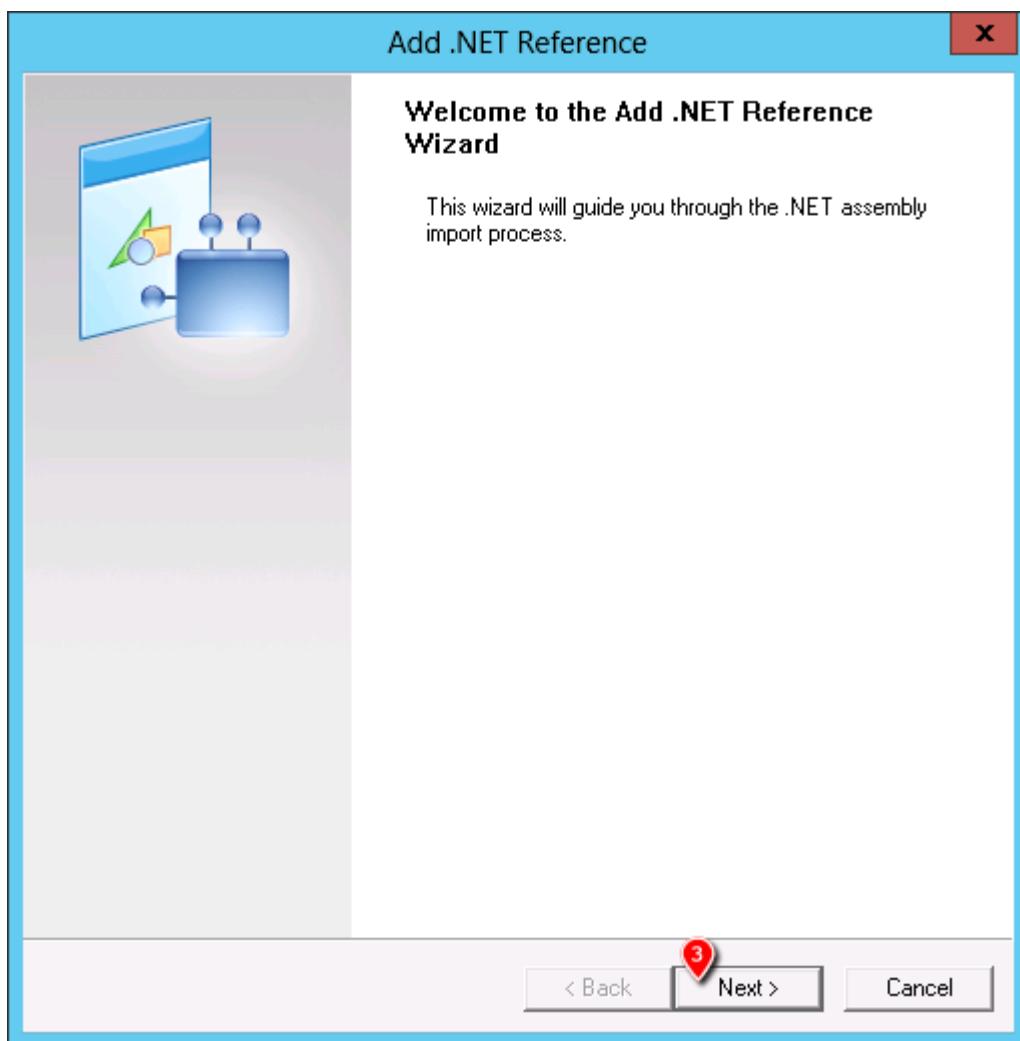
Import .NET Reference into Service Connect

To import .NET References:

1. In the Epicor Service Connect Administration Console, expand the **Connectivity** node.

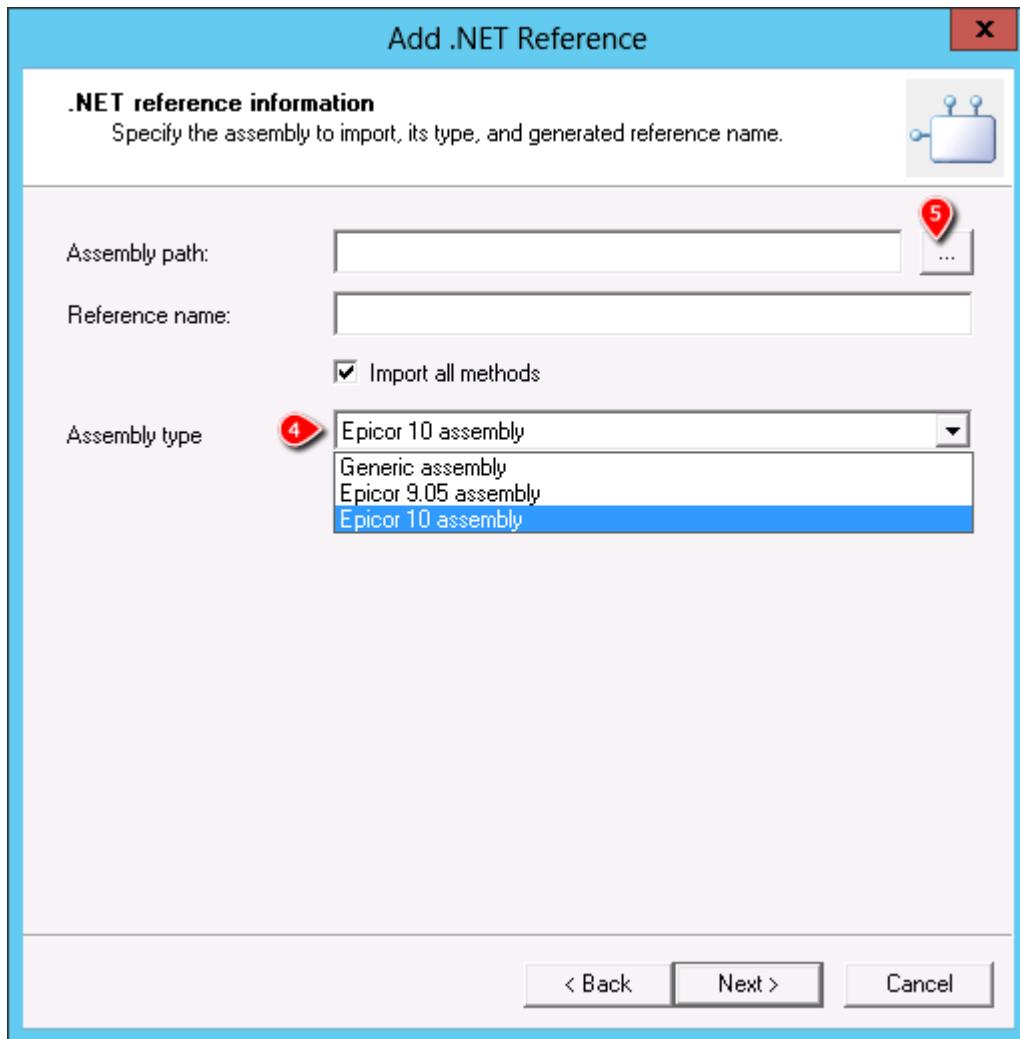


2. Right-click the **.NET References** node and select **Add Reference...**.
3. On the **Add .NET Reference** wizard welcome screen, click **Next**.

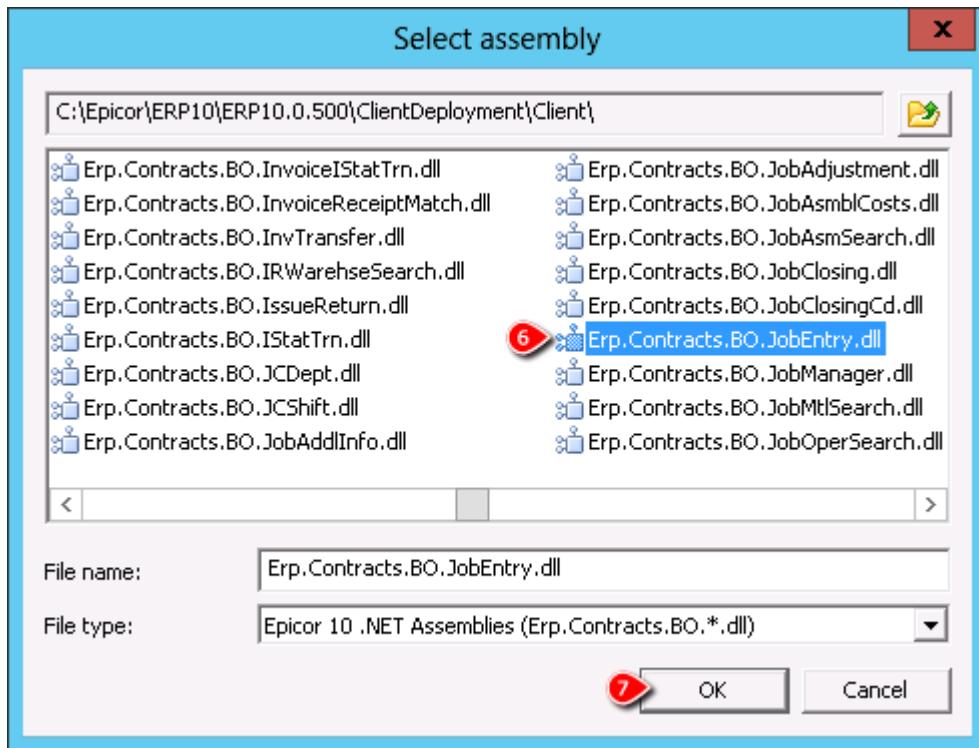


4. On the **.NET reference information** page, click the **Assembly type** drop-down list and select **Epicor 10 assembly**.

Select this type when you add a reference to Epicor 10 Business Objects.



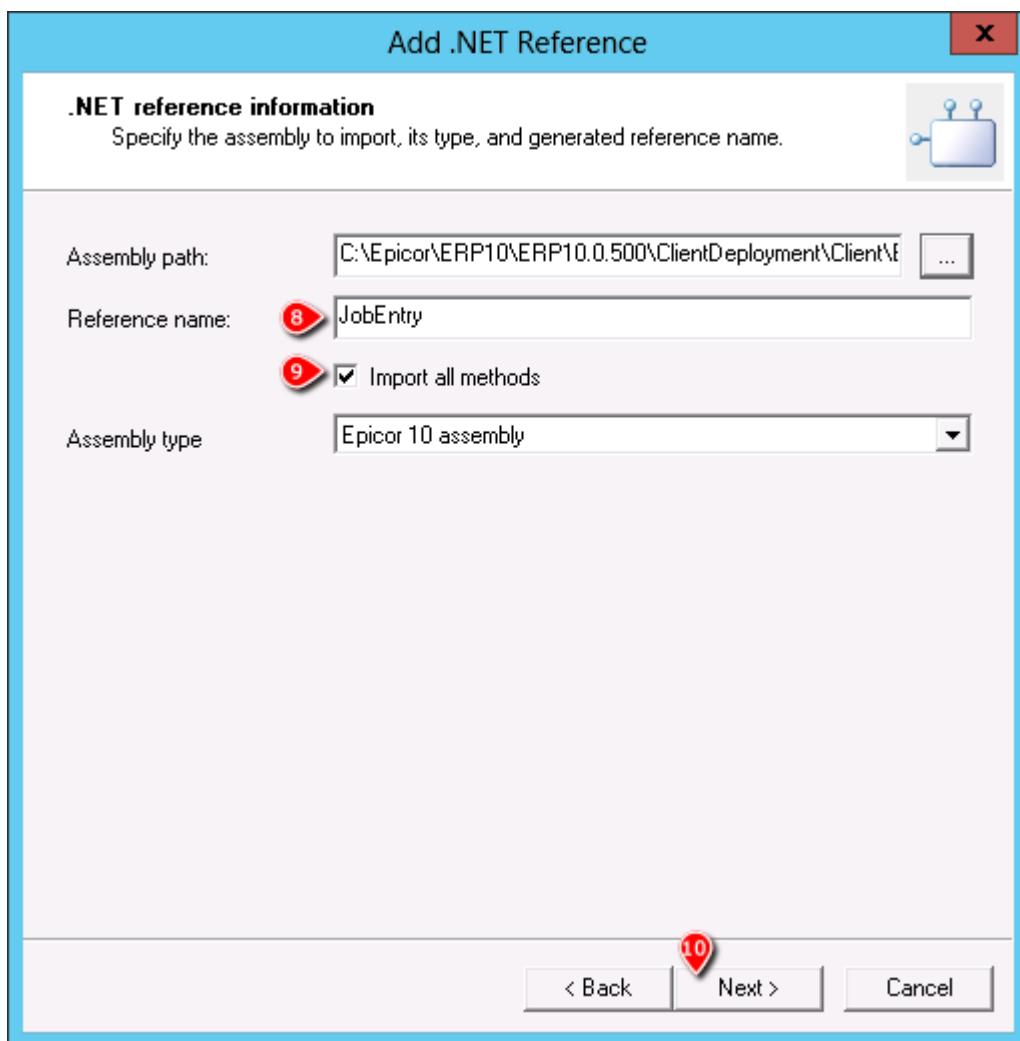
5. Next to the **Assembly path** field, click the ... (browse) button.
6. In the **Select assembly** window, navigate to and select the .NET assembly file.



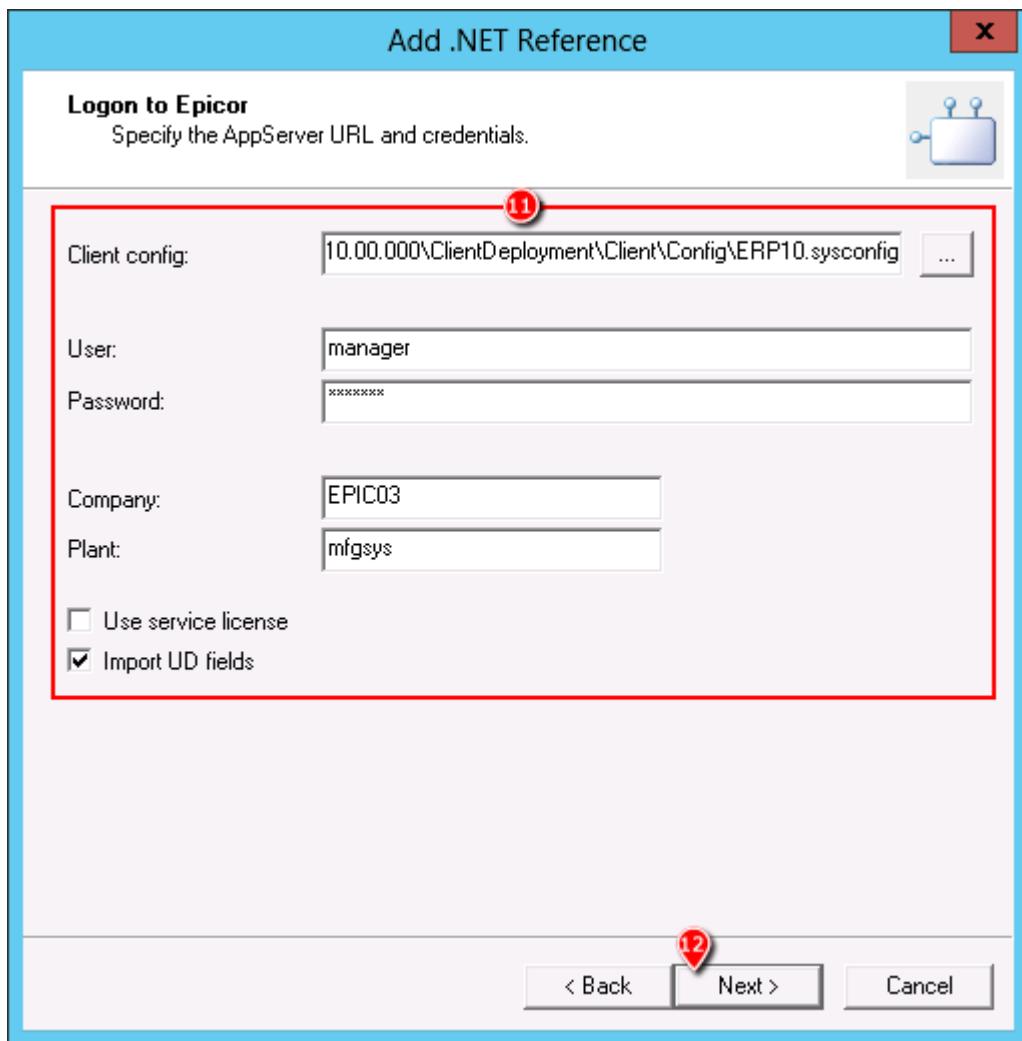
7. Click OK

The **Assembly path** field now displays the path to the assembly. This is the path on the Service Connect server.

8. Notice the **Reference name** field automatically displays the .NET reference name. This name will be displayed in the imported .NET references list.

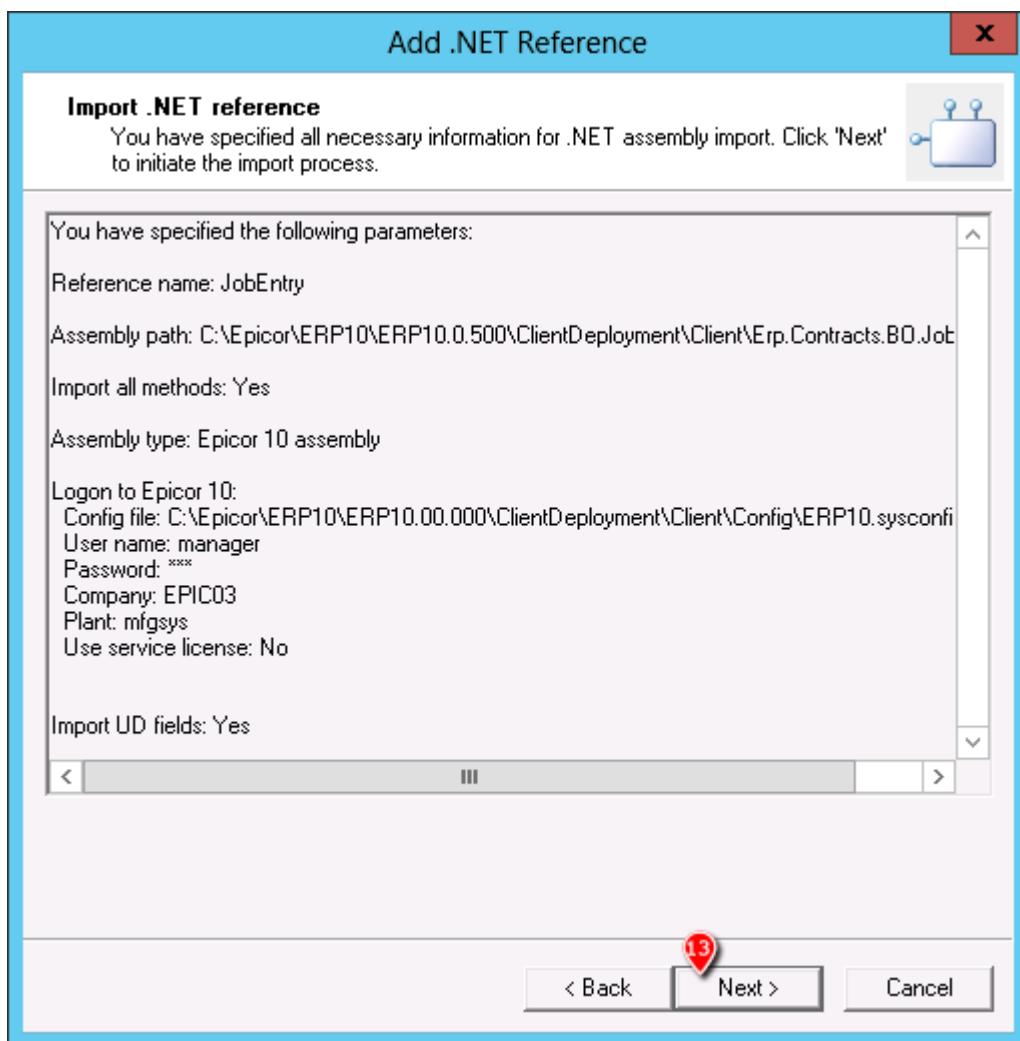


9. Select the **Import all methods** check box to import all .NET reference methods.
If you do not select the Import all methods check box, you can add the methods one-by-one later and configure them individually.
10. Click **Next**.
11. Enter the logon credentials applicable to your environment.

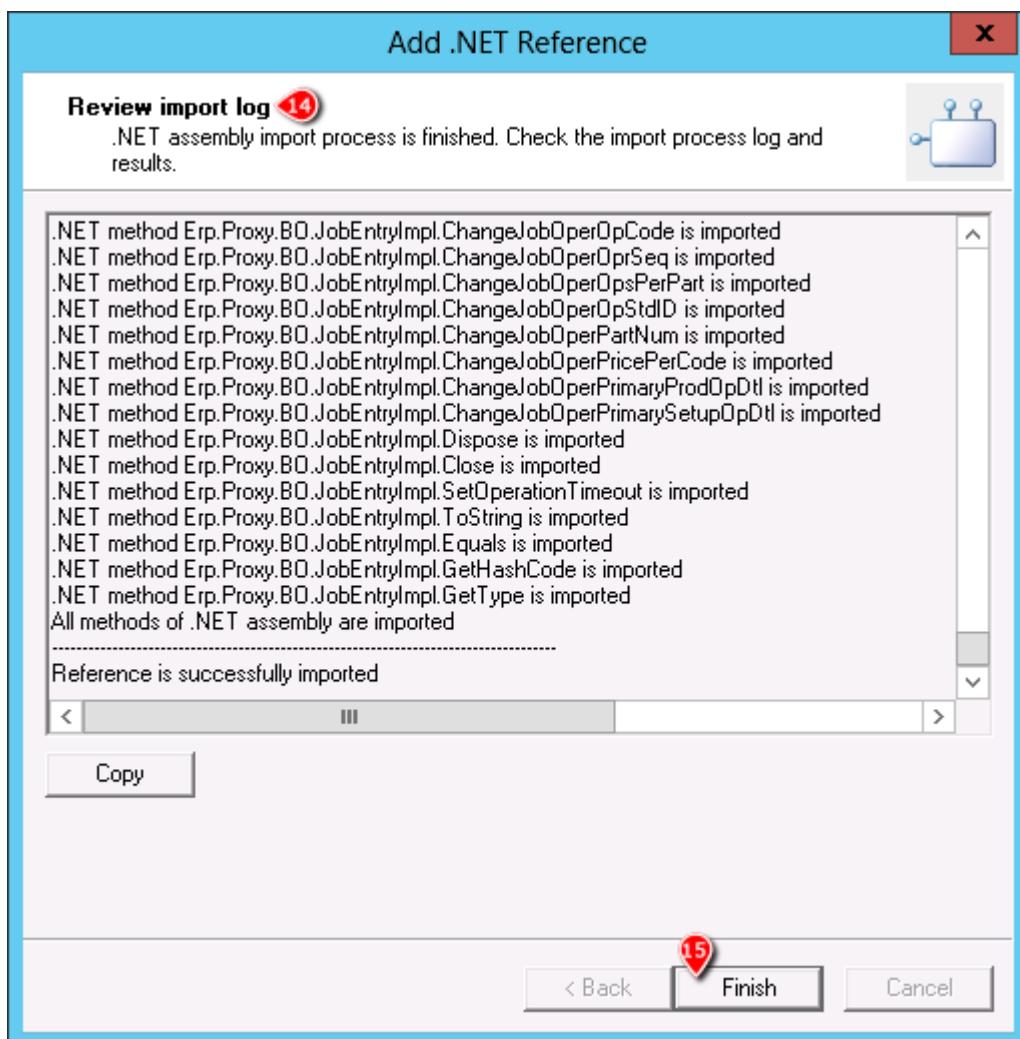


For details, refer to the Epicor 10 .NET Reference Setup procedure in the Connectivity Components chapter.

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.



If the .NET Reference import failed, you can click the **Back** button to go back and adjust the reference settings, or, you can click **Cancel** to abort the import operation.

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

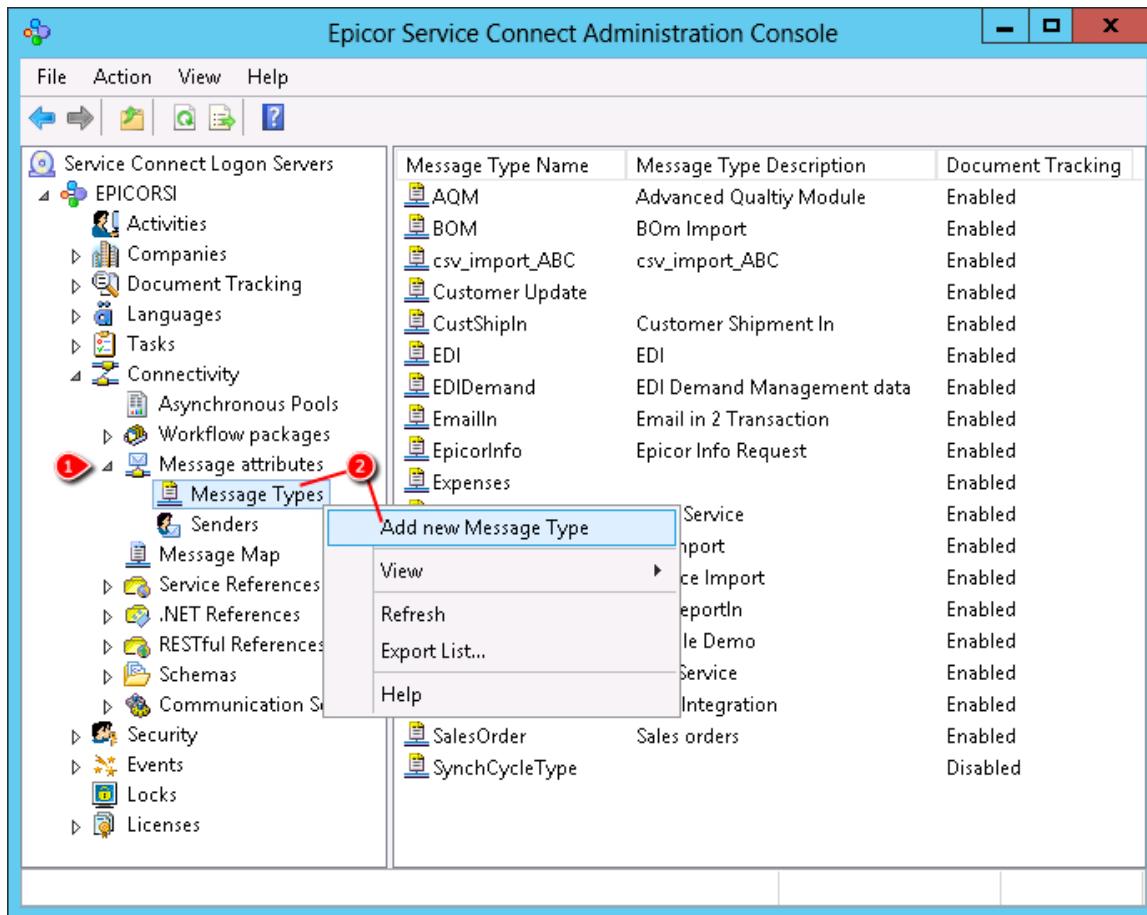
You can use the Test context menu item on .NET reference methods to call particular methods from the ESC Administration Console to ensure you specified reference settings correctly. For example, you can test the GetList method to verify response accuracy of an imported reference. For detailed instructions on how to test a method, refer to the Connectivity Components Chapter.

Add a Message Type

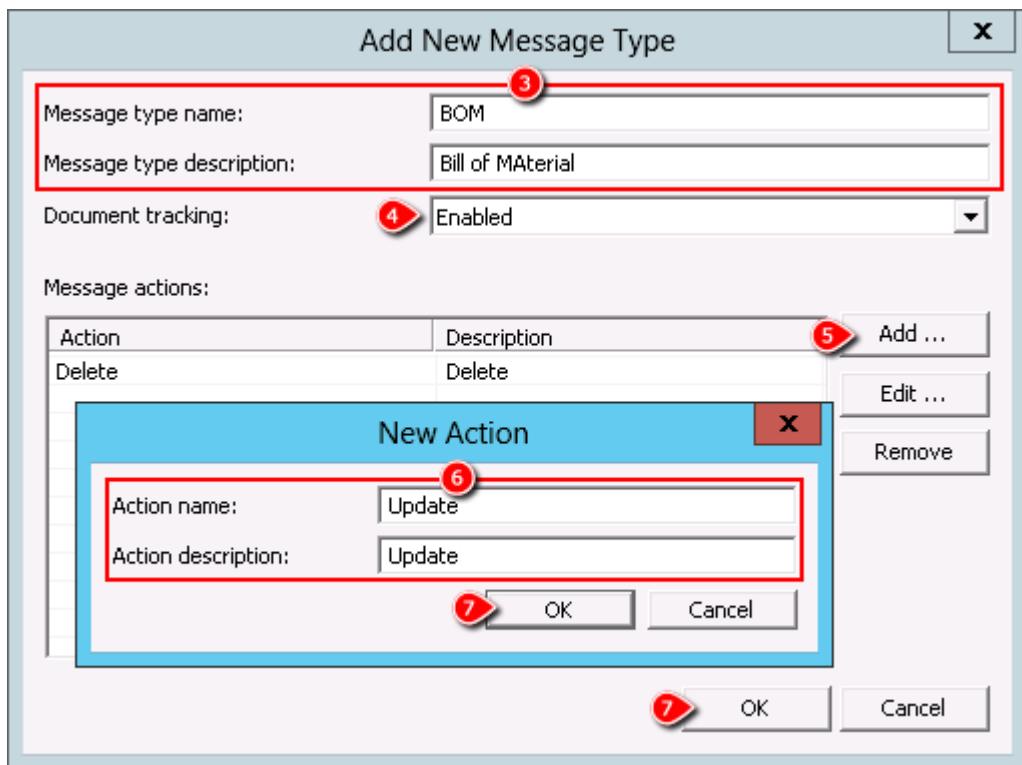
Use Message Types to classify the different documents you plan to process in Service Connect.

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 **Add New Message type** window, enter a **Message type name** and **Message type description** that reflect the types of documents the workflow will handle.



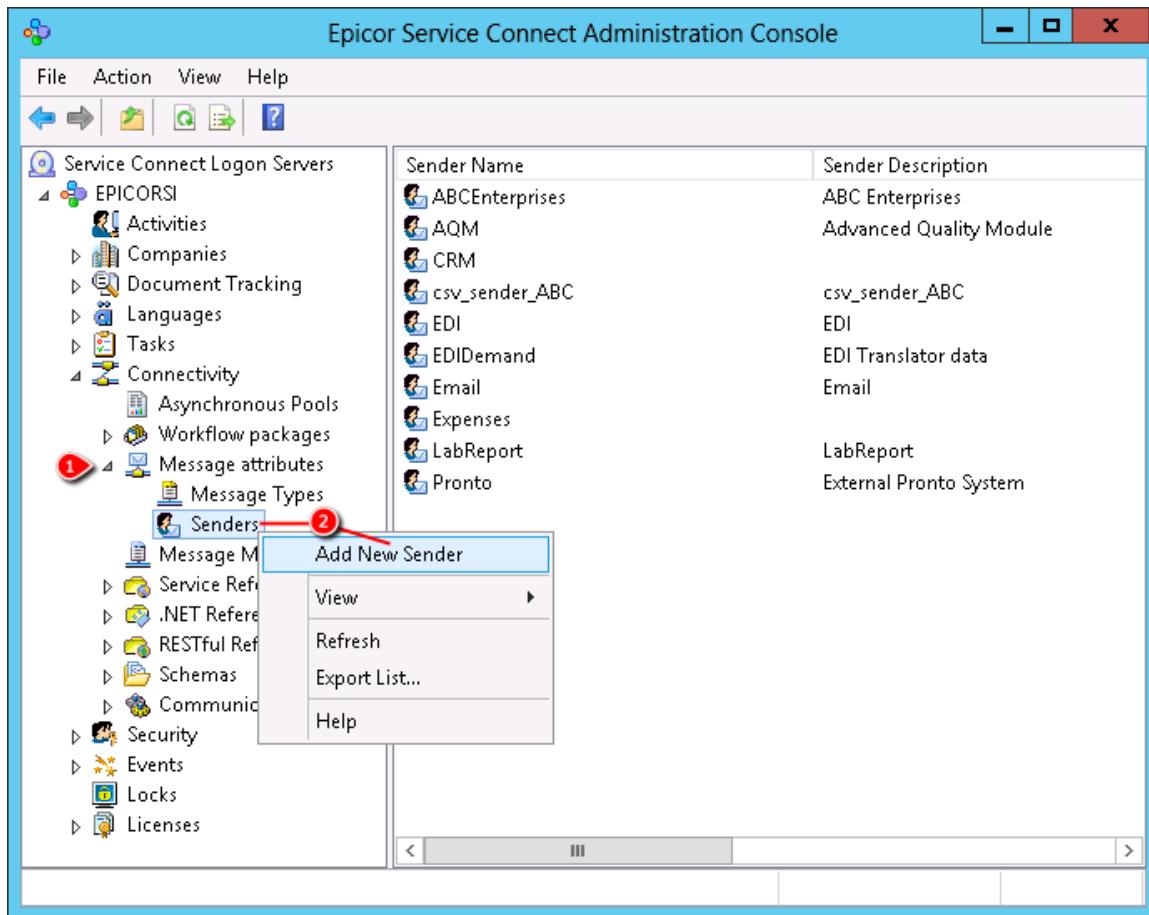
4. Select **Enabled** in the **Document tracking** field to be able to view the processing of documents associated with the message type.
5. Click **Add**.
6. In the New Action window, enter an **Action** and **Description** that indicate the action the workflow will take when a document assigned to this message type enters the system.
7. Click **OK** until you exit all dialog boxes.

Add a Sender

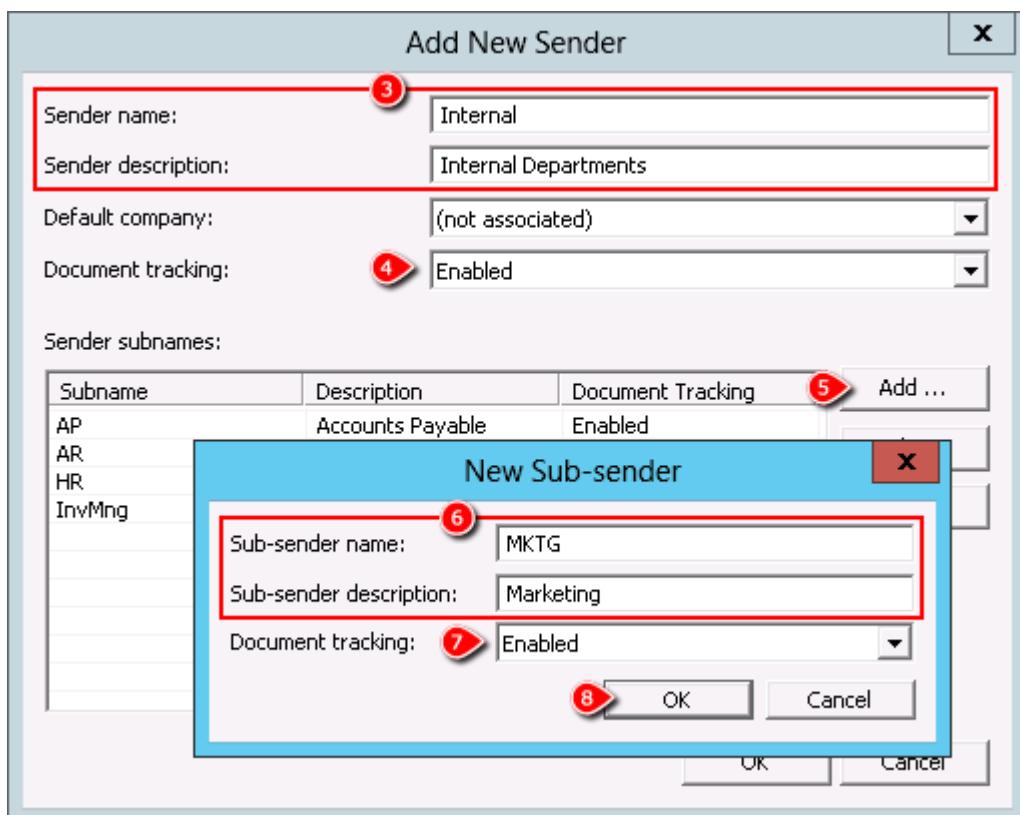
Use Senders to define the origin of the documents sent to Service Connect. The origin can be an application name, your company name, or a computer name. Within a Sender, you can also define sub-names to more narrowly define a document's origin.

To add a Sender:

1. In 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 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.

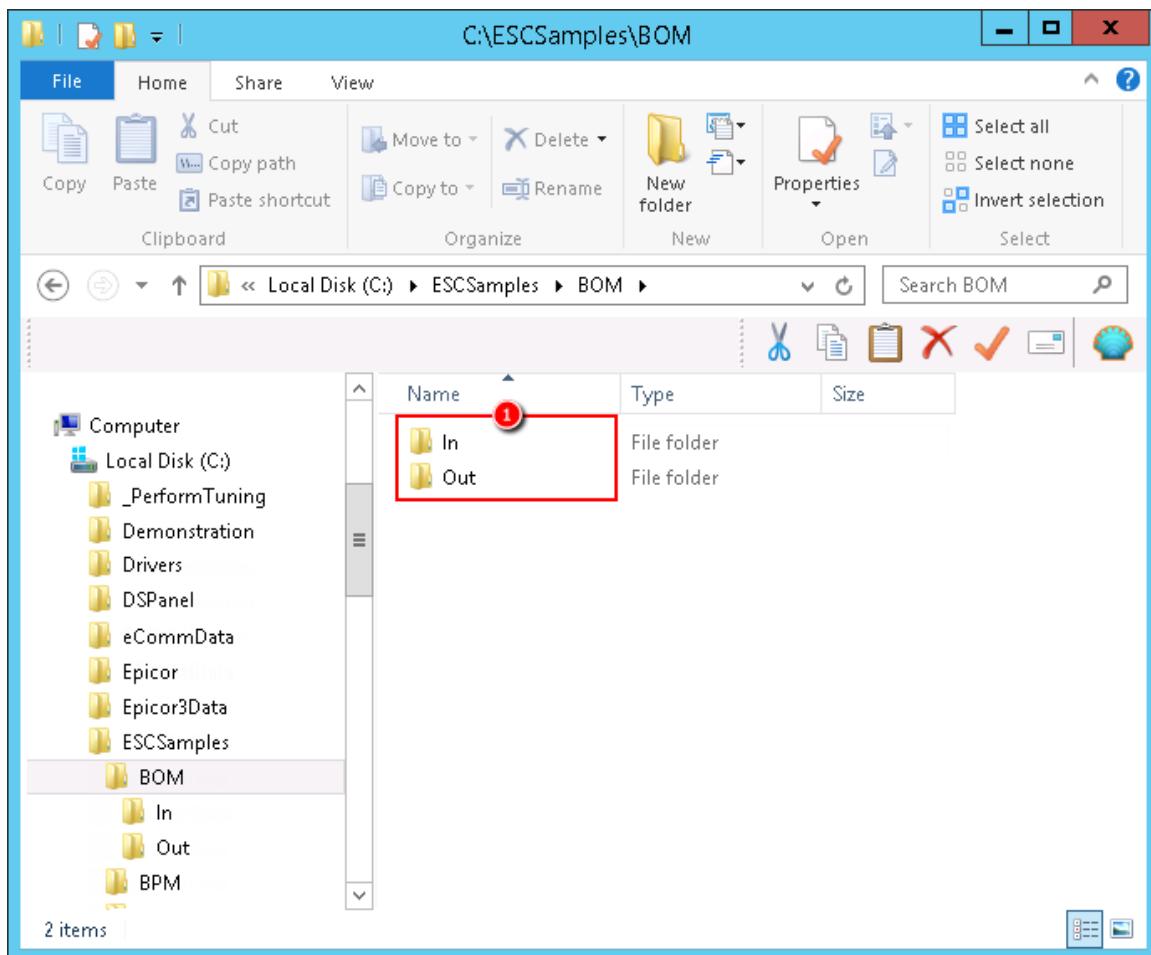
Create Input and Output Channels

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 the Connectivity Components chapter.

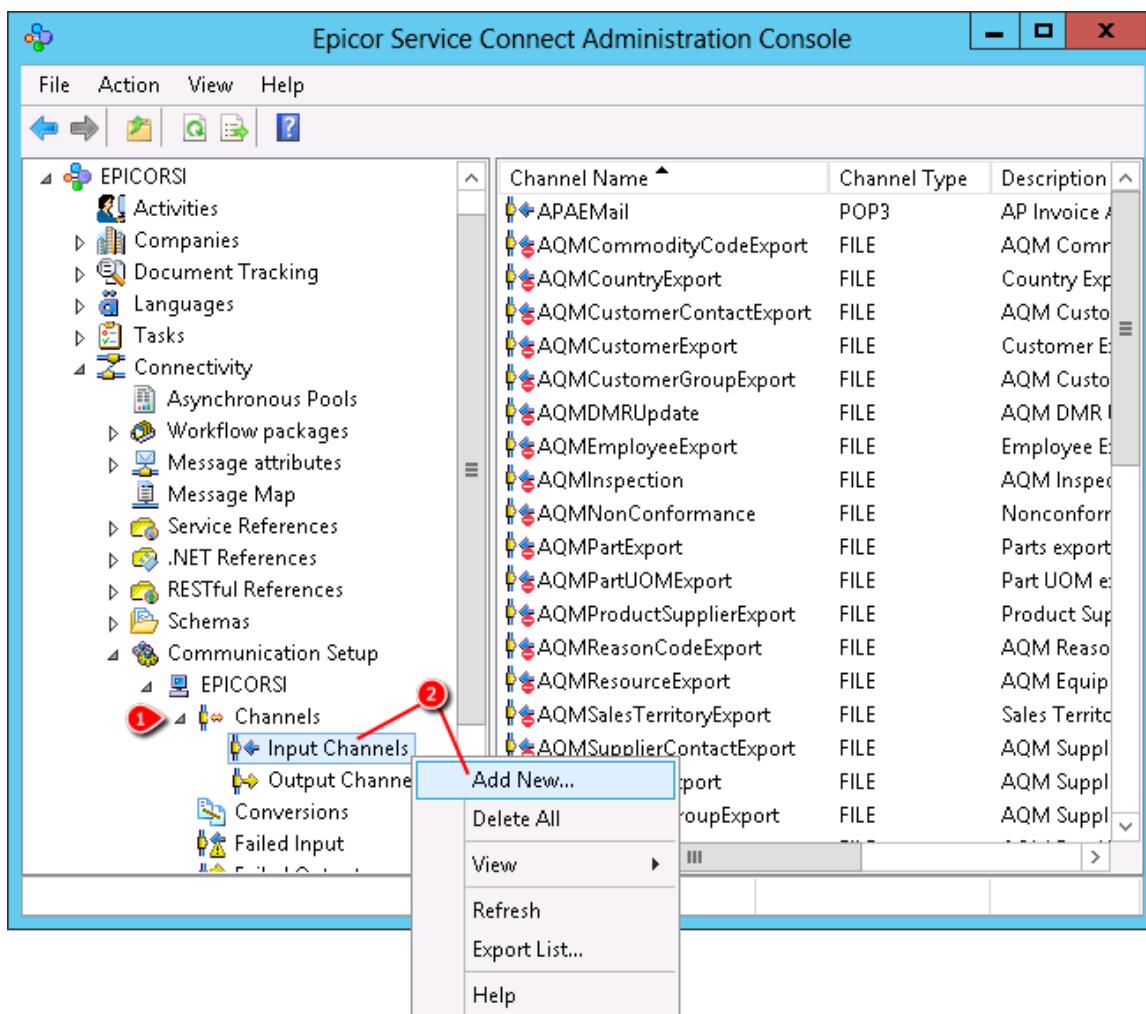
Create an Input Channel

To add an input channel:

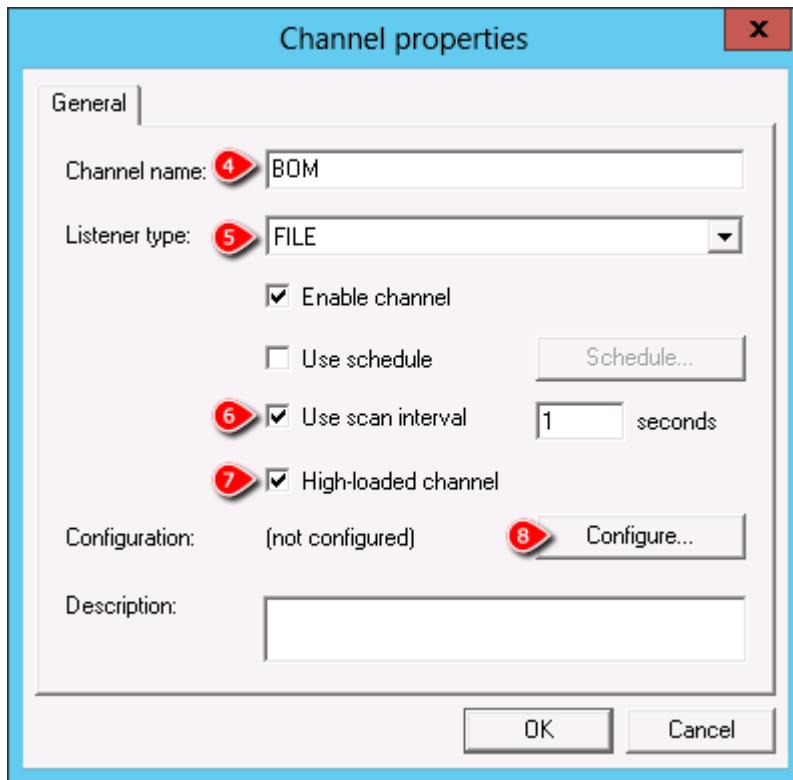
1. Open Windows Explorer and create a folder on a file system the Service Connect server can access. For example, create **C:\ESCSamples\BOM\In** and **C:\ESCSamples\BOM\Out**.



2. In the Epicor Service Connect Administration Console, expand the following nodes: **Connectivity > Communication Setup > < machine name> > Channels**.

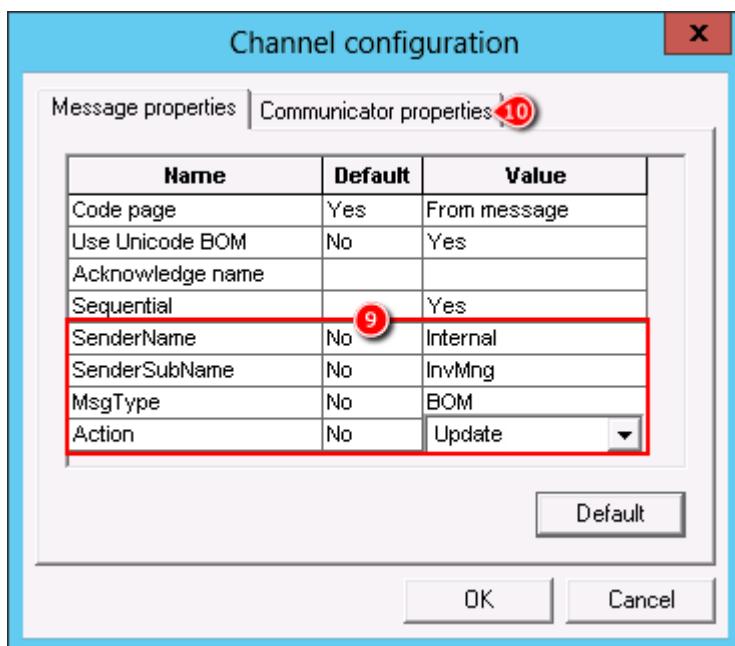


3. Right-click **Input Channels** and select **Add New**.
4. In the **Channel properties** window, enter a **Channel name**.



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**.

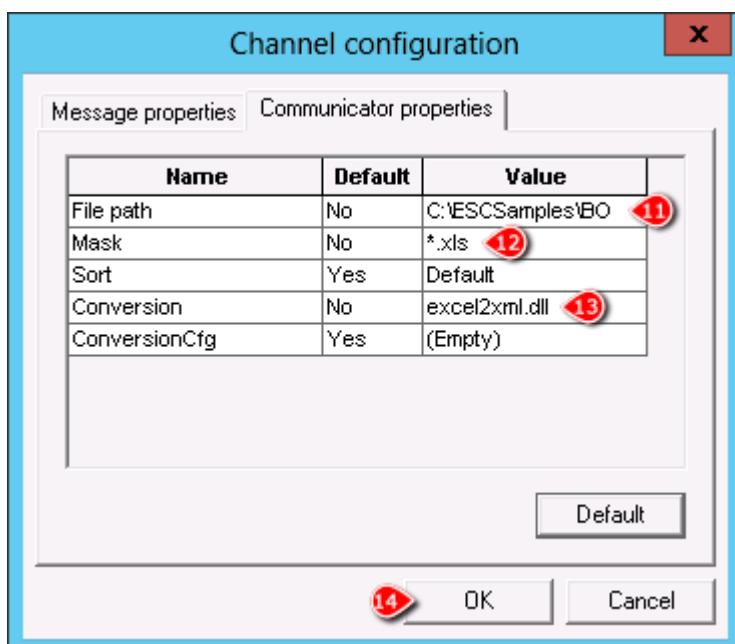
When this option is selected, the system will try to allocate a separate thread for the channel.
Review the Communication Setup and Channels sections in the Connectivity Components chapter for more information on how to allocate a separate thread for the channel.
8. Click **Configure**.
9. In the **Channel 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. Open the **Communicator properties** tab.
11. Enter the **File path** to the new folder created at the beginning of this procedure for the file input.

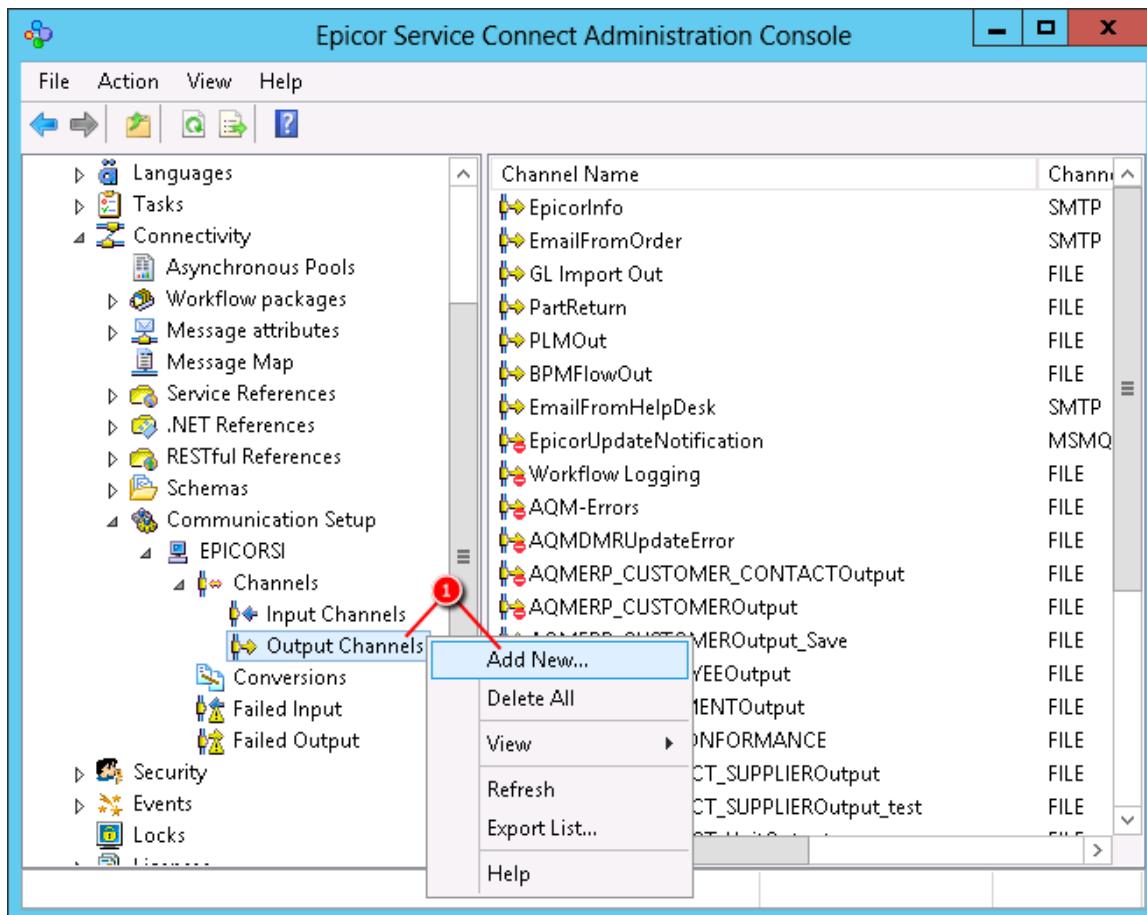


12. Enter ***.xls** as the **Mask**.
13. Select **excel2xml.dll** as the **Conversion**.
14. Click **OK** until you exit all dialog boxes.

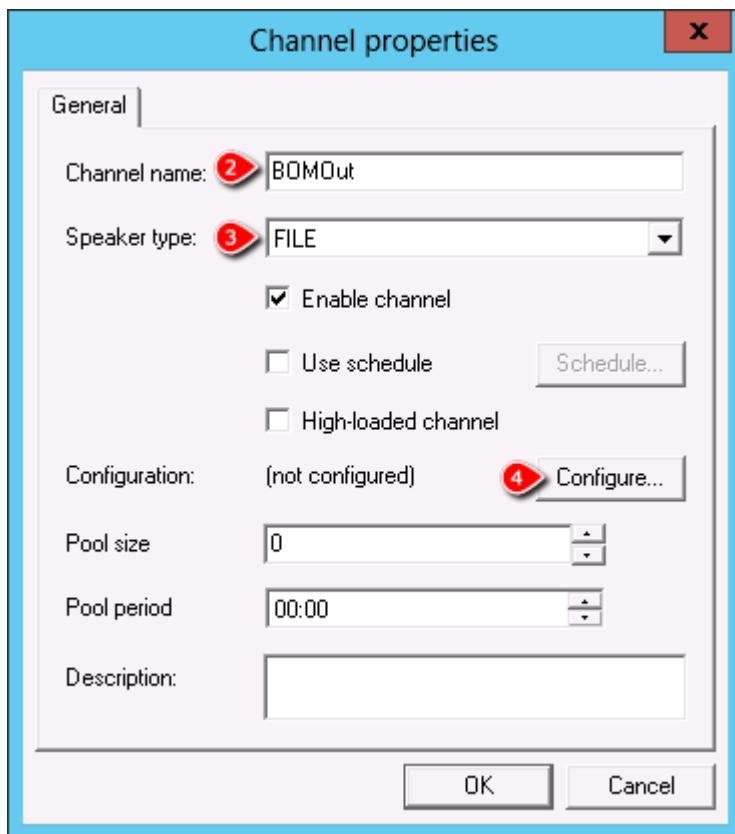
Create an Output Channel

To add an output channel:

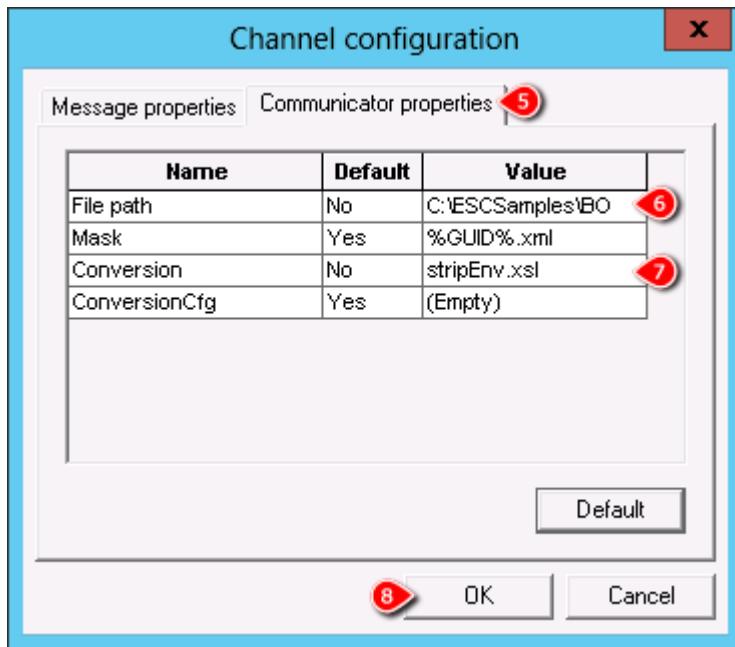
1. Right-click **Output Channels** and select **Add New...**.



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, open the **Communicator properties** tab.



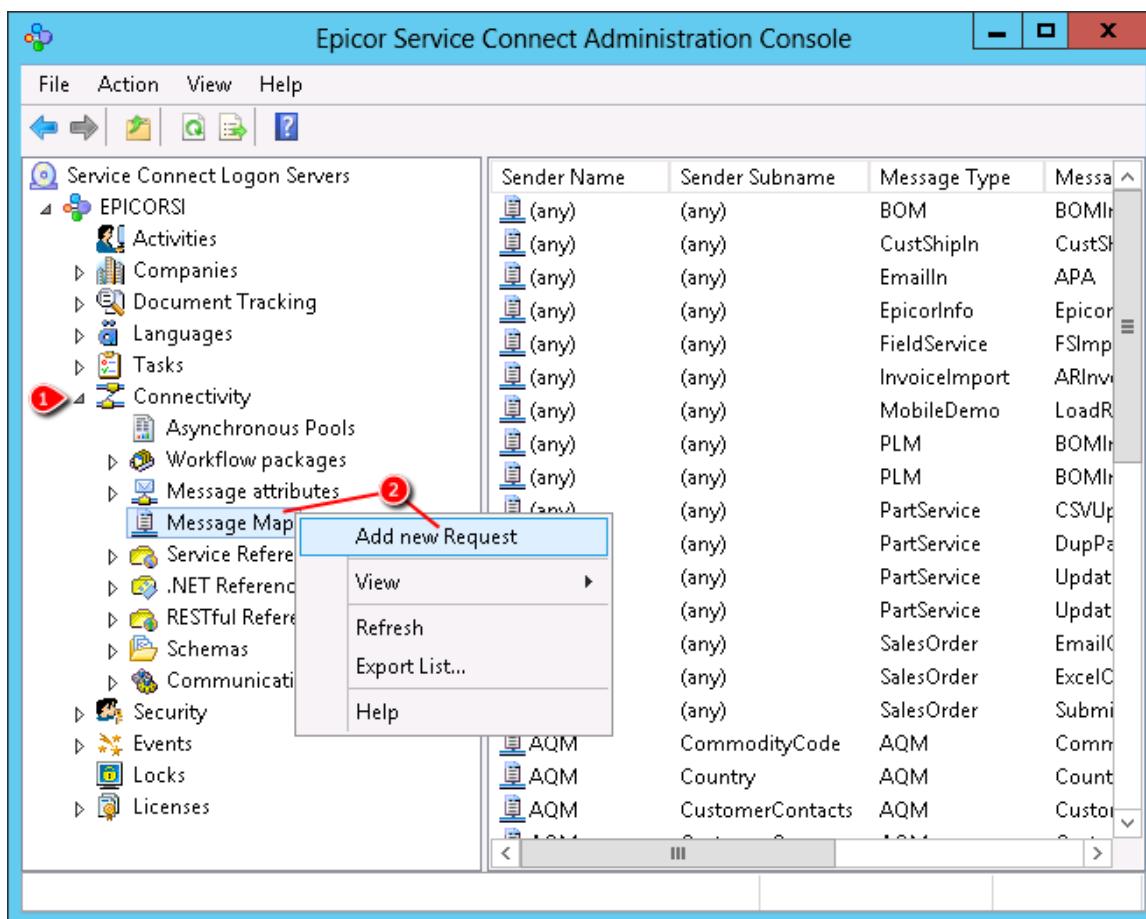
6. Enter the **File path** to the folder where you want to publish the output.
7. Select **stripEnv.xsl** as the **Conversion**.
8. Click **OK** until you exit all dialog boxes.

Create the Message Map

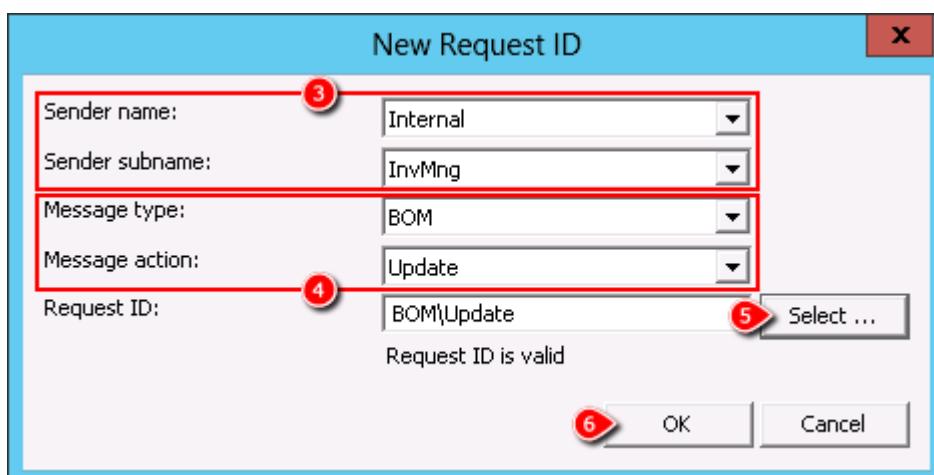
Message maps are used to route a document once the document is processed in an input channel. Service Connect uses the message map to send the document to the appropriate workflow based on the Sender and Message Type.

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.

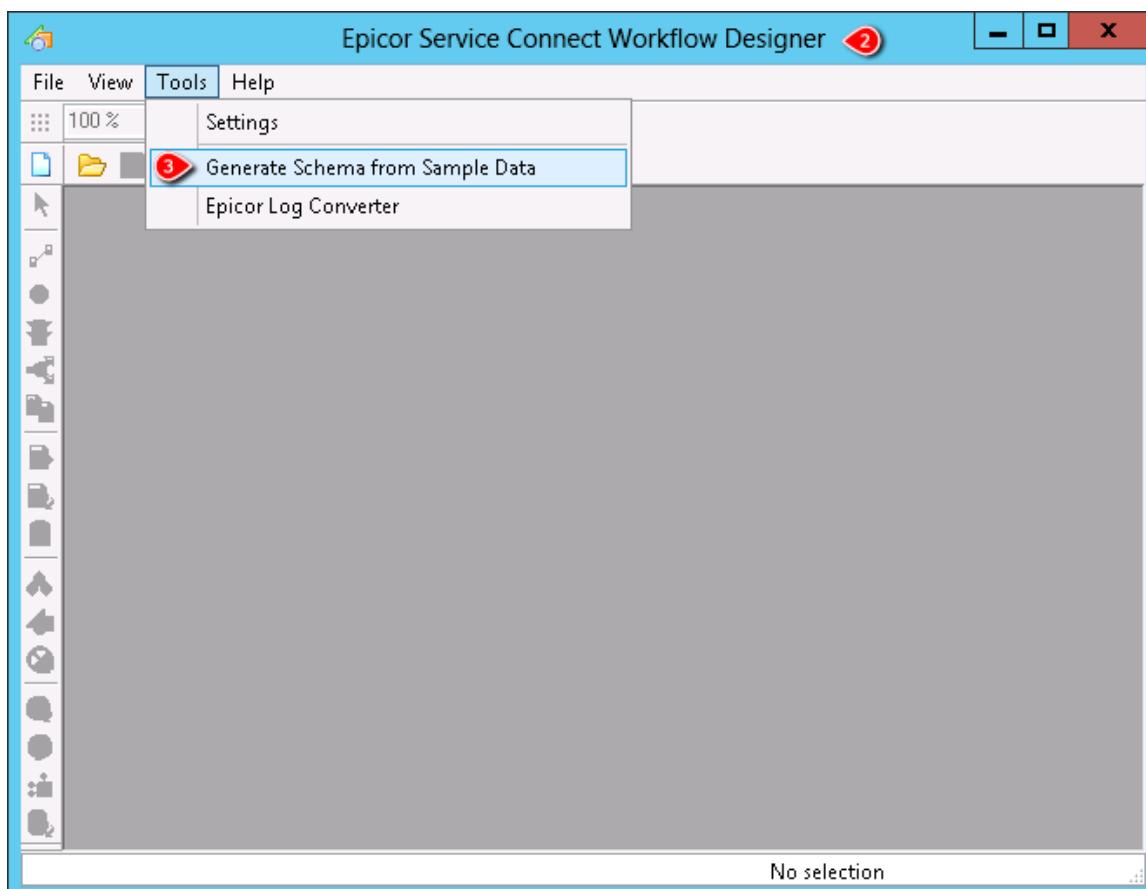
Service Connect can only process Excel files without protection.

To generate the schema and import it into Service Connect:

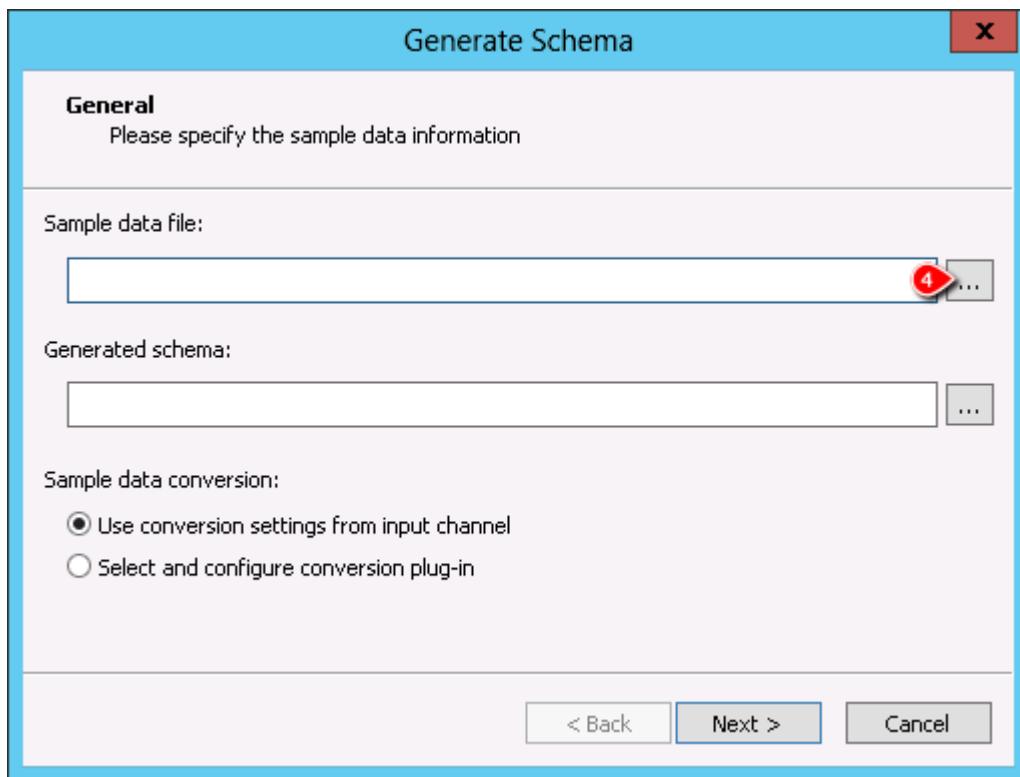
1. Create or locate a spreadsheet that you can use as sample similar to the one shown here.

	A	B	C	D	E	F	G	H	I	J
1	GROUP_ID	GROUP_ID_DESC	ASSEMBLY_PART_NBR	ASSEMBLY_PART_DESC	REVISION	COMPONENT_NBR	COMPONENT_DESCR	SEQ_NBR	QTY	DUE_DATE
2	E-200	E-200 Sage Rods	P1010	Sage 4wt 8 1/2 Rod	1	E200-P1010-ADD	Mandrell	10	1	9/4/2015
3	E-200	E-200 Sage Rods	P1010	Sage 4wt 8 1/2 Rod	1	E200-P1010-BMI	Carbon Fibre	20	1	9/4/2015
4	E-200	E-200 Sage Rods	P1010	Sage 4wt 8 1/2 Rod	1	E200-P1010-PRI	Guides	30	1	9/4/2015
5	E-200	E-200 Sage Rods	P1010	Sage 4wt 8 1/2 Rod	1	E200-P1010-PIC	Ribbon	40	1	9/4/2015

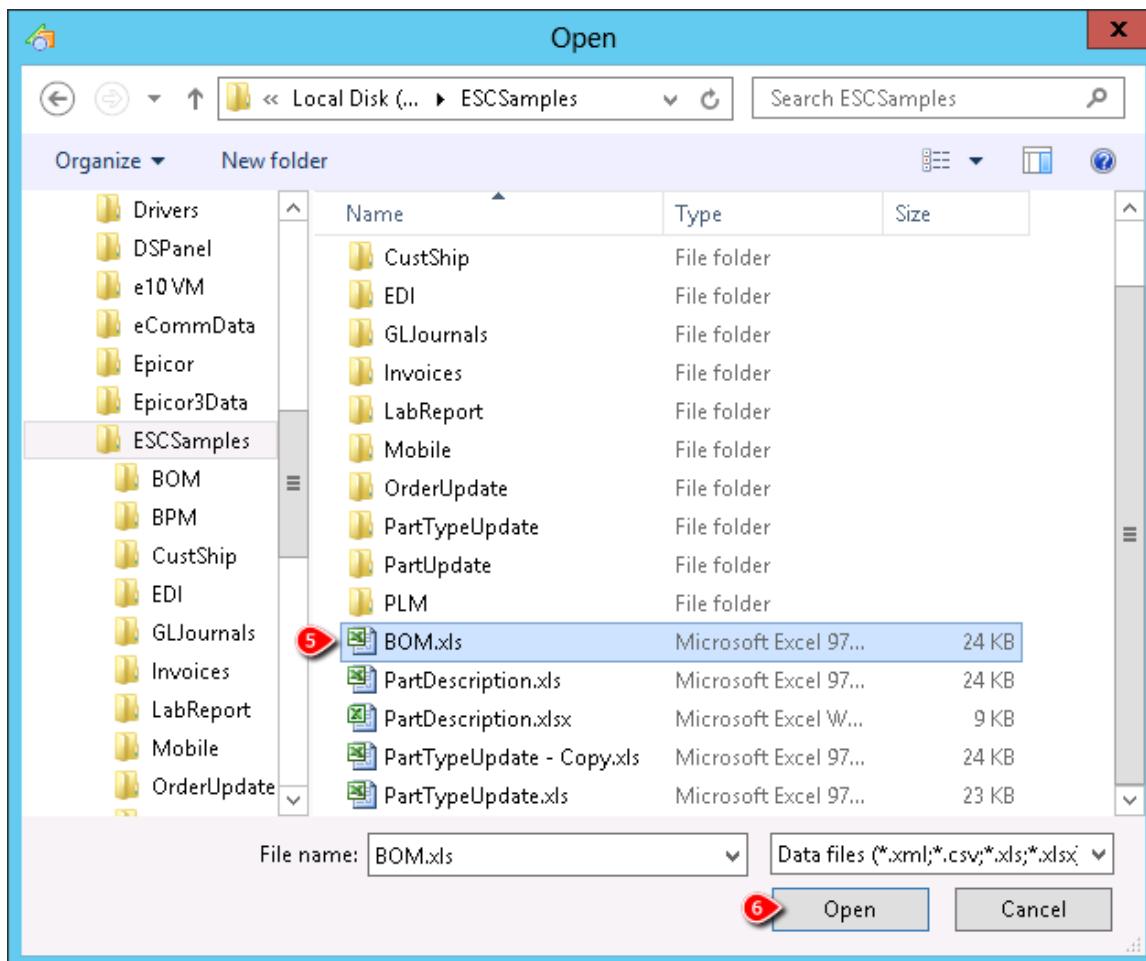
2. Log into the **Epicor Service Connect Workflow Designer**.



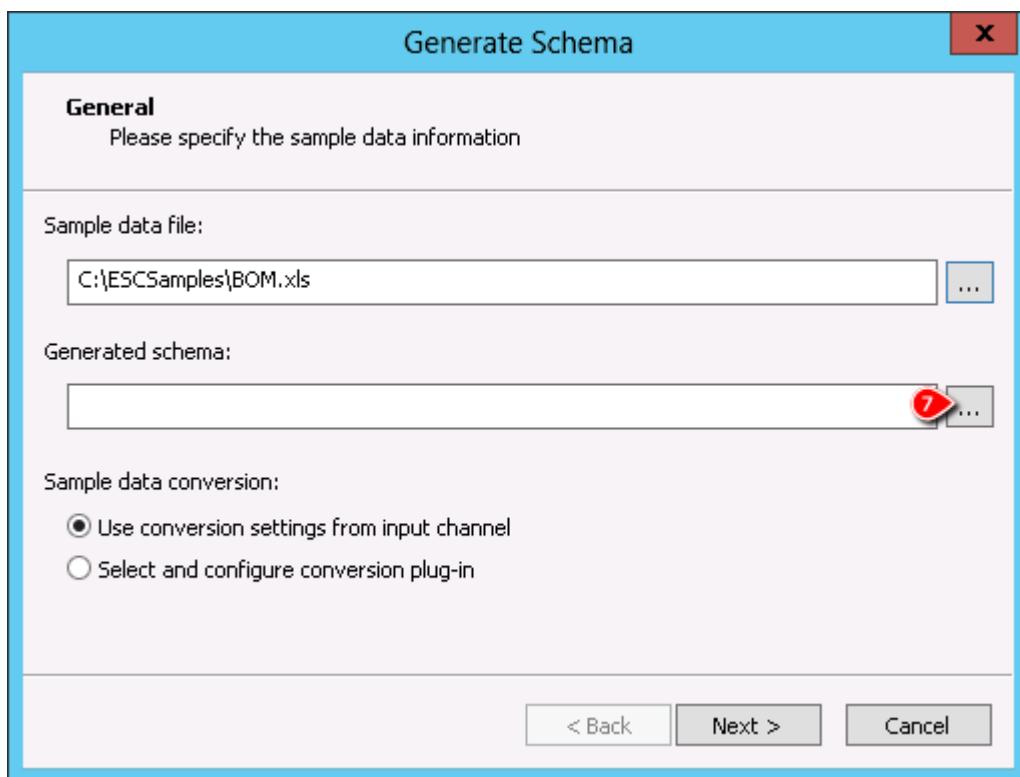
3. From the **Tools** menu, select **Generate Schema from Sample Data**.
4. The **Generate Schema** window displays. Next to the **Sample data file** field, click the browse button.



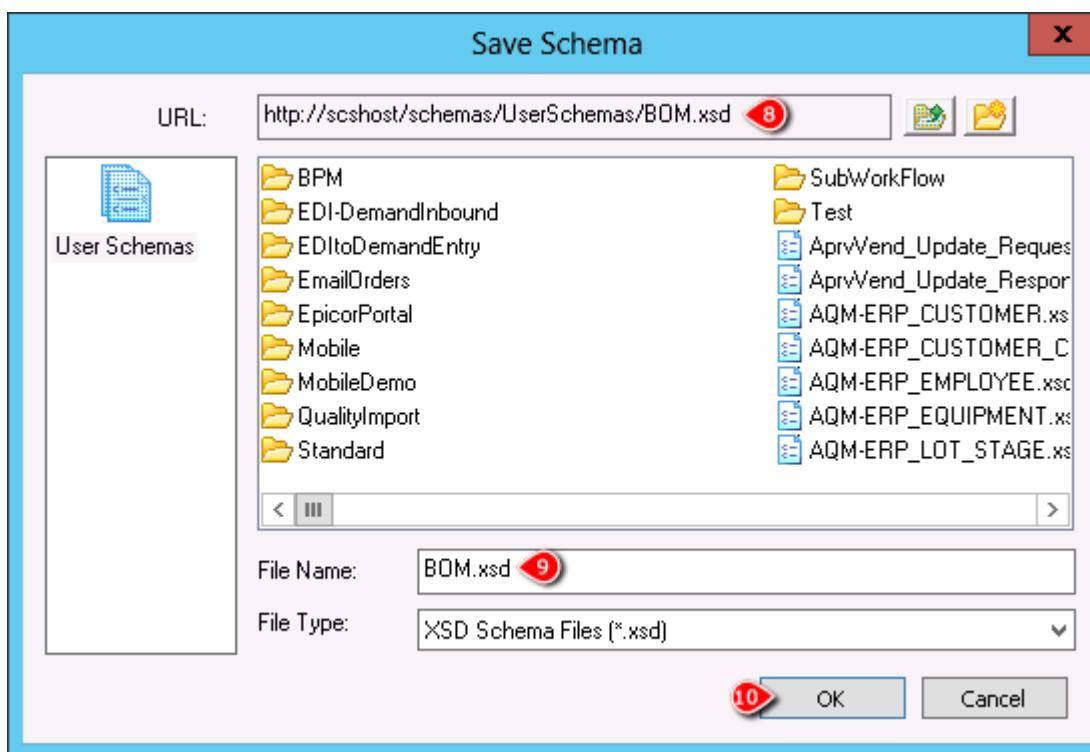
5. In the standard **Open** window, browse to and select the sample Excel file.



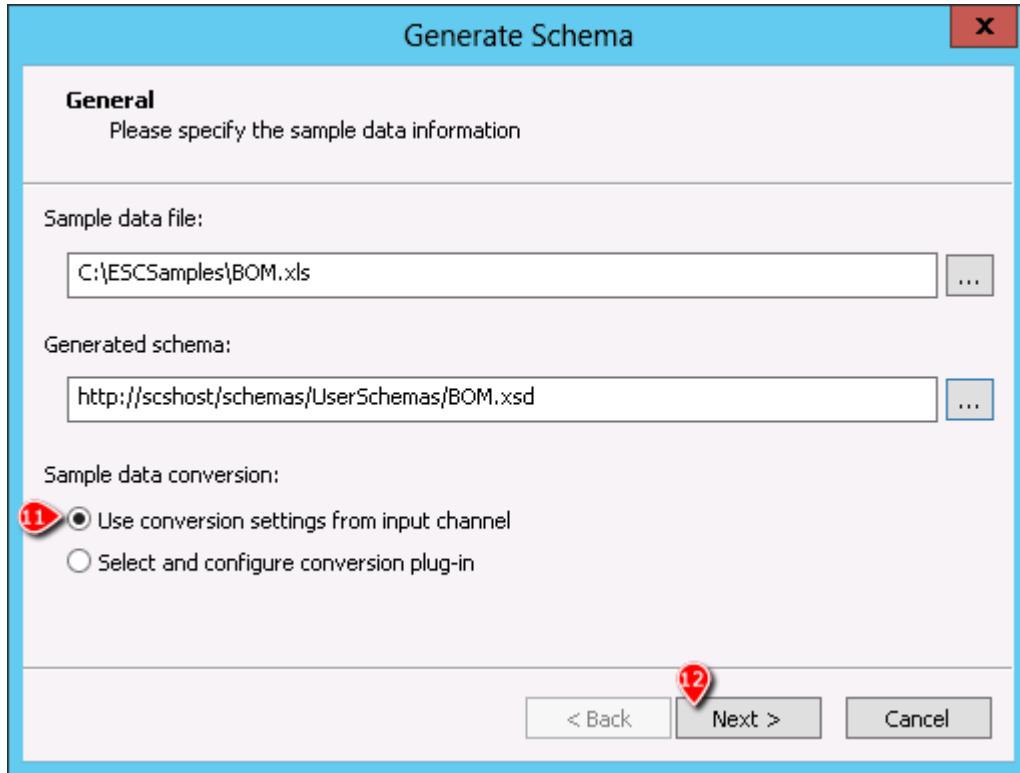
6. Click **Open**.
7. On the **Generate Schema** screen, next to the **Generated schema** field, click the ... (ellipse) button.



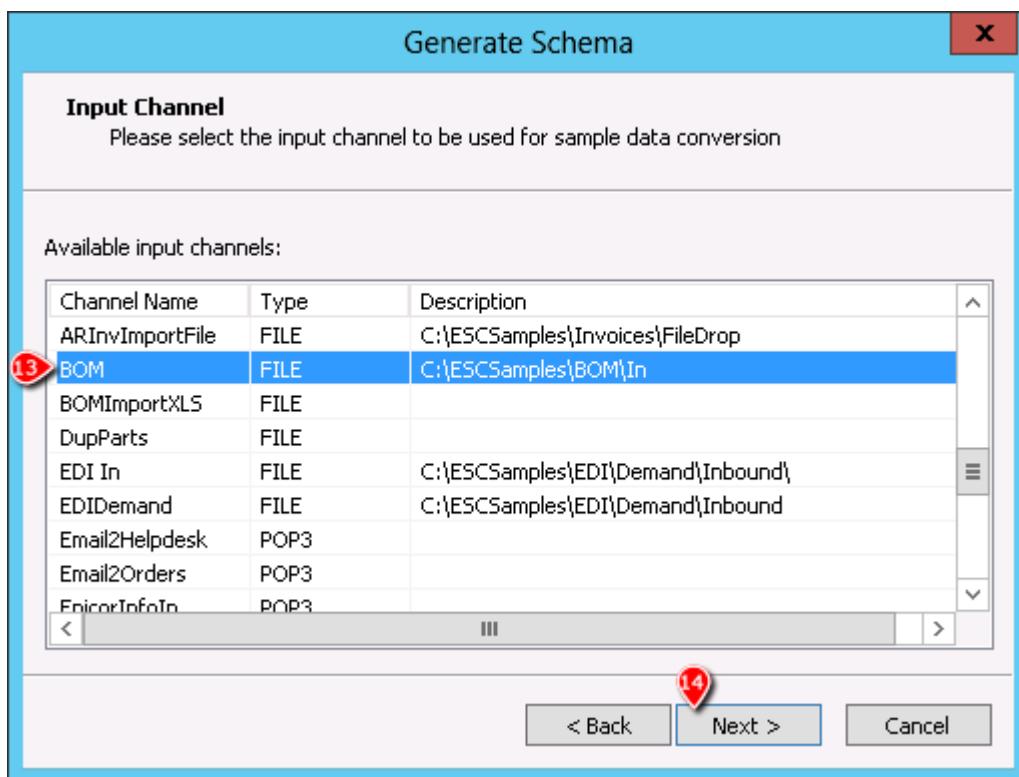
8. In the **Save Schema** dialog, navigate to the folder where to save the schema.



9. In the **File Name** field, accept the default schema file name or enter a different 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.

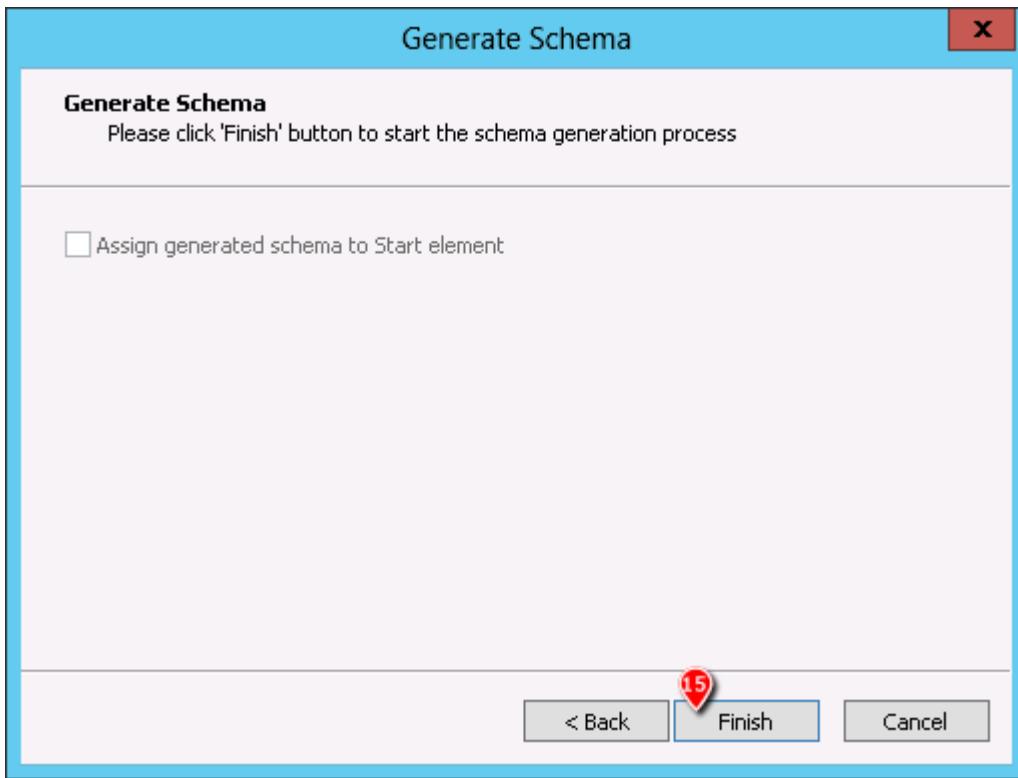


12. Click **Next**.
13. From the **Available input channels** list, select a FILE channel which uses the **excel2xml.dll** conversion. In this example, you select a channel named **BOM**.



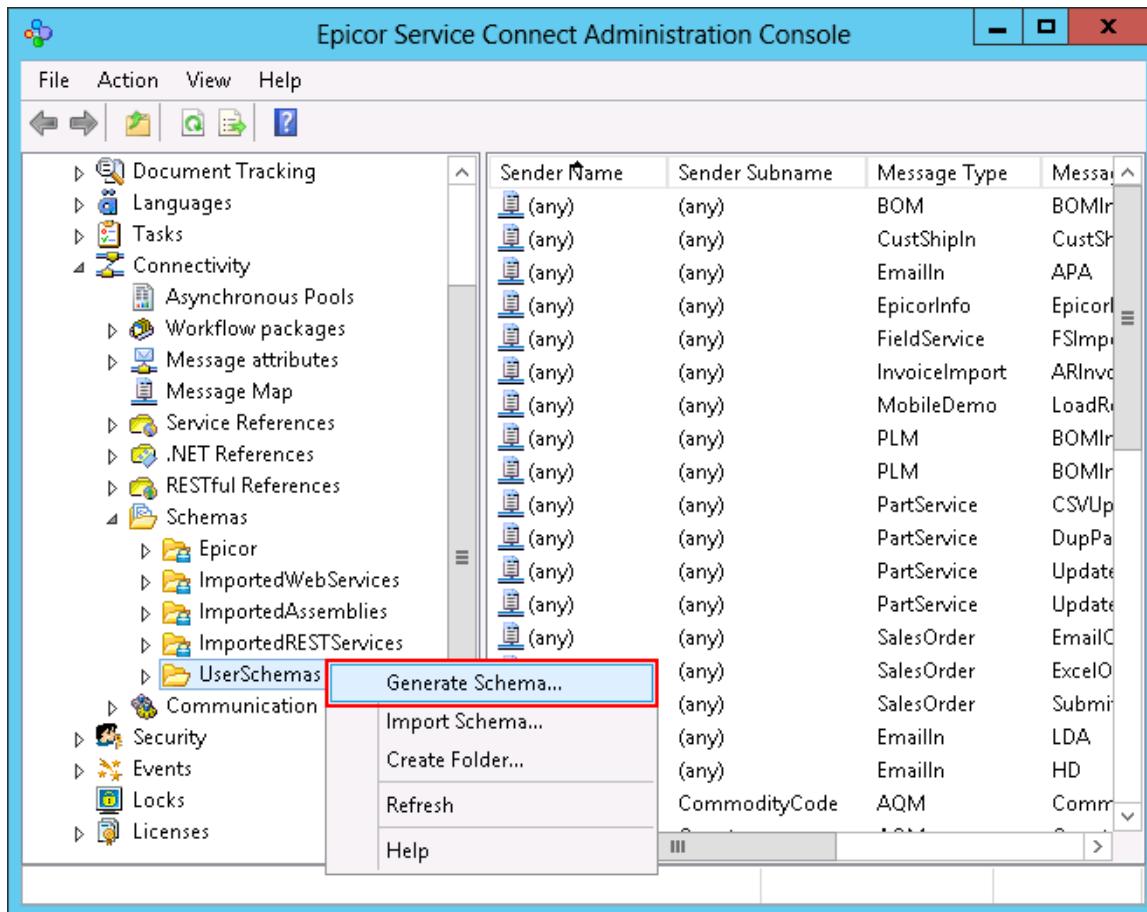
14. Click **Next**.

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 alternatively generate a schema from the **Connectivity > Schemas > User Schemas** node of the Service Connect Administration Console as well.



Epicor ERP Tracing

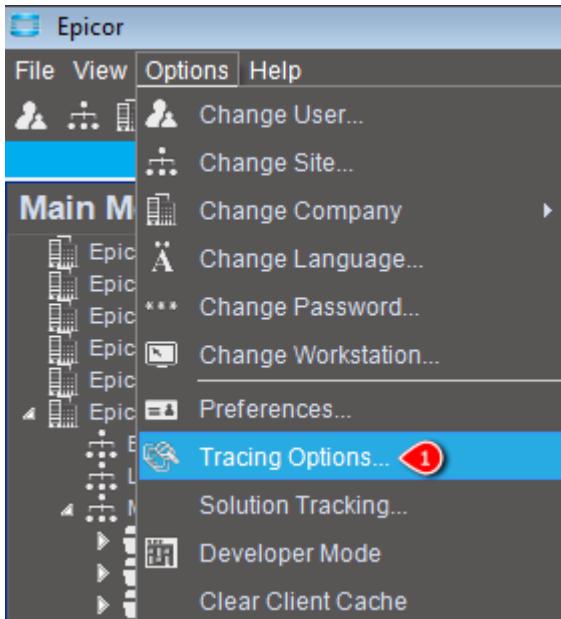
Use the Tracing Options Form window to set up a trace log that captures all public business object calls, which includes calls the client initiates as well as calls from one object to another object. When you activate this log, any public business calls sent to the server are recorded in the log.

Start a Trace Log

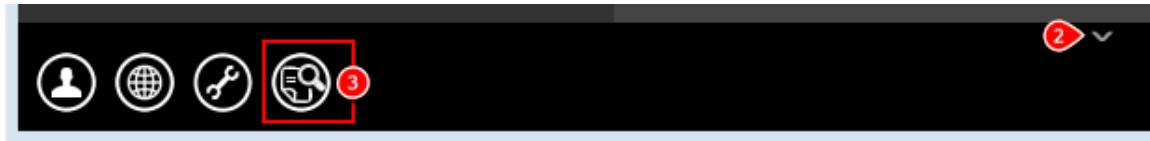
You use the tracing log to track Epicor ERP business object calls. You can then use this information to see how successful operations are performed within the application, so you can map the business logic as needed to a Service Connect workflow.

To start a trace log:

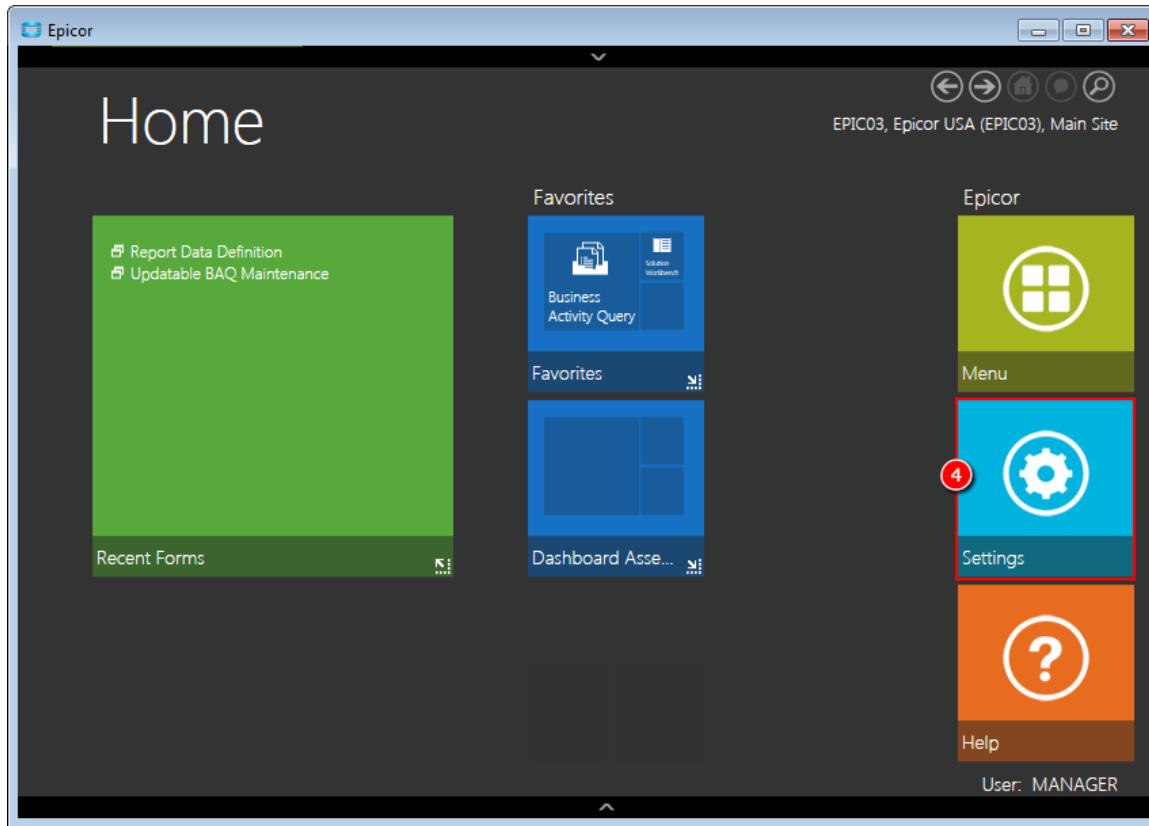
- When you run the application using the **Classic** interface, you activate the trace log from the Main Menu. Click **Options > Tracing Options**.



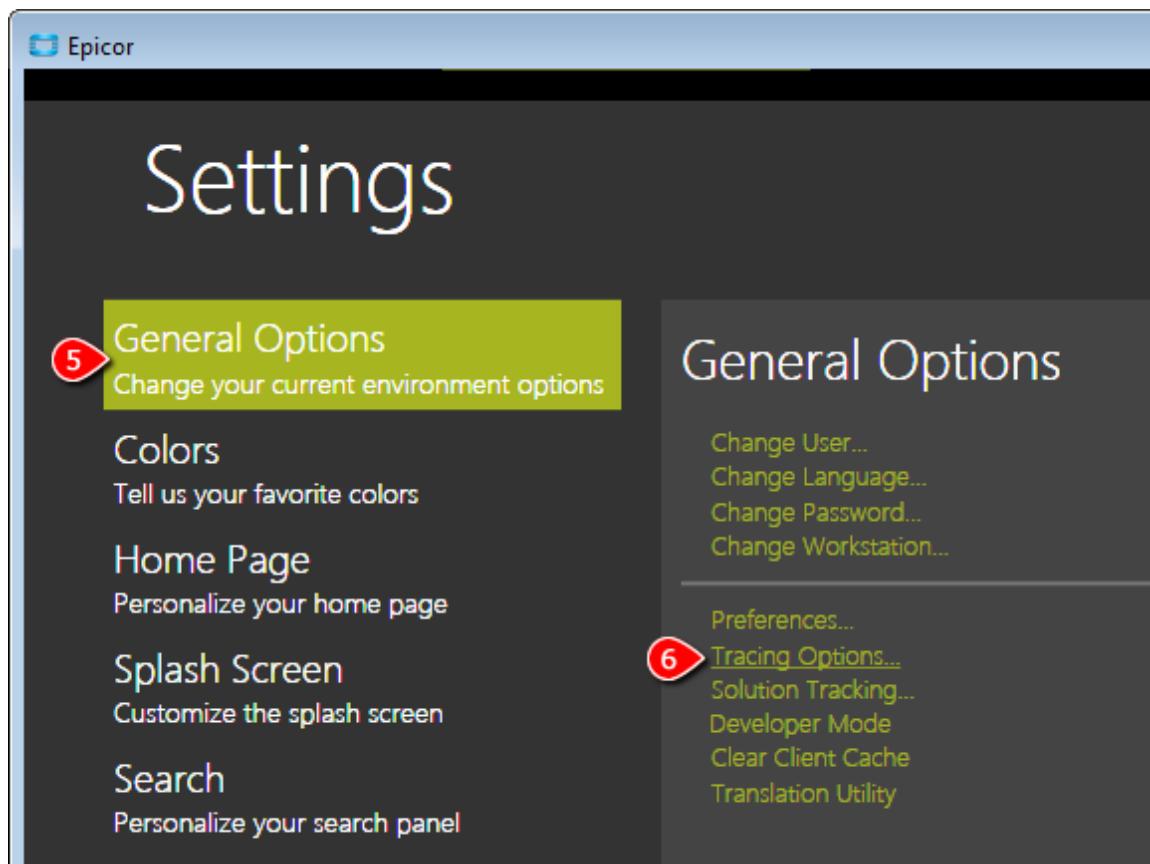
2. When you run the application using the **Modern Shell** interface, you can activate the trace log in a couple ways. Click the **Down Arrow** at the bottom of the window to display the toolbar.



3. Now click the **Tracing Options** button.
4. You can also activate the tracing log from the **Home** menu. Click the **Settings** tile.

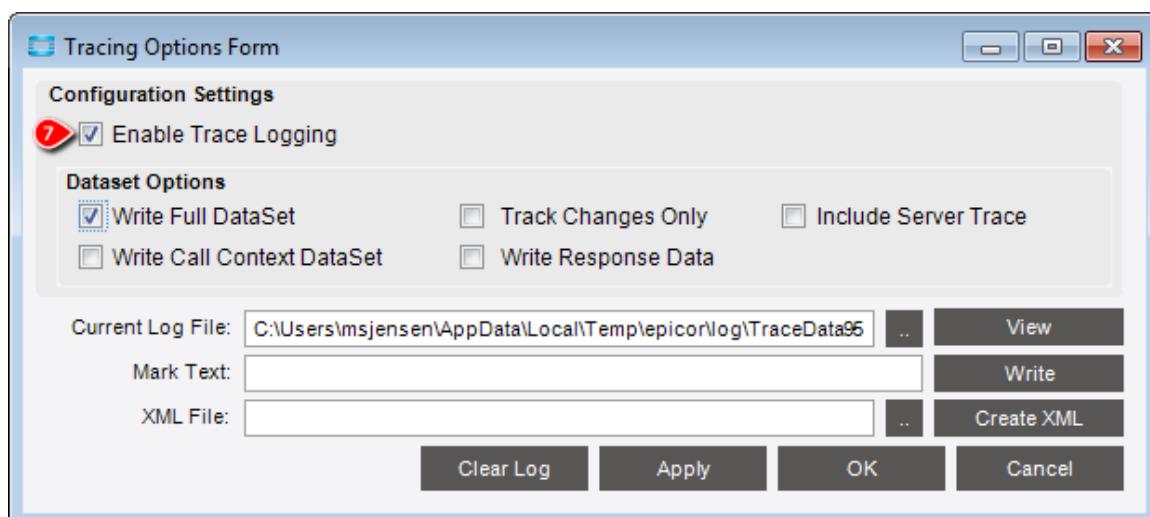


5. Click the **General Options** setting.



6. Select **Tracing Options**.

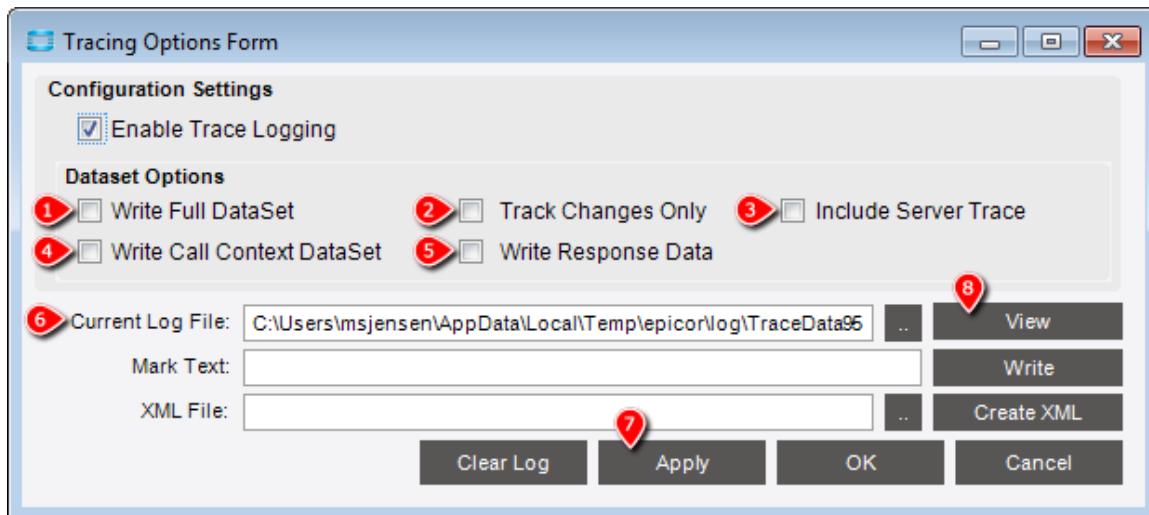
7. The **Tracing Options Form** displays. To activate the trace log, select the **Enable Trace Logging** check box.



Select Log Options

You can now define what data the trace log will capture.

1. Select the **Write Full DataSet** check box if you want to record all data within the tracing log. If this option is not selected, only header information is stored within the log.



2. Select the **Track Changes Only** check box if you only want changes to the dataset recorded within the tracing log. All changes to columns in the dataset are then stored within the log.
3. Activate the **Include Server Trace** check box when you want to track the client's interaction with the server. This creates a <serverTrace> node within trace packets (<tracePacket>) in the client tracing log. Use the database activity gathered in this section to review how the client installation may be affecting the performance of the server.

You can add server profiles and traces to the client log. When you select the Include Server Trace check box, the client log captures these additional options. To add these profiles and traces to the client log, update the .sysconfig file that launches the client installation. You can also customize what the tracing log tracks by creating a client configuration file that contains additional tracing options and logging levels. These custom options are used when you activate the client tracing log.

For more information, review the Performance Tuning Guide in the application help. The Custom Trace Logs section documents how you add these server profile and custom trace options.

4. Use the **Write Call Context Dataset** check box to include Business Process Management (BPM) table values on the trace log. This information provides the data context for a call each time a call is sent between the client and the server. This information is useful for developing BPM method directives, as you can intercept these calls to run additional processing to verify data and other custom functions. For more information about Business Process Management, review the Epicor ICE Tools User Guide.

5. Numerous method calls occur where the data is passed down, modified, not written to the database, and then returned to the client. Select the **Write Response Data** option to include these database transactions on the trace log.
6. All business object calls from this point forward are entered into the log file shown in the **Current Log File** field. If your system administrator activates the client log through **User Account Security Maintenance**, the default directory path defined on the user account displays in this field. However you can enter a different directory path in this field or click the **Browse (...)** button to find and select it. After you click **Apply** or **OK**, this custom directory path becomes the default location that stores the generated log files for this client.

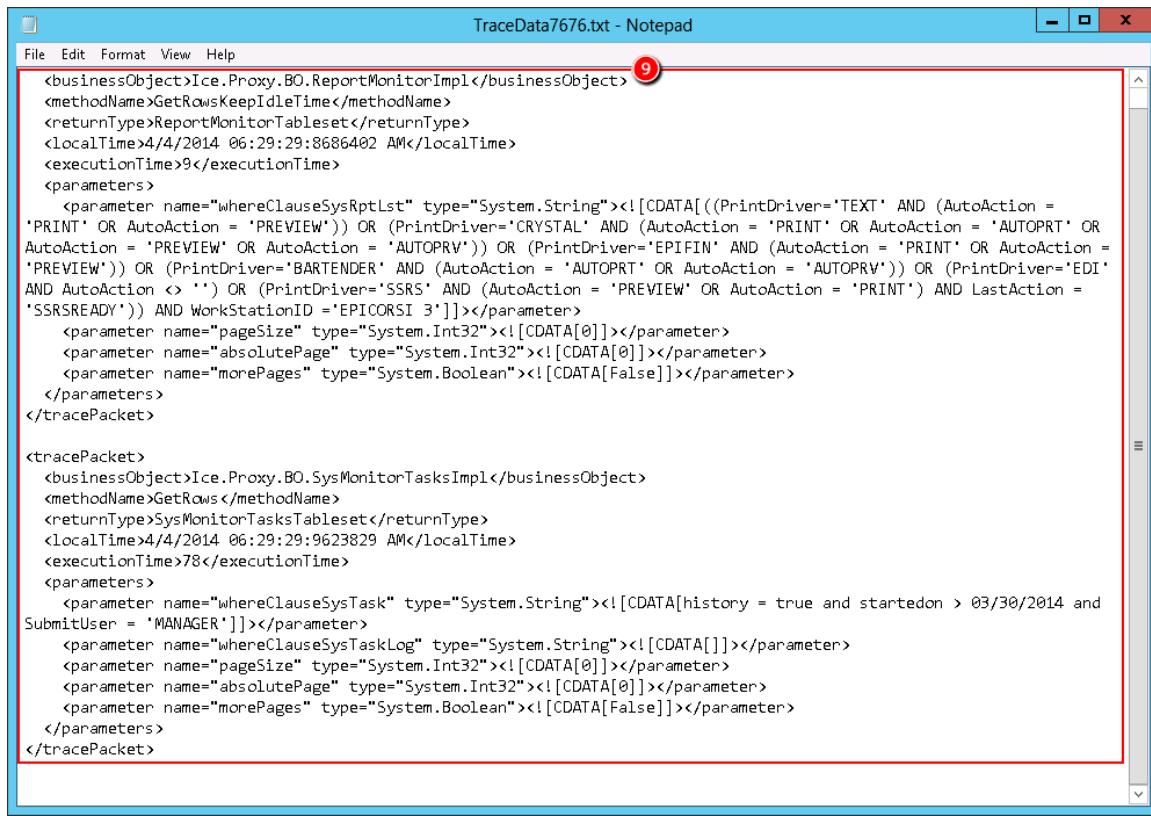
To learn how to activate the trace log through your user account, review the User Account Security Maintenance > Tracing topics in the Epicor ERP application help.

7. Click **Apply**.

The application begins a trace log.

8. Click **View** to review the log file.

9. The trace log may look similar to the one shown here.

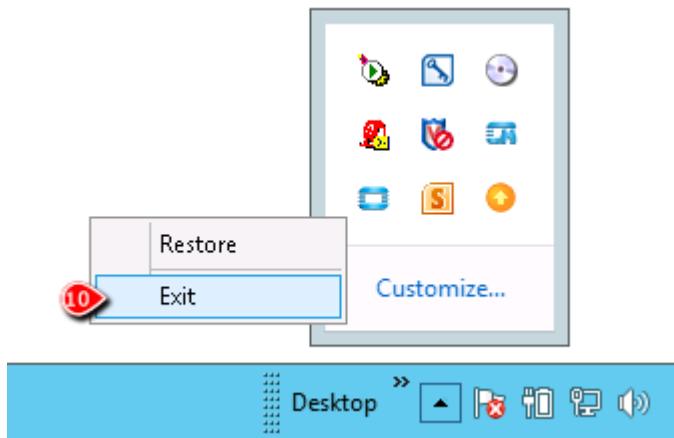


```

<businessObject>Ice.Proxy.BO.ReportMonitorImpl</businessObject>
<methodName>GetRowsKeepIdleTime</methodName>
<returnType>ReportMonitorTableset</returnType>
<localTime>4/4/2014 06:29:29:8686402 AM</localTime>
<executionTime>9</executionTime>
<parameters>
  <parameter name="WhereClauseSysRptLst" type="System.String"><![CDATA[((PrintDriver='TEXT' AND (AutoAction = 'PRINT' OR AutoAction = 'PREVIEW')) OR (PrintDriver='CRYSTAL' AND (AutoAction = 'PRINT' OR AutoAction = 'AUTOPRT' OR AutoAction = 'PREVIEW' OR AutoAction = 'AUTOPRV')) OR (PrintDriver='EPIFIN' AND (AutoAction = 'PRINT' OR AutoAction = 'PREVIEW')) OR (PrintDriver='BARTENDER' AND (AutoAction = 'AUTOPRT' OR AutoAction = 'AUTOPRV')) OR (PrintDriver='EDI' AND AutoAction <> '') OR (PrintDriver='SSRS' AND (AutoAction = 'PREVIEW' OR AutoAction = 'PRINT') AND LastAction = 'SSRSREADY')) AND WorkStationID ='EPICOR\$3'']]></parameter>
  <parameter name="pageSize" type="System.Int32"><![CDATA[0]]></parameter>
  <parameter name="absolutePage" type="System.Int32"><![CDATA[0]]></parameter>
  <parameter name="morePages" type="System.Boolean"><![CDATA[False]]></parameter>
</parameters>
</tracePacket>
<businessObject>Ice.Proxy.BO.SysMonitorTasksImpl</businessObject>
<methodName>GetRows</methodName>
<returnType>SysMonitorTasksTableset</returnType>
<localTime>4/4/2014 06:29:29:9623829 AM</localTime>
<executionTime>78</executionTime>
<parameters>
  <parameter name="WhereClauseSysTask" type="System.String"><![CDATA[history = true and startedon > 03/30/2014 and SubmitUser = 'MANAGER']]></parameter>
  <parameter name="WhereClauseSysTaskLog" type="System.String"><![CDATA[]]]></parameter>
  <parameter name="pageSize" type="System.Int32"><![CDATA[0]]></parameter>
  <parameter name="absolutePage" type="System.Int32"><![CDATA[0]]></parameter>
  <parameter name="morePages" type="System.Boolean"><![CDATA[False]]></parameter>
</parameters>
</tracePacket>

```

10. 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**.

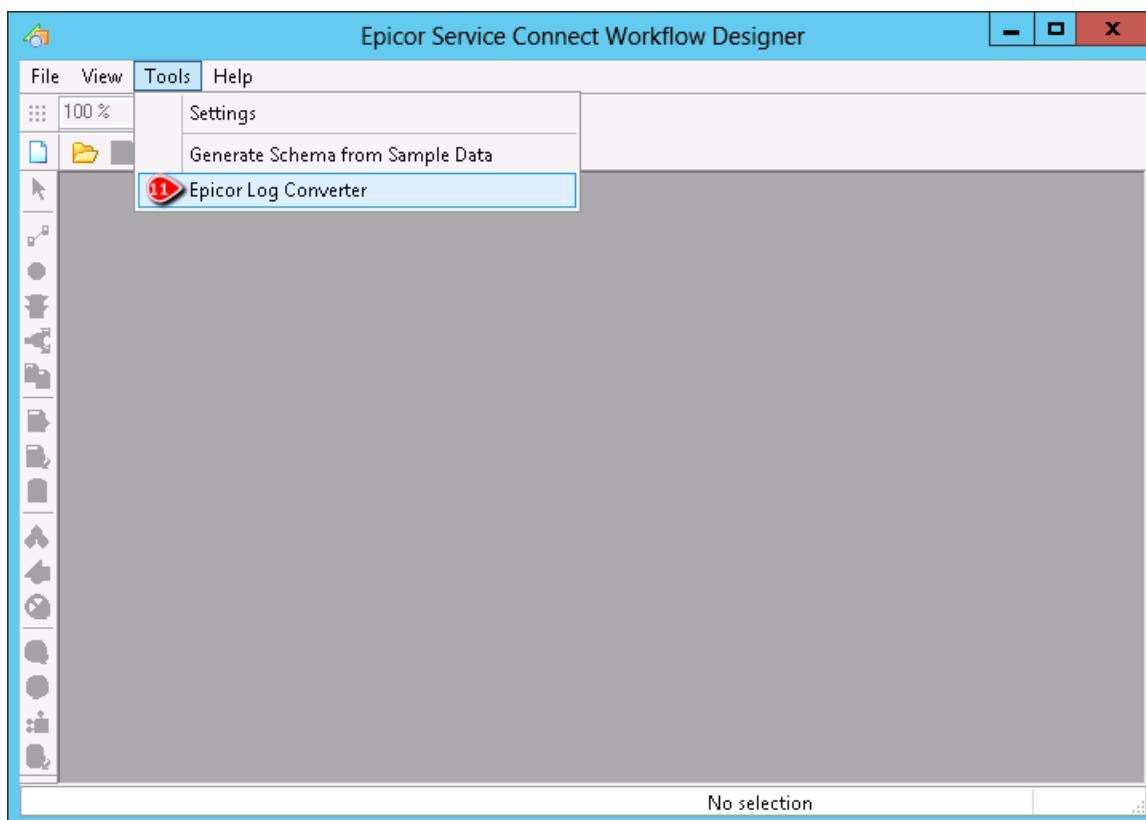


You can restart the System Monitor from the following menu path:

Menu Path: System Setup > System Maintenance > System Monitor

This
program
is not
available
in
Epicor
Web
Access.

11. 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.

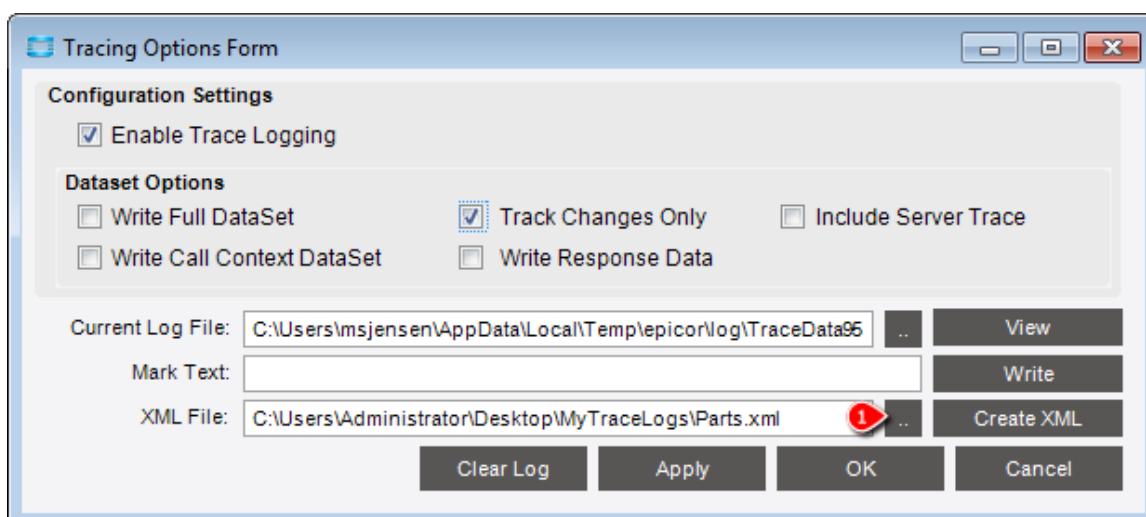


Create an XML Trace Report

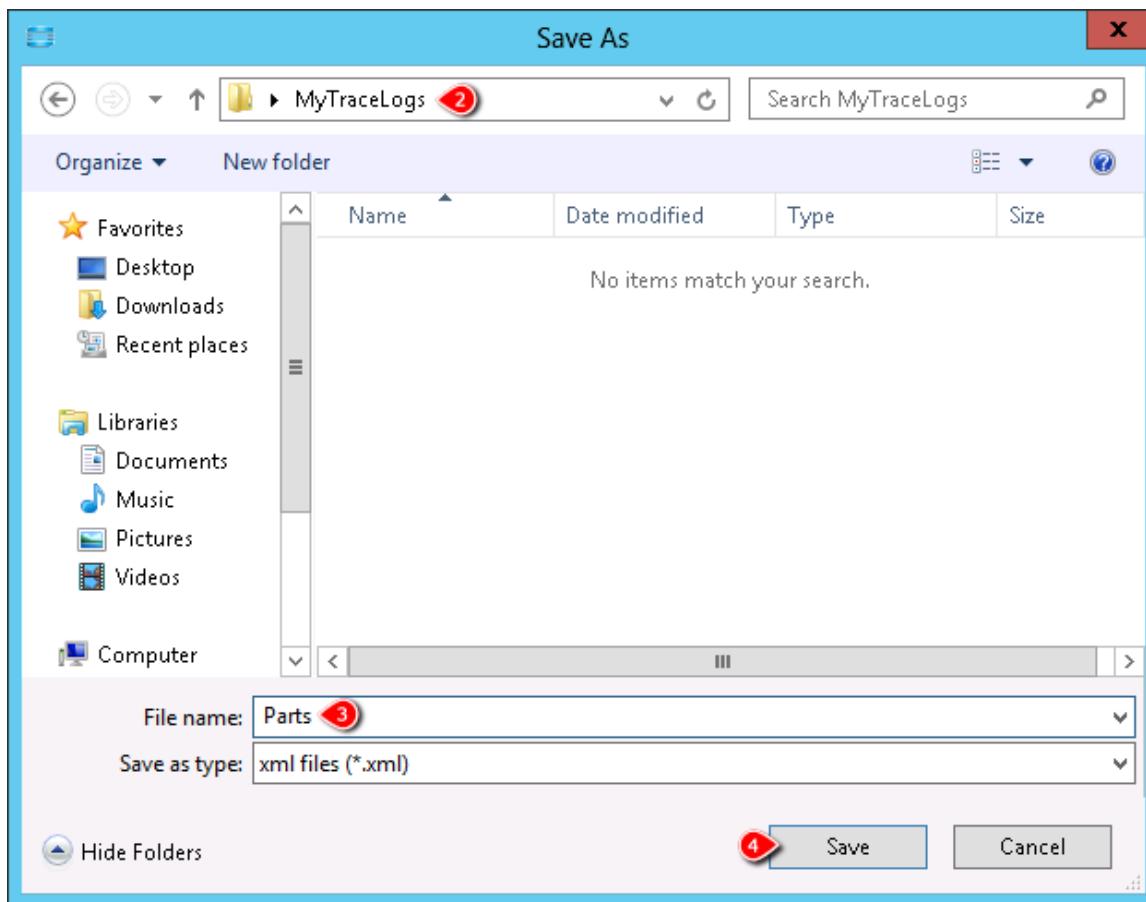
You can save the trace file to an XML file. The system generates an XSL stylesheet so the XML file is formatted in a report layout.

To create an XML trace report:

1. In the **Tracing Options Form**, click the **Browse (...) button**.



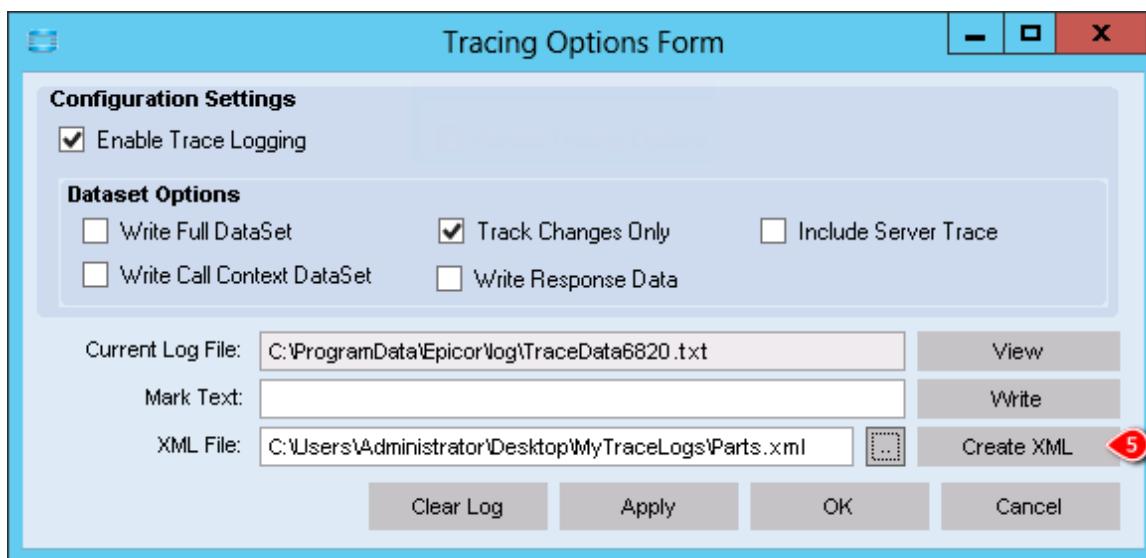
2. Browse to a location on your computer where you want to create the file.



3. Enter the file name.

4. Click **Save**.

5. 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.

The screenshot shows a Windows Internet Explorer window titled 'C:\Users\Administrator\Desktop\MyTraceLogs\Parts.xml - Windows Internet Explorer'. The page displays an XML trace. At the top, there is a table with columns for 'Parameter Name', 'Type', 'Data Set', and 'Value'. The first row shows 'pageSize' as 'System.Int32' with value '0'. The second row shows 'absolutePage' as 'System.Int32' with value '0'. The third row shows 'morePages' as 'System.Boolean' with value 'False'. Below this, two rows are listed under task 90: 'Ice.Proxy.BO.SysMonitorTasksImpl GetRows' at timestamp 85 on 4/4/2014 12:52:5. The parameters for this row are the same as the header table. Below these, another row is listed under task 91: 'Ice.Proxy.BO.ReportMonitorImpl GetRowsKeepIdleTime' at timestamp 9 on 4/4/2014 12:53:2. The parameters for this row are also the same as the header table.

Parameter Name	Type	Data Set	Value
pageSize	System.Int32		0
absolutePage	System.Int32		0
morePages	System.Boolean		False

Parameter Name	Type	Data Set	Value
whereClauseSysTask	System.String		history = true and startedon
whereClauseSysTaskLog	System.String		
pageSize	System.Int32		0
absolutePage	System.Int32		0
morePages	System.Boolean		False

Parameter Name	Type	Data Set	Value
whereClauseSysRptLst	System.String		((PrintDriver='TEXT' AND (AutoAction = 'PRINT' OR AutoAction = 'AUTOPRT' OR AutoAction = 'PREVIEW' OR AutoAction = 'PREVIEW')) OR (PrintDriver='BARTENDER' AND AutoAction <> '') OR (PrintDriver='SSRS' AND (AutoAc WorkStationID = 'EPICORSI 4'))
pageSize	System.Int32		0

If you send the XML file to someone else, be sure to include the file named TracePacket_Default.xsl.

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 3.0 Tools User Guide.

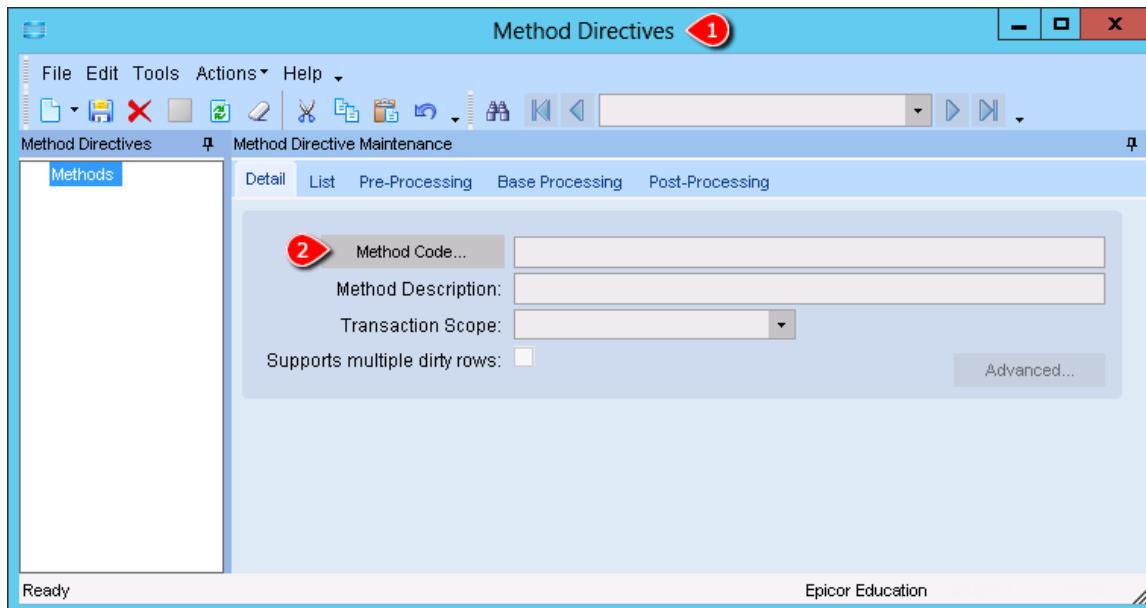
For the technical details on ESC and Epicor ERP BPM setup, refer to the Epicor Service Connect Installation and Implementation Guide > 2.9 Service Connect Interoperation with Epicor ERP section.

Locate a Business Object Method

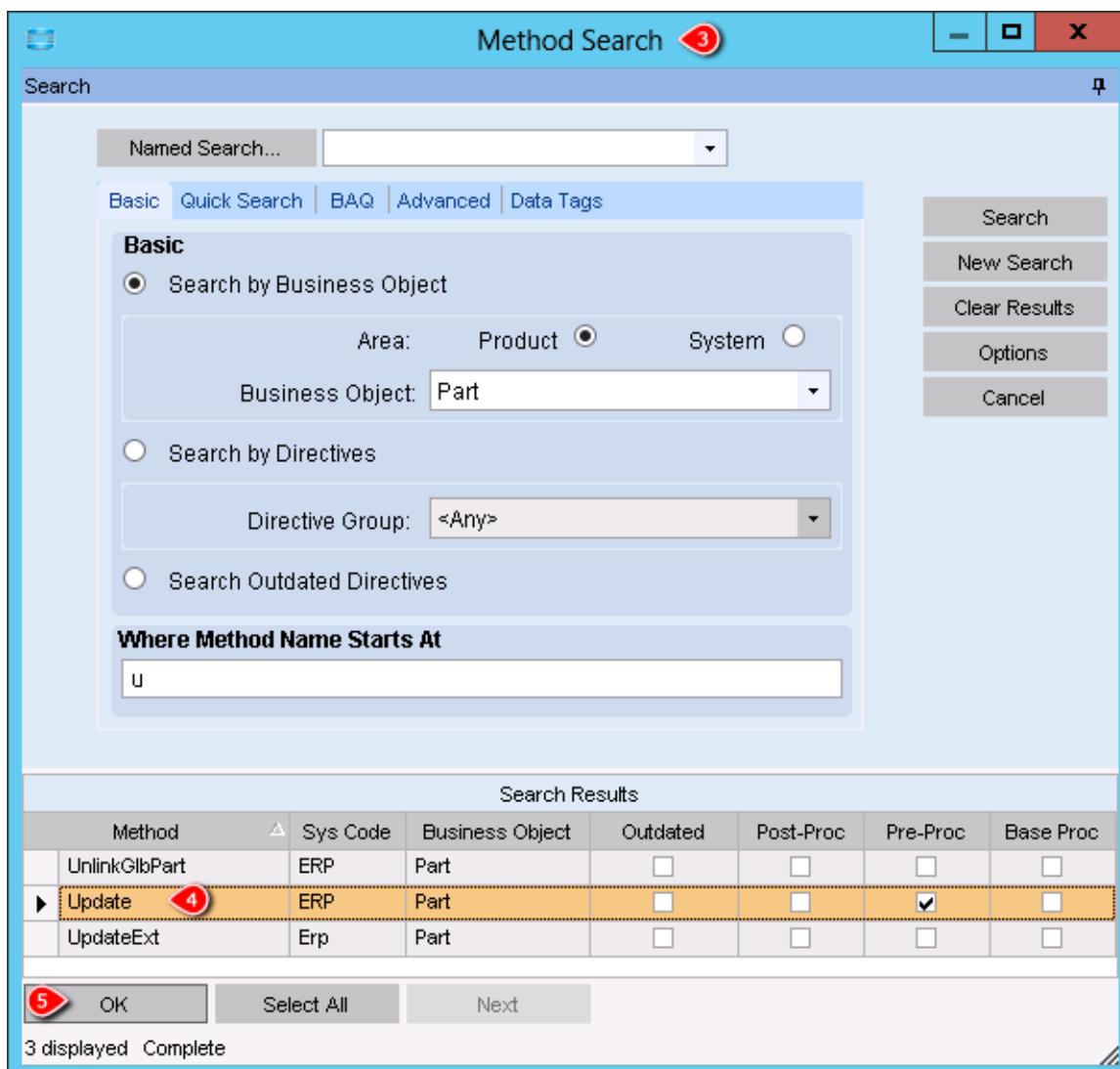
The first step to create a method directive is to identify the business object method that will trigger the BPM workflow.

To locate a business object method:

1. In Epicor ERP, open **Method Directives** using the following menu path: **System Management > Business Process Management > Method Directives Maintenance**.



2. Click **Method Code**.
3. Use the **Method Search** to locate a business object method.



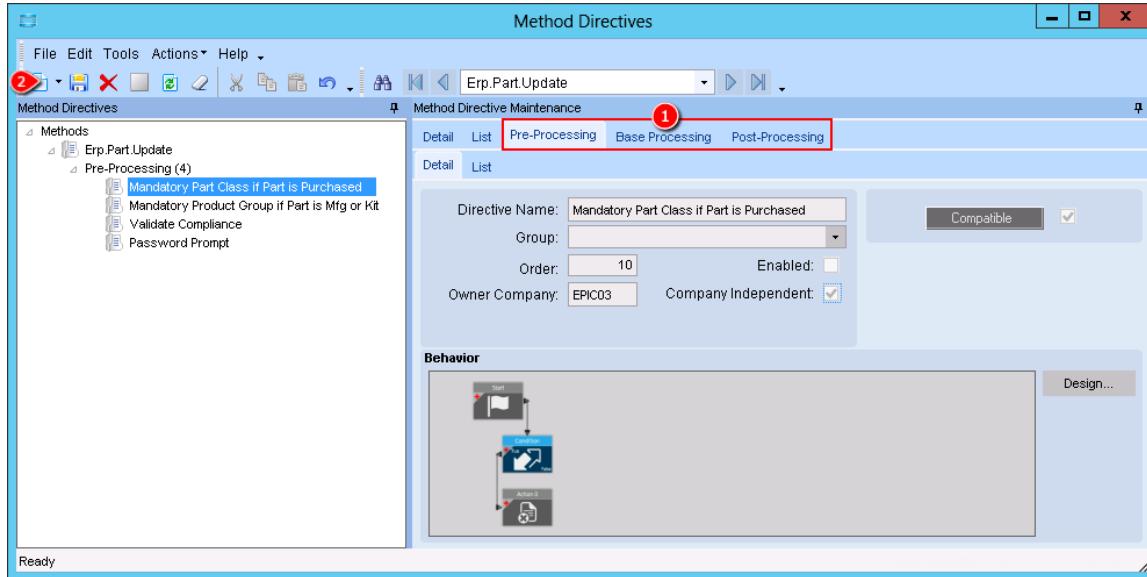
4. Select one or more methods.
5. Click **OK**.

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.

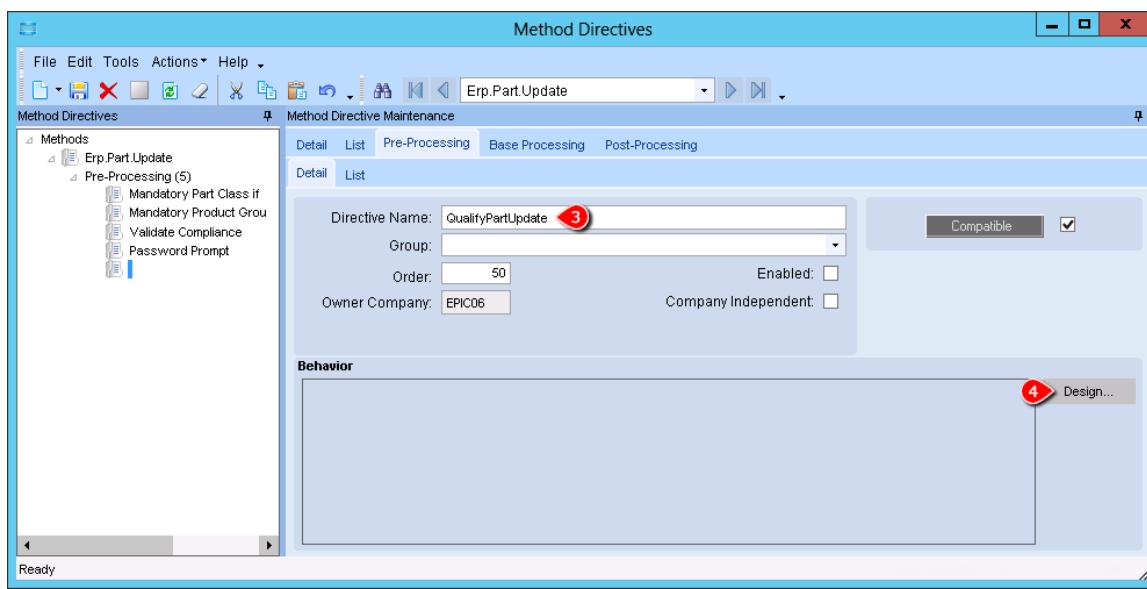
To create an action that calls a Service Connect workflow:

- 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.



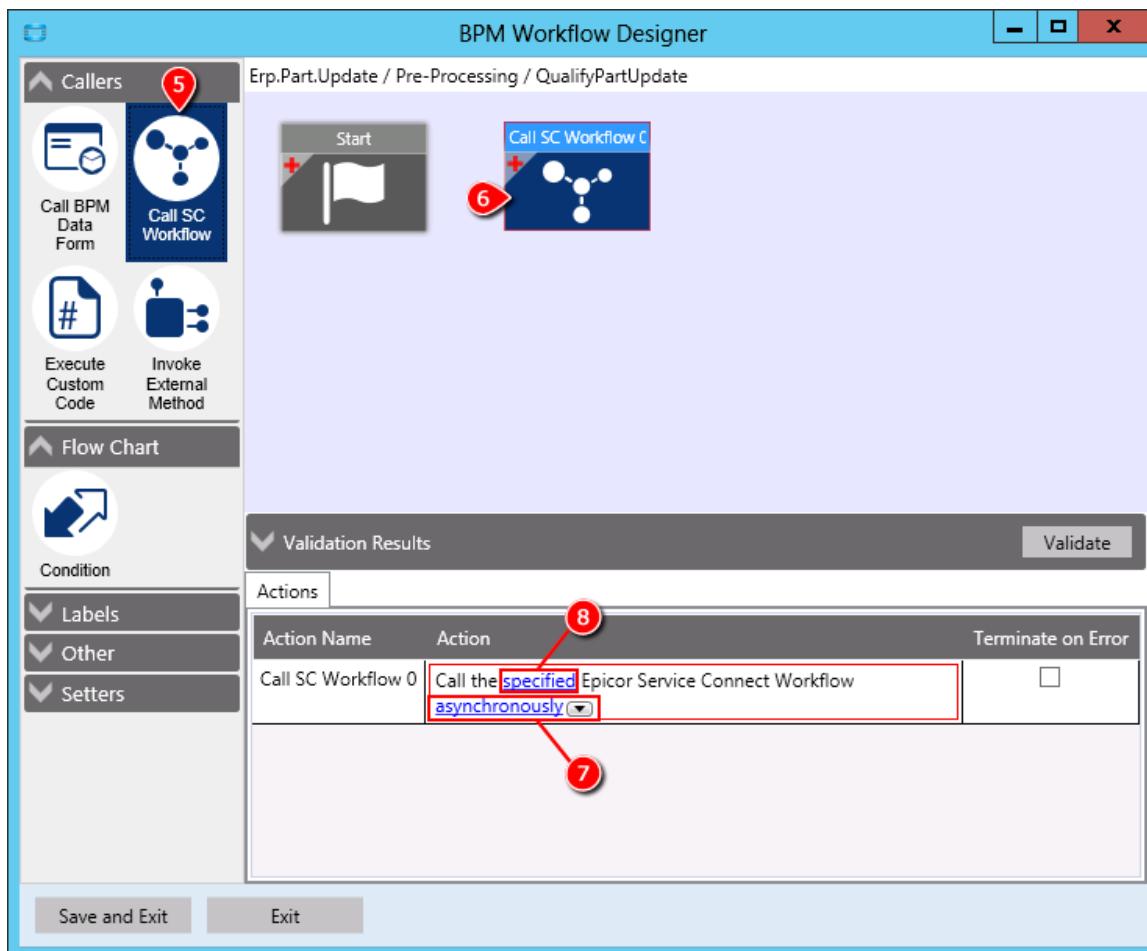
2. Click **New**.

3. Enter a **Directive Name**.

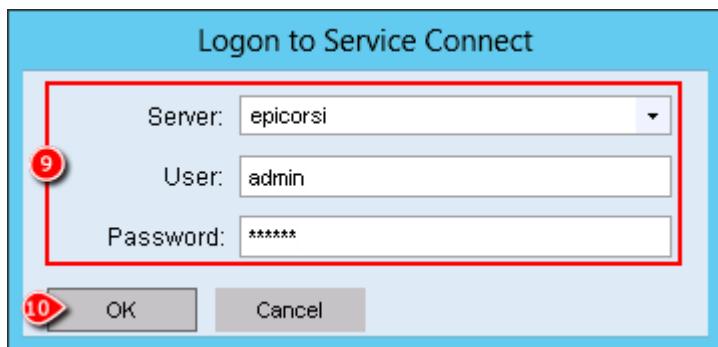


4. Click **Design**.

5. In the BPM Workflow Designer left pane, from the **Callers** group, click **Call SC Workflow**.



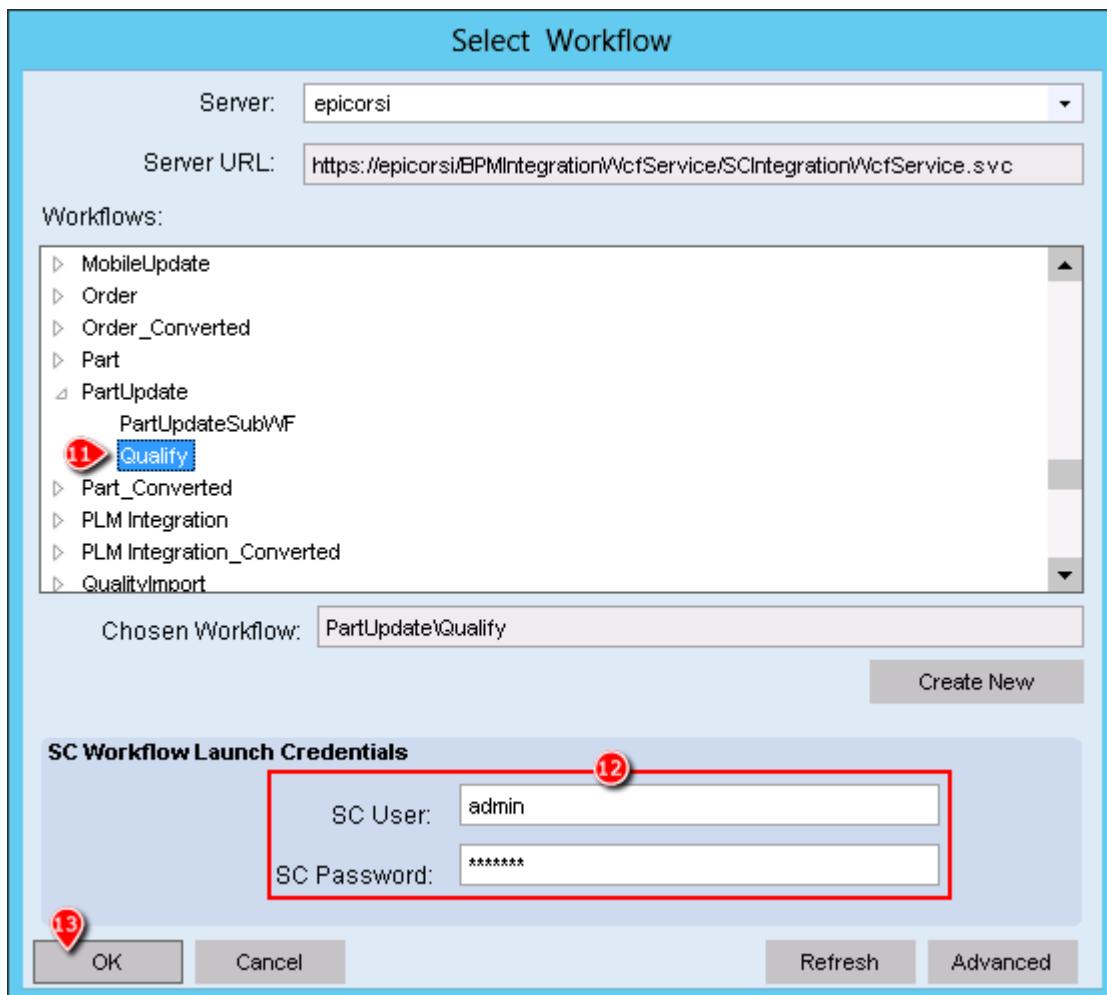
6. Drag and drop the item to the BPM Workflow Designer right pane.
7. In the **Actions** pane, in the **call the specified Epicor Service Connect Workflow asynchronously** phrase, click the word **asynchronously** to toggle between synchronous and asynchronous execution.
Asynchronous workflow calls are executed according to the schedule defined in the BPM Action Process program. Synchronous workflow calls are made when the action executes.
8. Click the word **specified**.
9. 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.



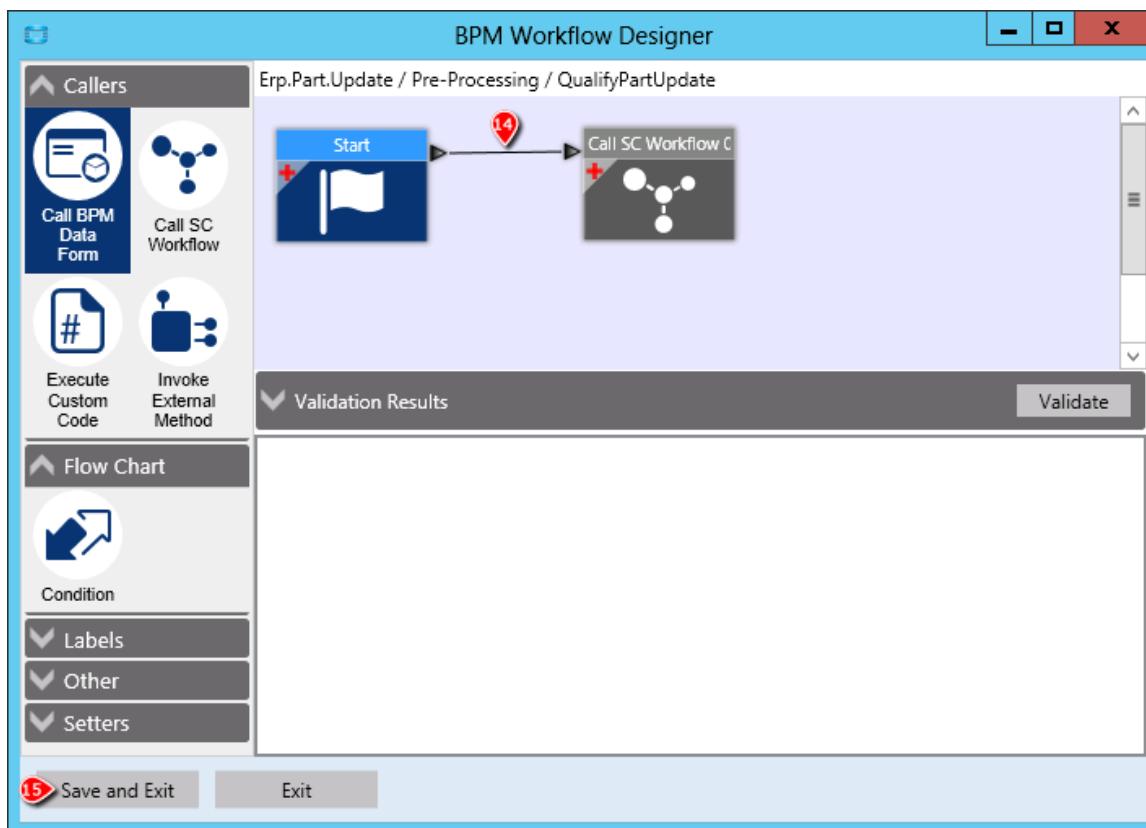
10. Click **OK**.

The system loads all the Service Connect workflow packages available on the server.

11. In the **Select Workflow** window, browse to the workflow you want to call from the method directive in the **Workflows** field.



12. Enter the **SC User** and **SC Password**.
13. Click **OK**.
14. Connect the **Start** element to the **Call SC Workflow** element.



15. Click **Save and Exit**.

BPM Integration Completion

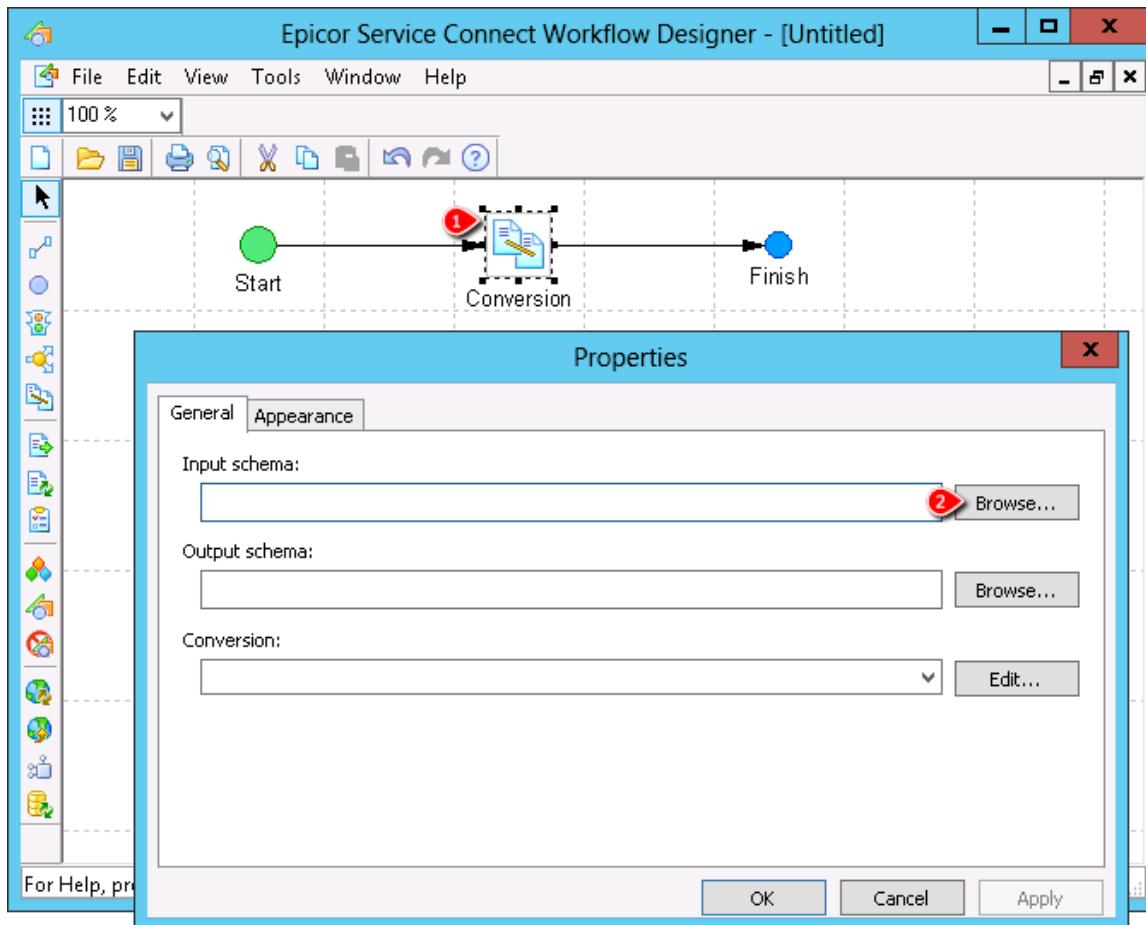
When the BPM directive executes, Epicor ERP sends the pre-update dataset for the business object method to Service Connect. Since Epicor ERP calls the Service Connect workflow directly, it is not necessary to set up message attributes, a message map, or an input channel to route the information.

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 workflow element, or if the first workflow element in the workflow is not a Conversion, you can assign the schema to the Start workflow element.

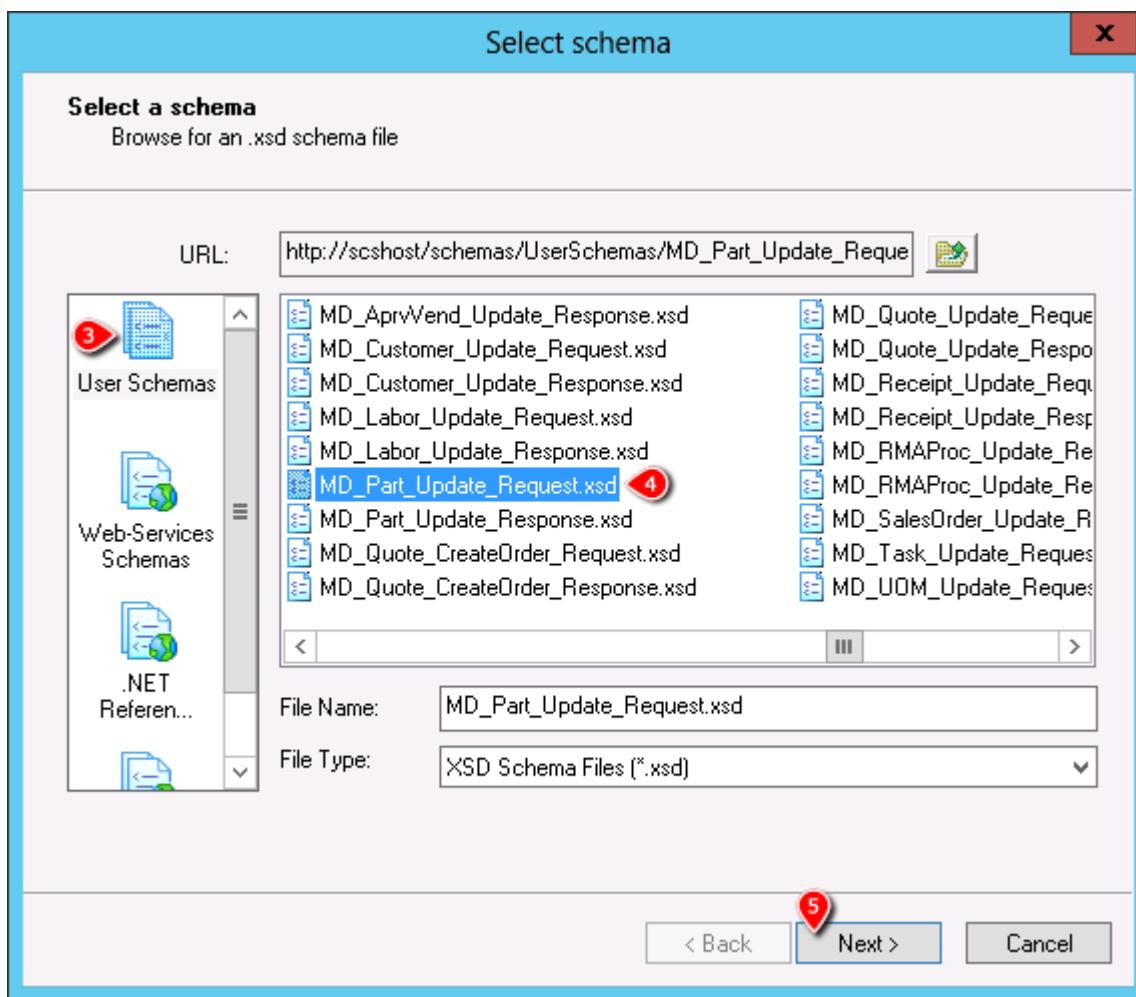
Example

Using the Part.Update method, complete the integration:

1. In the Epicor Service Connect Workflow Designer, right-click the **Conversion** workflow element that immediately follows Start and select **Properties**.



2. In the **Properties** window, click **Browse** to select the Input schema.
3. In the **Select Schema** window, in the left pane, click **User Schemas**.

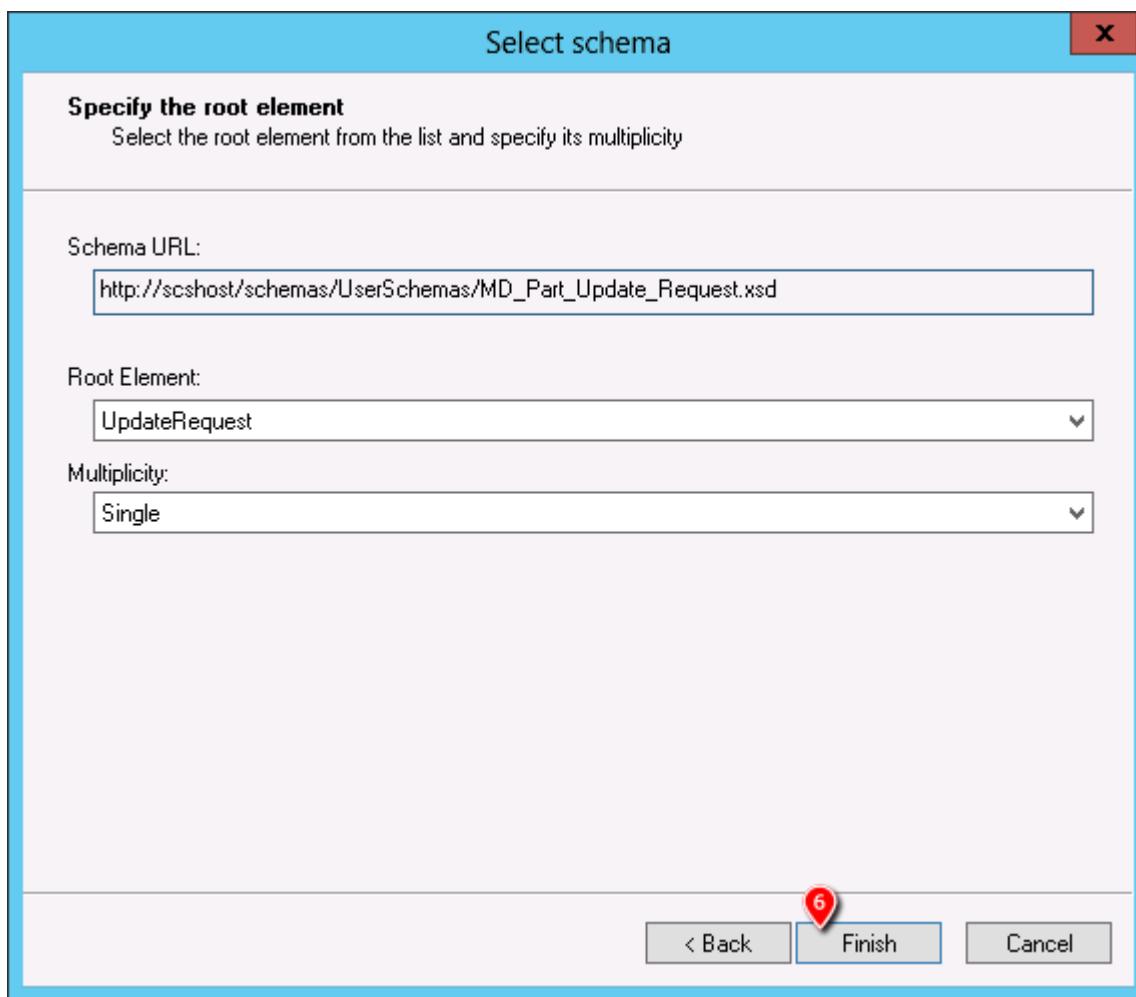


4. Browse to and select the request schema for the business object method.

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 workflow element. Refer to the Workflow Designer chapter for more information on how to define Conversions.

Exposing Info Prompt

If a BPM directive in ICE3 returns Info Prompt, it is exposed within the result of the call.

The message that lists all the field names and their values is added to Service Connect Event Log with the following Event Properties:

Property	Value
Type	Information, Warning, Error depending on the value in the Severity field of Info Message
Category	System
Description	For example: Description: Epicor BPM Info Prompt(s) received: Test
Data	Info Prompt data

To view the log, in the Administration Console, expand the **Events** node.

If Info Prompt directive has been assigned to BO method and data for that Info prompt was not provided by ESC in request in the CallContextIn node then response will contain data of BO method and new error section with the name of the Info Prompt form.

If a BO in ICE3 has a BPM assigned, and that BPM has several Info Prompt actions or several BPMs on a single method result in several Info Prompts, you can preliminarily specify the data for the upcoming Info Prompts you know about.

If the data was not passed for a specific Info Prompt, ESC logs the information you can use to specify the data.

If Info Prompt request was received after a call to the BO, then if ESC is not able to feed the data for the BPM Info Prompt using data from the Optional Context, ESC treats this as a fatal error which does not trigger BO re-calling.

ESC supports mandatory Info Prompts.

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.

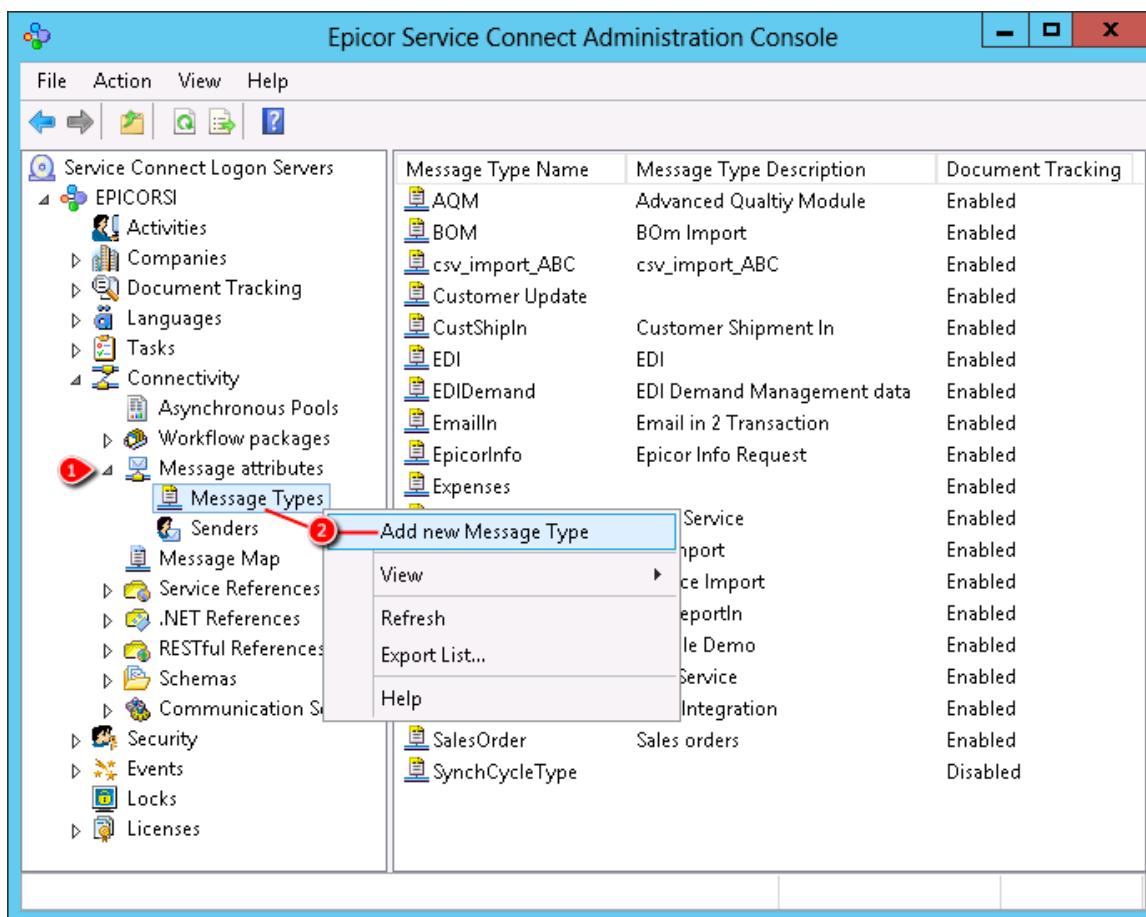
- Define a Message Type.
- Add a Sender.
- Create a Message Map.
- Add a Service Reference to the application Web Services.
- Add an Input and an Output Channel.
- Generate a schema to handle incoming data.

Add a Message Type

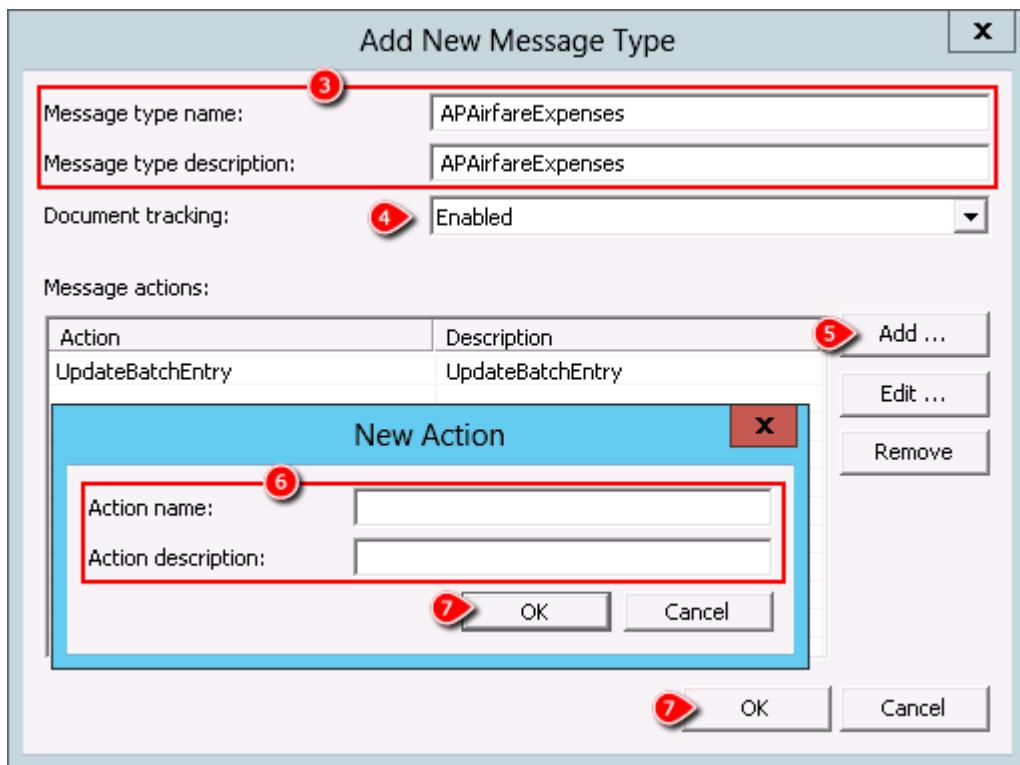
Use Message Types to classify the different documents you plan to process in Service Connect.

To add a message type:

1. In the ESC Administration Console, expand the **Connectivity > 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.



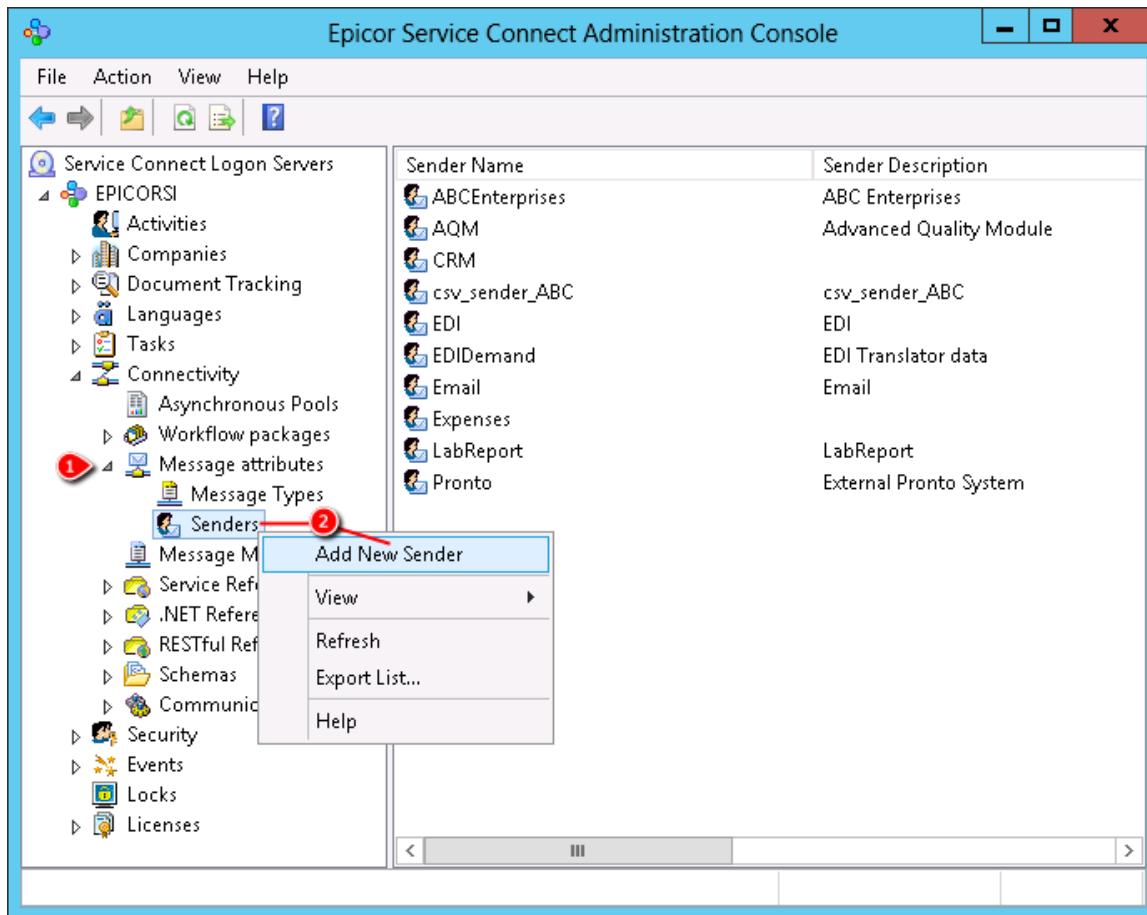
4. In the **Document tracking** field, select **Enabled** to be able to view the processing of documents associated with the message type.
5. Click **Add**.
6. Enter an **Action** and **Description** that indicate the action the workflow will take when a document assigned to this message type enters the system.
7. Click **OK** until you exit all dialog boxes.

Add a Sender

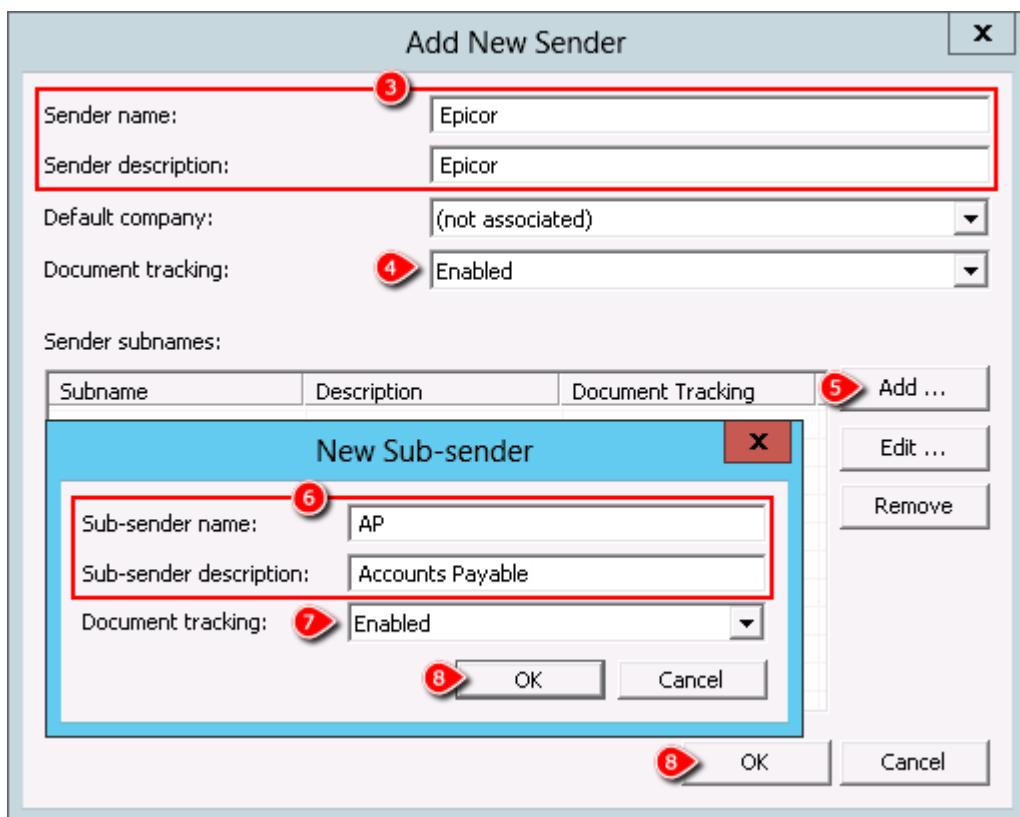
Use Senders to define the origin of the documents sent to Service Connect. This can be an application name, your company name, or a computer name. Within a Sender, you can also define sub-names to more narrowly define a document's origins.

To add a Sender:

1. In the ESC Administration Console, expand the **Connectivity > 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.



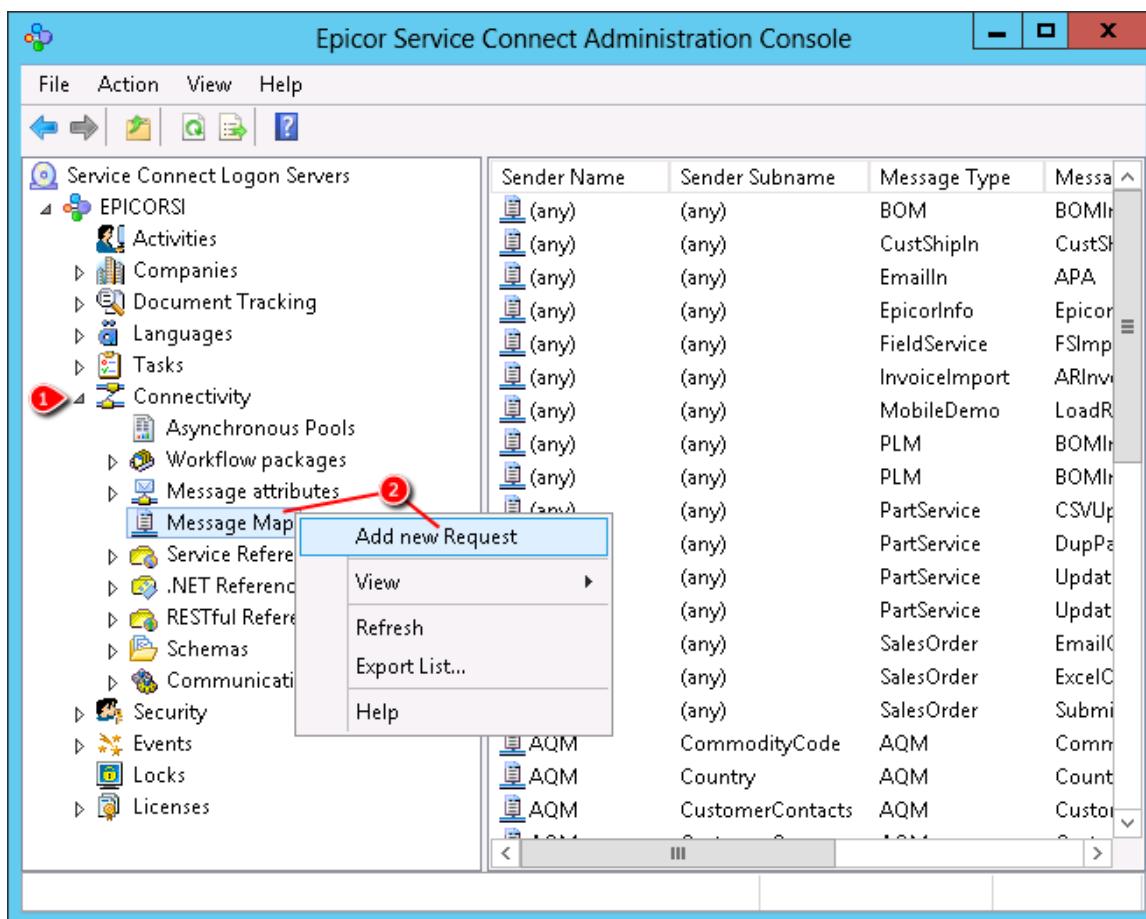
4. Select **Enabled** in the **Document tracking** field, to be able to view the processing of documents associated with this Sender.
5. Click **Add**.
6. Enter a **Sub-sender name** and **Sub-sender description**.
7. Set the **Document Tracking** field to **Enabled**.
8. Click **OK** until you exit all dialog boxes.

Create a Message Map

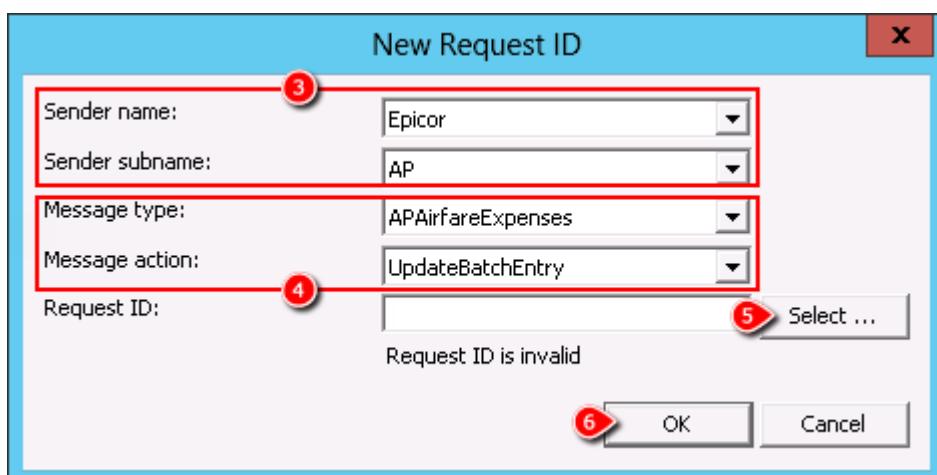
Message maps are used to route a document once the document is processed in an input channel. Service Connect uses the message map to send the document to the appropriate workflow based on the Sender and Message Type.

To create a message map:

1. 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 field blank and complete it later.
6. Click **OK**.

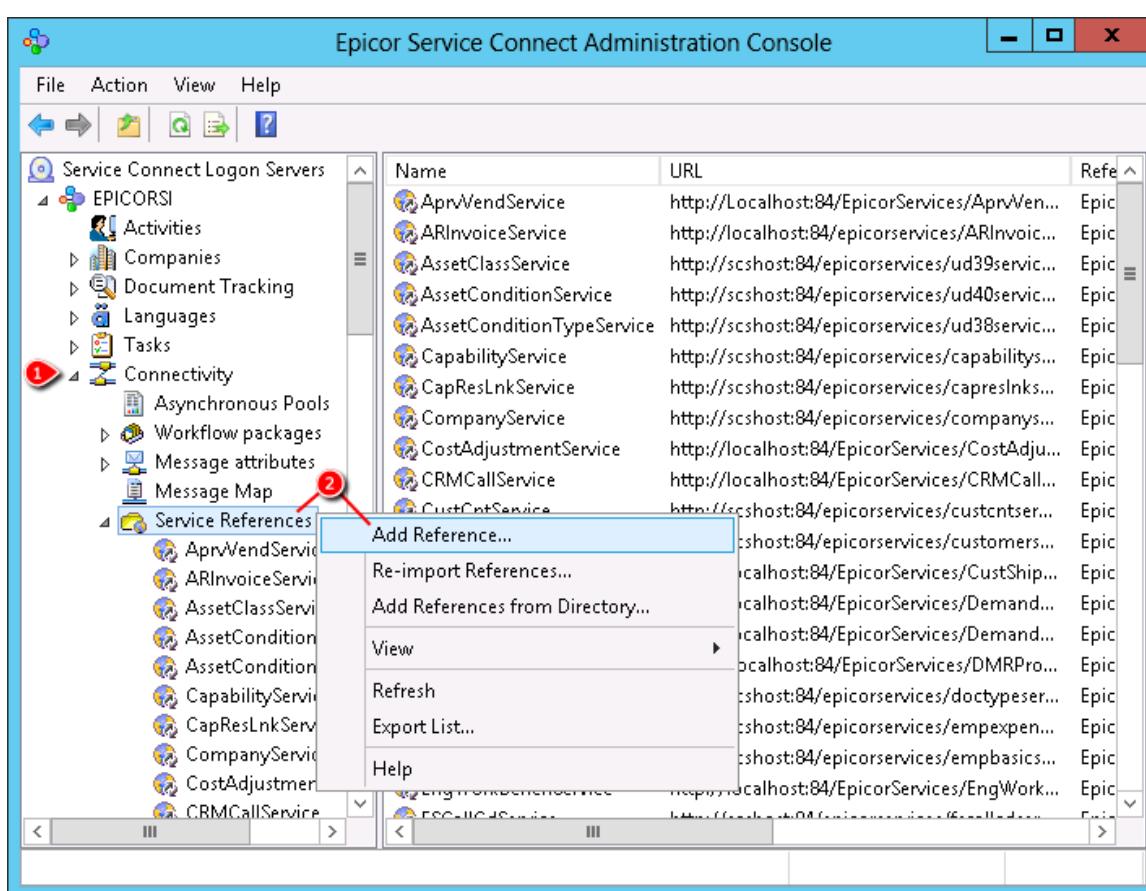
The result of your entries is that when a document is received in an input channel (yet to be defined), Service Connect will route the document to the workflow selected as the request ID if the sender name, subname, message type, and message action defined on the input channel match the entries in the message map.

Import Service References

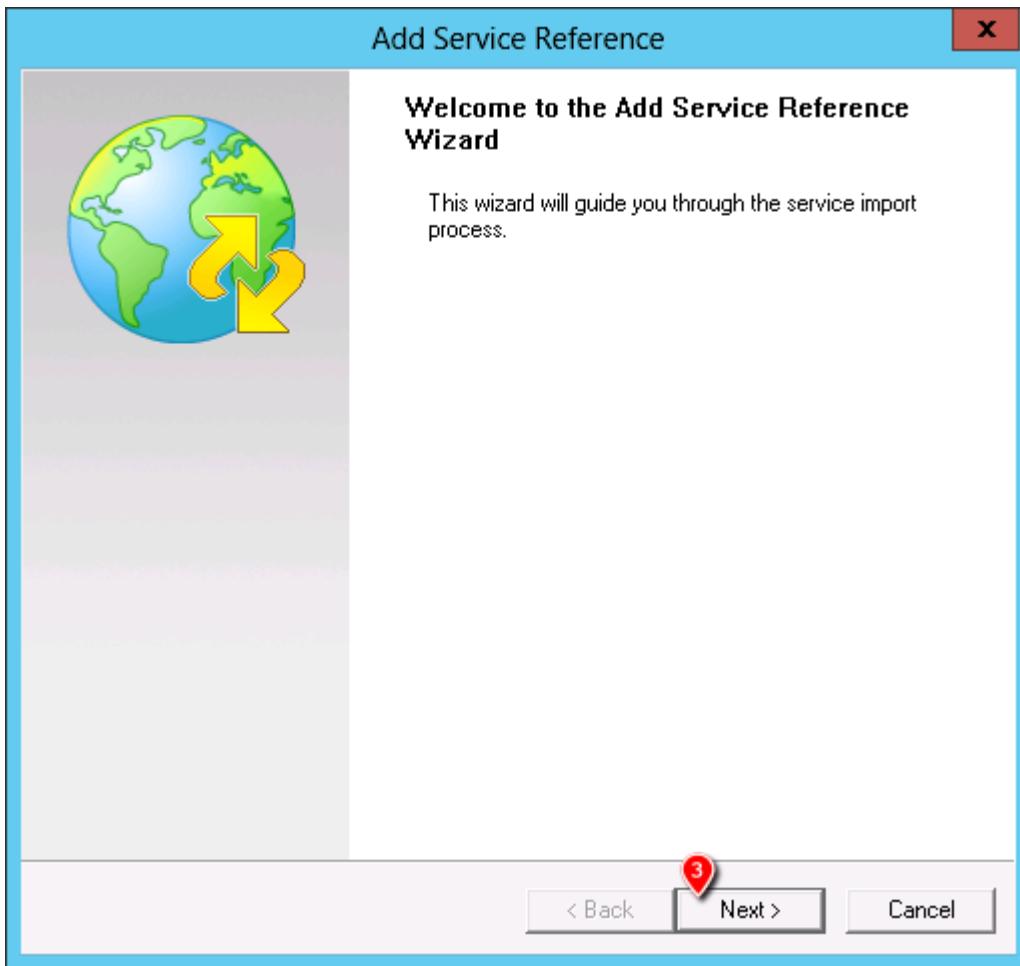
Service references are links to the web services or Windows Communication Framework® (WCF) services of applications outside Service Connect. After you import service references into the ESC Administration Console, you can call web service methods from inside a workflow.

To add a service reference:

1. In the ESC Administration Console, expand the **Connectivity** node.

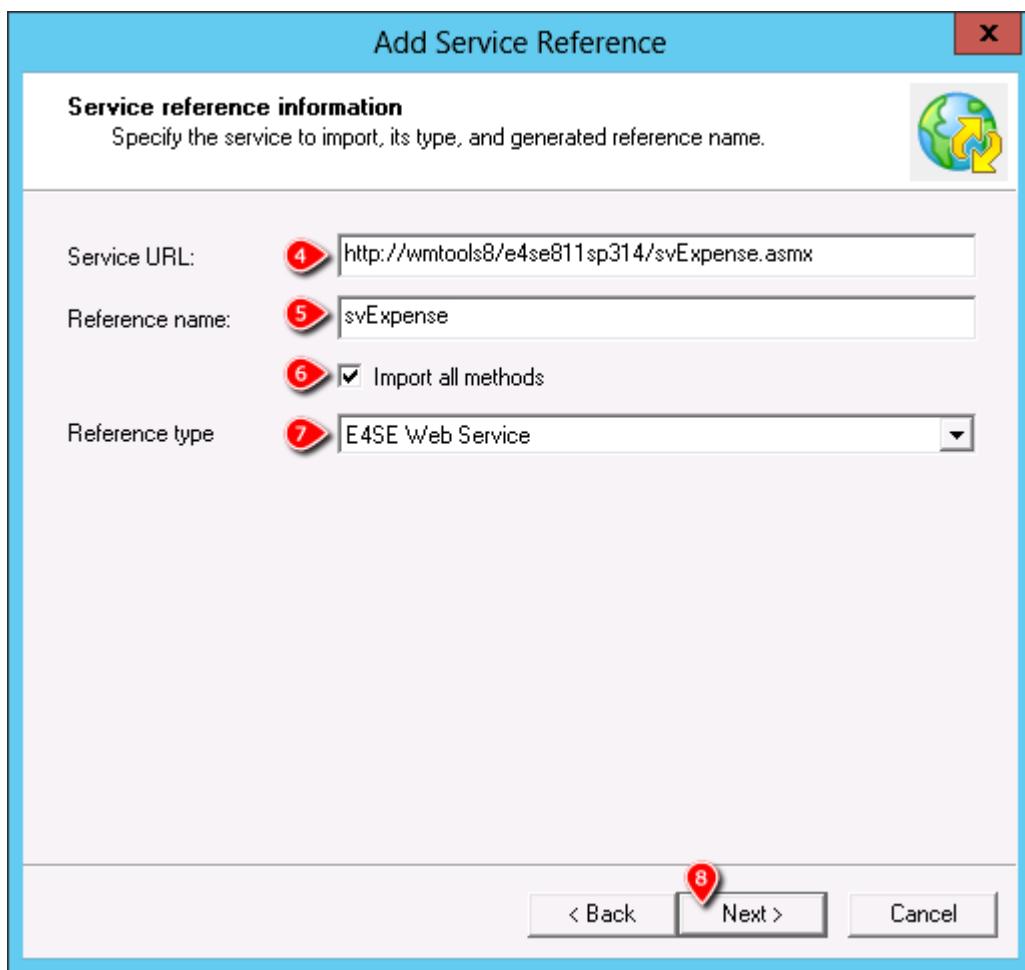


2. Right-click **Service References** and select **Add Reference...**
3. Click **Next** in the first wizard screen.



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.

4. Enter the **Service URL**.



5. View the default reference name.

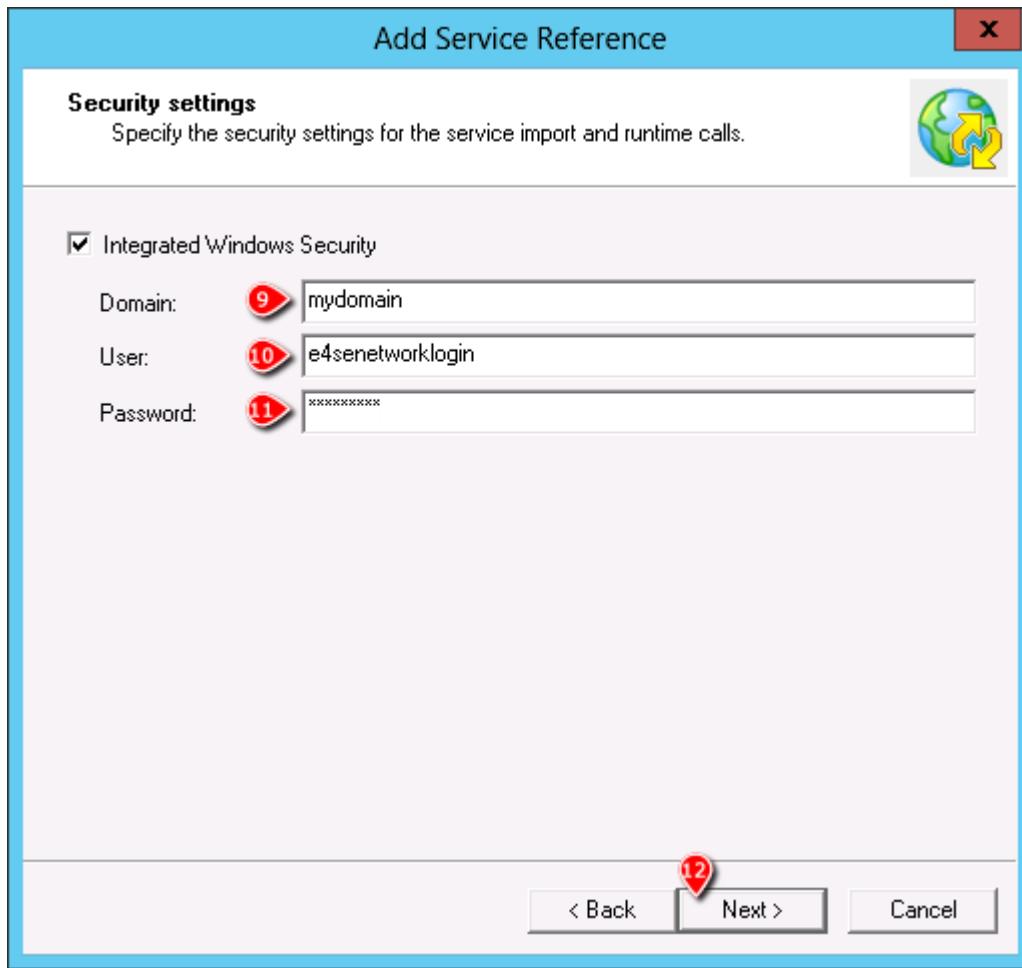
The name will be used to identify the service reference in the ESC Administration Console and will prefix the web service methods in Web Method calls.

6. Leave the **Import all methods** check box selected.

7. Select **E4SE Web Service** as the **Service Reference Type**.

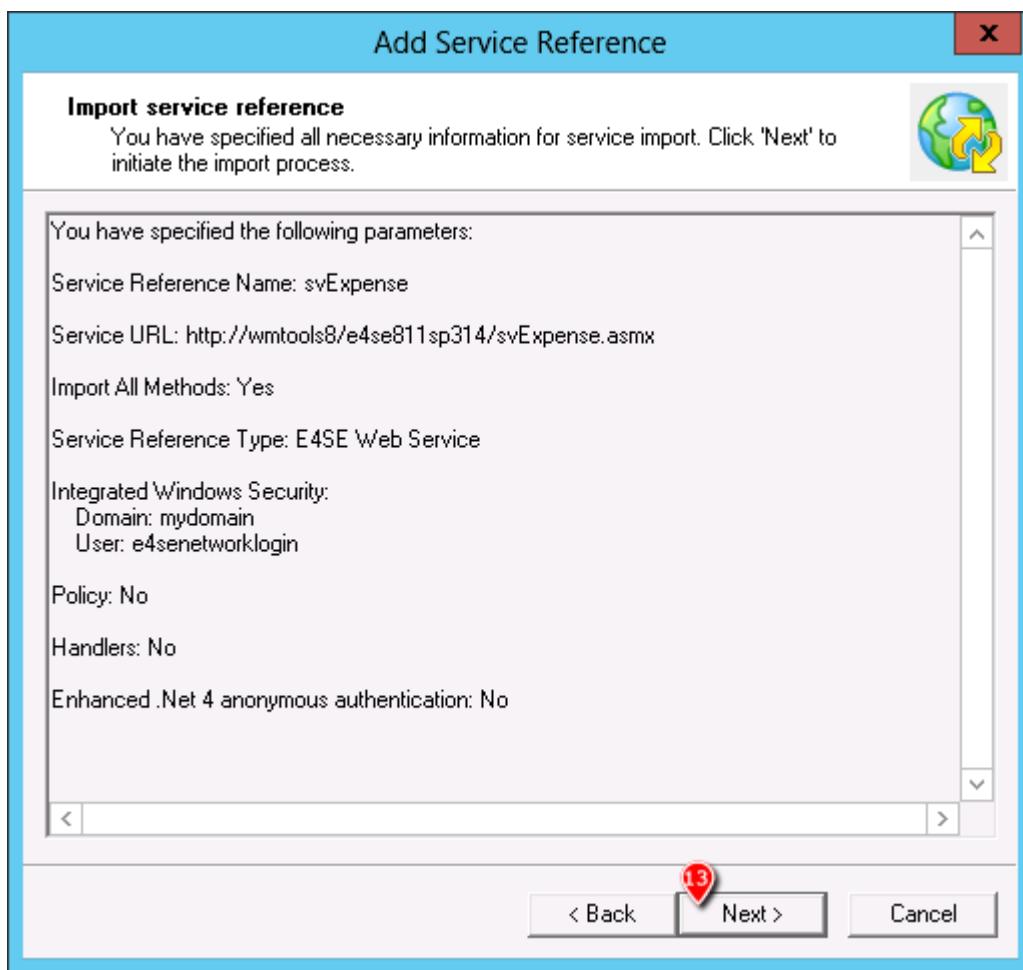
8. Click **Next**.

9. 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.

10. Enter the **User**.
11. Enter the **Password**.
12. Click **Next**.
13. Review the web service import information and click **Next**.



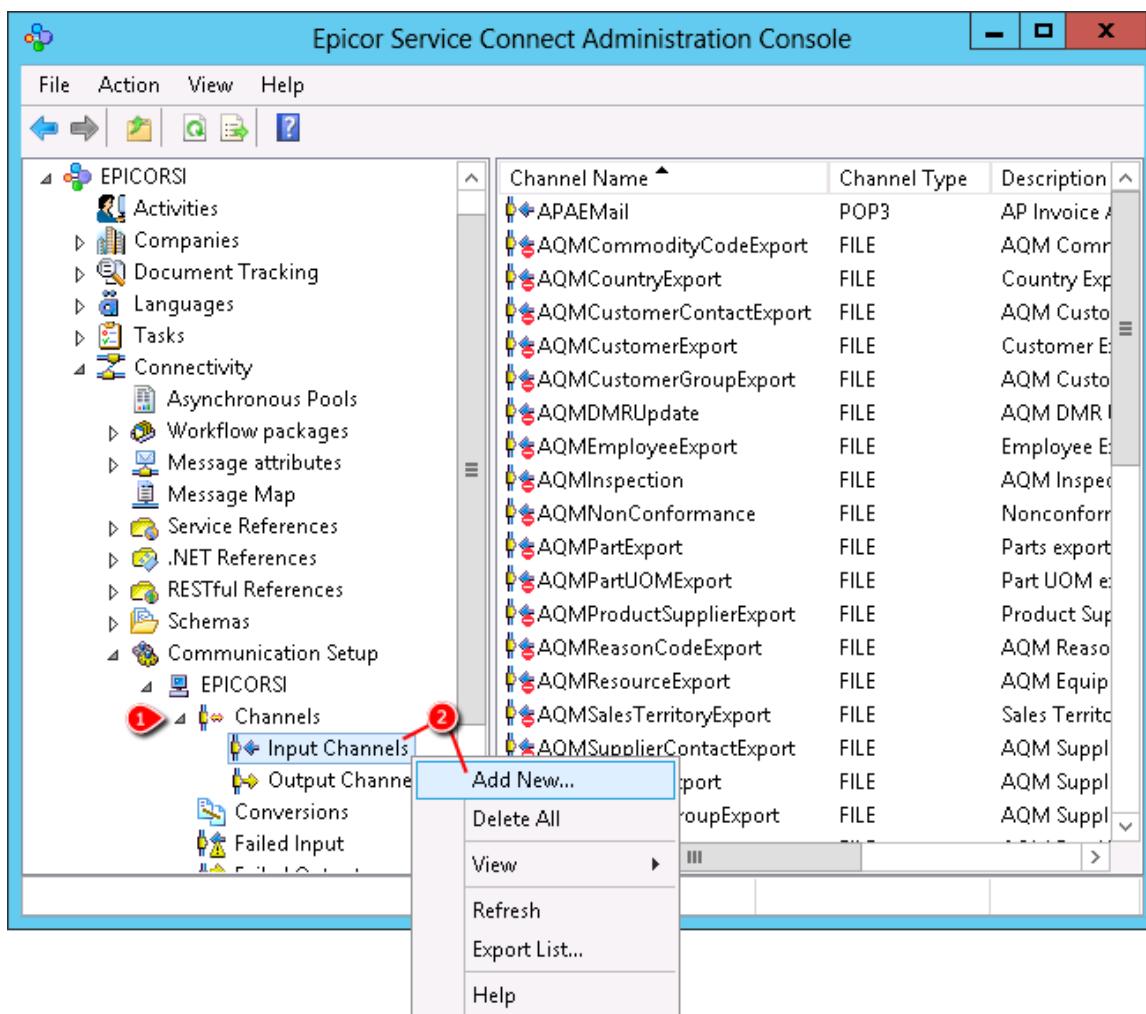
- 14.** Click **Finish** after the import is complete.

Add an Input Channel

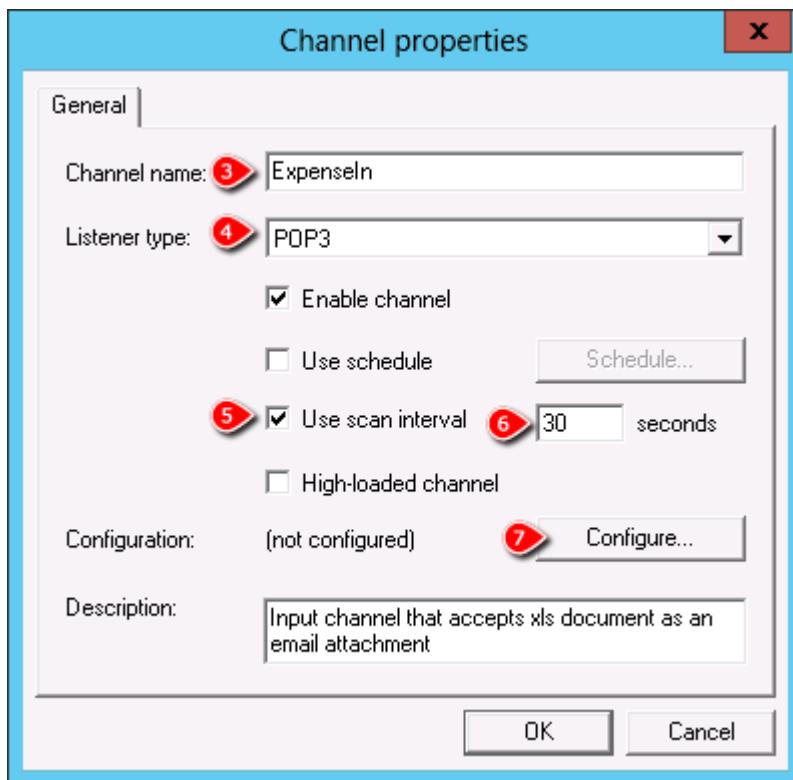
An input channel serves as an entry point into the workflow. The following steps show you how to create an input channel that accepts a Microsoft Excel document as an email attachment. The input channel monitors the inbox of a POP3 mail account. For more information on the other types of input channels Service Connect can monitor, refer to Chapter 3: Connectivity Components.

To create an input channel:

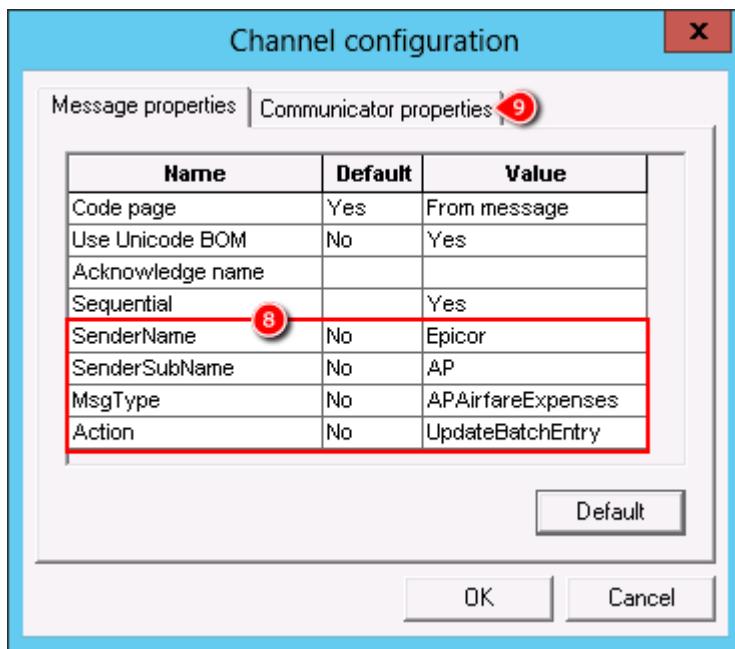
1. In Epicor Service Connect Administration Console, expand the **Connectivity > Communication Setup >** server name > **Channels** node.



2. Right-click **Input Channels** and select **Add New**.
3. In the **Channel properties** window, enter a **Channel name**.



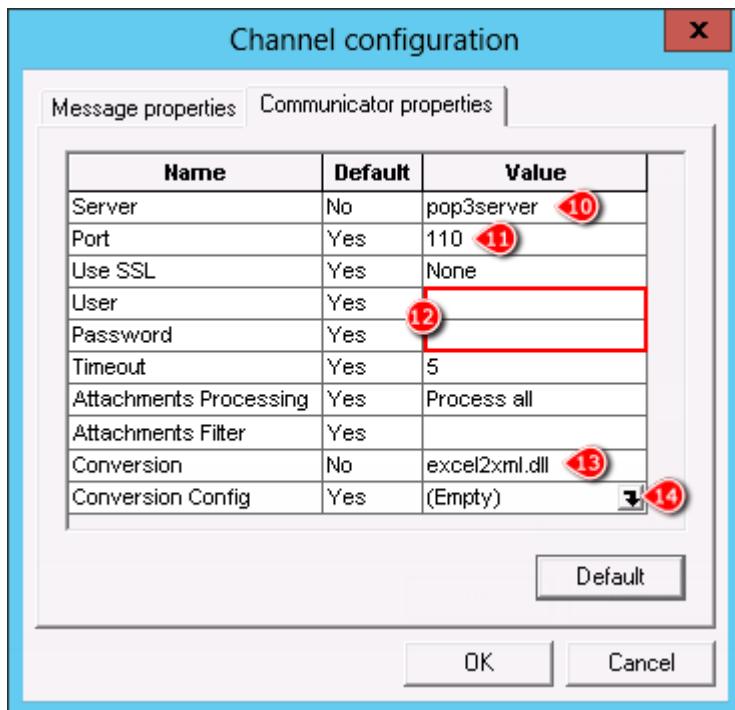
4. Select **POP3** as the **Listener type**.
5. Select the **Use scan interval** check box.
6. Increase the interval value to **30** seconds.
Some POP3 services block requests from a specific address if it connects to the mailbox too often.
7. Click **Configure**.
8. In the **Channel configuration** window, for the following fields, select the values used for the message map created earlier:
 - SenderName
 - SenderSubName
 - MsgType
 - Action



9. Open the **Communicator properties** tab.

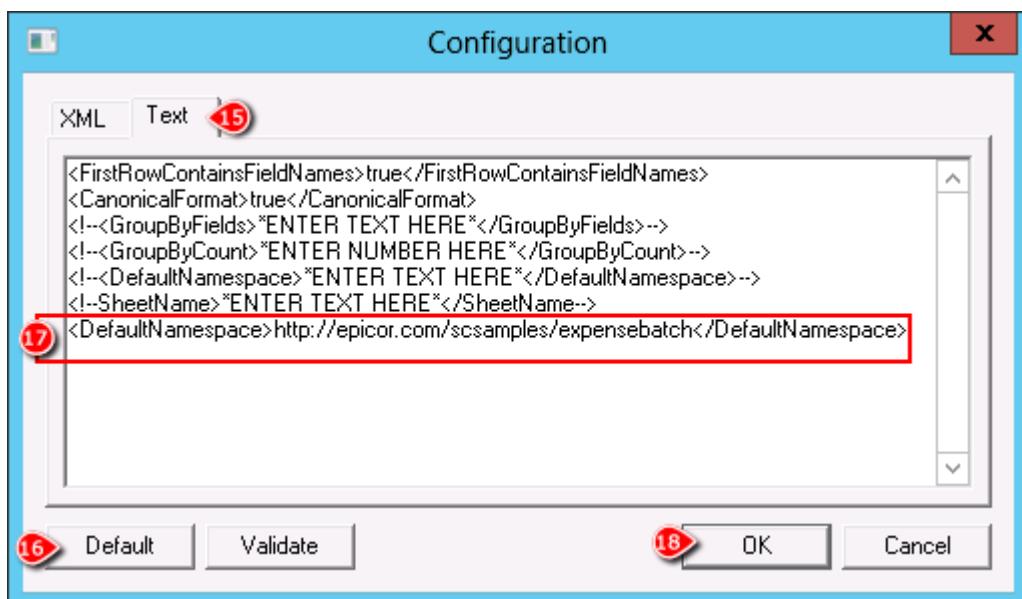
10. Enter the **Server** name of a POP3 mail server.

If the server is within a trusted domain, you can enter just the server name. If the server is outside of 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 POP3 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 Value field in the **Conversion Config** row, then click the down arrow button.
You must click the field first to display the down arrow button.
15. In the **Configuration** window, open the **Text** tab.



16. Click the **Default** button.
17. Add the following line to the configuration: **<DefaultNamespace>
http://epicor.com/scsamples/expensebatch </DefaultNamespace>**.
Typing this value into the 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.
18. Click **OK** until you exit all dialog boxes.

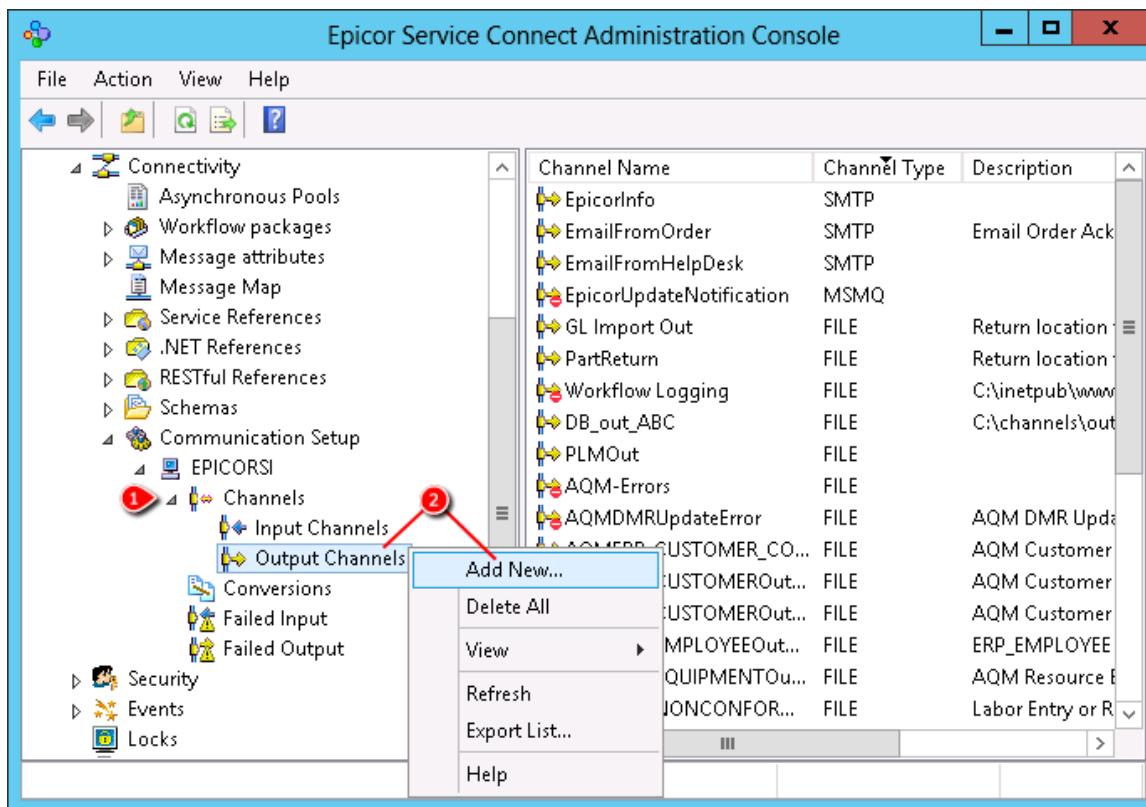
Add an Output Channel

An output channel serves as an exit point for documents that leave the workflow. If the workflow performs its functions without sending documents to other applications from Service Connect, an output channel may not be required.

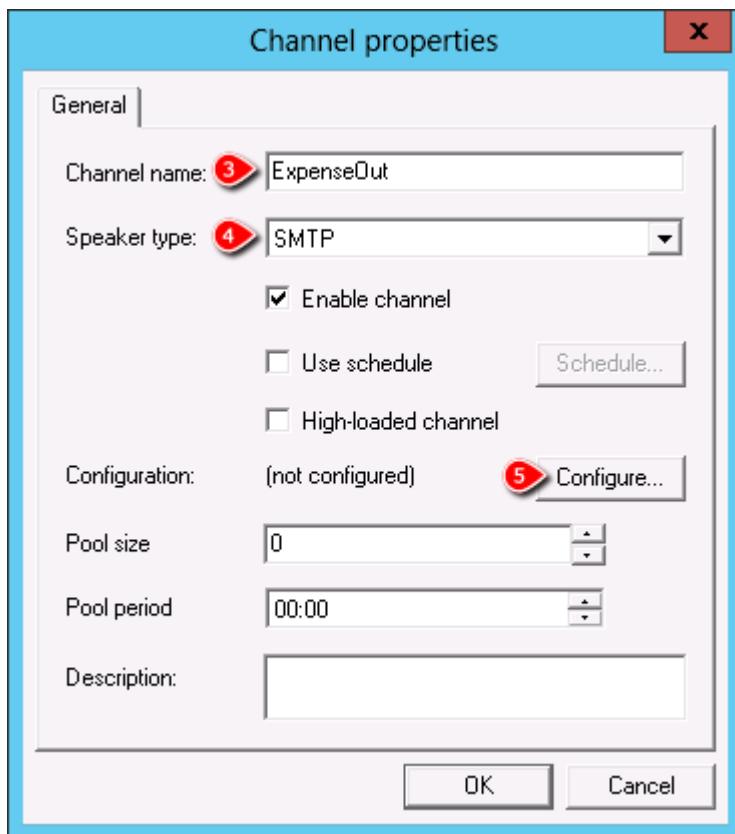
The following procedure shows you how to add an output channel you can use to send an email to someone to notify them about the workflow status. For information about the other types of output channels Service Connect can use to send information, refer to Chapter 3: Connectivity Components.

To add an output channel:

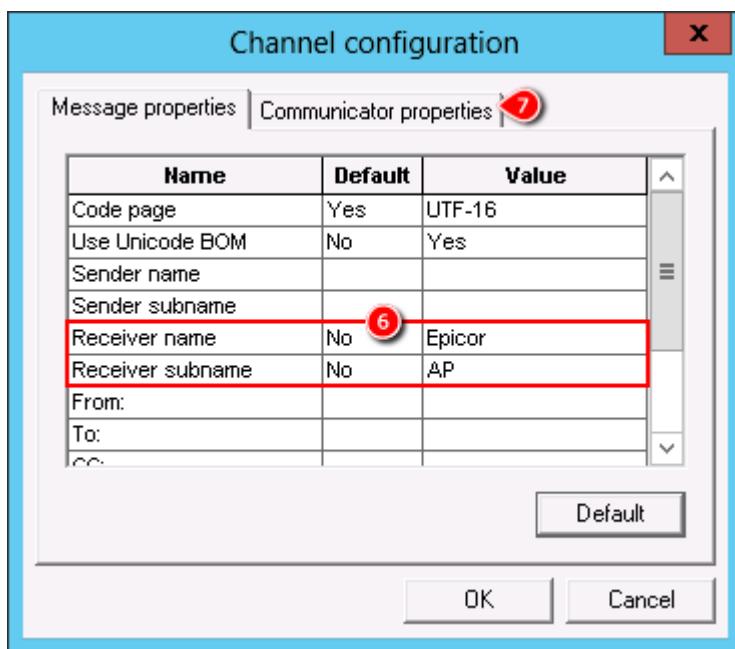
1. In Epicor Service Connect Administration Console, expand the **Connectivity > Communication Setup > server name > Channels** node.



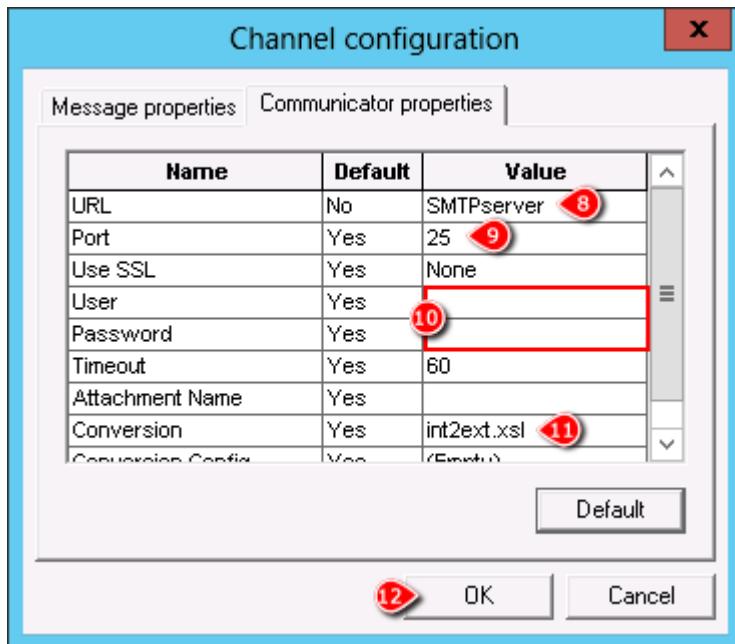
2. Right-click **Output Channels** and select **Add New...**.
3. In the **Channel properties** window, enter a **Channel name**.



4. Select **SMTP** as the **Speaker type**.
5. Click **Configure**.
6. In the **Channel configuration** window, select the appropriate sender name and subname as the **Receiver name** and **Receiver subname**.



7. Open the **Communicator properties** tab.
8. 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.

9. 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.

10. If the SMTP server is outside the trusted domain, enter the **User** and **Password** for an account on the server.
11. Select **int2ext.xsl** as the **Conversion**.
12. 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.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Last_Name	First_Name	Inv_Nbr	Inv_Date	Date_Booked	Travel_Date	Airline	City_Routing	Fare	Bill	Reason1	Reason2	Svc_Fee_Description	Resource_Code
2	Baird	Melanie	307348	10-Apr	10-Apr	11-May	KLM	LGB JFK LGB	\$370.00	N	Sale Demo		Dom Tkt	BAIRD
3	Baird	Melanie	307348	10-Apr			Invco Tel Fee		\$30.00	N	Sale Demo			BAIRD
4	Baird	Melanie	307348	10-Apr			Invco Tel Fee		\$10.00	N	Sale Demo			BAIRD
5	Burge	Norman	307698	14-Apr	13-Apr	17-May	Mega	LAX JFK LAX	\$405.74	N	Sale Demo		Dom Tkt	BURGEN
6	Burge	Norman	307698	14-Apr			Invco Tel Fee		\$40.00	N	Sale Demo			BURGEN
7	Burge	Norman	307698	14-Apr			Invco Tel Fee		\$40.00	N	Sale Demo			BURGEN

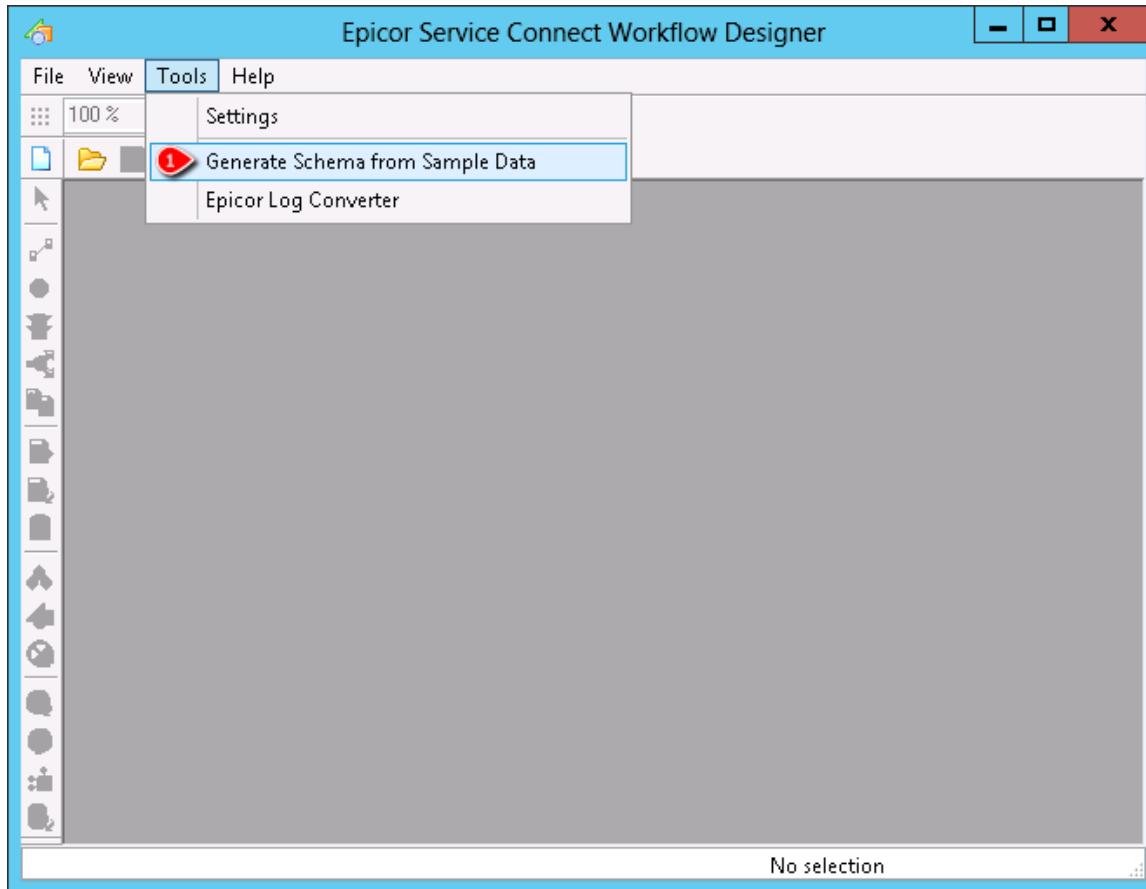
The spreadsheet columns include:

- **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

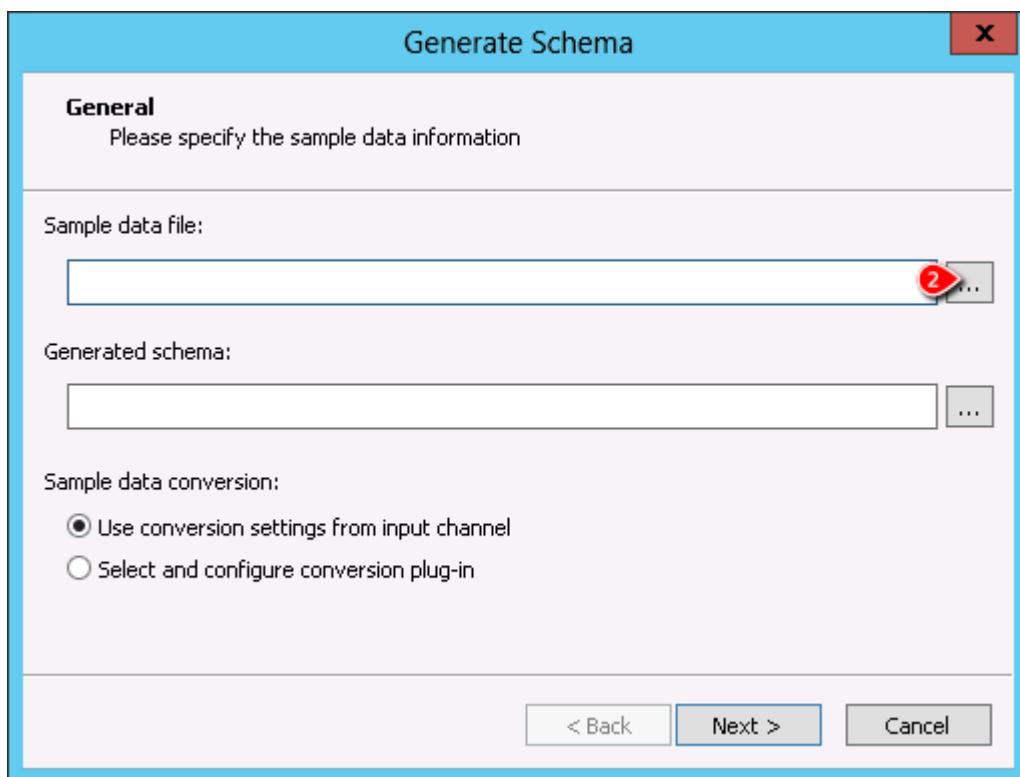
To generate a user schema:

1. In the Workflow Designer, from the **Tools** menu, select **Generate Schema from Sample Data**.

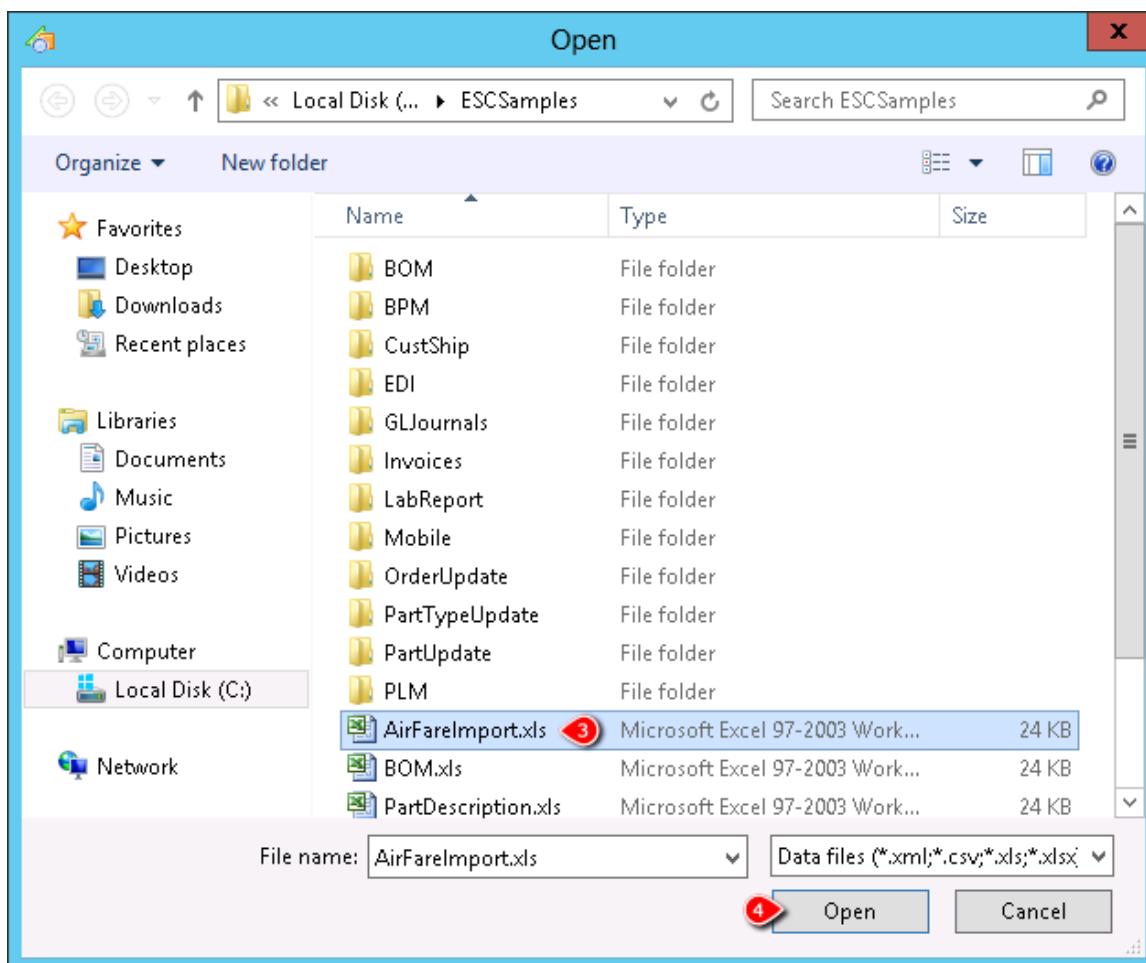


The **Generate Schema - General** window displays.

2. Next to the **Sample data file** field, click the browse (...) button.

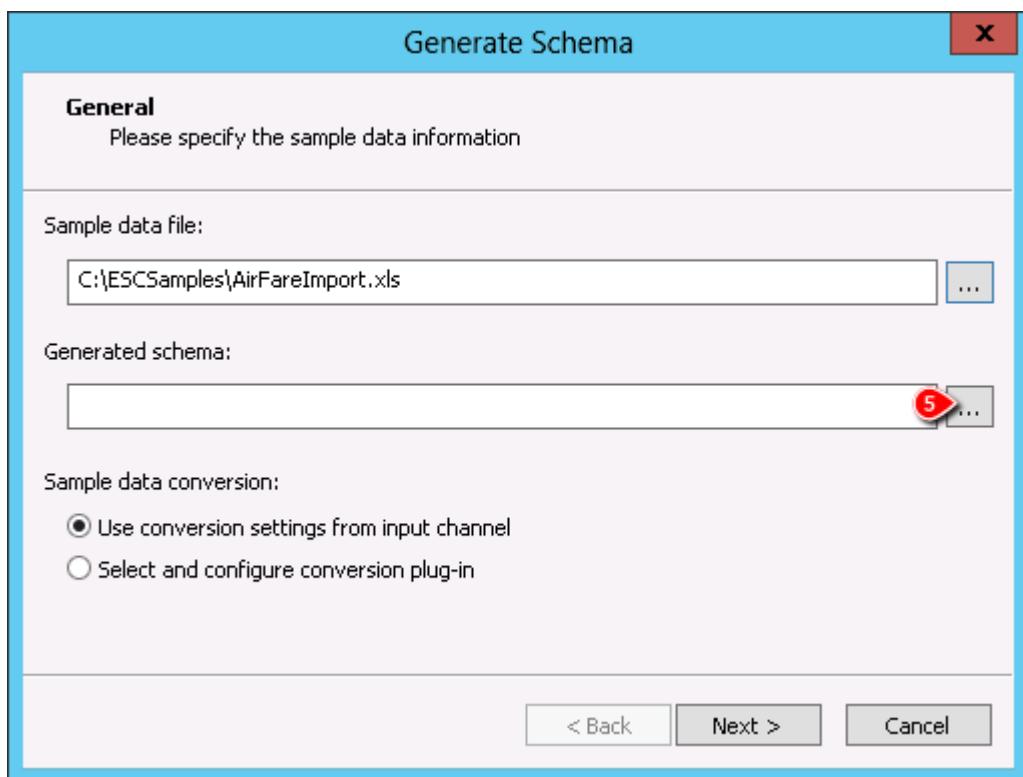


3. In the **Open** window, browse to and select the Excel file.

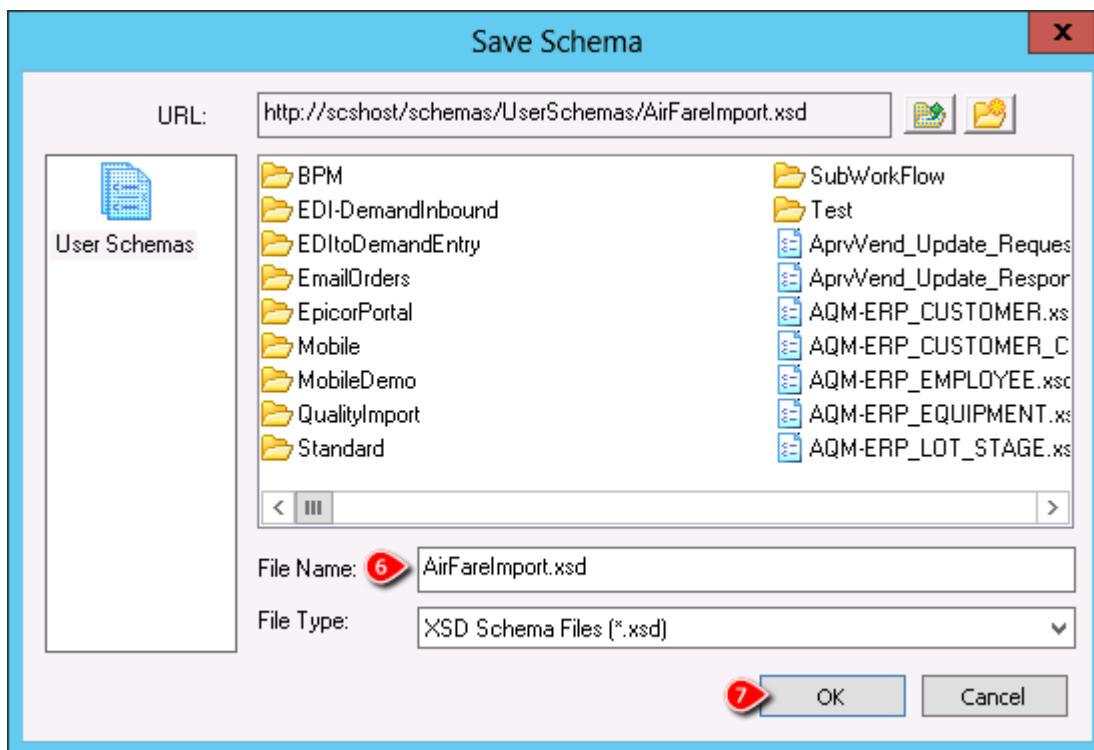


4. Click **Open**.

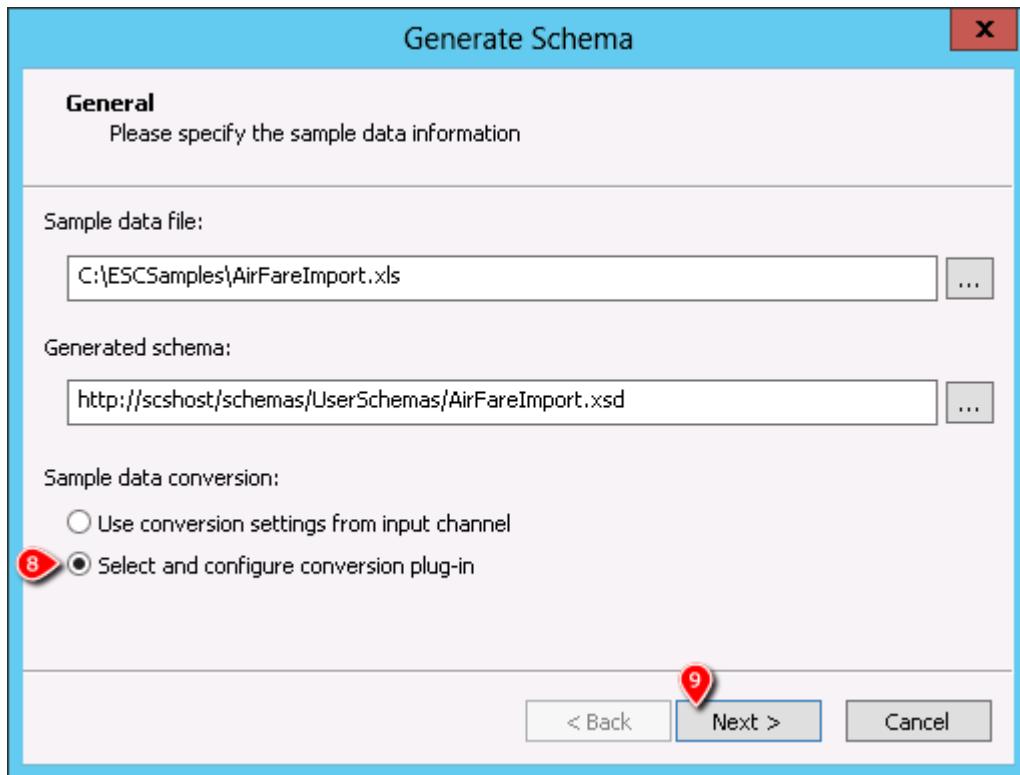
5. In the **Generate Schema - General** window, next to the **Generated schema** field, click the browse (...) button.



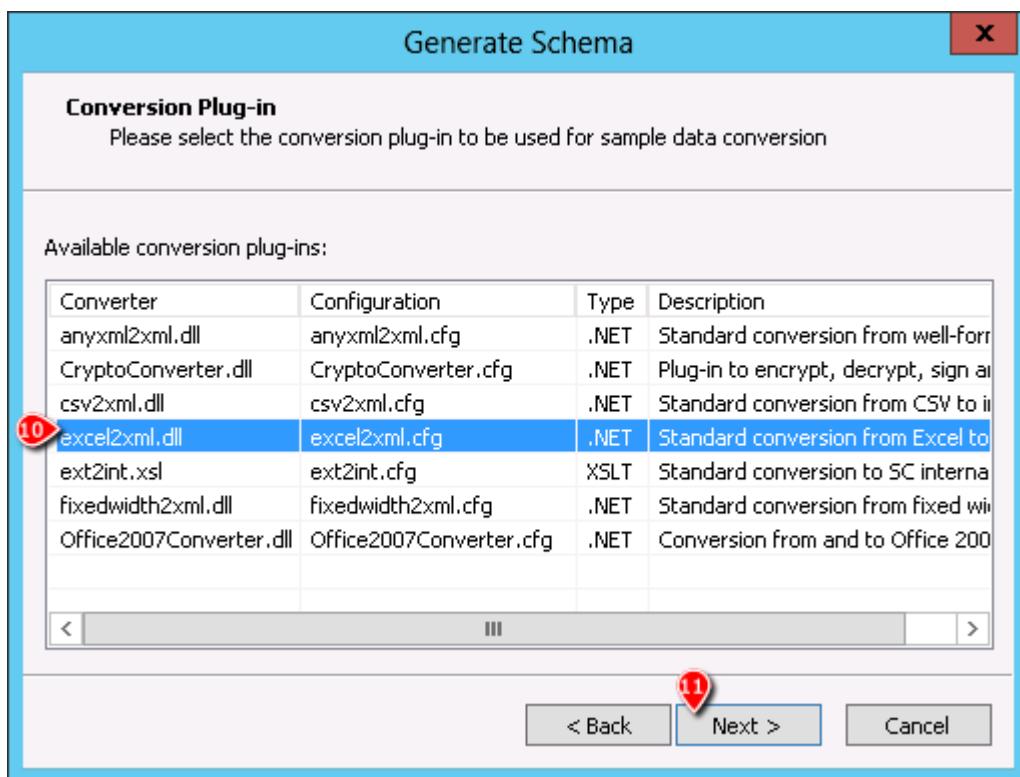
6. In the **Save Schema** window, in the **File name** field, enter a name for the schema file.



7. Click **OK**.
8. Select the **Select and configure conversion plug-in** option.

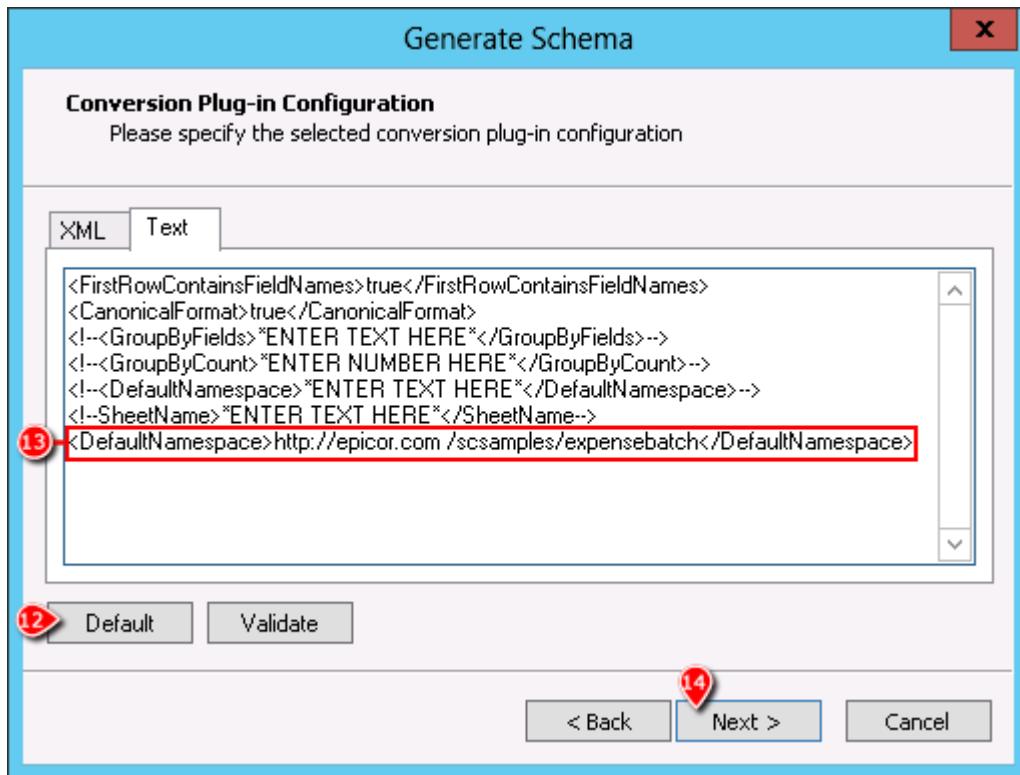


9. Click **Next**.
10. In the **Generate Schema - Conversion Plug-In** window, select **excel2xml.dll**.



11. Click **Next**.

12. In the **Generate Schema - Conversion Plug-in Configuration** window, open the **Text** tab and click **Default**.

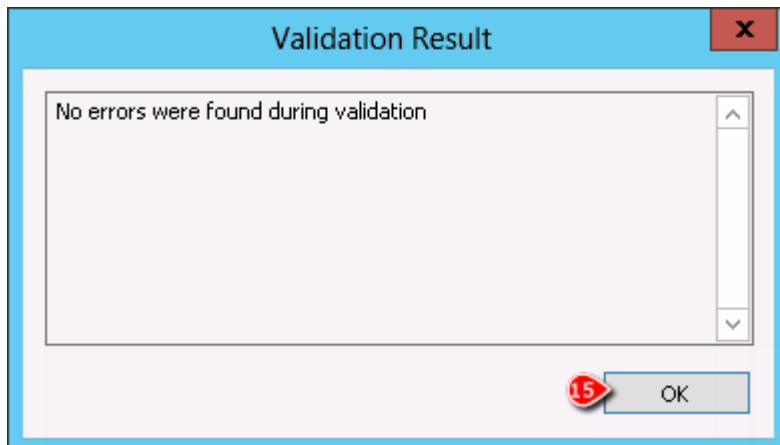


13. Add **<DefaultNamespace>http://epicor.com /scsamples/expensebatch </DefaultNamespace>**.

Entering this value as the plug-in configuration 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.

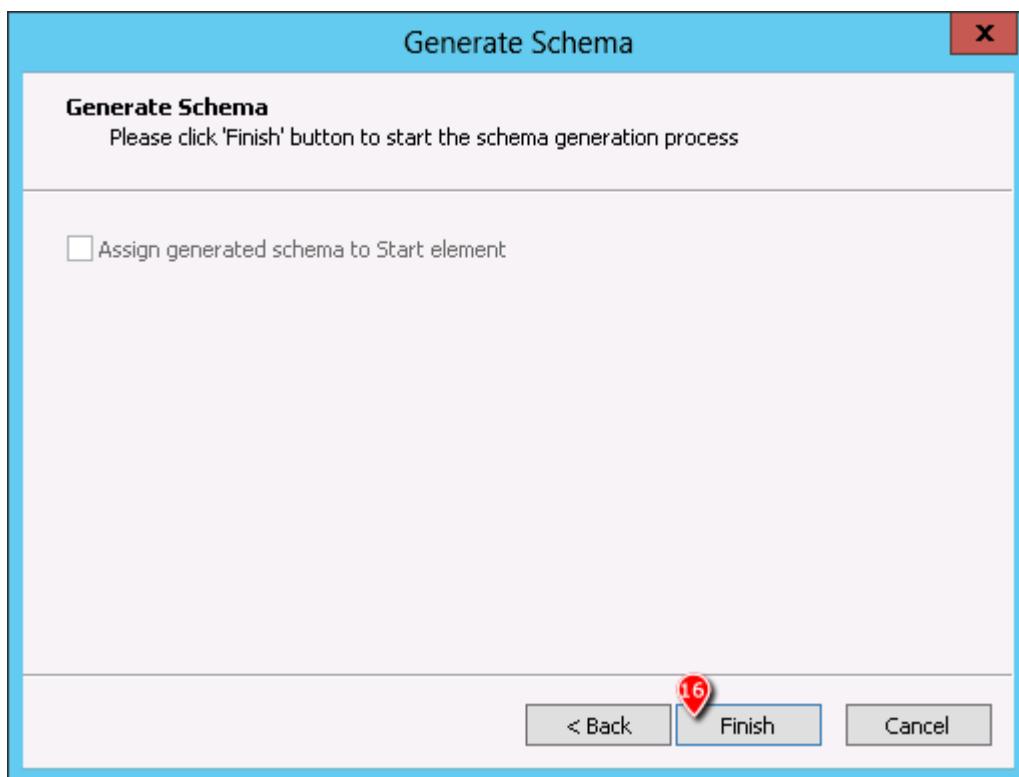
14. Click **Next**.

15. To the **Validation Result** warning, click **OK**.



Each column header in the spreadsheet becomes an element beneath the row element.

16. In the **Generate Schema - Generate Schema** window, click **Finish**.



The schema is created and Service Connect recognizes it as a user-generated schema. 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.

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.

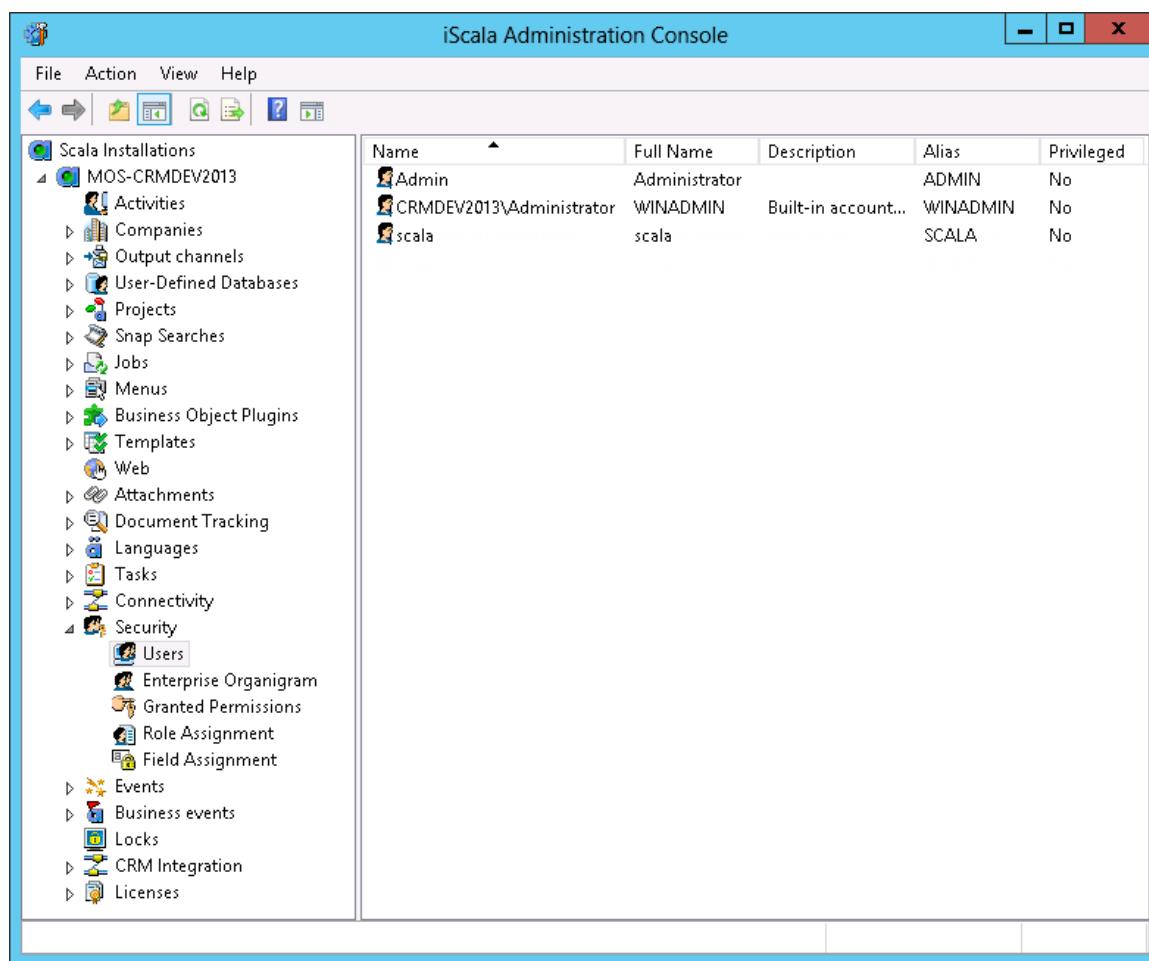
When Service Connect is installed as part of iScala, some data and integration parameters will require migration. Most of this process is done automatically during installation, but some issues require user attention.

Security

To set up and maintain security in your iScala system, the iScala Administration Console provides the following features:

- User Accounts
- Enterprise Organigram
- Permission Settings
- Role Assignment

In the iScala 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 iScala Administration Console tree in several places:

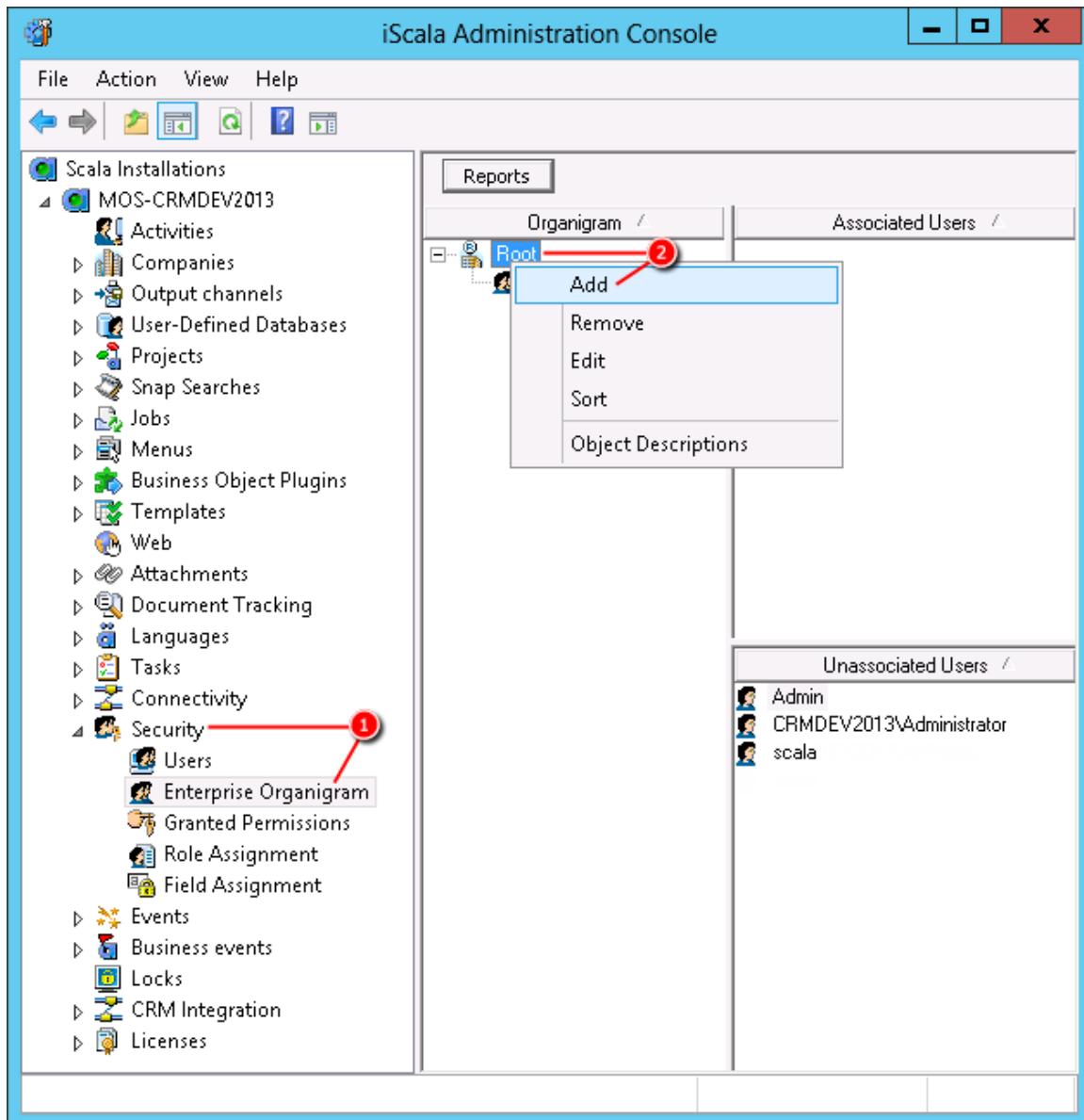
- **As root nodes** – To manage security items, users, and user activities in all the companies of the current installation.
- **Under each company node in the tree** – To manage company-specific security items, users, and activities.

The ability to view these settings depends on your administration rights.

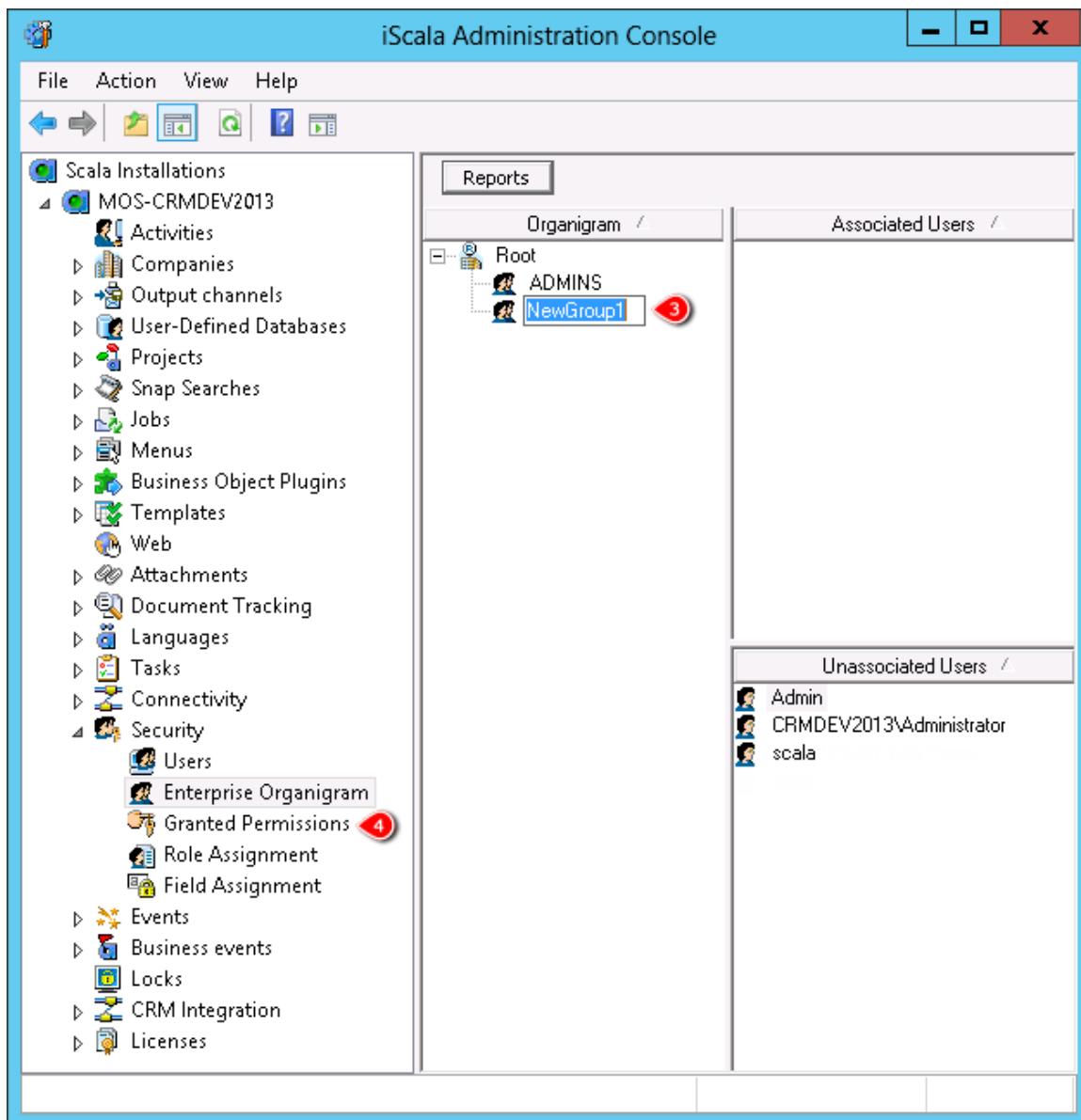
Grant Access Rights

To grant access rights in iScala:

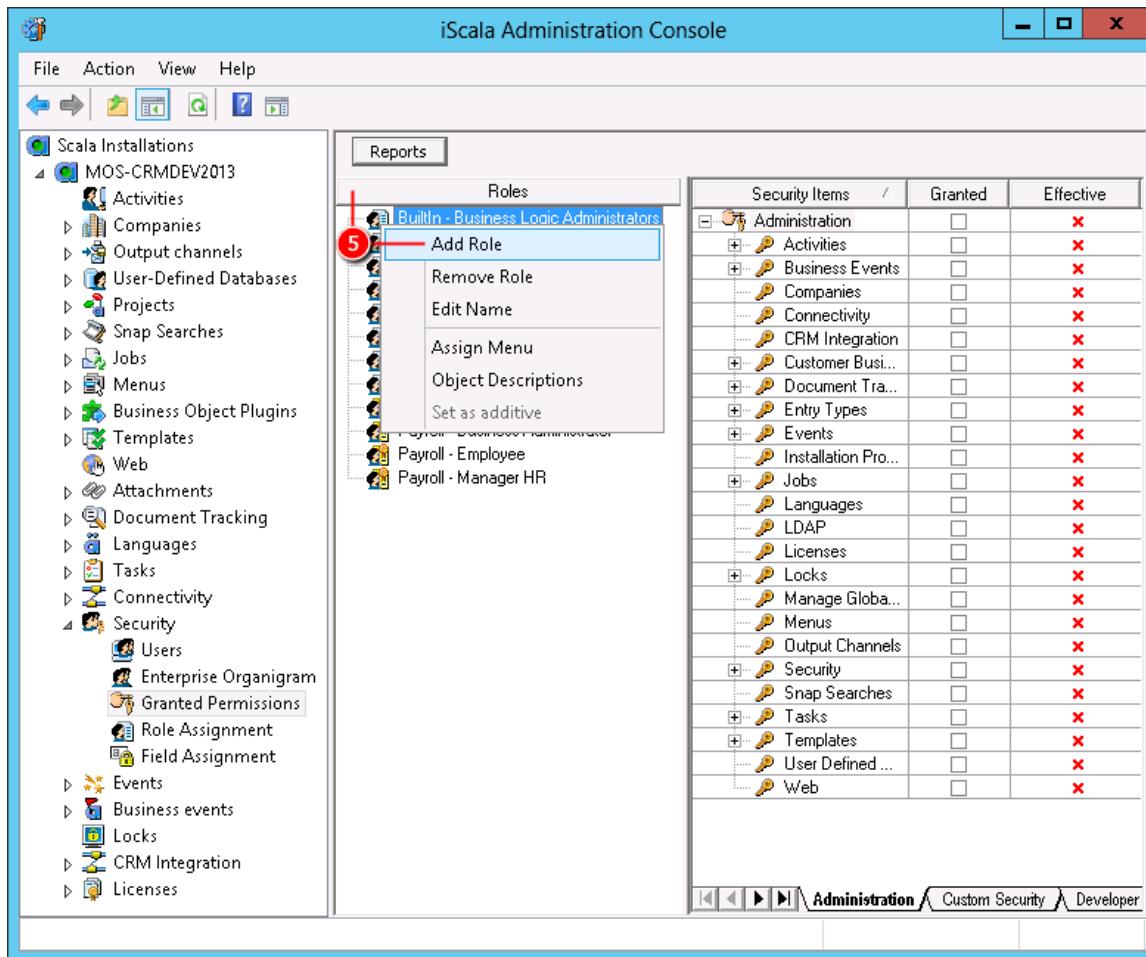
1. In the iScala Administration Console tree view, expand the **Security** node and select **Enterprise Organigram**. All users in the current installation of iScala are grouped hierarchically. An organigram is a diagram of the company's structure.



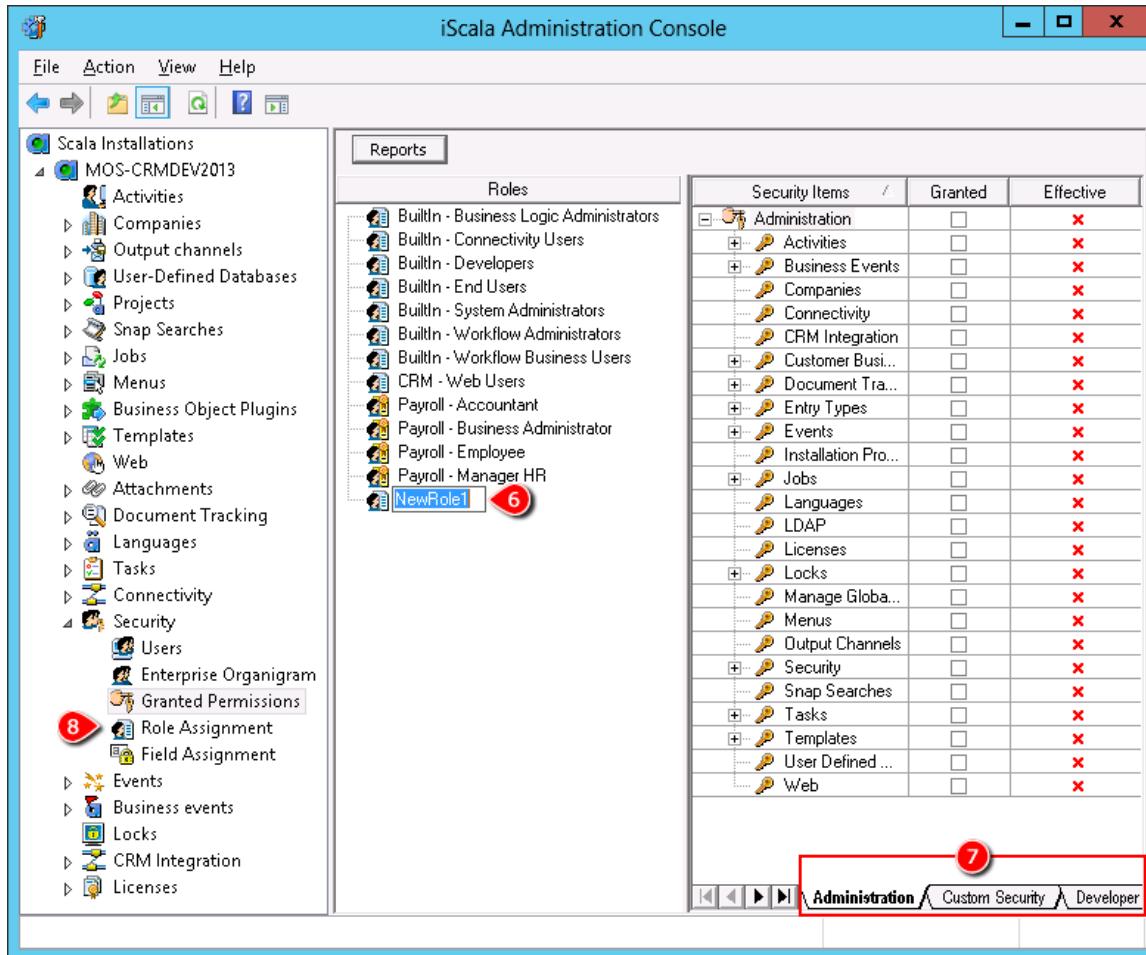
2. In the middle pane, right-click **Root** and select **Add** to create a group.
3. Type the group name to replace the default NewGroup1 text.



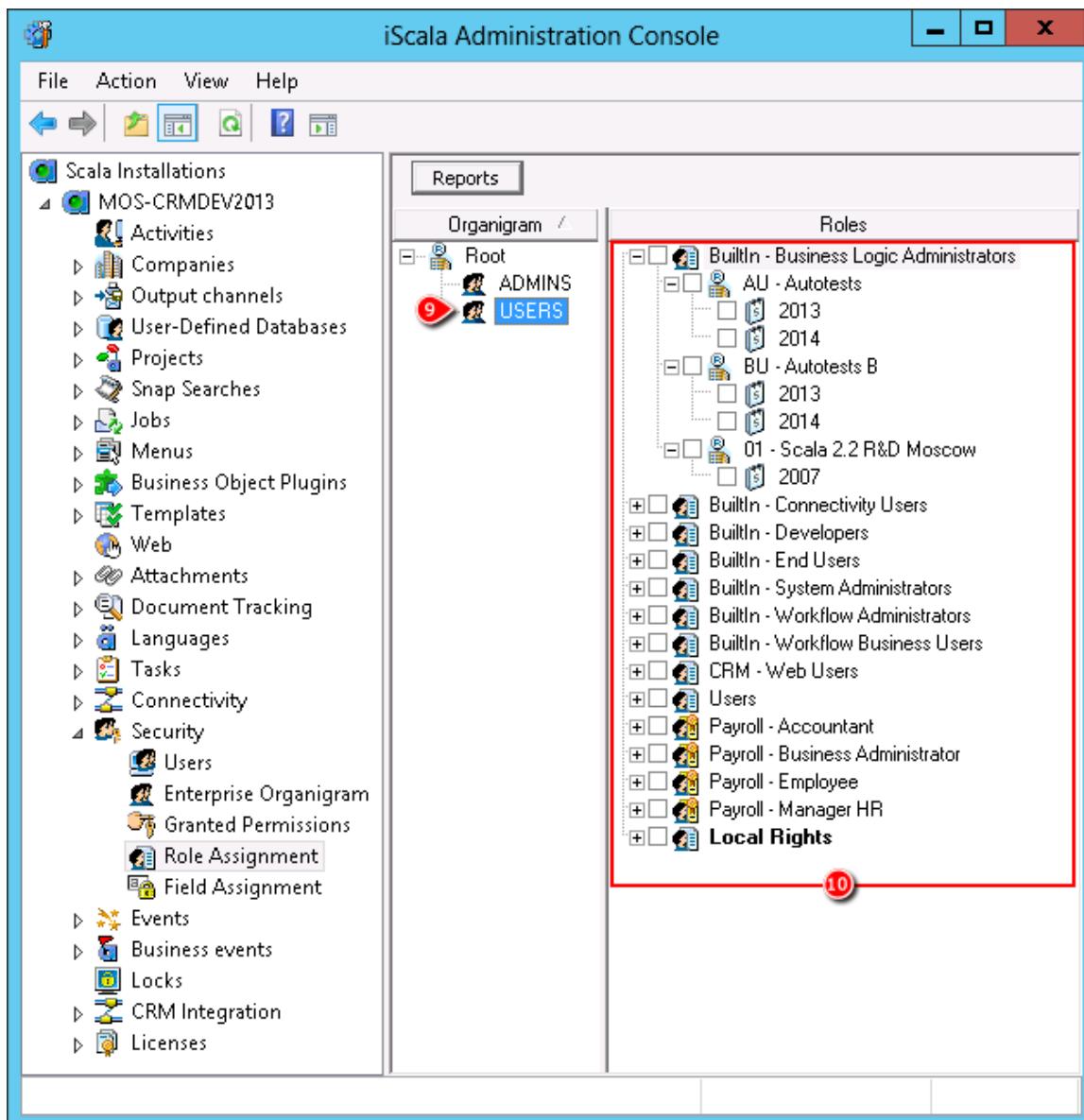
4. In the tree view, select the **Granted Permissions** node.
5. In the middle pane, right-click the **Roles** list and select **Add Role** to create a new role.



6. Type the role name to replace the default NewRole1 text.



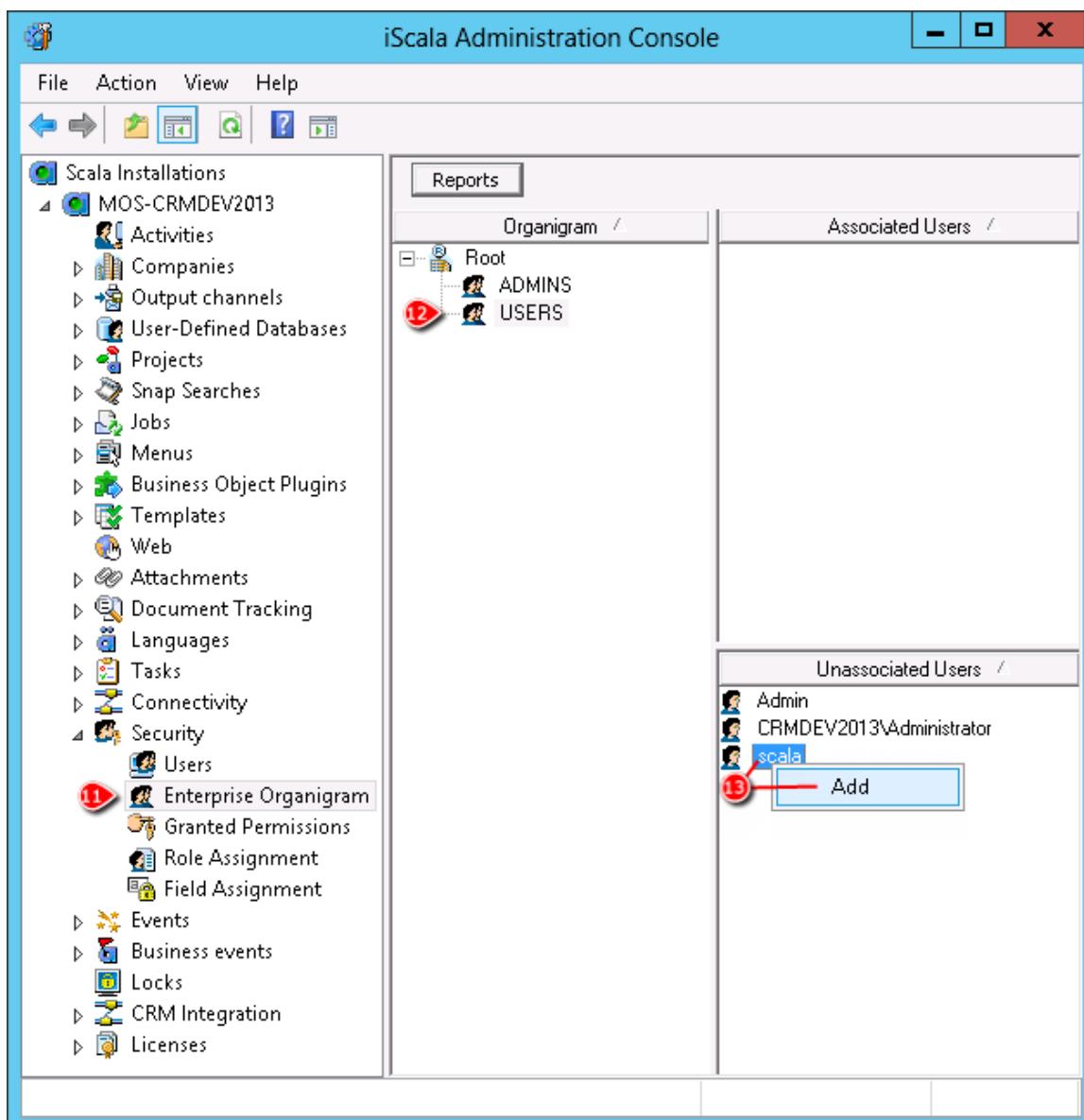
7. To set permissions for the new role, click the various tabs at the bottom of the right pane and grant rights to specific security items.
Alternately, you can find an already existing role that has the required permissions. Examples of built-in roles include System Administrators, Workflow Administrators, Workflow Business Users, and End Users.
8. In the tree view, select the **Role Assignment** node.
9. In the middle pane, click the user group in the **Organigram** tree



10. Select the role or roles check boxes in the **Roles** pane to map the organigram user group to the role or roles.

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.

11. In the tree view, select the **Enterprise Organigram** node.
12. In the center pane under **Organigram**, click the group to which you want to add the user.



13. 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

You have three options to add users:

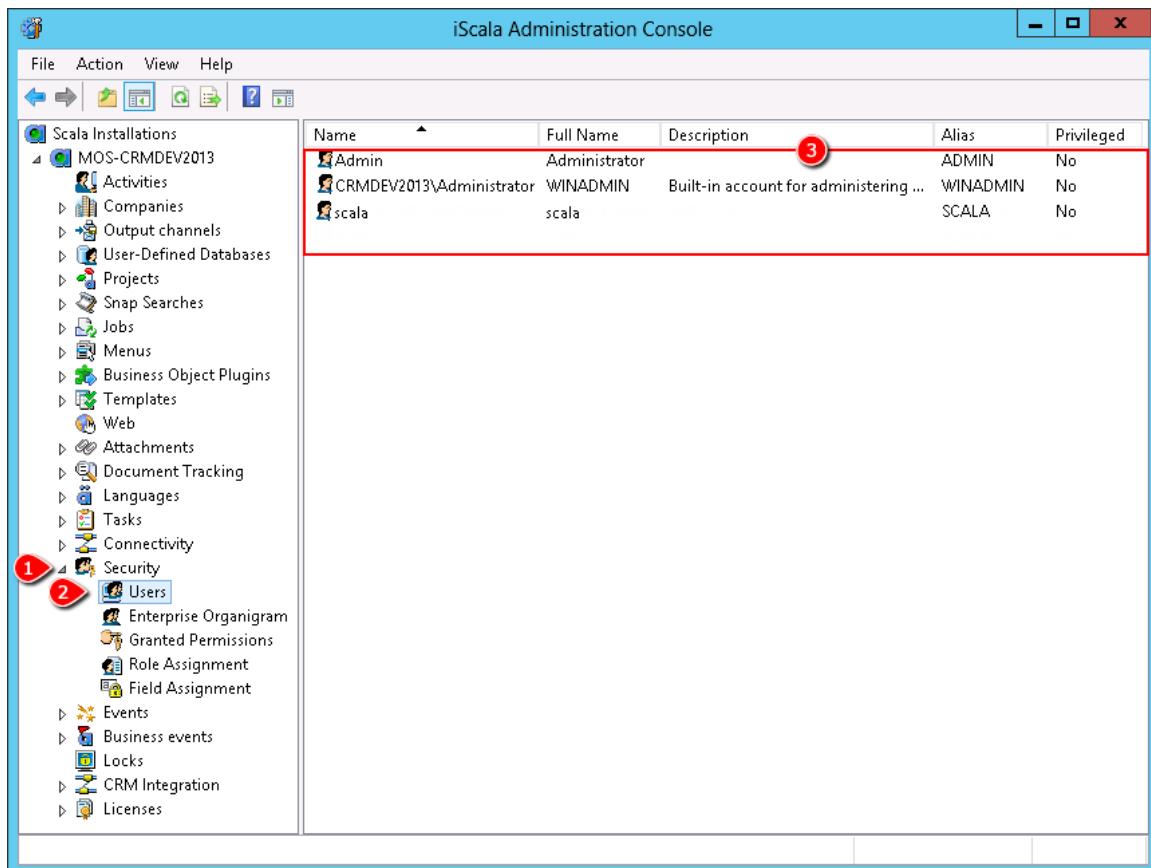
- Create a new user
- Import a Windows user or a group of Windows users

- Import a list of users from a file; for example, you can 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.



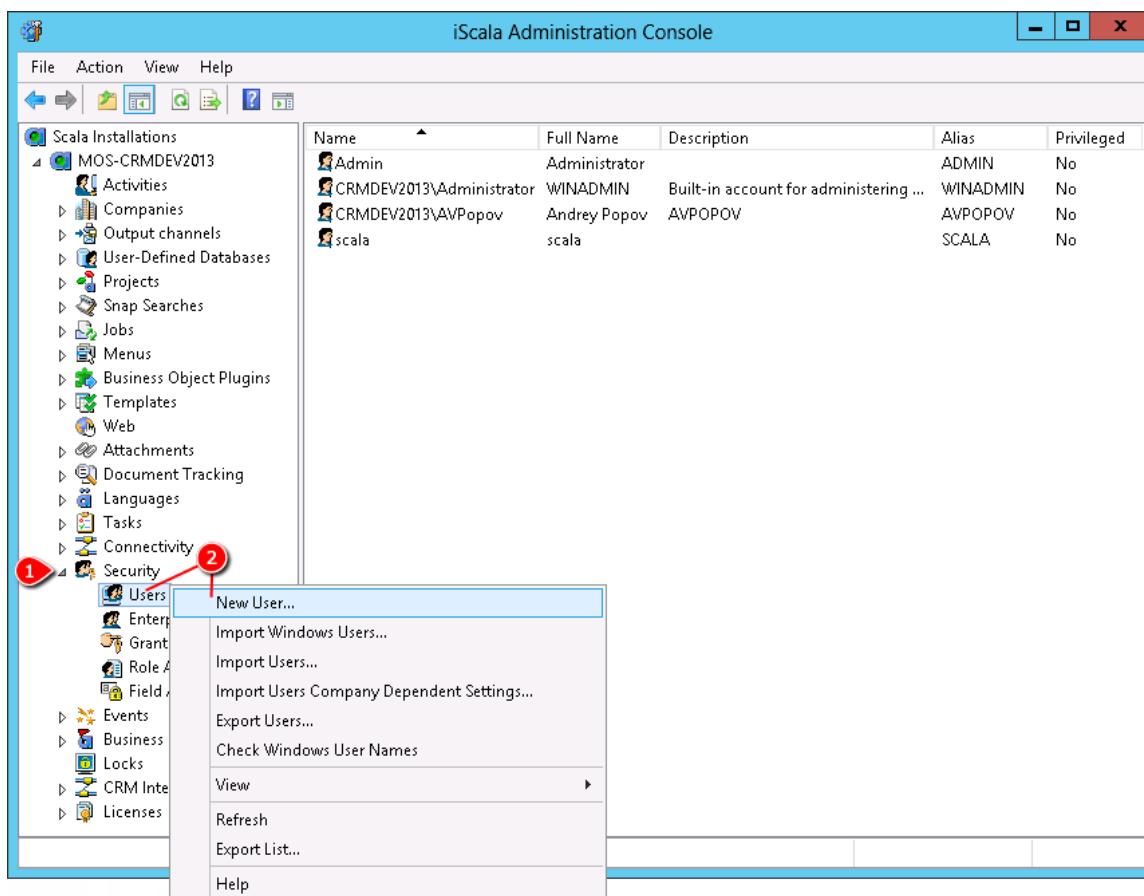
2. Click the **Users** node. The list of all user accounts displays in the right pane.
3. Review the user information, including the **Name**, **Full Name**, **Description**, **Alias**, and **Privileged**.

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

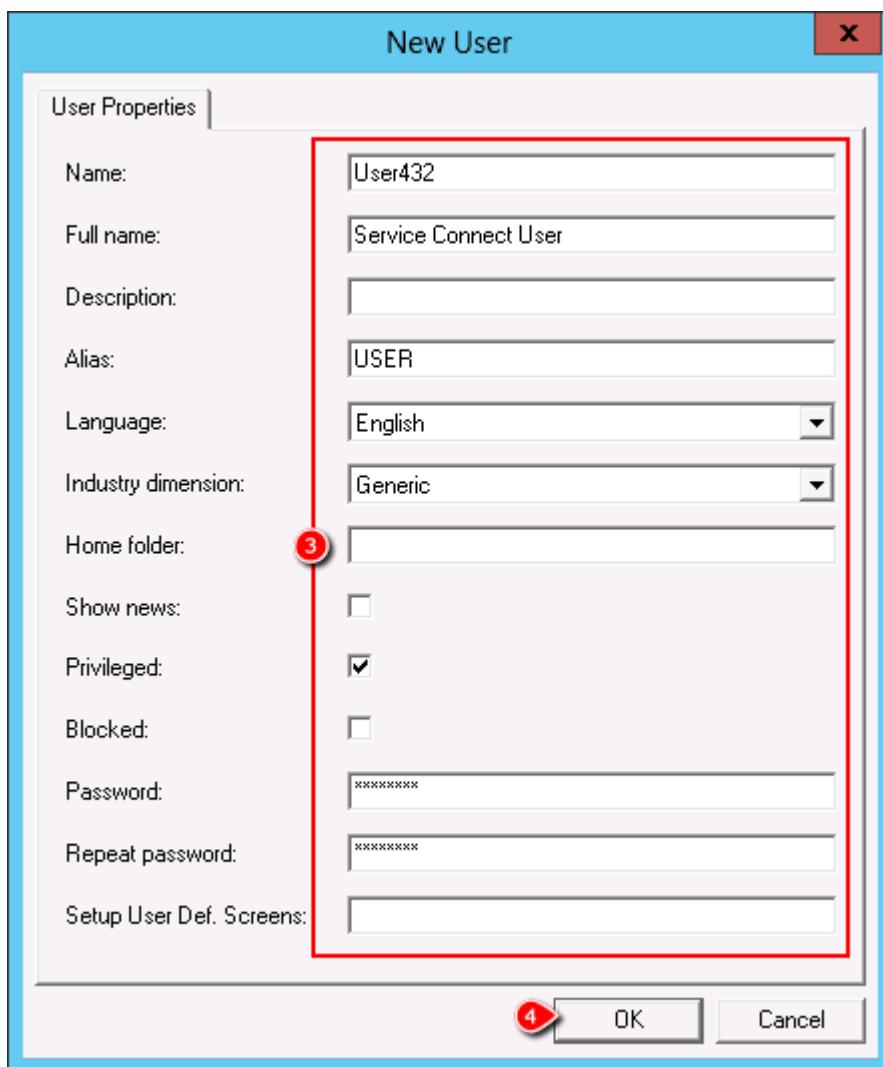
Use the following steps to create a new user:

1. In the Administration Console, expand the **Security** node.



2. Right-click the **Users** node and select **New User...**.

3. Set the **User Properties**.

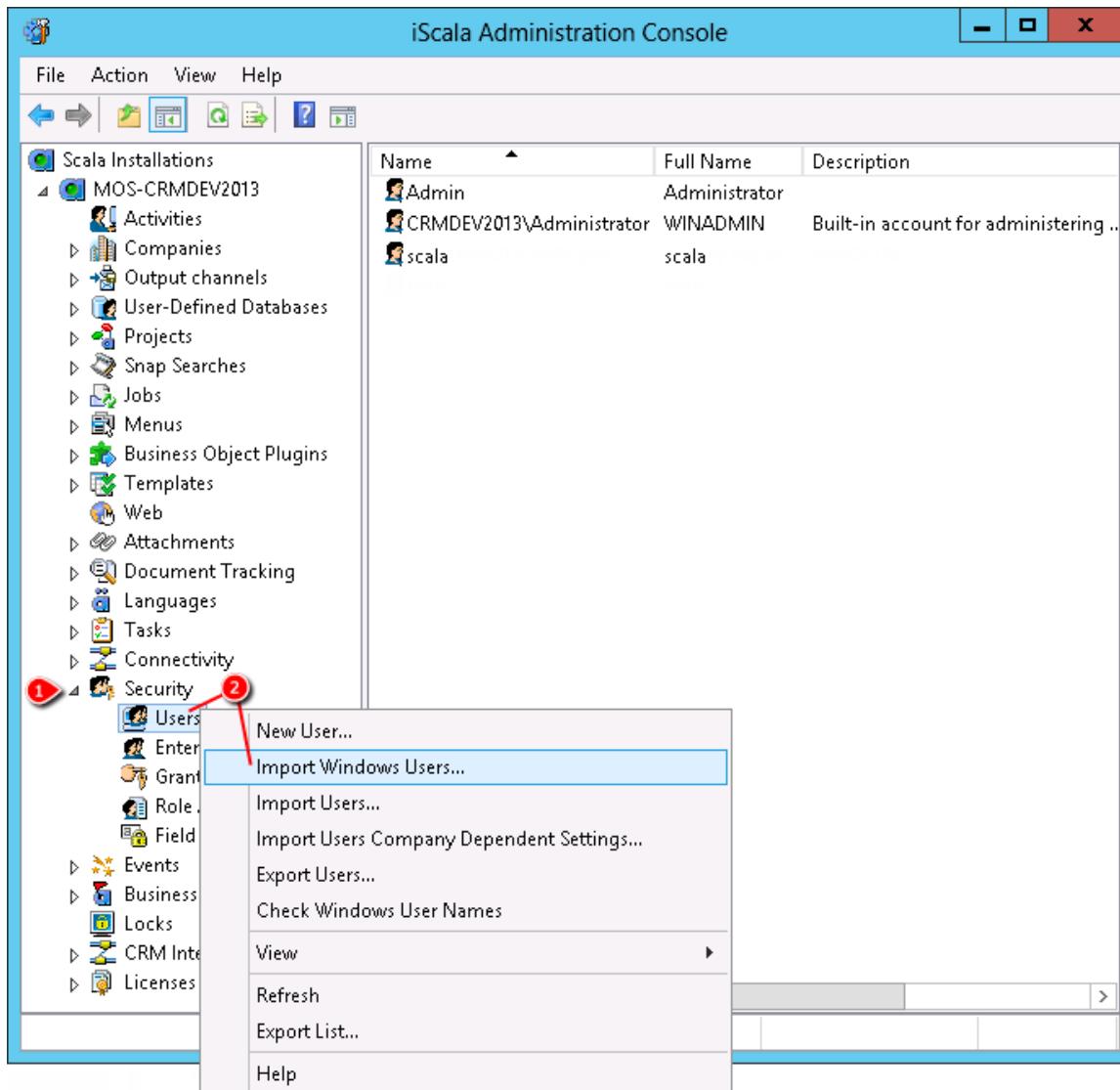


4. Click **OK**.

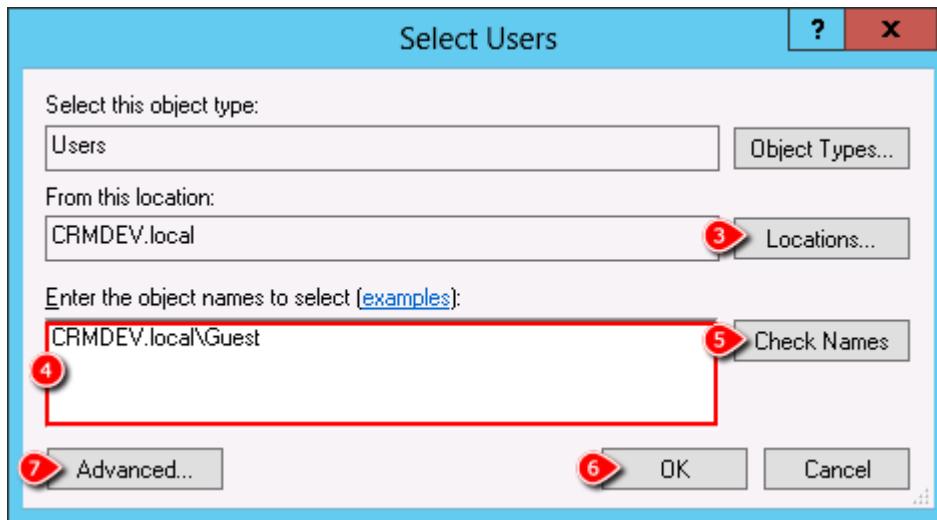
Import a Windows User

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.

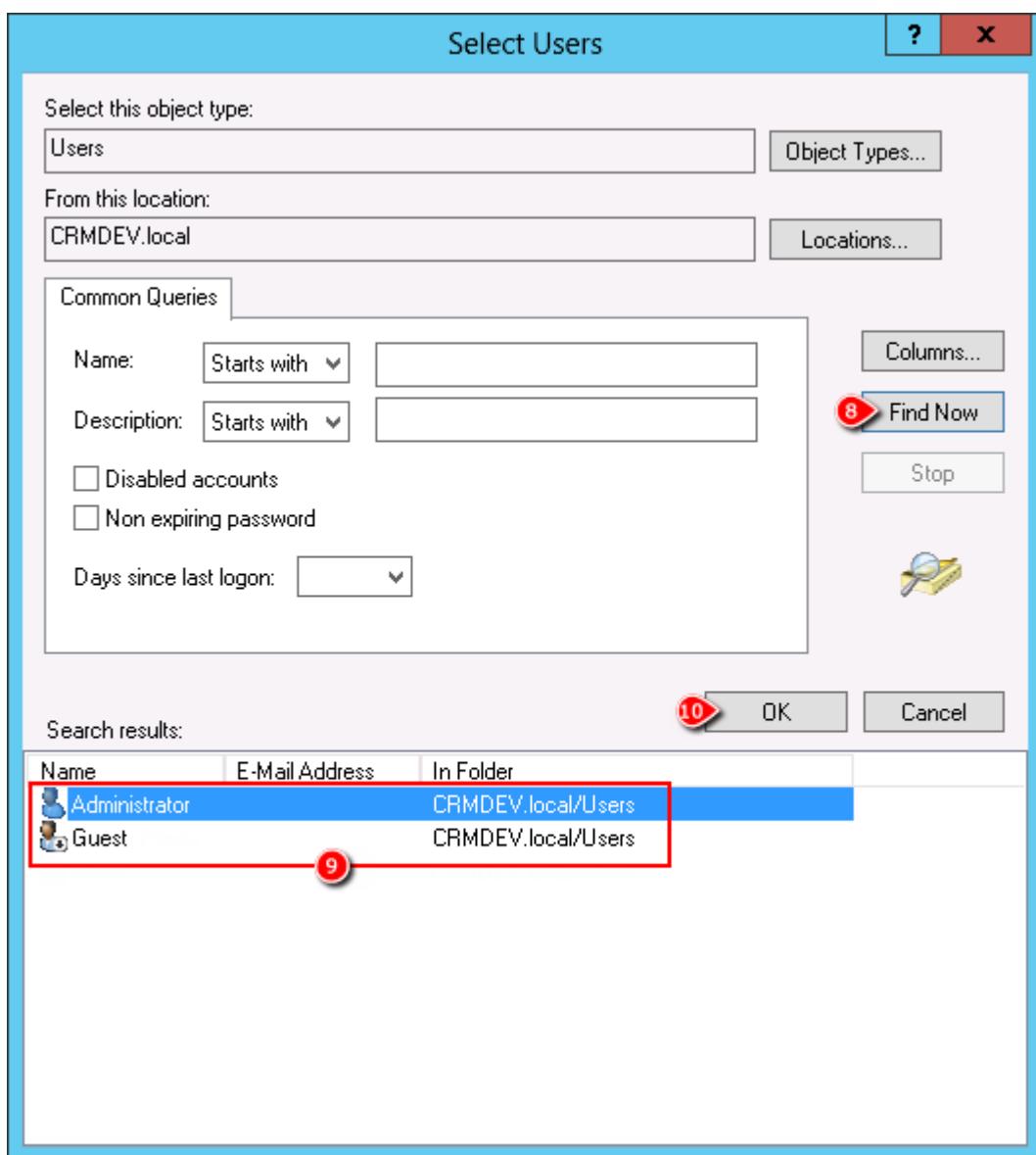


4. Type the user names in the **Enter the object names to select** field.

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.
8. In the **Select Users** window, 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.

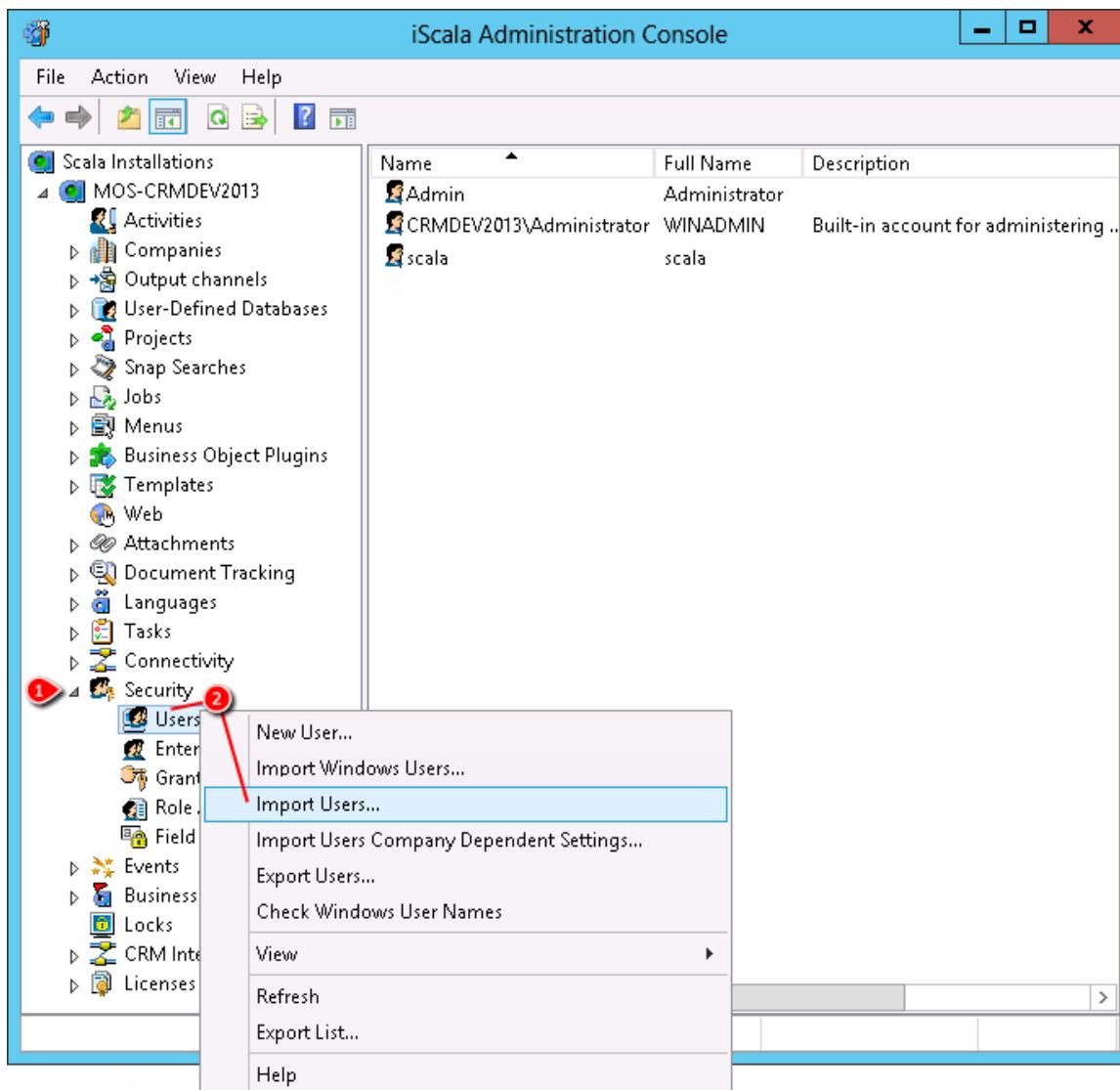
9. Select the user or users from the **Search results** list.

10. Click OK.

Import a List of Users from a File

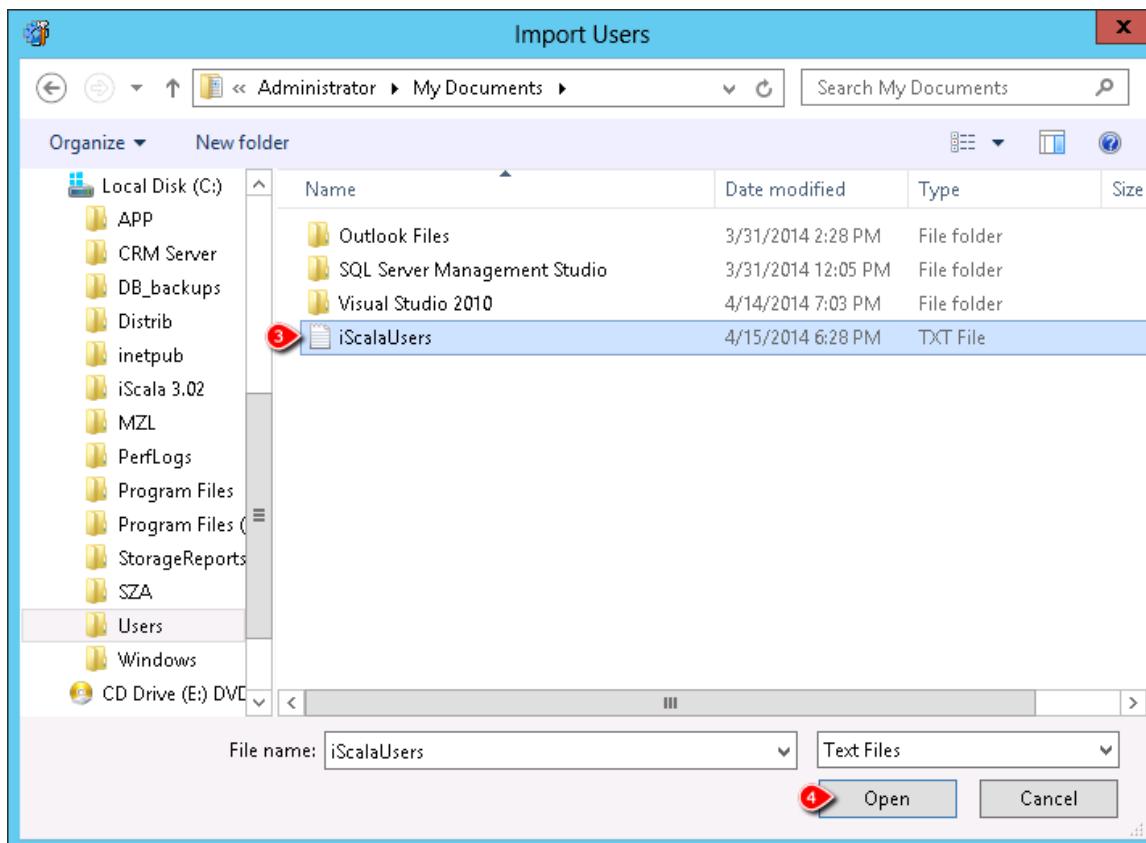
Use the following steps to import a single user or multiple users from a text file:

1. In the Administration Console, expand the **Security** node.



2. Right-click **Users** and select **Import Users...**.

3. In the **Import Users** window, locate the text file that contains the user list you want to import.

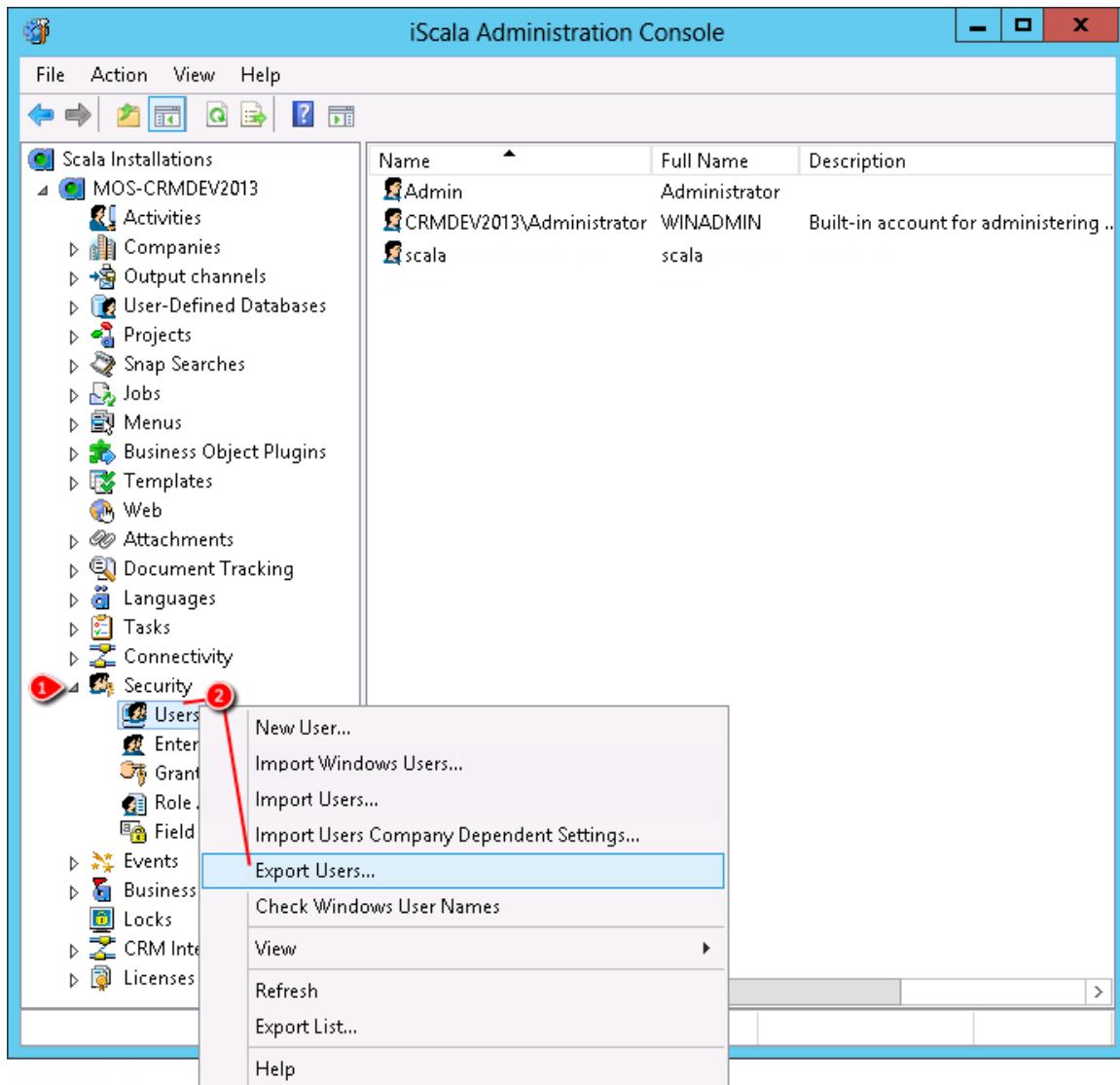


4. Click Open.

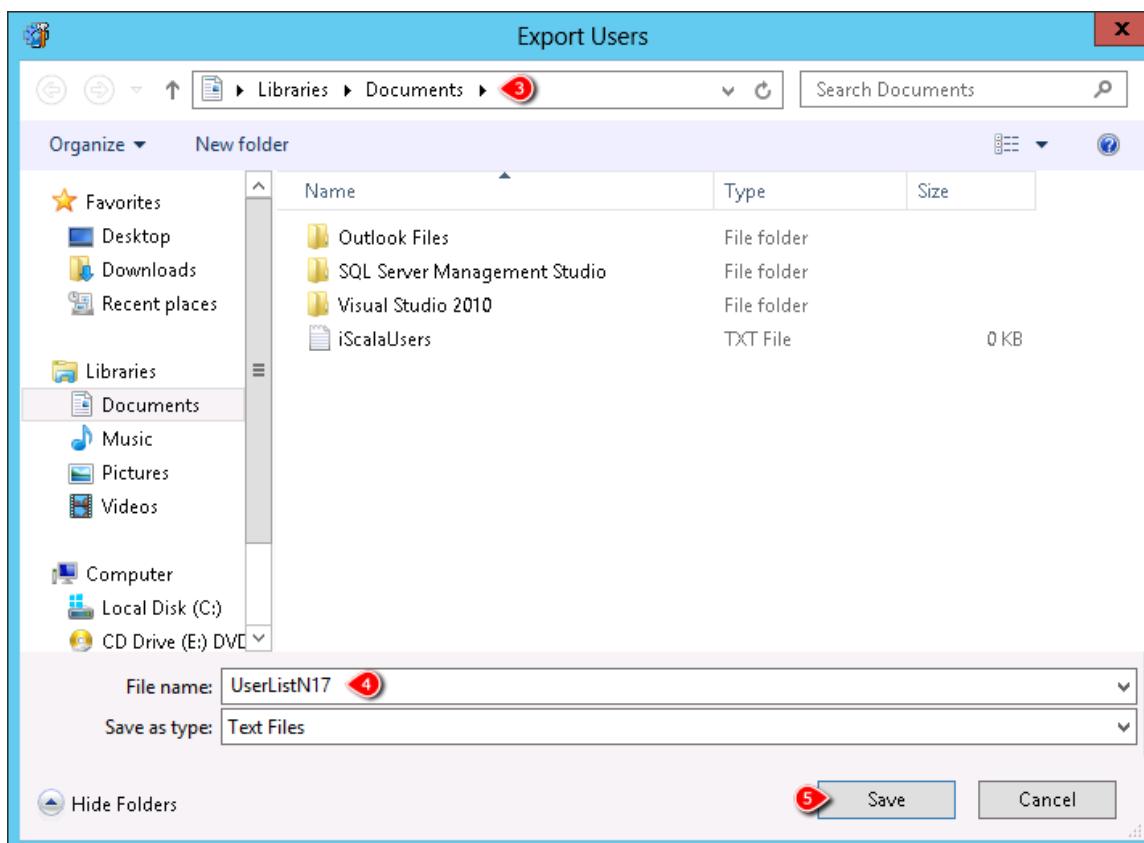
Export Users

Use the following steps to export a users list from Service Connect.

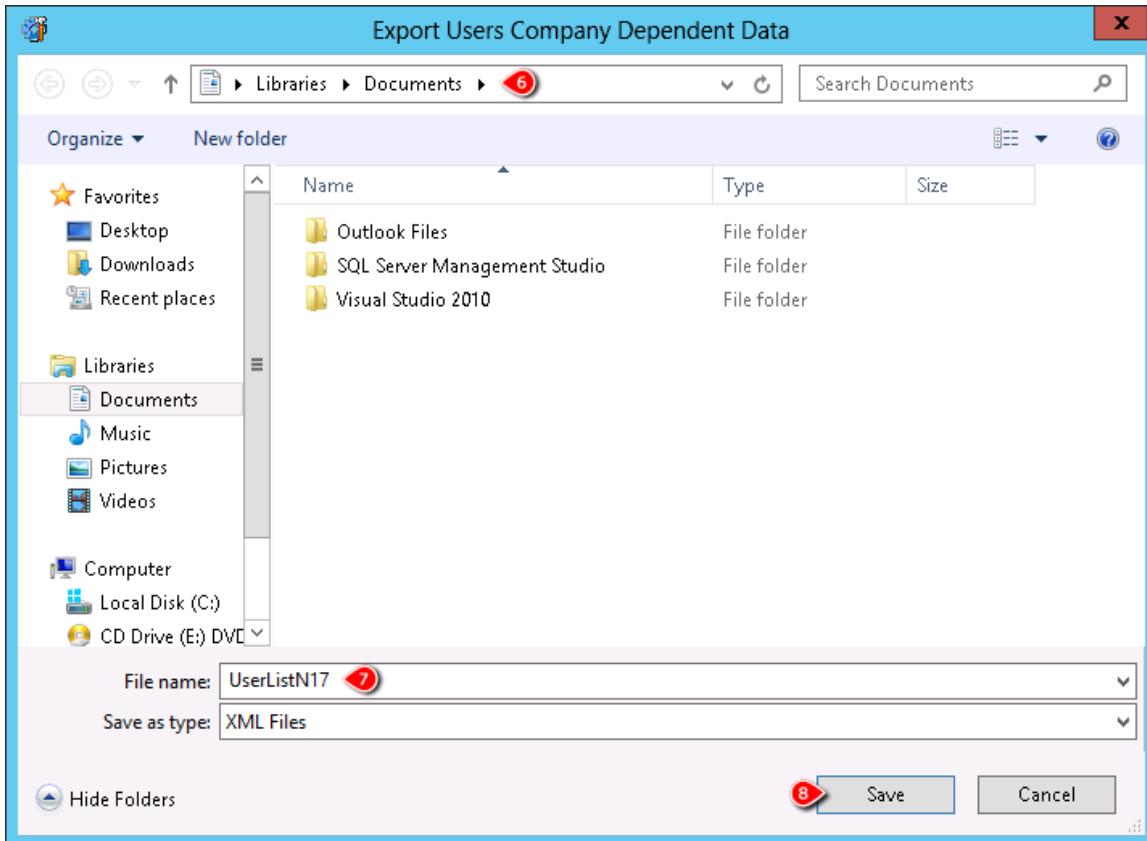
1. In the Administration Console, expand the **Security** node.



2. Right-click **Users** and select **Export Users**.
3. In the **Export Users** window, select the appropriate location for the user text file.



4. Enter the **File name**.
5. Click **Save**.
6. In the **Export Users Company Dependent Data** window, select the appropriate location to export the users company dependent data file.



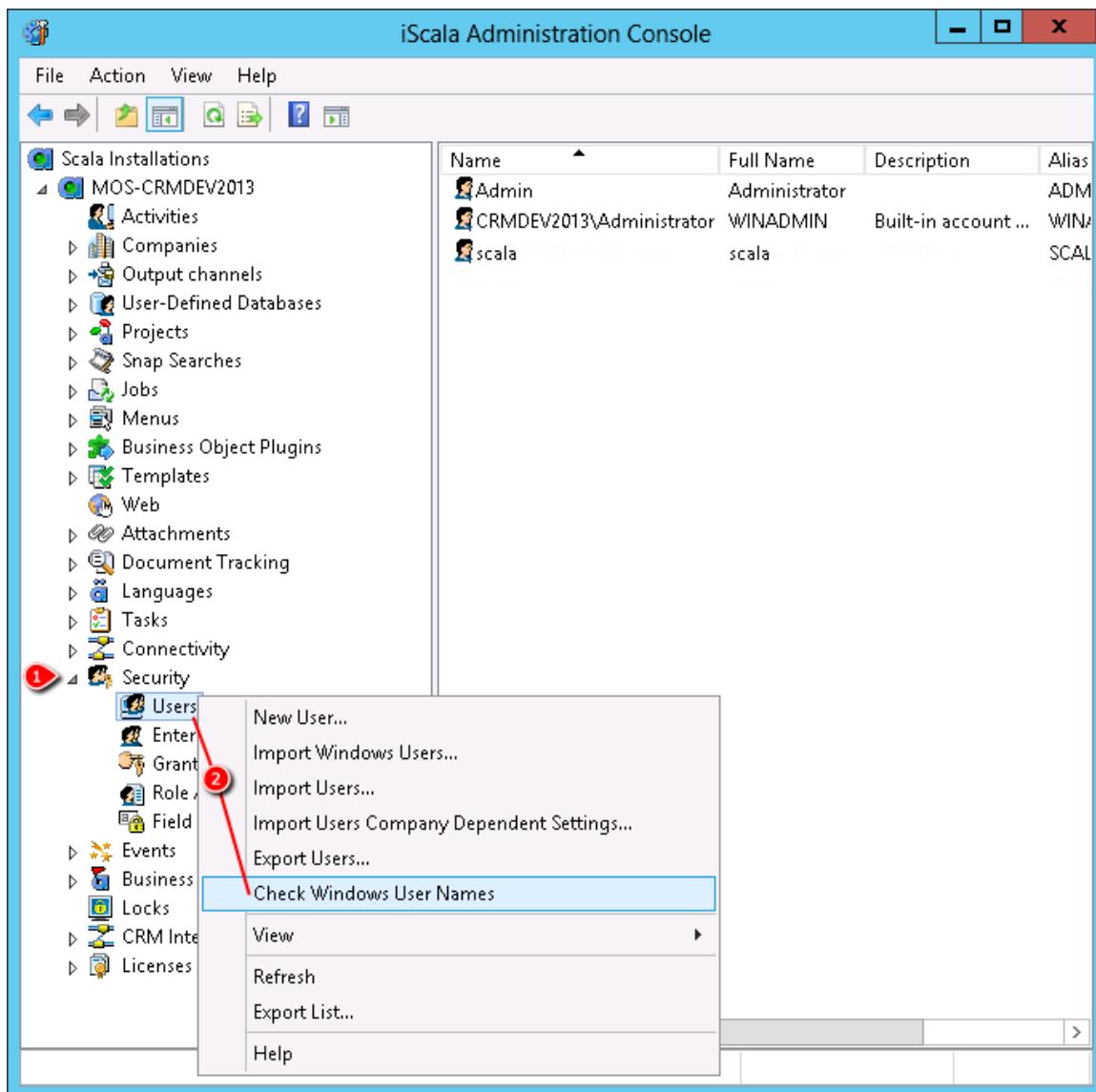
7. Enter the **File name**.

8. Click **Save**.

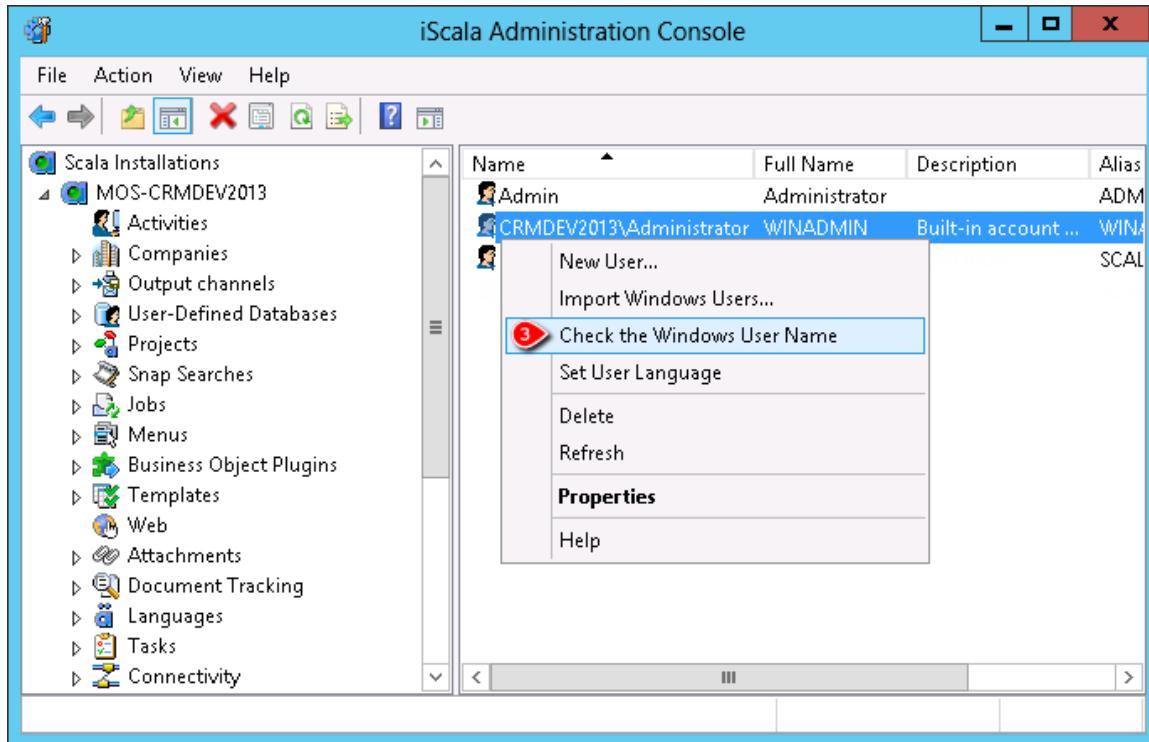
Check Windows User Names

Use the following steps to check windows user names.

1. To check all user names, expand the Security node.



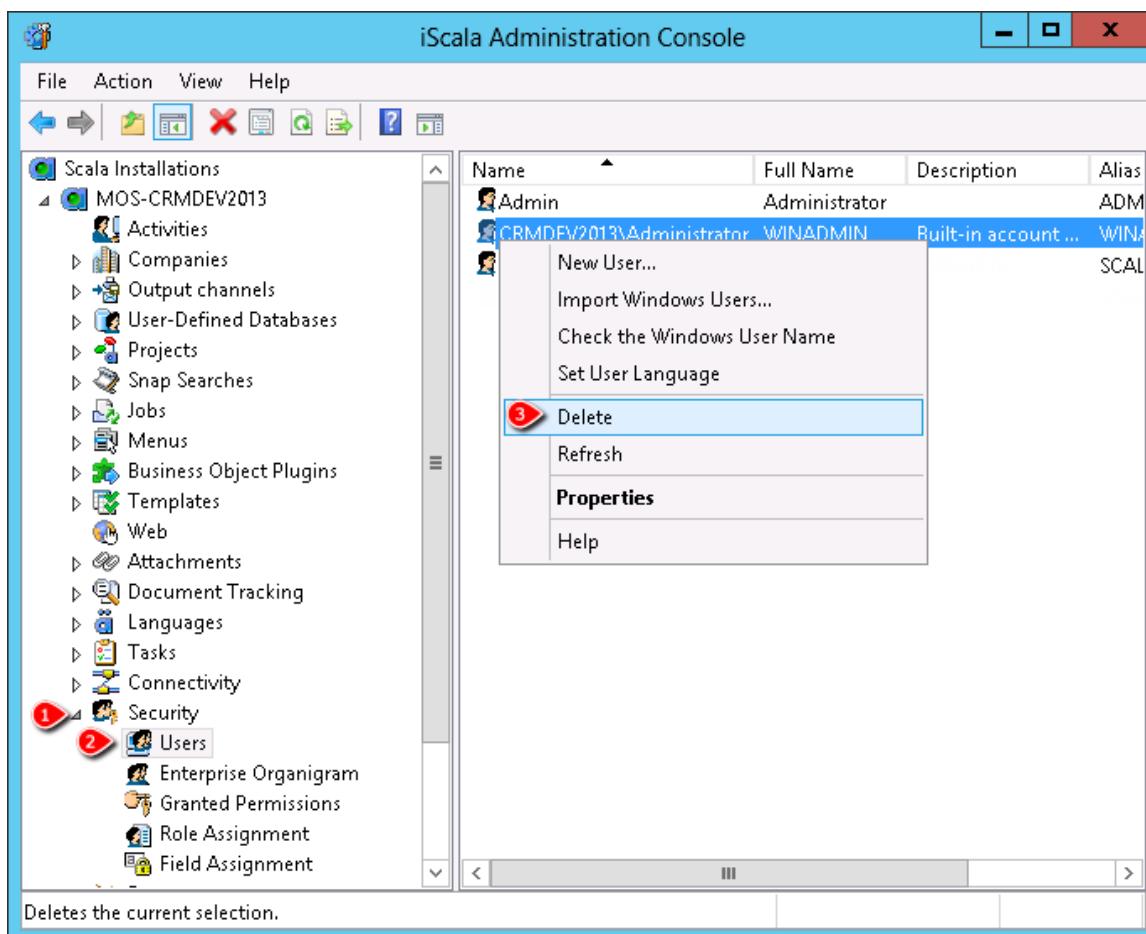
2. Right-click the **Users** node and select **Check Windows User Names**.
3. To check a particular user name, right-click that user in the list of users in the right pane and select **Check the Windows User Name**.



Delete a User

Use the following steps to delete a user.

1. In the Administration Console, expand the **Security** node.



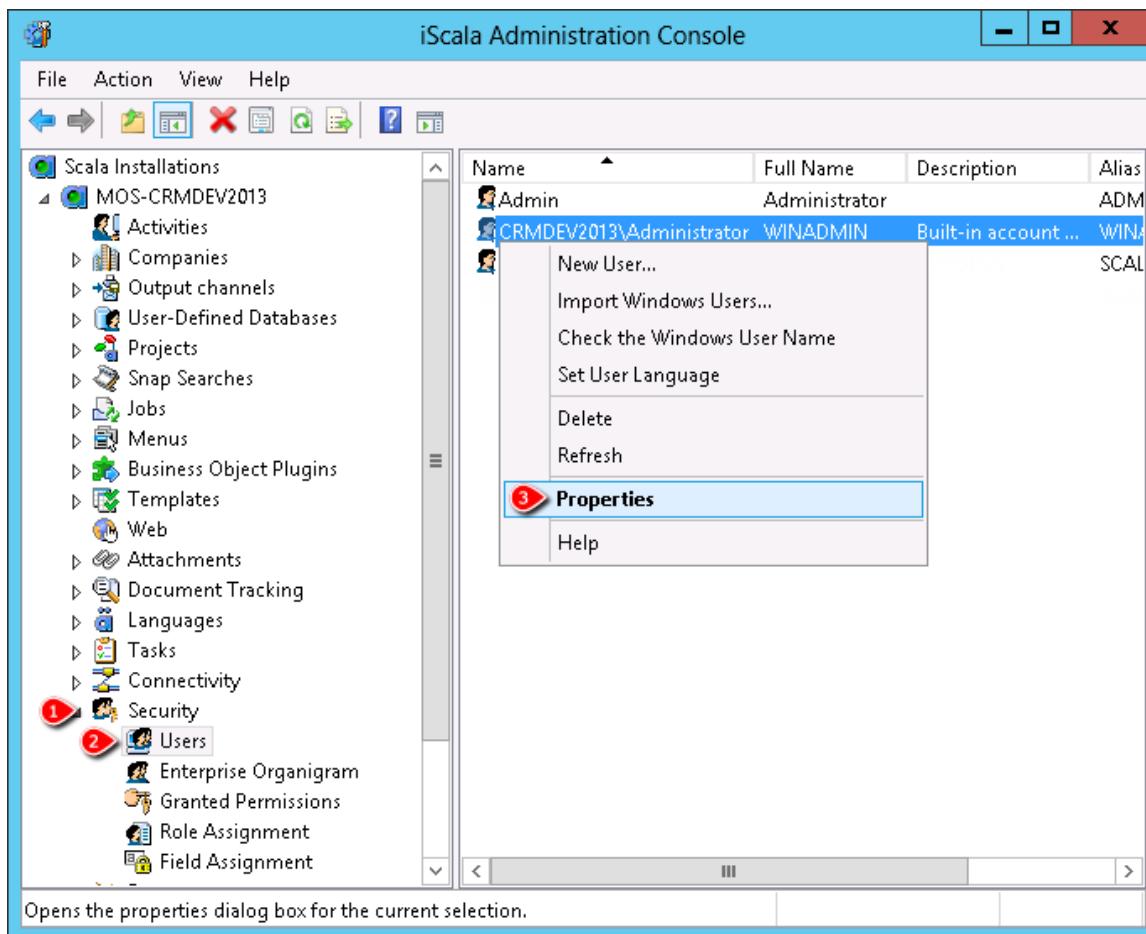
2. Select **Users**.

3. In the right pane, right-click the user you want to delete and select **Delete** from the menu.

Manage User Properties

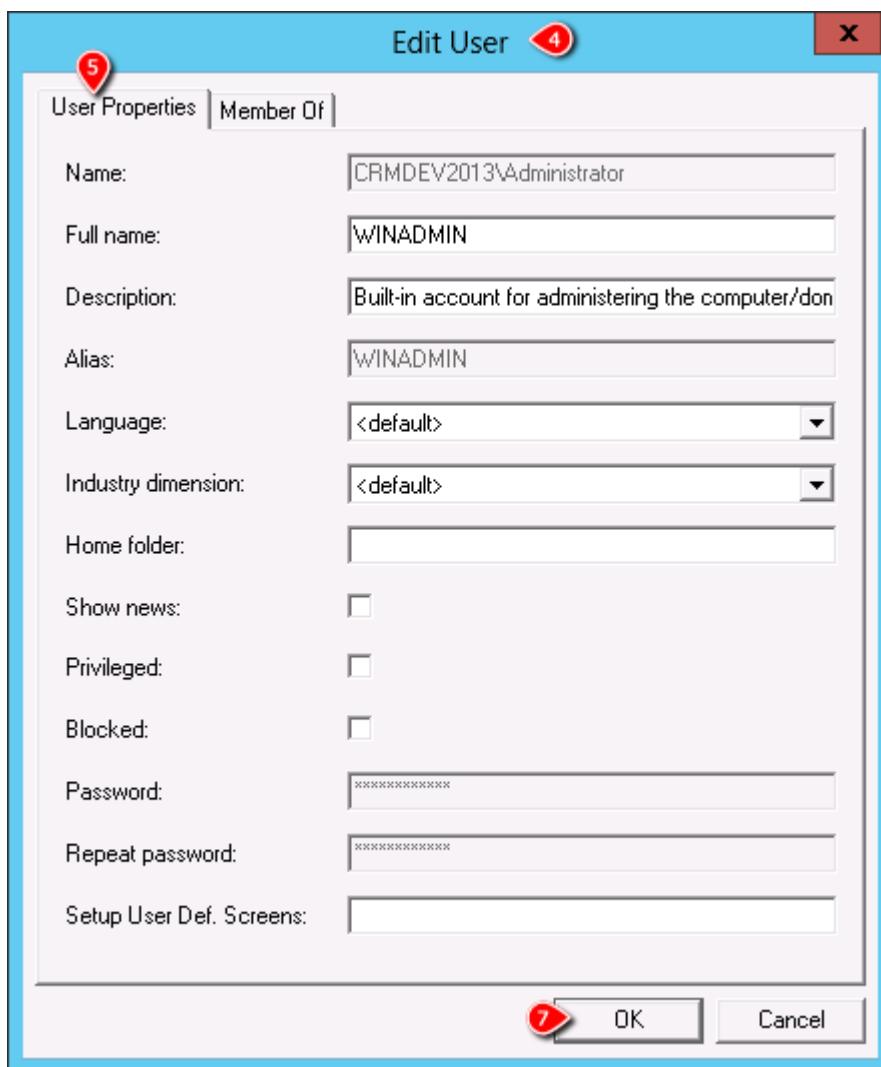
To view or edit user properties:

1. In the Administration Console, expand the **Security** node.



2. Select **Users**.

3. In the user list in the right pane, right-click the user whose properties you want to view and select **Properties**.



4. The **Edit User** window displays.

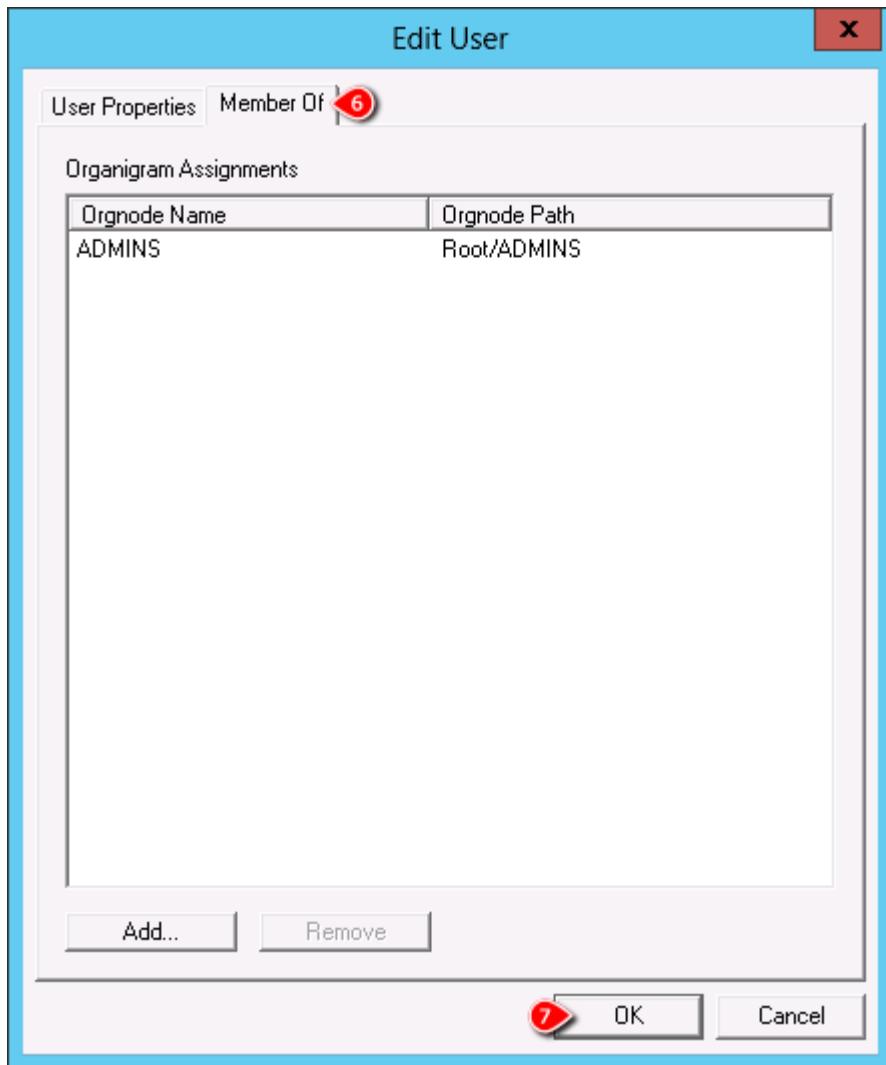
These user properties are general for a user within all companies. You can also set company-specific user properties. Review the Company-Specific User Properties section for more details.

5. 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 output 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.

6. On the **Member Of** tab, view the list of user groups to which the user belongs.



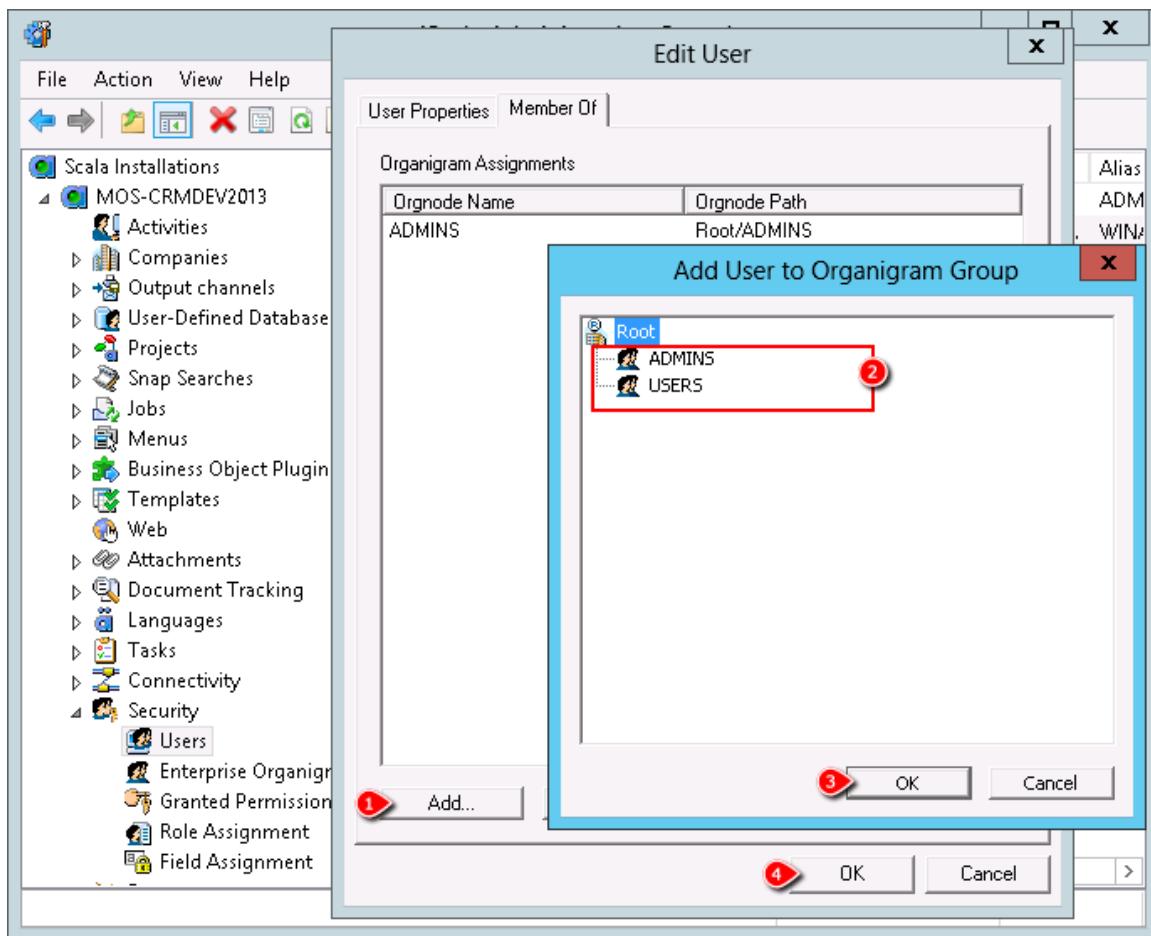
For the instructions on how to assign the user to or remove from a group refer to the next procedure.

7. If you change any user properties, click **OK** to save the changes.

Assign a User to a User Group

In the user properties dialog, you can add a user to an Enterprise Organigram group and remove the user from a group.

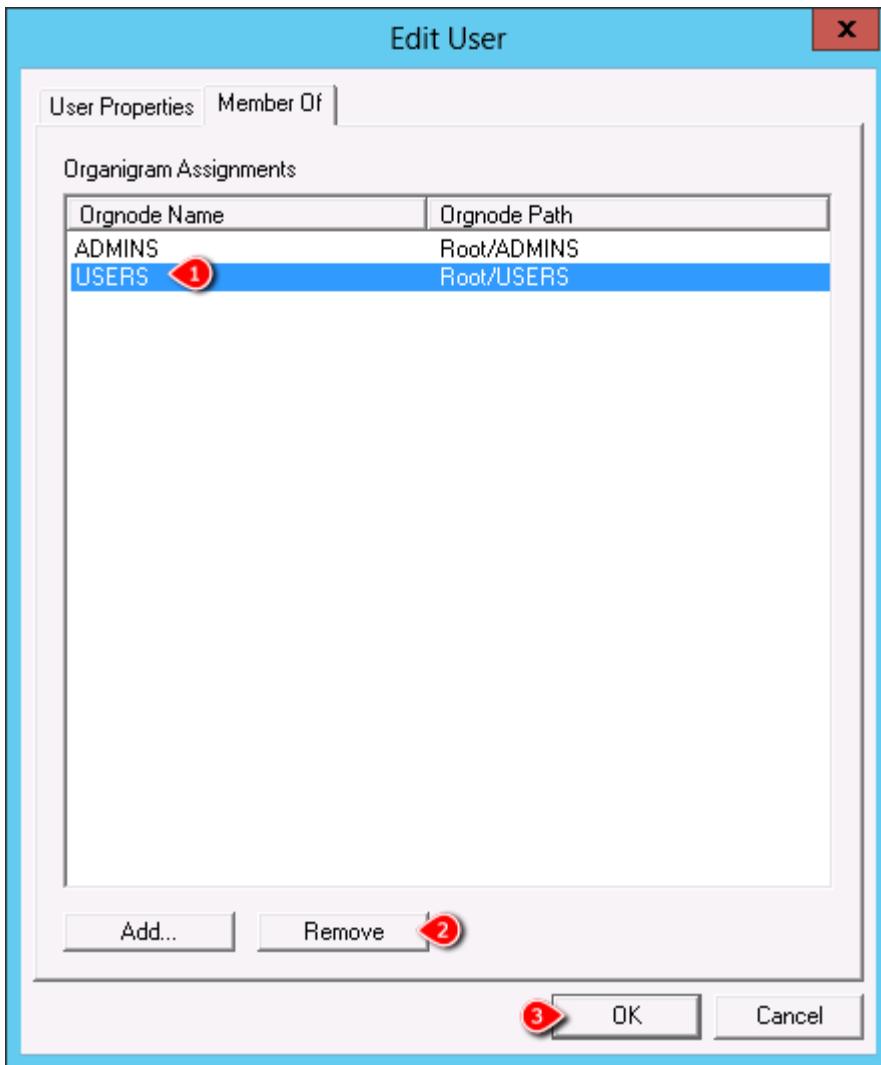
1. In the **Edit User** window, on the **Member Of** tab, click the **Add** button.



2. In the **Add User to Organigram Group** window, select the group to which you want to add the user.
3. Click **OK**.
4. In the **Edit User** window, click **OK**.

Remove a User from a User Group

1. In the **Edit User** window, on the **Member Of** tab, select the group from which you want to remove the user.



2. Click the **Remove** button.
3. In the **Edit User** window, click **OK**.

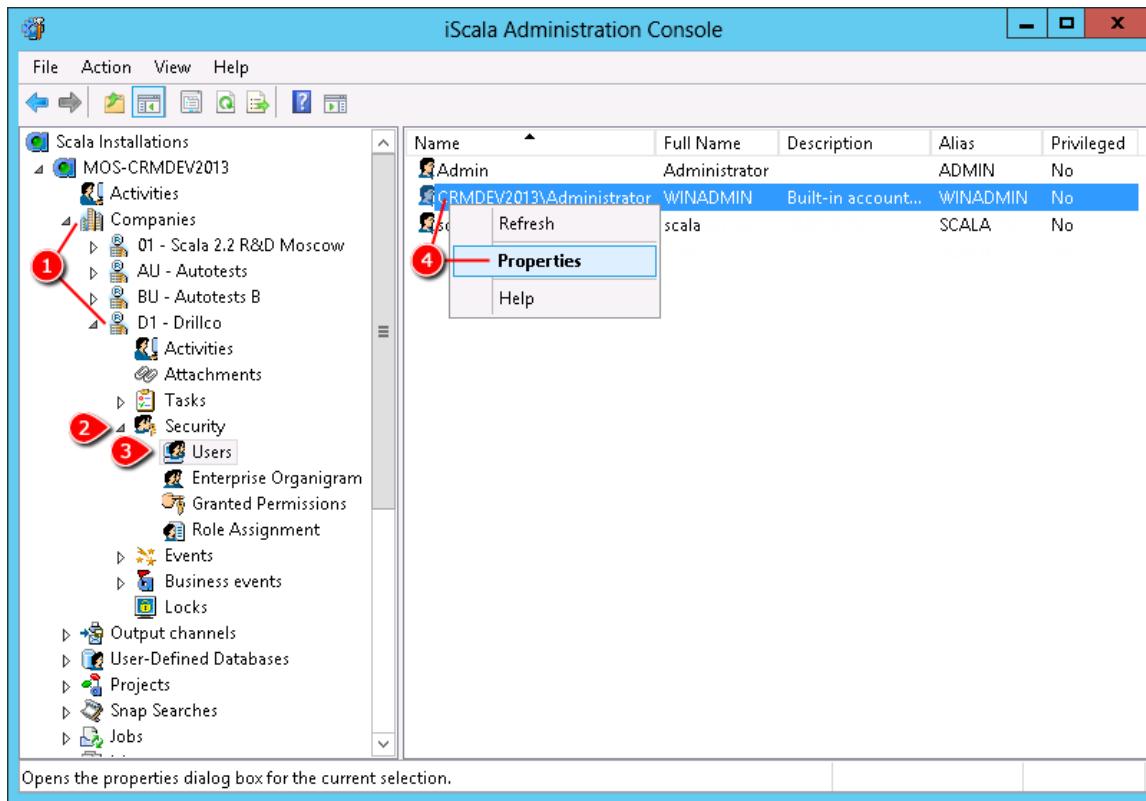
You can adjust several other iScala-specific user properties in iScala WinDS, using **System Utilities > User Management > User Properties**.

Edit Company-Specific User Properties

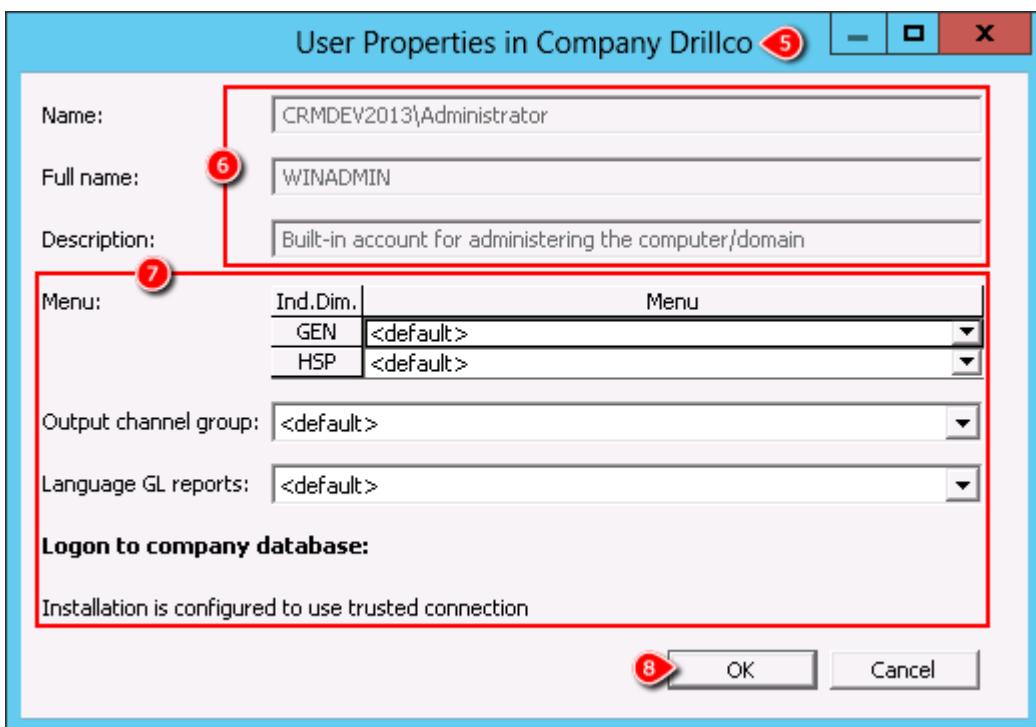
Company-specific user properties override the general properties set in Security > Users.

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.
3. Select **Users**.
4. 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**.
5. The **User Properties in Company <Company Name>** window displays.



6. You cannot change the user's **Name**, **Full name**, or **Description**. These properties are read-only.
7. 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.⁵
 - **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.

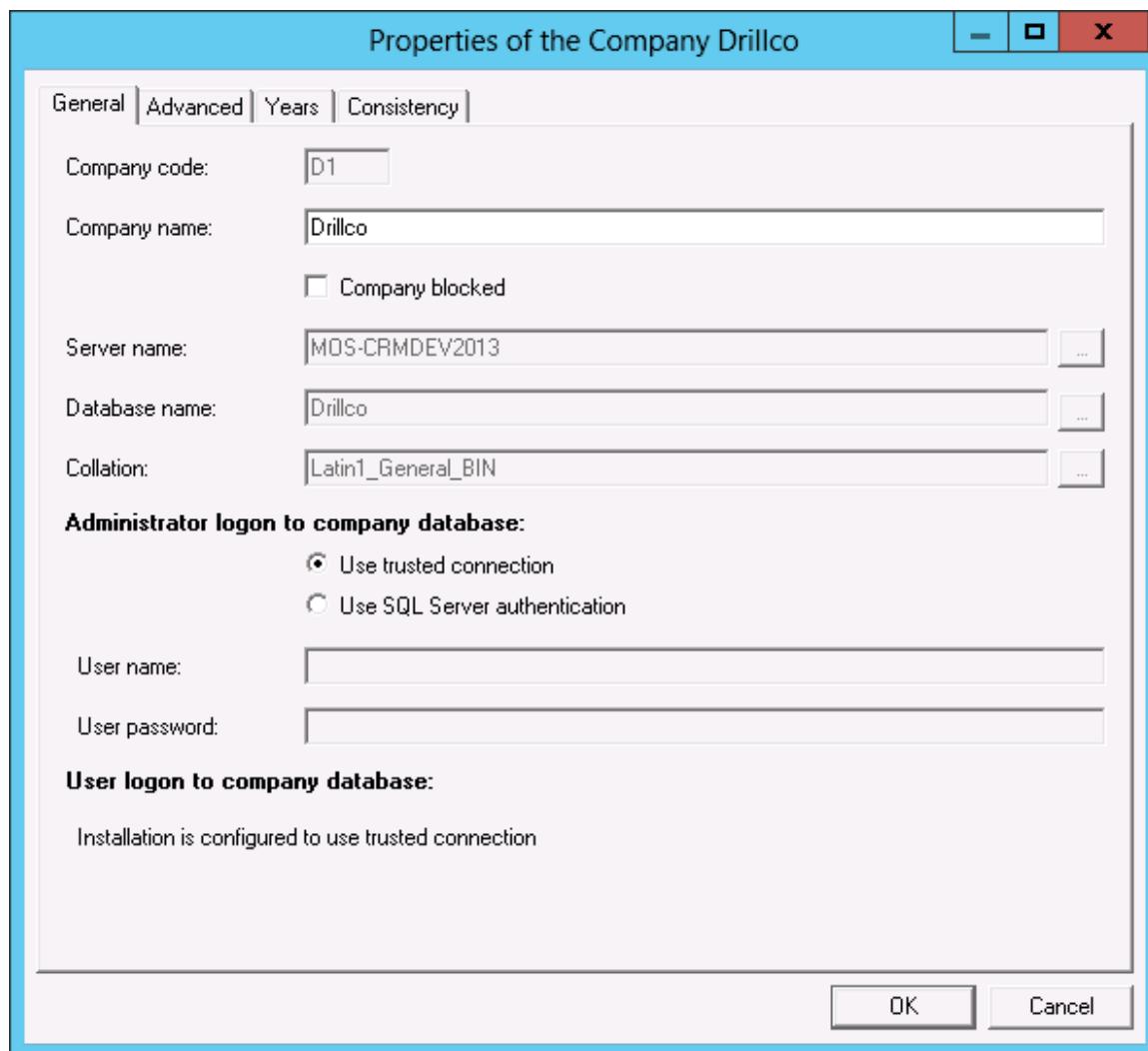
When SQL Server Authentication is used, you can enter a SQL username and password iScala will use when it makes SQL calls for the current user. This SQL username and password should not be known to the end-user; only to the system administrator. If the end-user knows the credentials iScala uses to make 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.

8. Click **OK** to save the changes, if any.

Company Parameters

Company parameters are accessed from the Company node.



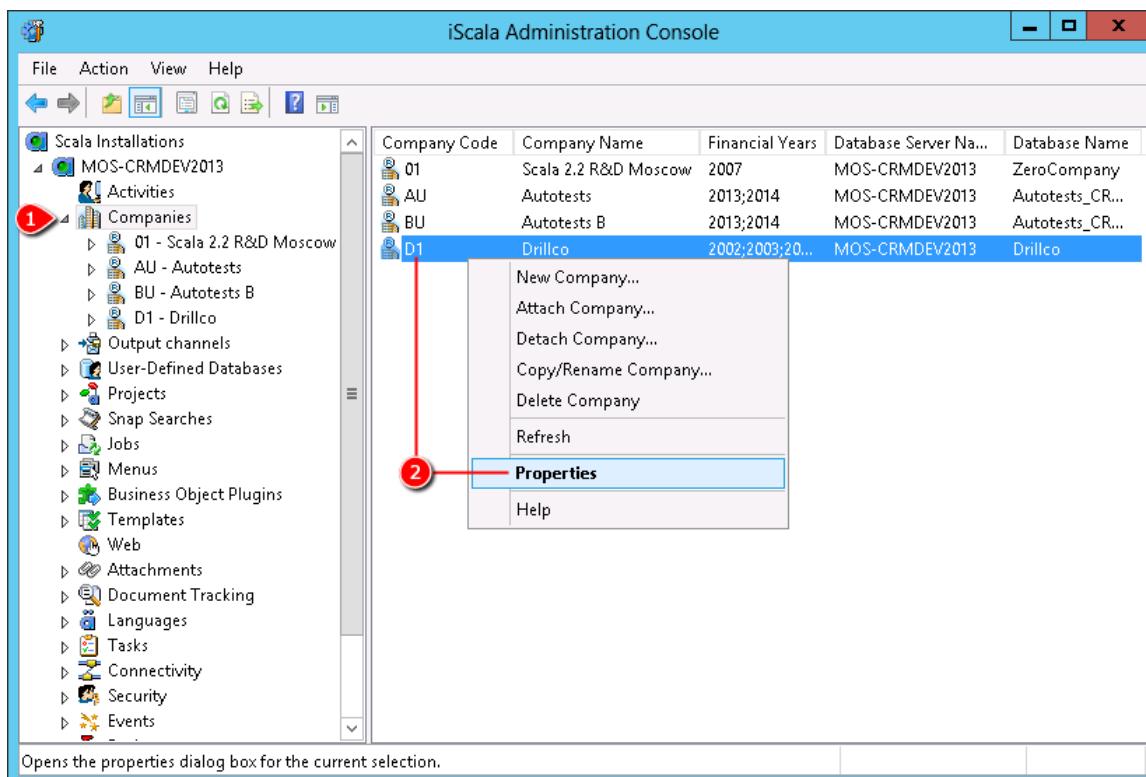
You can perform the following actions in the Company properties dialog:

- View general company properties.
- Block a company from logging in.

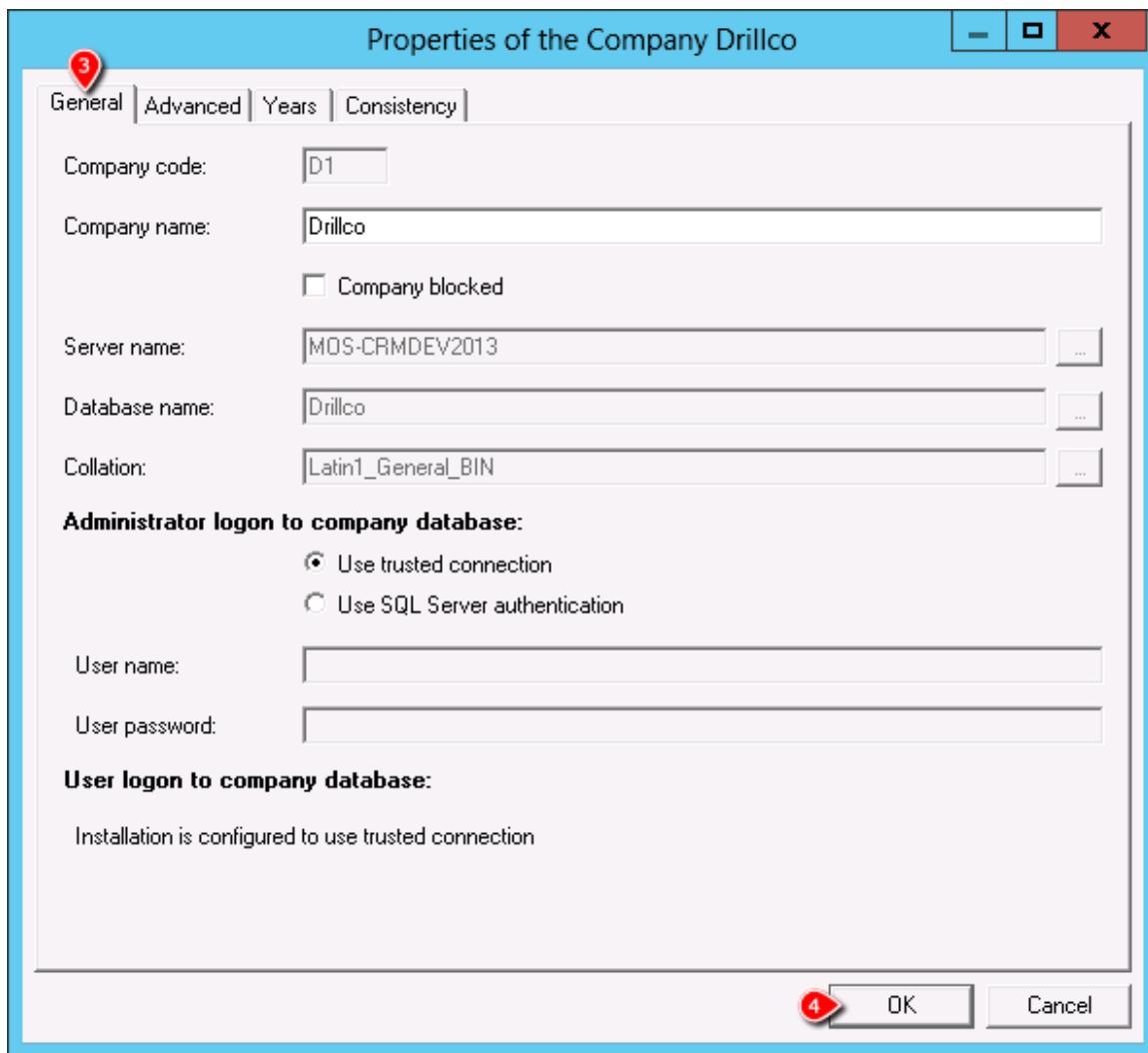
- Set the administrator logon and user logon to the company database.
- Set the company default menu.
- Specify company login parameters.
- Add and delete company financial years.
- Set a financial period for an existing year.
- View lists of installed modules and system objects for the selected financial year.

View General Company Properties

1. Expand the **Companies** node.



2. Right-click the company from the list and select **Properties**.
3. Open 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.

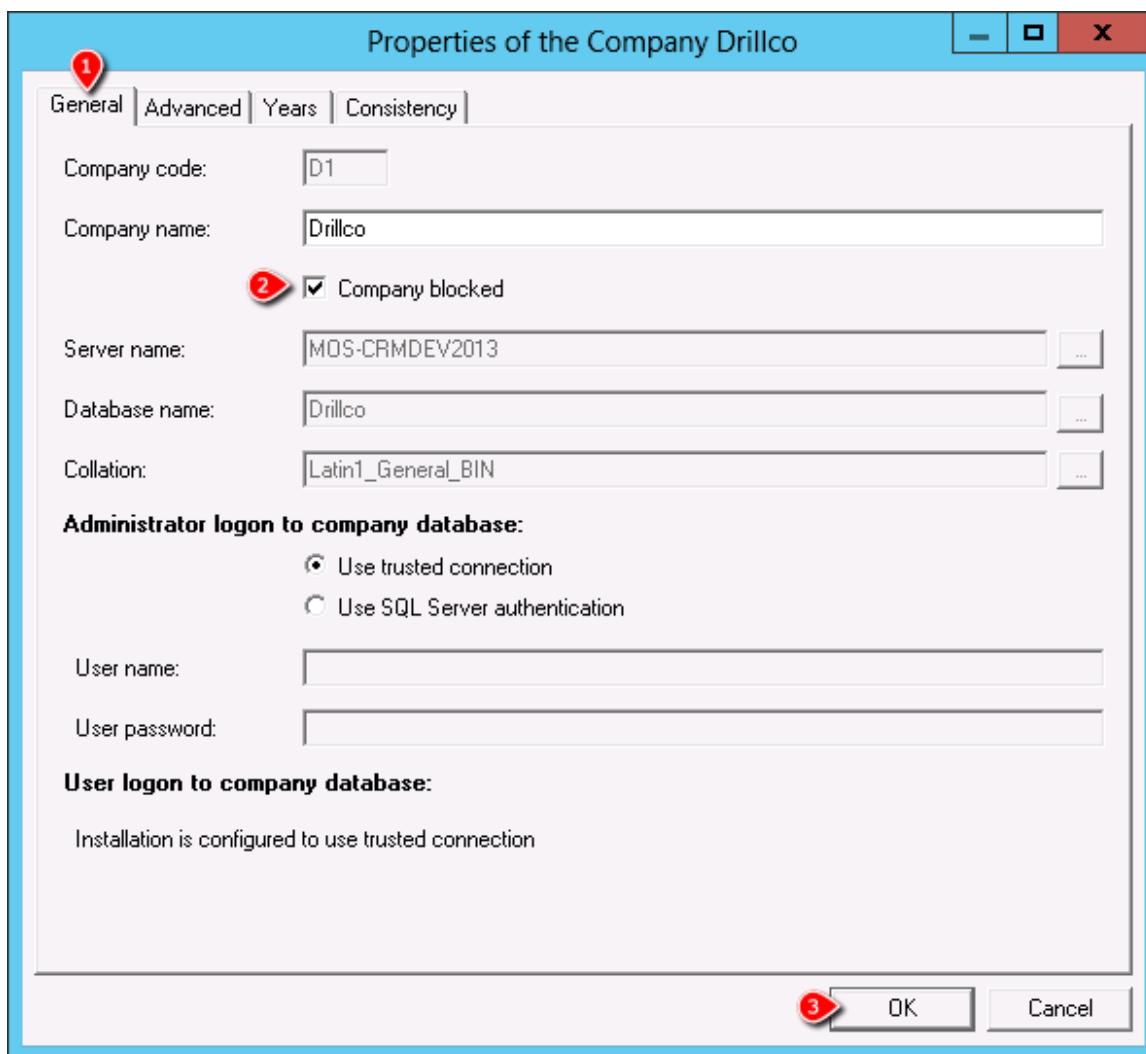


4. Click **OK**.

Block a Company

To block a company from logging in:

1. Open the **General** tab of the **Properties of the Company** dialog.

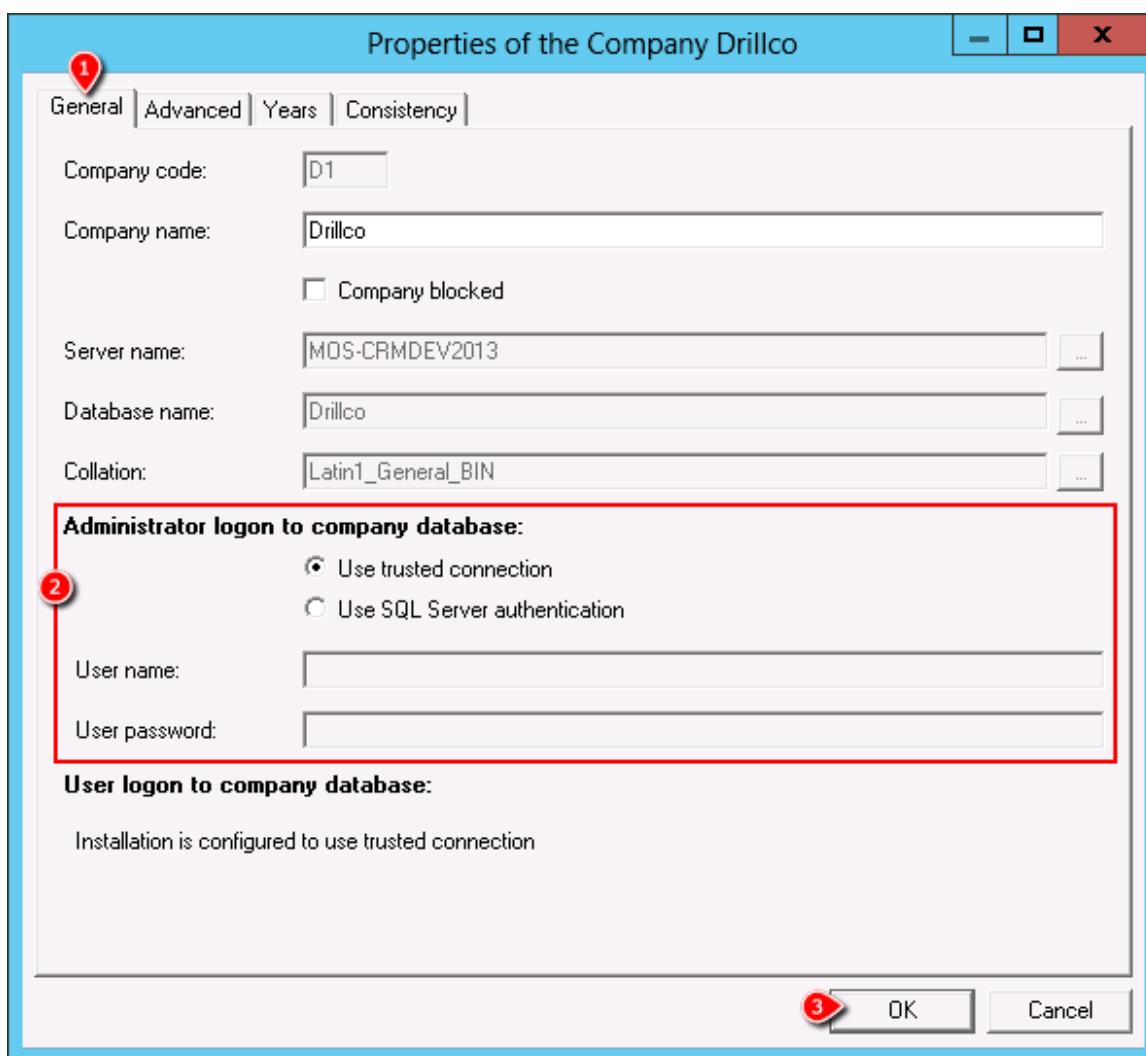


2. Select the **Company blocked** check box.

3. Click **OK**.

Set an Administrator Logon to a Company Database

1. Open the **General** tab of the Properties of the Company dialog box.



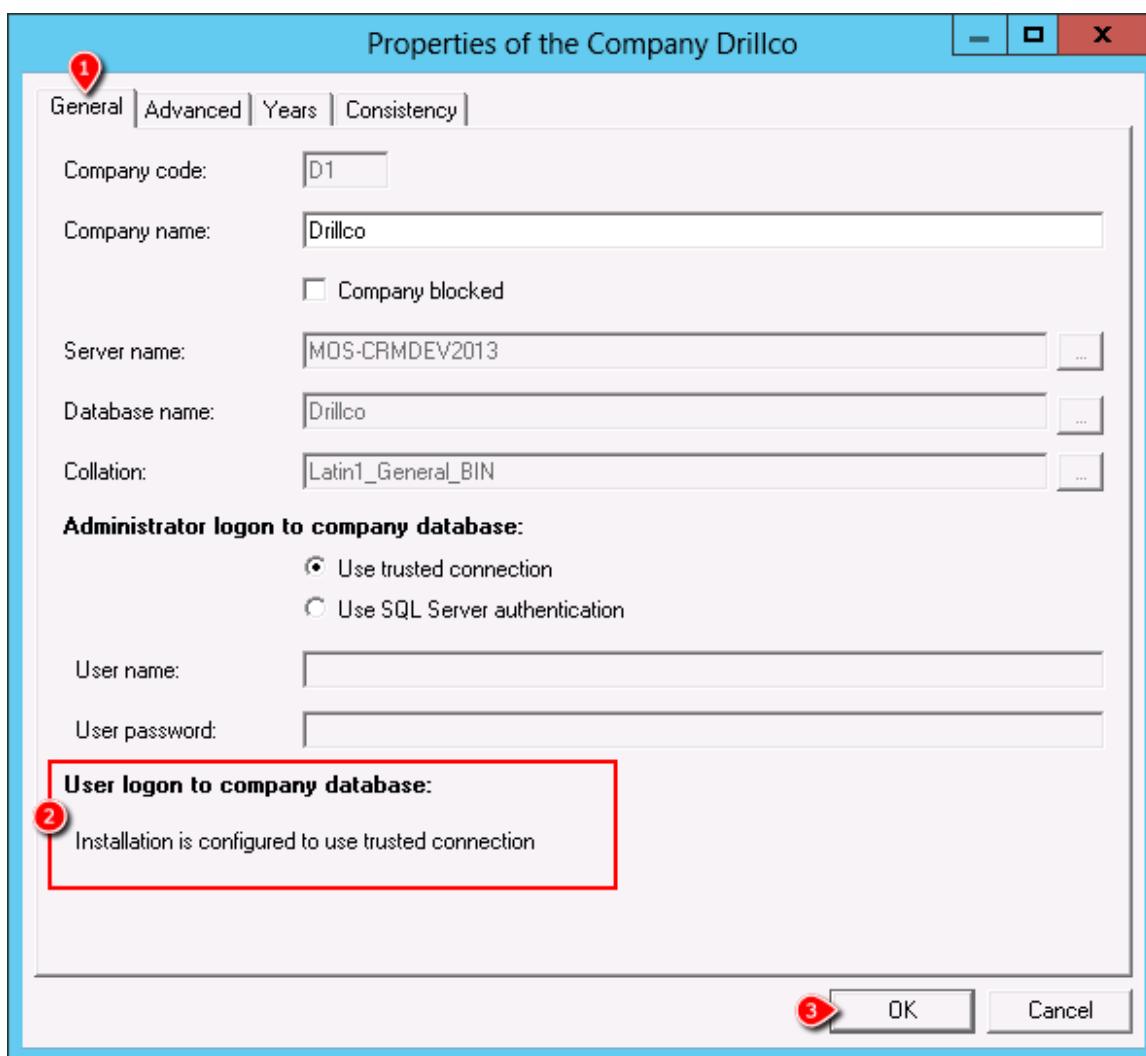
2. In the **Administrator logon to company database** section, specify the account the iScala Administration Console uses to access the company database. This account can be a trusted connection or SQL Server Authentication.
3. Click **OK**.

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. Open the **General** tab of the Properties of the Company dialog box.

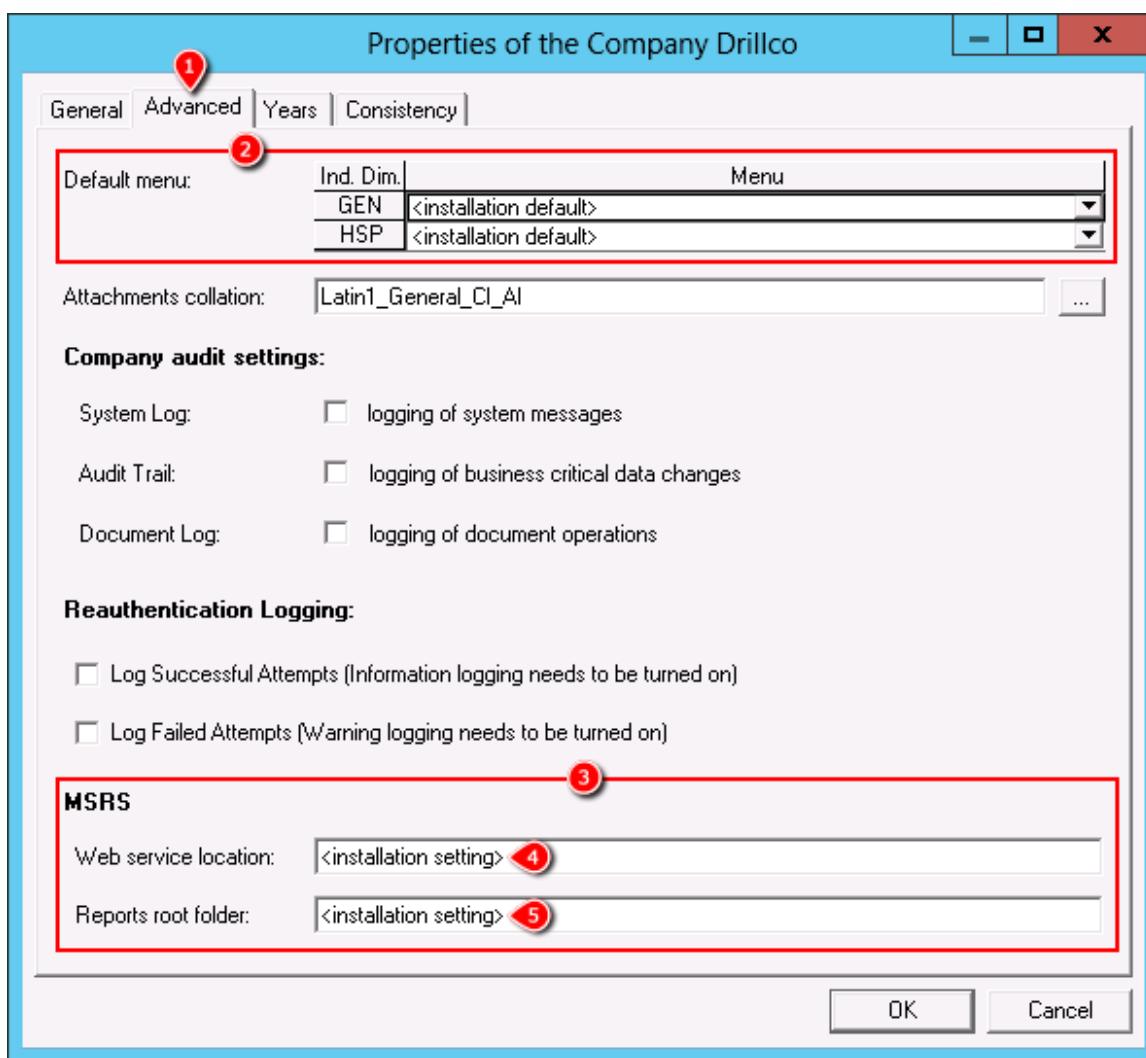


2. In the **User logon to company database** section, specify whether users connect to the iScala Company database via a trusted connection or using SQL Server Authentication.
3. Click **OK**.

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. Open the **Advanced** tab on the Properties of the Company dialog.

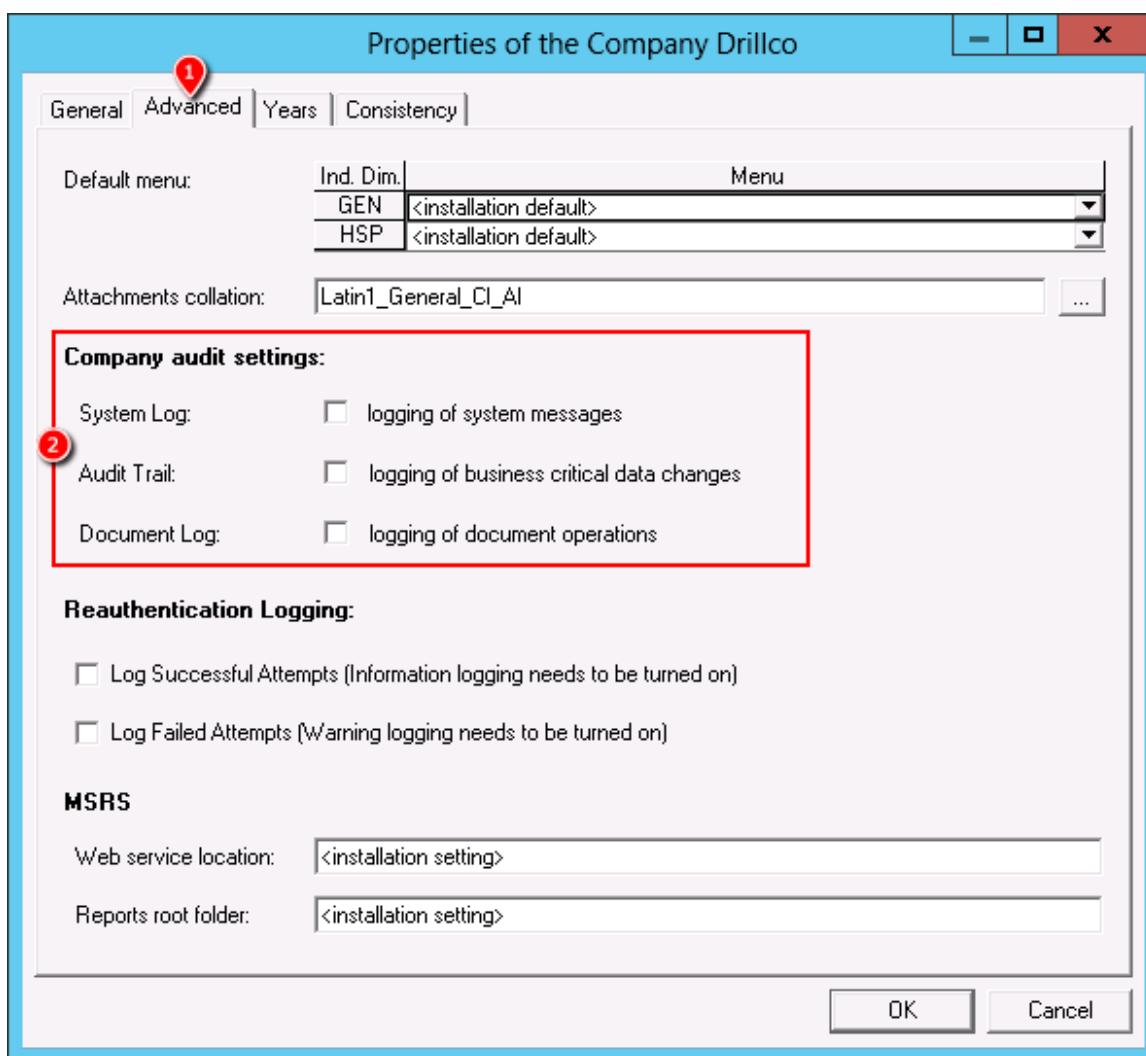


2. Under the **Default menu** section, view or edit the default menus for the company for various industry dimensions.
3. You can override iScala **MSRS** installation-level properties on the Company basis by defining Microsoft Reporting Services Service settings in the MSRS group.
If you have several 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. Open the **Advanced** tab on the Properties of the Company dialog box.

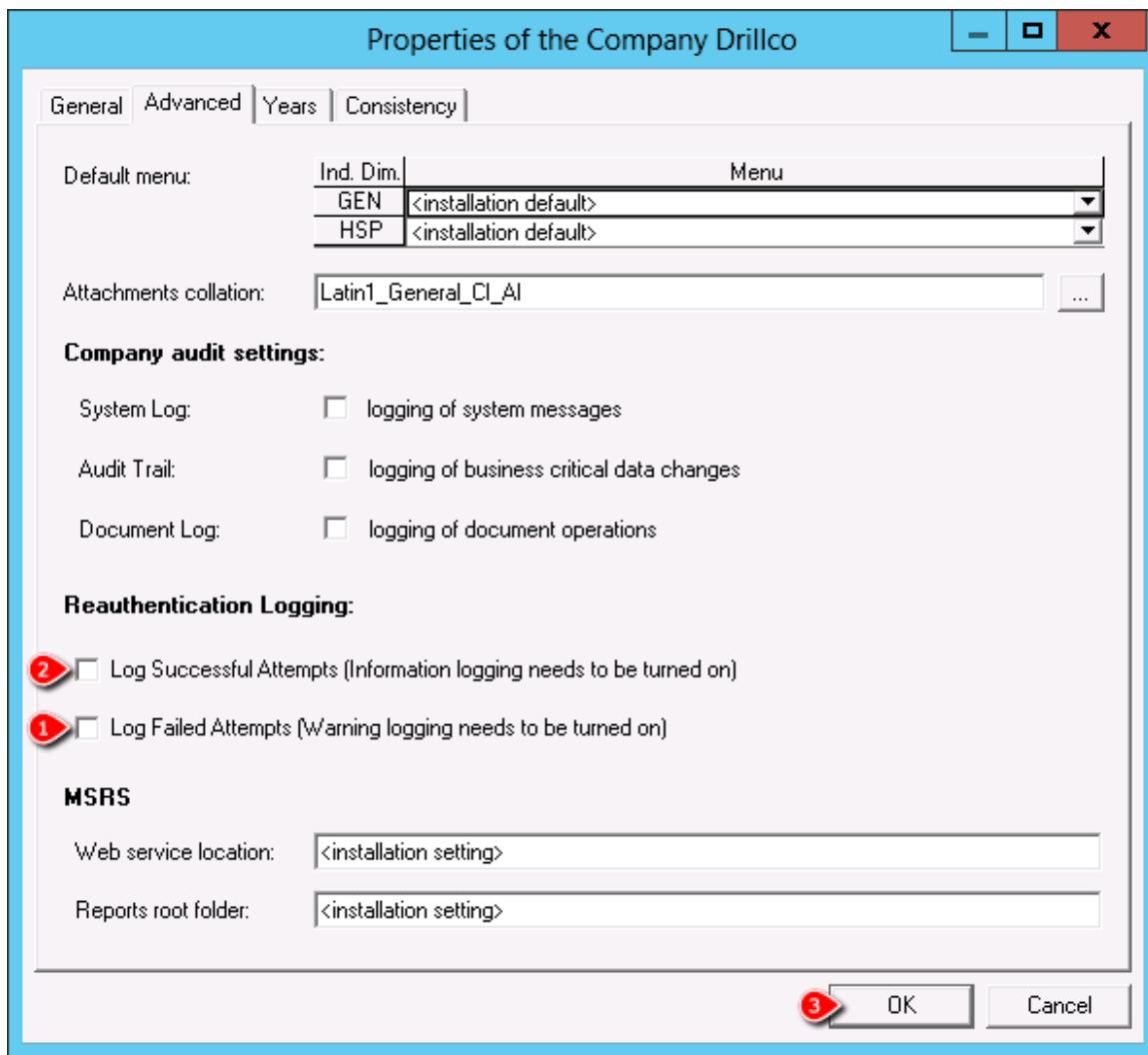


- In the **Company audit settings** section, view or change the log settings by selecting or clearing the check boxes. Check boxes include:
 - System Log** - Enables the logging of system messages.
 - Audit Trail** - Enables the logging of business-critical data changes.
 - Document Log** - Enables the logging of document operations.

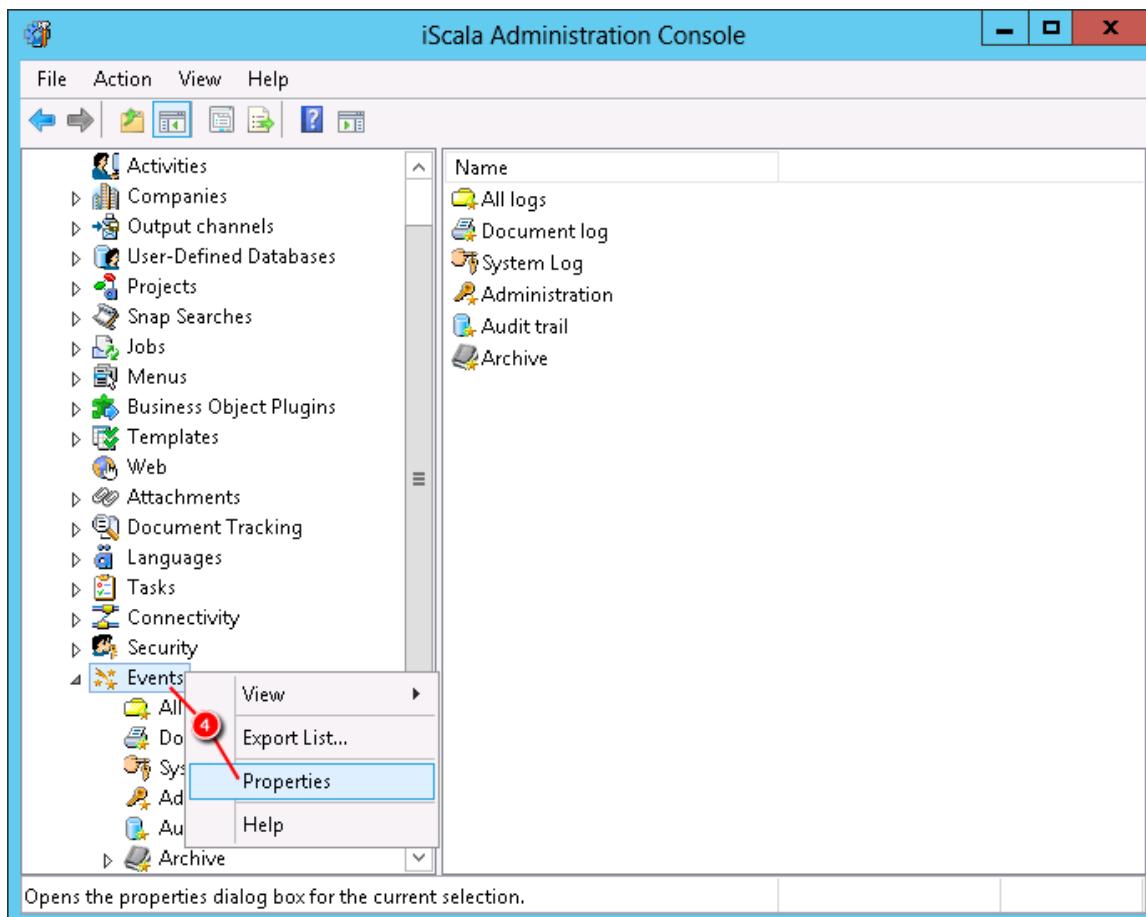
Set Logging of Action Reauthentication Attempts

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.

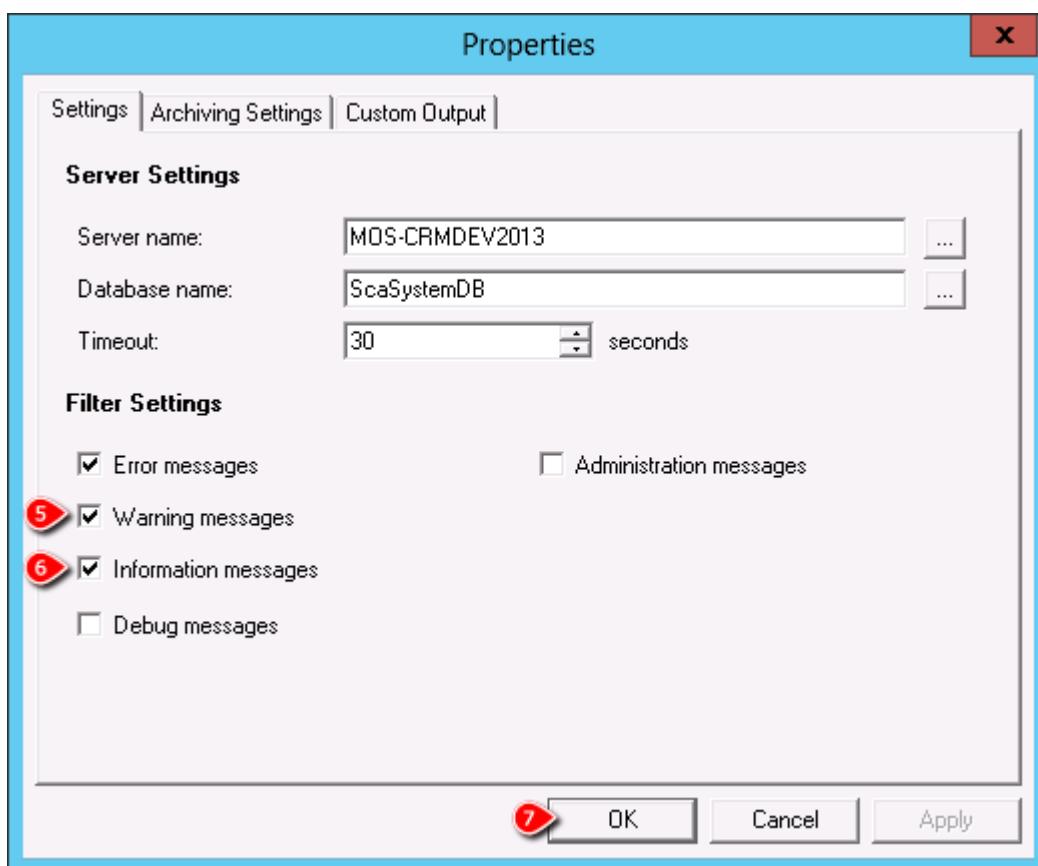
- To enable failed Reauthentication attempts logging, in the company properties dialog, on the **Advanced** tab, select the **Log Failed Attempts** check box.



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**.



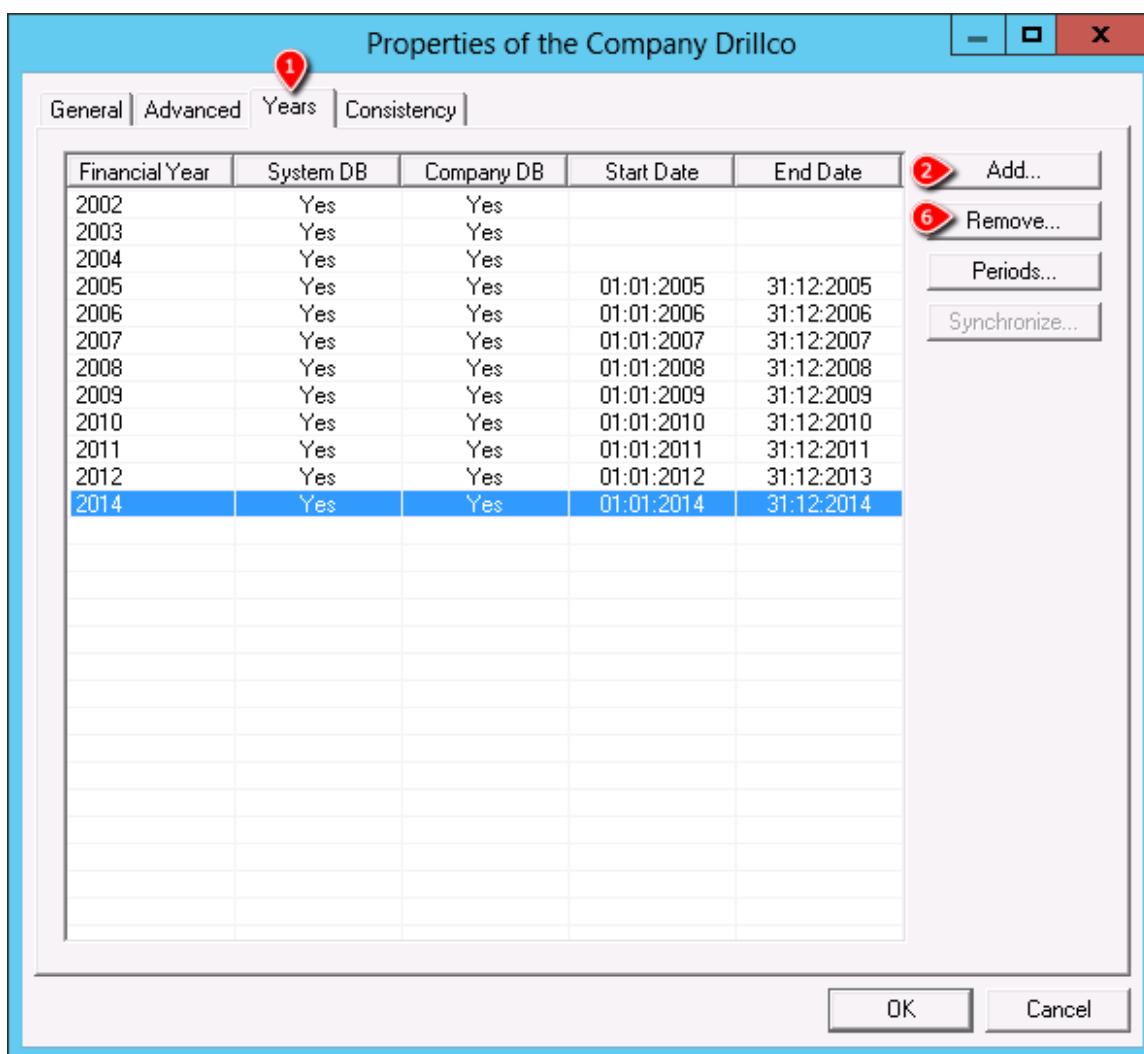
5. For failed Reauthentication attempts logging, in the **Properties** window, under **Filter Settings**, select the **Warning messages** check box.



6. For successful Reauthentication attempts logging, select the **Information messages** check box.
7. Click **OK**.

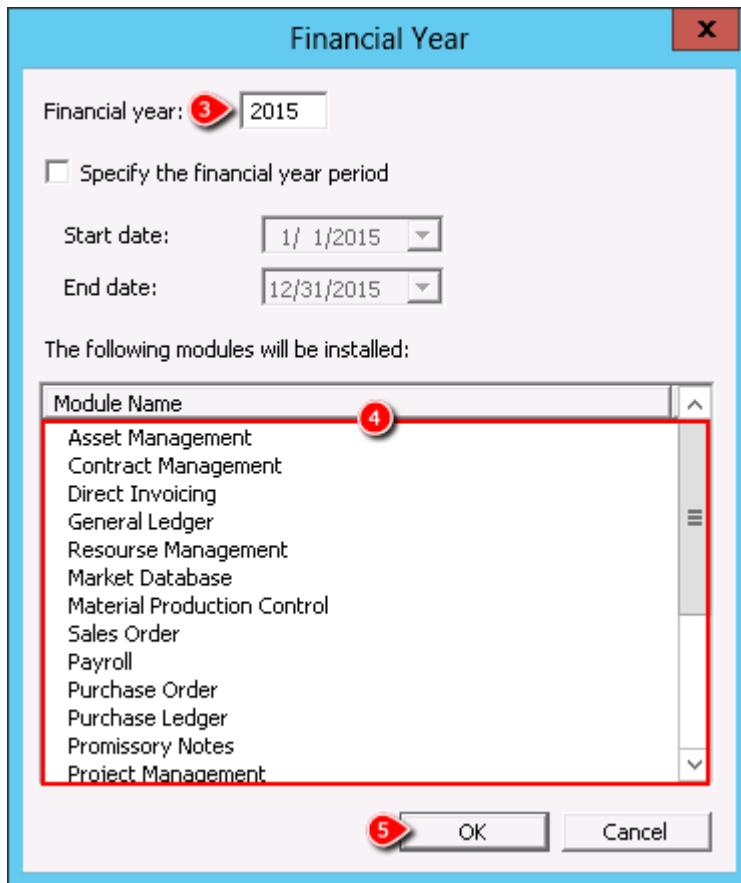
Manage the Financial Years for the Company

1. In the **Properties of the Company <company name>** window, open 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.



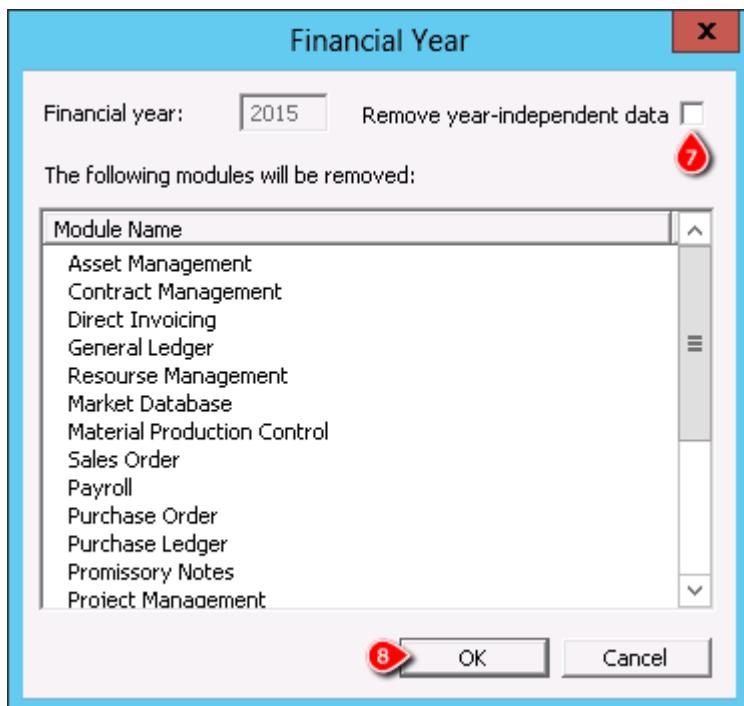
4. In the **Module Name** section, view the list of installed iScala modules.

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.

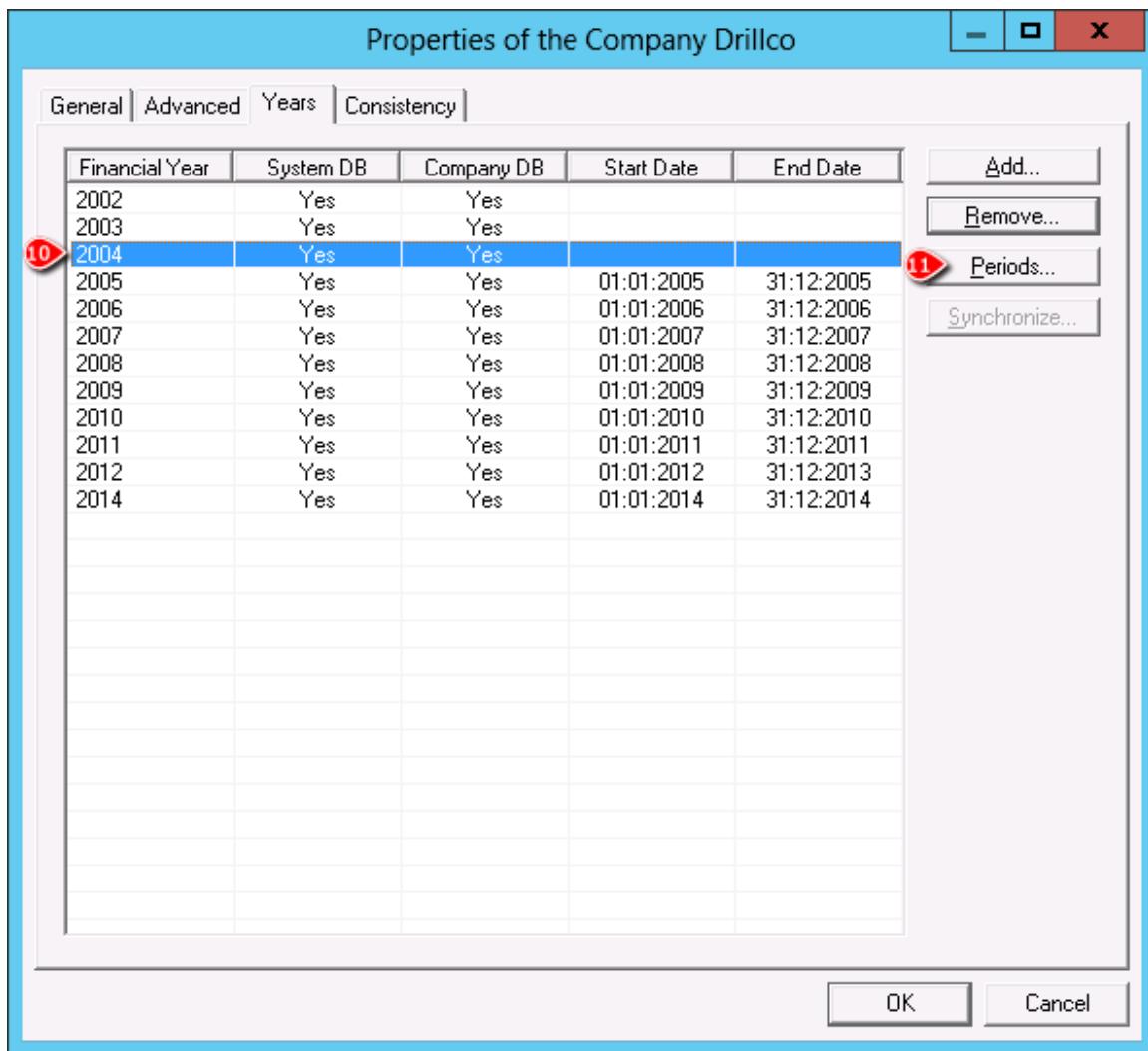
5. Click **OK**.

Wait till the process of creating a new financial year is complete and the new year displays in the list.

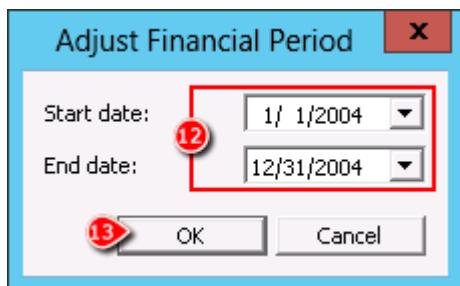
6. To remove a financial year from the **Years** list, select the year you want to remove and click the **Remove** button.
7. Specify whether you want to **remove year-independent data**.



8. Click **OK**.
9. To the confirmation messages, click **Yes**.
10. To set a financial period for an existing year, select the year in the grid.



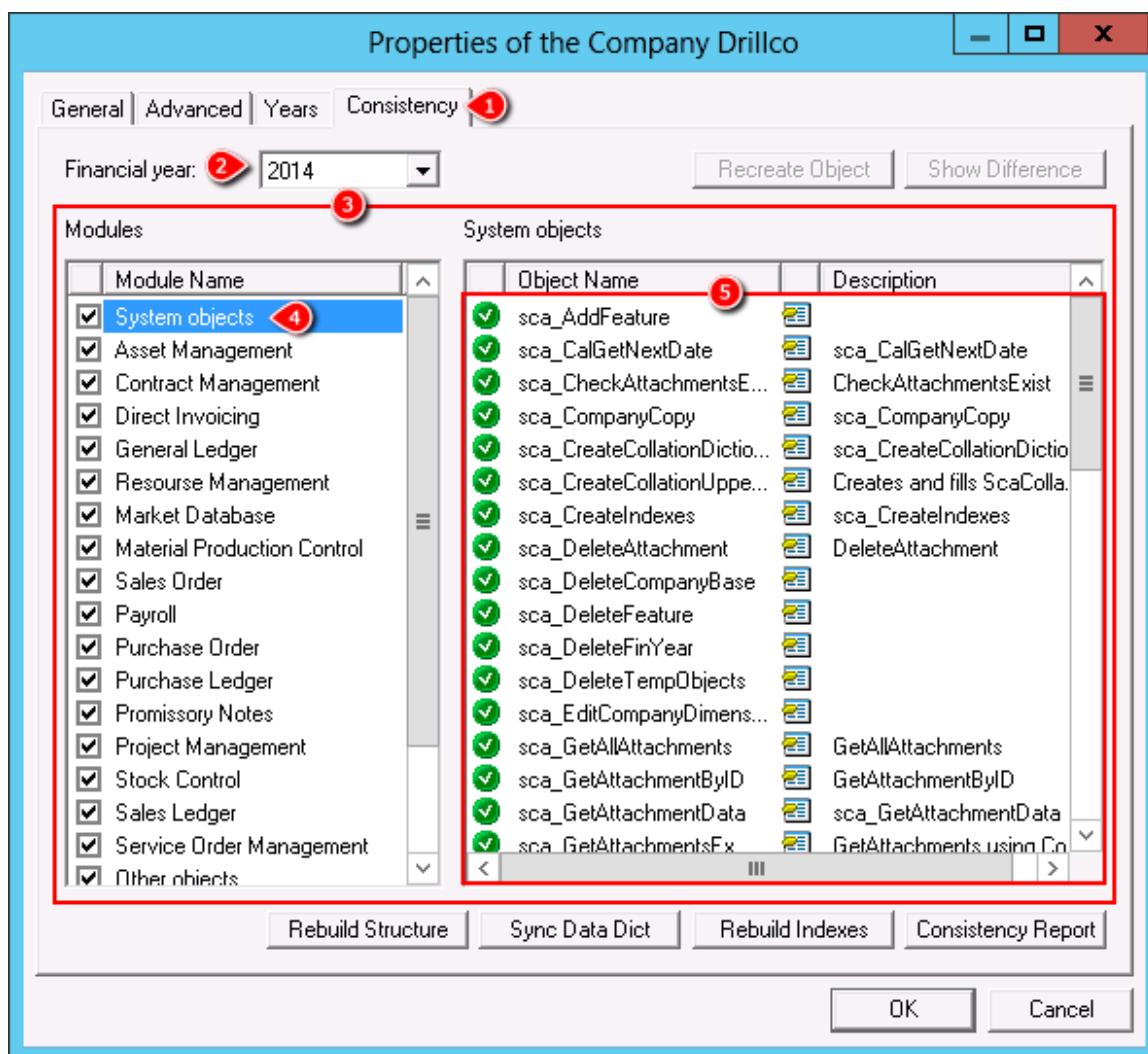
11. Click **Periods...**.
12. Specify the period **Start Date** and **End Date**.



13. Click **OK**.

View Lists of Installed Modules and System Objects

1. In the **Properties of the Company <company name>** window, open the **Consistency** tab.

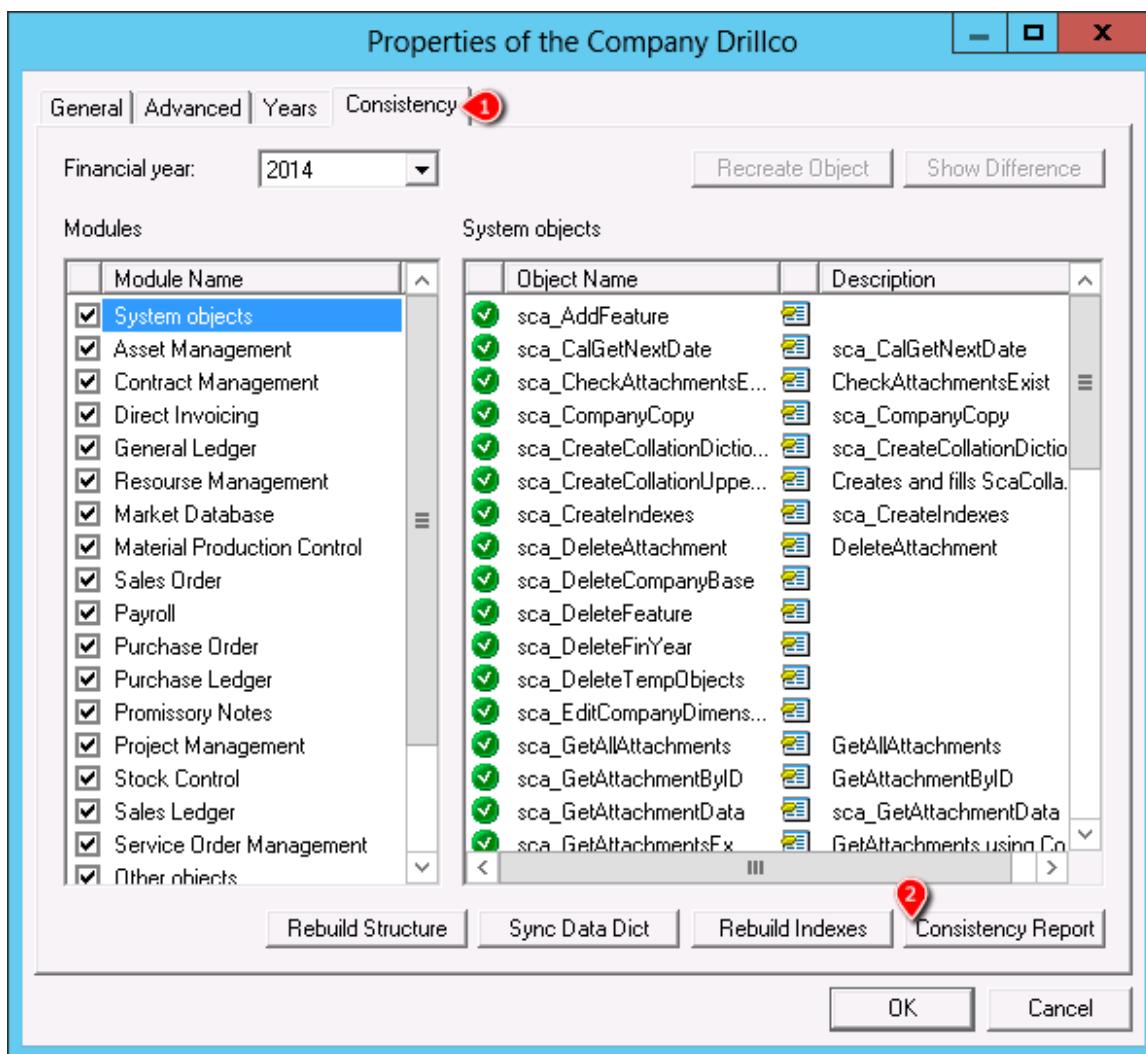


2. Select the **Financial year**.
3. View the list of installed iScala modules and system objects for the selected year.
4. Select a module from the **Modules** list.
5. The list of all **System objects** available for this module displays in the right pane. For each System object, the following information displays:

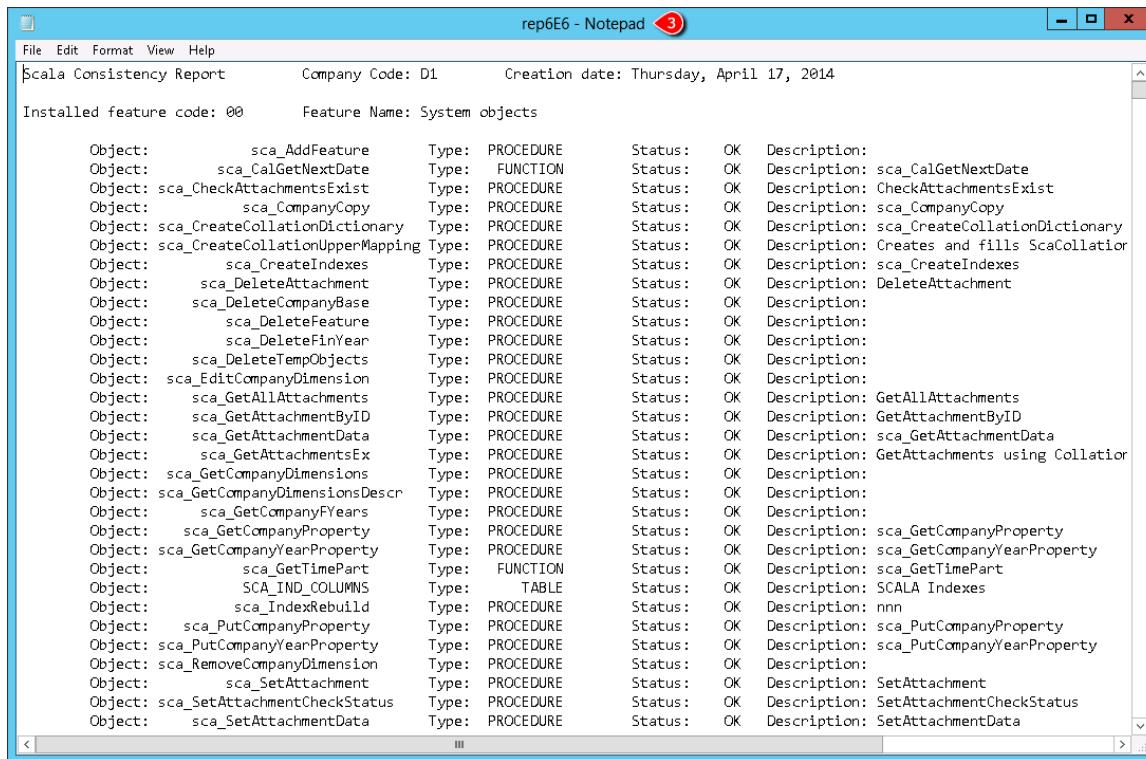
Column	Values	Description
Object status	  	Object does not exist. Structure or index of the Object differs from the system database data dictionary. Object does not fall into one of the above groups.
Object type	Procedure, Function, Table, or View	
Object description		Object description

Run the Consistency Report

1. In the **Properties of the Company <company name>** window, open the **Consistency** tab.



2. Click the **Consistency Report** button.
3. The report is displayed in Notepad.



```

Scala Consistency Report      Company Code: D1      Creation date: Thursday, April 17, 2014
Installed feature code: 00      Feature Name: System objects

Object: sca_AddFeature      Type: PROCEDURE      Status: OK      Description:
Object: sca_CalGetNextDate    Type: FUNCTION       Status: OK      Description: sca_CalGetNextDate
Object: sca_CheckAttachmentsExist  Type: PROCEDURE    Status: OK      Description: CheckAttachmentsExist
Object: sca_CompanyCopy      Type: PROCEDURE      Status: OK      Description: sca_CompanyCopy
Object: sca_CreateCollationDictionary  Type: PROCEDURE    Status: OK      Description: sca_CreateCollationDictionary
Object: sca_CreateCollationUpperMapping  Type: PROCEDURE    Status: OK      Description: Creates and fills ScaCollation
Object: sca_CreateIndexes     Type: PROCEDURE      Status: OK      Description: sca_CreateIndexes
Object: sca_DeleteAttachment  Type: PROCEDURE      Status: OK      Description: DeleteAttachment
Object: sca_DeleteCompanyBase  Type: PROCEDURE      Status: OK      Description:
Object: sca_DeleteFeature     Type: PROCEDURE      Status: OK      Description:
Object: sca_DeleteInYear      Type: PROCEDURE      Status: OK      Description:
Object: sca_DeleteTempObjects  Type: PROCEDURE      Status: OK      Description:
Object: sca_EditCompanyDimension  Type: PROCEDURE      Status: OK      Description:
Object: sca_GetAllAttachments  Type: PROCEDURE      Status: OK      Description: GetAllAttachments
Object: sca_GetAttachmentByID  Type: PROCEDURE      Status: OK      Description: GetAttachmentByID
Object: sca_GetAttachmentData  Type: PROCEDURE      Status: OK      Description: sca_GetAttachmentData
Object: sca_GetAttachmentsEx   Type: PROCEDURE      Status: OK      Description: GetAttachments using Collation
Object: sca_GetCompanyDimensions  Type: PROCEDURE      Status: OK      Description:
Object: sca_GetCompanyDimensionsDescr  Type: PROCEDURE      Status: OK      Description:
Object: sca_GetCompanyFYears    Type: PROCEDURE      Status: OK      Description:
Object: sca_GetCompanyProperty  Type: PROCEDURE      Status: OK      Description: sca_GetCompanyProperty
Object: sca_GetCompanyYearProperty  Type: PROCEDURE      Status: OK      Description: sca_GetCompanyYearProperty
Object: sca_GetTimePart        Type: FUNCTION       Status: OK      Description: sca_GetTimePart
Object: SCA_IND_COLUMNS       Type: TABLE         Status: OK      Description: SCALA Indexes
Object: sca_IndexRebuild      Type: PROCEDURE      Status: OK      Description: nnn
Object: sca_PutCompanyProperty  Type: PROCEDURE      Status: OK      Description: sca_PutCompanyProperty
Object: sca_PutCompanyYearProperty  Type: PROCEDURE      Status: OK      Description: sca_PutCompanyYearProperty
Object: sca_RemoveCompanyDimension  Type: PROCEDURE      Status: OK      Description:
Object: sca_SetAttachment     Type: PROCEDURE      Status: OK      Description: SetAttachment
Object: sca_SetAttachmentCheckStatus  Type: PROCEDURE      Status: OK      Description: SetAttachmentCheckStatus
Object: sca_SetAttachmentData  Type: PROCEDURE      Status: OK      Description: SetAttachmentData

```

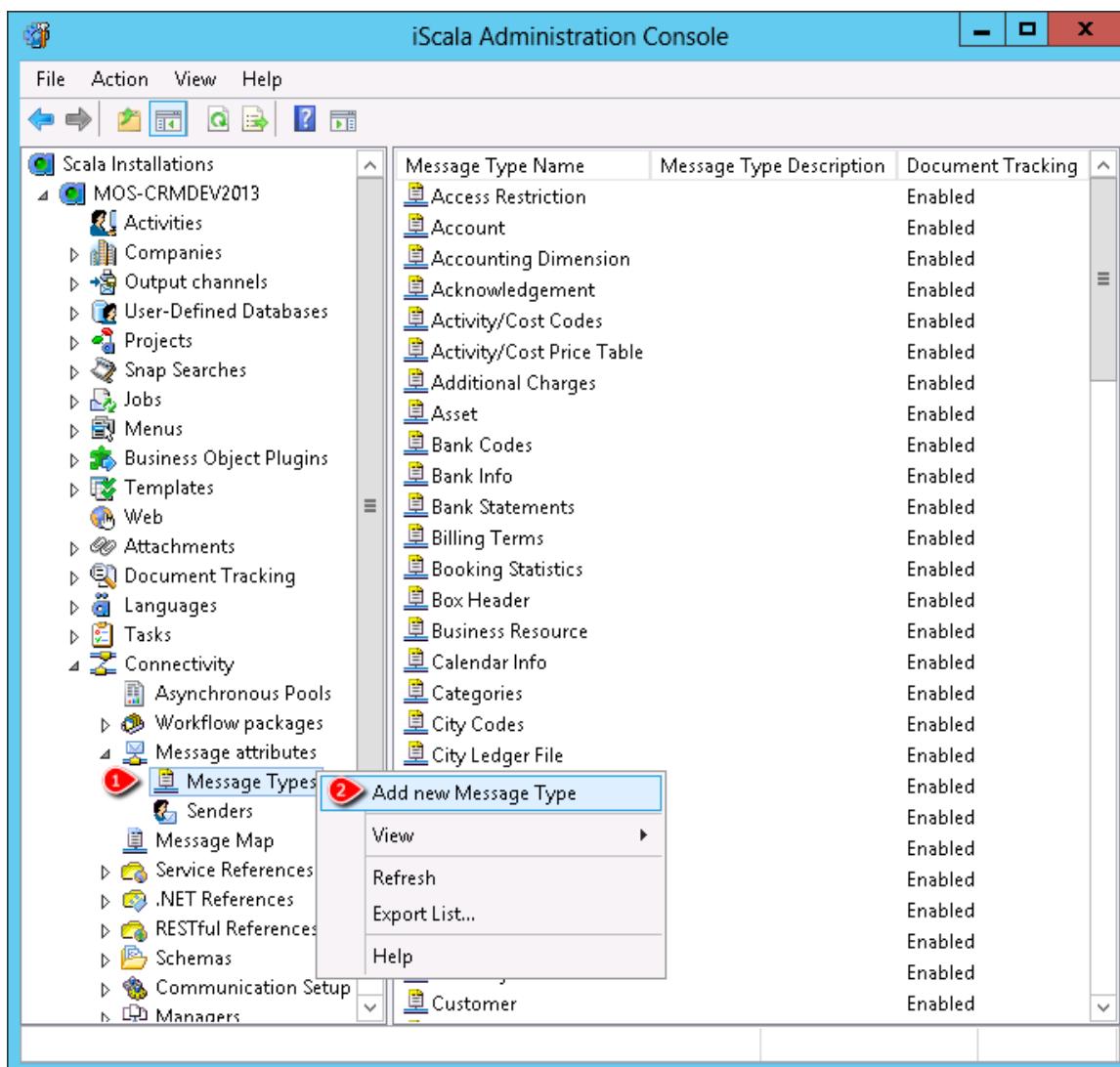
Channel Parameters

Channel parameters are accessed through the iScala Administration Console.

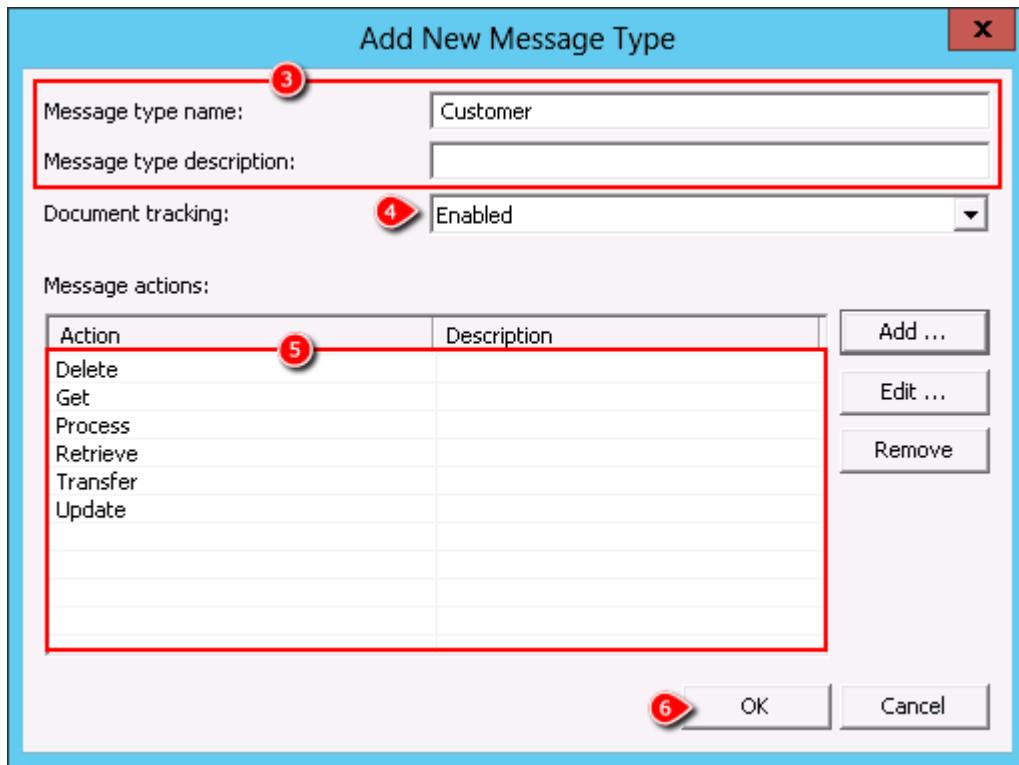
You can use the Administration Console to create a Message Type, Sender, and Message Map which connects the message type and sender with the workflow. The iScala Administration Console is also used to define the input file channel.

Create a Message Type

1. In the iScala Administration Console, expand the **Connectivity > Message Attributes > Message Types** nodes.



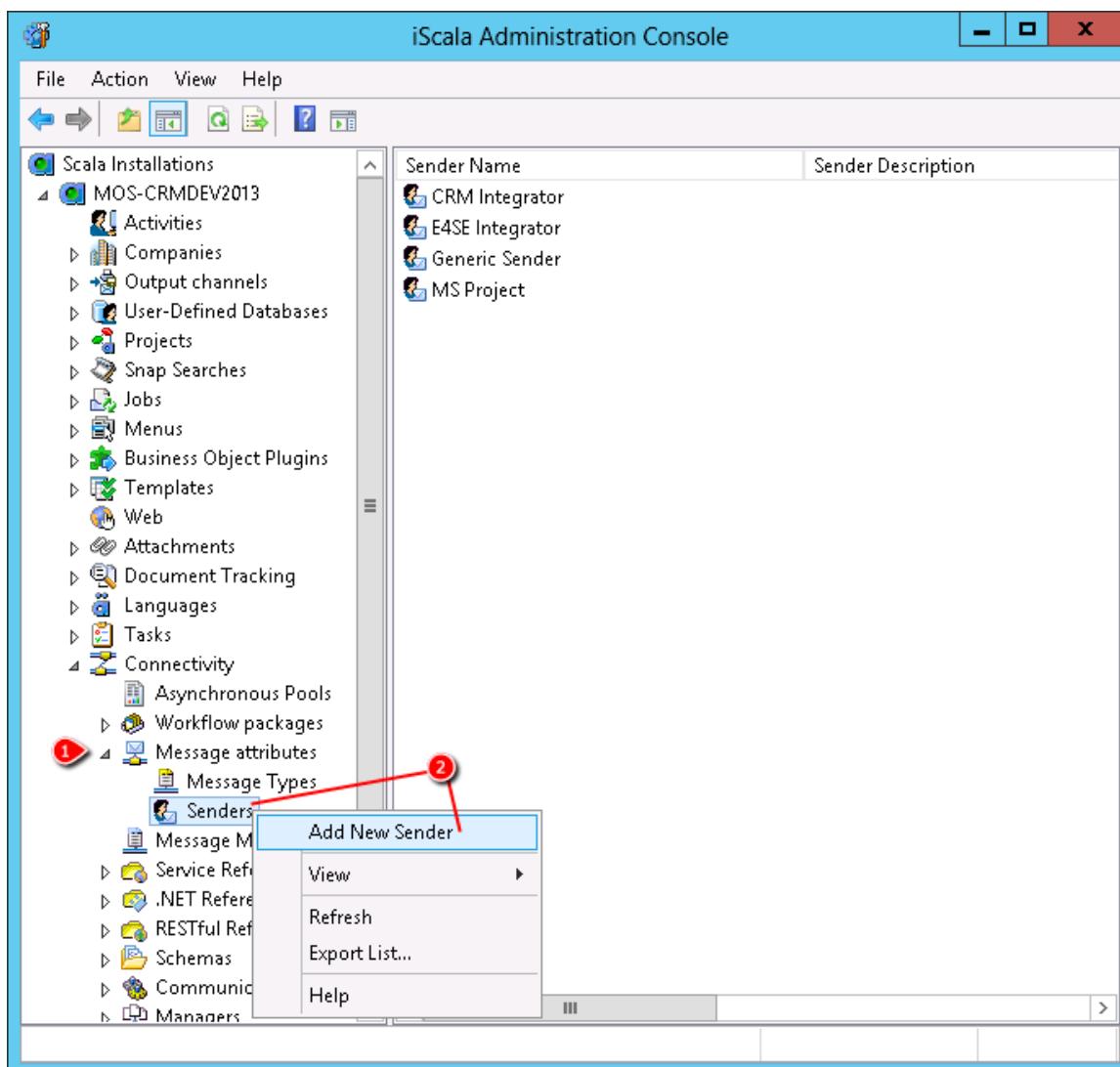
2. Right-click **Message Types** and select **Add new Message Type**.
3. Enter a **Message type name** and **Message type description**.



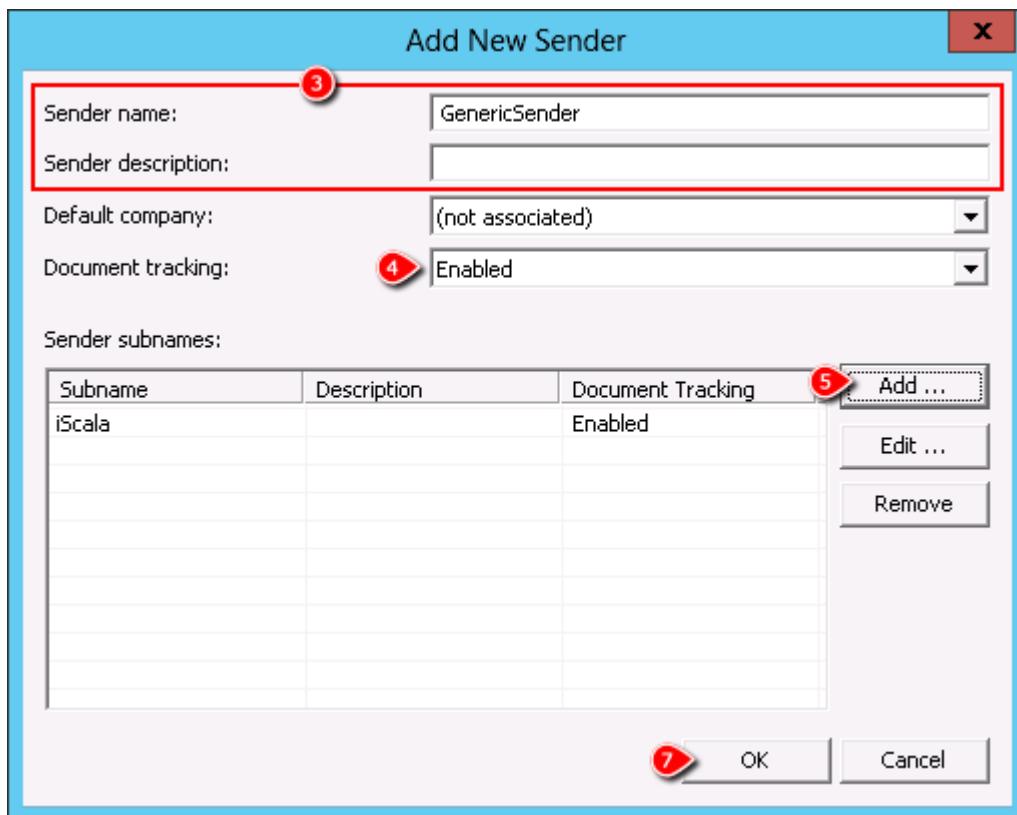
4. Select **Enabled** in the **Document Tracking** field.
5. Add a **Message Action**.
Actions act as descriptions of mapped processes.
6. Click **OK**.

Create a Sender

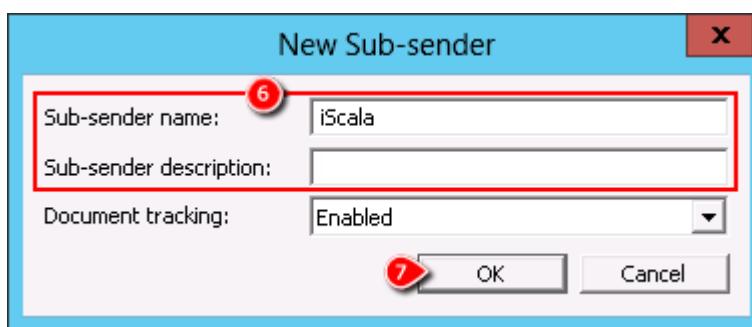
1. In the iScala Administration Console, expand the **Connectivity > Message Attributes** node.



2. Right-click **Senders** and select **Add New Sender**.
3. Enter a **Sender name** and **Description**.



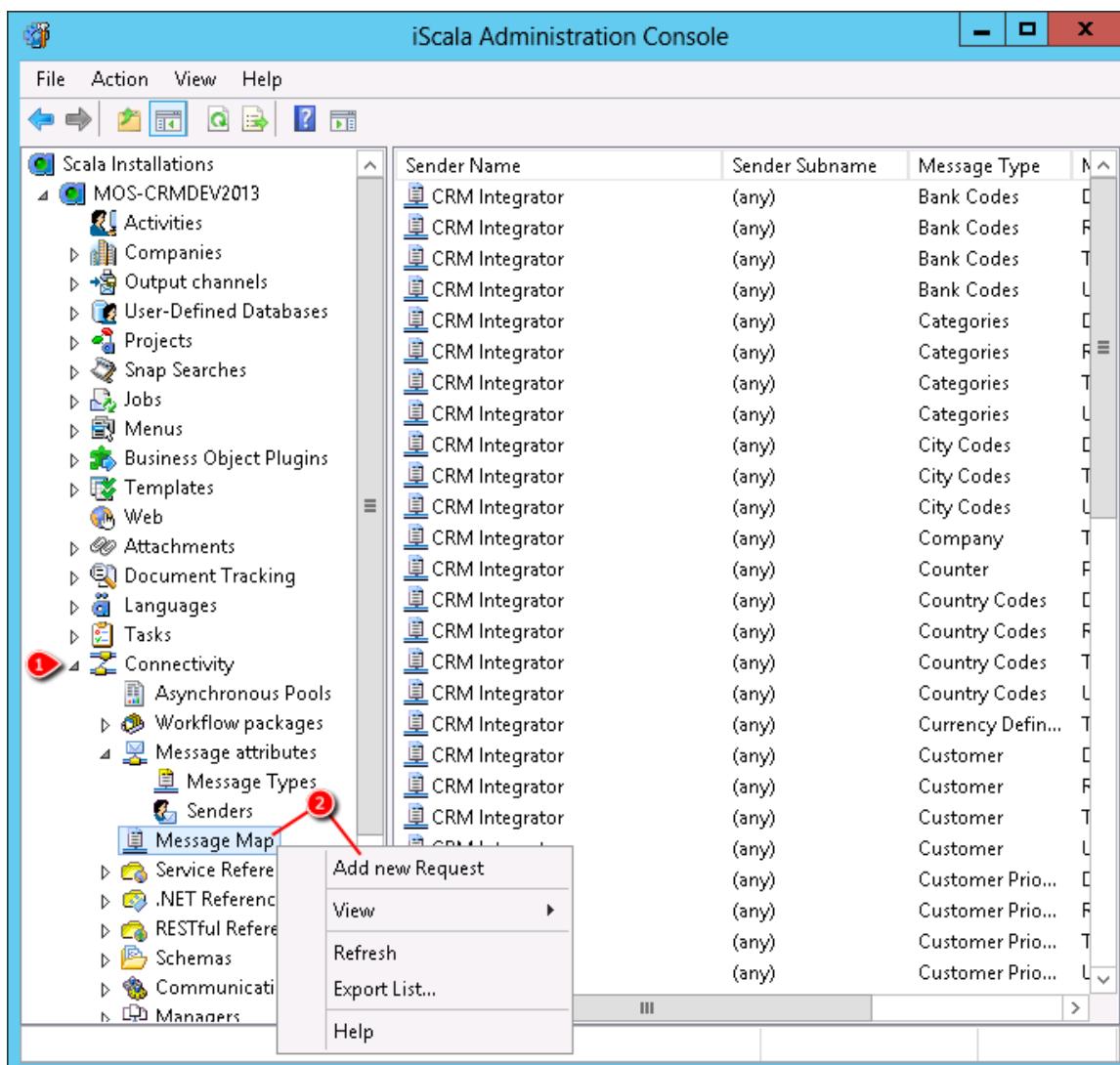
4. Select **Enabled** in the **Document tracking** field.
5. Click the **Add** button in the **Sender subnames** section.
6. Enter a **Sub-sender name** and **Sub-sender description**.



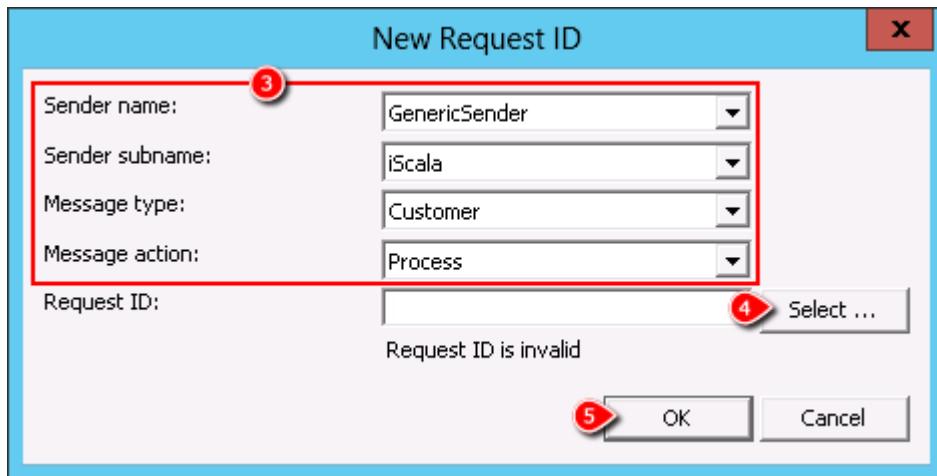
7. Click **OK**.

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. In the **New Request ID** dialog, select the **Message Type** and **Sender** information.

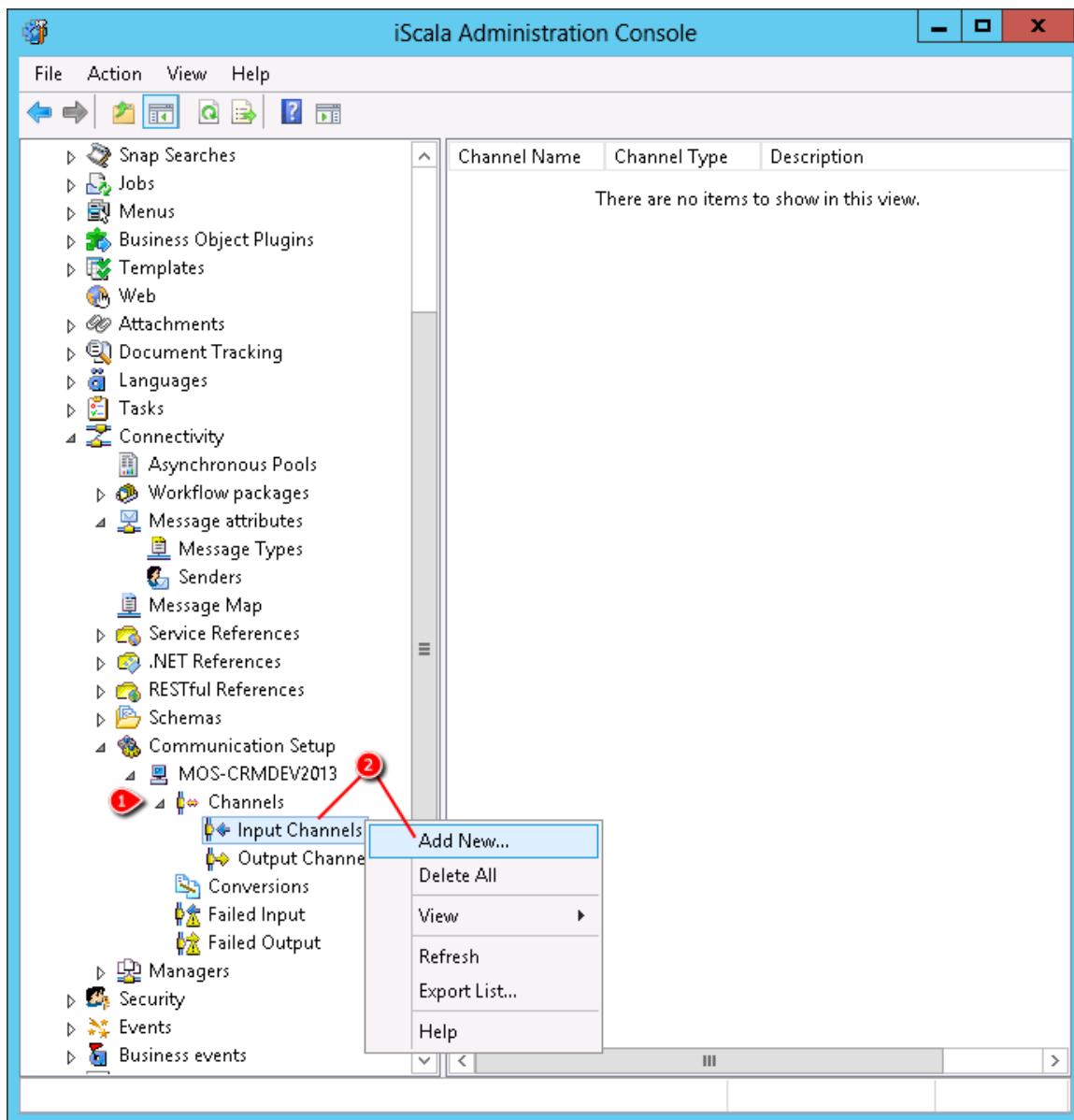


4. If you have already created a workflow, click **Select** to select your workflow from the list.

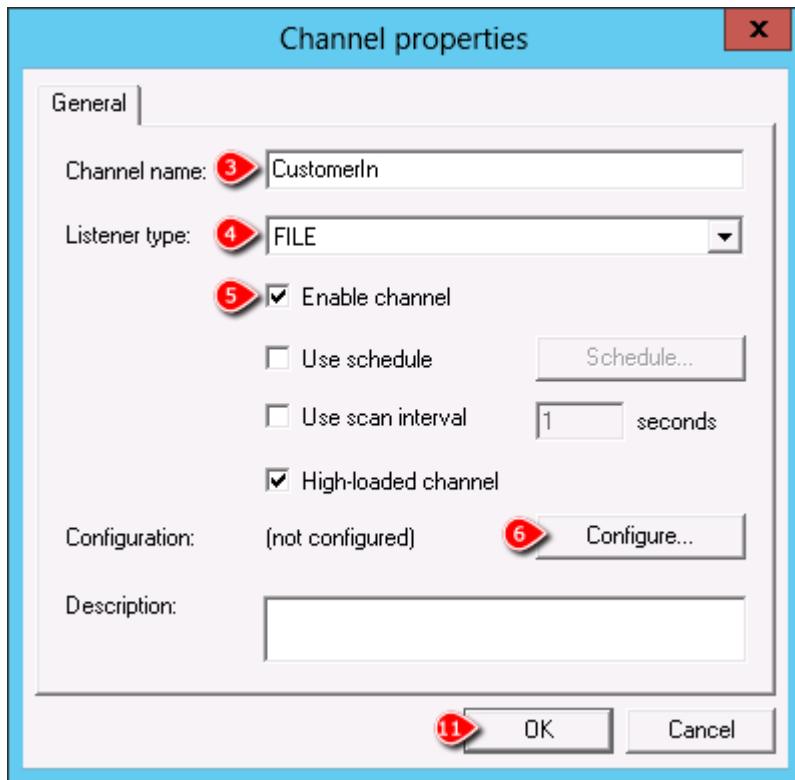
5. Click **OK** until you exit all dialog boxes.

Create File Input Channel

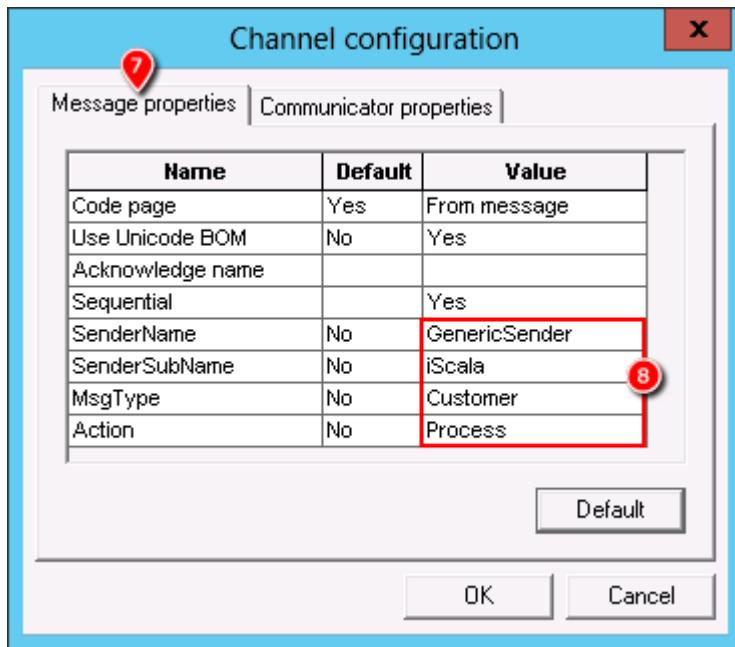
1. In the iScala Administration Console, expand the **Connectivity > Communication Setup > <server name> > Channels** node.



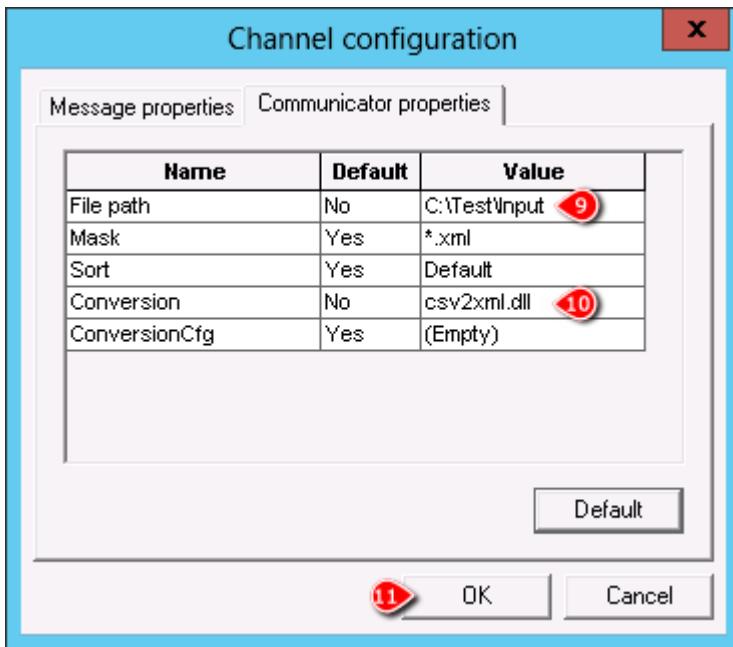
2. Right-click **Input Channels** and select **Add New**.
3. In the **Channel properties** window, enter a **Channel name**.



4. Select **FILE** as the **Listener type**.
5. Select the **Enable channel** check box.
6. To set channel properties, click **Configure**.
7. In the **Channel configuration** window, open the **Message properties** tab.



8. 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
9. On the **Communicator properties** tab, specify the **File path** to a folder Service Connect can monitor, for example, C:\Test\Input.



10. Select the appropriate **Conversion**.
11. 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.

Index

.net call example 405
.net import finalization 128
.net references 119

A

activity views 279
add .net methods 144
add a .net reference 120
add a channel 190
add a connect message type 584
add a connect sender 586
add a functoid to a conversion 457
add a message map 256
add a message map (routing) 94
add a message type 90, 544
add a new view 280
add a reference to the rules 480
add a restful reference manually 153
add a sender 92, 546
add a service reference 97
add an input channel 594
add an output channel 259, 599
add multiple .net references 134
add multiple users 31
add operation 166
add resource 164
add resource group 163
add service connect activities to the visual studio toolbox 365
add service methods 108
apply a literal value 445
apply an action to all activities in a view 301
archiving options 270
assign a user to a user group 636
asynchronous pool 178
attribute nodes 440

B

block a company 643
boolean functoids 462
bpm exposing info prompt 583
bpm integration completion 580
break 428
break example 428

C

change the admin and epicor user passwords 23
channel parameters 659
channels 188
check windows user names 629
choice 387
choice example 387
combine the data 519

commands toolbar 309
communication setup items, workflows, service references, .net references or rest references restore 68
company parameters 641
complex nodes 439
condition 381
condition example 381
conditional functoid example 471
configure com channel options 198
configure file channel options 199
configure ftp (.net) channel options 204
configure ftp channel options 201
configure http channel options 207
configure ibm mq and ibm mq (.net) channel options 213
configure imap channel options 218
configure msmq channel options 216
configure pop3 channel options 217
configure schedule channel options 208
configure sftp channel options 202
configure smtp channel options 220
configure sonicmq channel options 221
configure the input channel 495
configure windows service bus channel options 222
connectivity administration 55
connectivity components 80
conversion 377
conversion example 377
conversion value functoids 477
conversions 224
convert an epicor log file 493
convert workflow 238
create a backup 55
create a connect message map 588
create a directive that calls a service connect workflow 576
create a message extension 509
create a message map 663
create a message type 659
create a new user 619
create a project in visual studio 364
create a sender 661
create a workflow 251
create a workflow package 82
create document schemas 557
create file input channel 665
create input and output channels 548
create rules 479
create the message map 555

D

date-time functoids 467
date-time functoids format string 470
db lookup functoid example 483
db operation example 414
define a sharepoint workflow 258
define company audit settings 647

define conversion rules in place 477
delete a user 631
des poster 535
document conversion 18
document nodes 439
document submission 15
document tracking 267, 507
drop nodes to the user section 525

E

edit company-specific user properties 638
edit menu 307
edit resource group, resource, or operation 169
enterprise integration overview 584
epicor 10 .net reference setup 126
epicor 9 .net reference setup 125
epicor erp tracing 565
epicor erp web services setup 536
epicor log converter 492
epicor logon 531
epicor service connect administration console 13
example 580
export users 626
exposing processing as web services 86

F

failed input 233
failed output 234
fake nodes 441
file menu 307
filter a view using business attributes 295
filter an existing view 285
filter log entries 50
functoids 456

G

general settings 505
generate a schema 498
generate a schema for an inbound document 602
generating schema from sample data 494
grant access rights 612
grant permissions 29

H

help menu 309

I

ibm mq (.net) channel options 215
ibm mq channel options 214
import .net reference into service connect 537
import a list of users from a file 625
import a restful reference automatically 171
import a windows user 621
import activities into esc administration console 303
import service references 590
import to and export from document tracking 302

integration overview 536
items toolbar 311

L

locate a business object method 575
log in to service connect 22

M

manage asynchronous pools 180
manage custom conversions 230
manage licenses 52
manage service connect services 41
manage the financial years for the company 651
manage user sessions 40
mathematical functoids 463
menu bar 307
message attributes 90
message attributes and message maps 88
message extensions 509
message extensions example 509
message maps 94
move a workflow between workflow packages 83
multiple .net references import finalization 140

N

node annotations and data types 441
node characteristics 441
node mappings 450
nodeset functoids 466

O

operators 465

P

poster 335
poster example 335
pre-defined conversion rules 479
process a task 263
process properties 505
process variables 521
process variables example 521

R

re-import .net references 146
re-import an individual web service 113
re-import service references 110
register servers 38
register the wf workflow in service connect 366
remove a user from a user group 637
requester example 367
rest operation 408
RESTful references 153
restore from a backup 61
run the consistency report 657

S

sample data example 495
 schemas 177
 security 611
 service connect and epicor enterprise 584
 service connect and epicor erp 536
 service connect and epicor erp business process management 575
 service connect and epicor iscala 611
 service connect installation management 38
 service connect workflow designer 14
 service references 97
 set an administrator logon to a company database 644
 set logging of action reauthentication attempts 648
 set number of attempts of attempts for failed output 235
 set the company default menu and override msrs properties 646
 set up a condition 381
 set up a conversion 377
 set up a link condition 453
 set up a node condition 447
 set up a poster 335
 set up a process variable 521
 set up a requester 368
 set up a splitter 312
 set up a sub-workflow 353
 set up a task 316
 set up a user 24
 set up a web method 396
 set up an input channel 191
 set up channel schedule 196
 set up conversion plug-in configuration options 226
 set up document tracking 268
 set up poster dynamic properties 347
 set up requester dynamic properties 371
 set up task dynamic properties 323
 set up the input channel 248
 set up the windows wf integration 364
 setup user accounts using roles 26
 sharepoint integration 247
 simple nodes 439
 special functoids 466
 special technique 610
 how to use the epicor for service enterprises debugger 610
 splitter 311
 standard toolbar 310
 string functoids 461
 sub-workflow 353
 sub-workflow example 353
 summary 79, 305, 535, 669

T

task 316
 task dynamic configuration example 329
 task example 316
 tasks 263
 test connectivity 115
 test methods 150
 the service connect environment 12
 tools menu 308
 trace log, options 569
 trace log, start 565

U

use a document type to filter a view 290
 use choice rules 387
 use condition rule 381
 use event logs 48
 use the container 515
 use the output channel from a poster 261
 use the rules file for the conversion functoid 482
 use the user properties tab 632
 user management 22
 user parameters 618

V

validate process 532
 view a list of all users 619
 view activities 271
 view general company properties 642
 view installation properties 43
 view lists of installed modules and system objects 656
 view mapped nodes only 451
 view menu 308
 view the user logon to a company database 645

W

web method example 396
 window menu 308
 windows workflow foundation call 363
 workflow converter 236
 workflow designer window 306
 workflows 80

X

xml mapper 435
 xml mapper graphical interface 437
 xml trace report 572



Additional information is available at the Education and Documentation areas of the EPICweb Customer Portal. To access this site, you need a Site ID and an EPICweb account. To create an account, go to <http://support.epicor.com>.