



ENOVIA SmarTeam

SmarTeam – Gateway Implementation Guide

© Dassault Systèmes, 2004, 2010. All rights reserved.

CATIA, ENOVIA, SMARTEAM and the 3DS logo are registered trademarks of Dassault Systèmes or its subsidiaries in the US and/or other countries.

PROPRIETARY RIGHTS NOTICE: This documentation is the property of Dassault Systèmes. This documentation shall be treated as confidential information and may only be used by employees or contractors of the Customer in accordance with the terms of the End-User License Agreement accepted by Customer.

Any use of the Licensed Program contained in this media or accompanying it, is subject to the terms of the End User License Agreement accepted by Customer. The Licensed Program is protected by international copyright laws and international treaties. Unauthorized use, reproduction and/or distribution of any of the Licensed Program, or any part thereof, may result in severe civil and/or criminal penalties, and will be prosecuted to the maximum extent possible under the law. Company names and product names mentioned herein are the property of their respective owners and certain portions of the Licensed Program contain elements subject to copyright owned by these entities. See the Documentation CD provided with the Licensed Program for details and/or additional terms and conditions relating to these entities.

Part Number: GWY-I4-200110

Table of Contents

Chapter 1: Introduction	1
SmarTeam – Gateway	2
Software Modules	2
SmarTeam – Gateway Architecture	3
Object Data Flow	3
Typical Integrations	3
Integration of Two SmarTeam Sites	4
Configuring object data transmission	4
Integrating SmarTeam with an ERP Application	4
Configuring object data transmission	4
Understanding Event-Driven Data Transmission	4
Standard SmarTeam Operation Event	5
User-Defined Gateway Event	6
Naming Conventions	9
Chapter 2: Preparing SmarTeam for Gateway Operations	10
Modifying Data Structures	11
Adding SmarTeam – Gateway Behavior	11
Creating an ECO Class	11
Creating a ECO Class Procedure	12
Modifying the Component Class	12
Using the ECO and Component Classes	12
Configuring the Database Connection for Gateway Operations	12
Defining a SmarTeam User-Defined Command for a User-Defined Gateway Event	13
Write the Script	13
Create a SmarTeam User-Defined Command	14
Defining a Microsoft® Messaging Queue	16
Chapter 3: Configuring SmarTeam – Gateway	18
Scenarios	18
Scenario 1: Outbound SmarTeam Component Object Data Transmission	19
Defining an Outbound Gateway Operation Mapping	19
Defining a Gateway Object Specification	22
Initiating and Verifying Data Transmission	23
Scenario 2: Outbound SmarTeam Assembly Data Transmission	24
SmarTeam Assembly	24
Defining an Outbound Gateway Operation Mapping	24
Procedure	24

Defining a Gateway Object Specification	26
Procedure	26
Initiating and Verifying Data Transmission	27
Scenario 3: Inbound SmarTeam Component Object Data Transmission	28
Defining Inbound SmarTeam Functionality	28
Procedure	28
Initiating and Verifying Data Transmission	29
Scenario 4: Using SmarTeam – Gateway API for Inbound Connectivity	30
Writing a Script	30
Sample Script	30
Setting the Script Hook	32
Chapter 4: Integrating with a Remote SmarTeam Site	34
Integration Architecture	34
Operation of Gateway software modules	34
Scenario 1: Sending a SmarTeam Component to a Remote SmarTeam Site	36
Configuring Source SmarTeam – Gateway for Outbound Flow	36
Configuring Remote SmarTeam – Gateway for Inbound Flow	36
Chapter 5: Integrating with BizTalk 2004/2006	37
Creating Schemas	37
Load Schemas into a Microsoft Visual Studio Project	38
Orchestration	39
Messages	39
Receive Node	39
Receive Port	40
Send Node	40
Send Port	40
Mapping	40
Set Strong Name Key	41
Build and Deploy Solution	42
Create Send Port	42
Create Receive Port	43
Final Installation Procedures	44

Chapter 1: Introduction

This document describes how to use SmarTeam – Gateway to integrate SmarTeam with Enterprise applications such as SAP and Oracle Applications in an ERP system, for a number of common data transmission scenarios. The purpose of such integration is to allow for event-driven transmission of SmarTeam object data to and from the integrated Enterprise applications using standard XML data transmission methods.

One main benefit of Enterprise Application Integration (EAI) for SmarTeam is the ability to provide up-to-date engineering information to Enterprise Resource Planning applications, such as SAP, about products being developed under a SmarTeam application and to receive management information from them. For example, when a SmarTeam object is created, SmarTeam transmits the relevant object information to the ERP application and management parameters for the newly created object are created by the ERP application and sent back to SmarTeam.

The document is aimed at integrators of SmarTeam in EAI systems and guides the integrator step-by-step through the integration process, including testing the integration at each stage.

It is assumed that the integrator has a basic knowledge of BizTalk operations and of the various connectivity solutions in an enterprise ERP system.

Note: Both Microsoft® BizTalk 2004 and Microsoft® BizTalk 2006 tools are supported in this release of SmarTeam – Gateway. Only screen captures for Microsoft® BizTalk 2004 are shown in this manual.

The descriptions in this manual apply to the standard SmarTeam SmDemo database. The description may vary for other SmarTeam databases.

All of the console functions and commands used in this document are defined in the SmarTeam Gateway Administration Guide (BizTalk). Please refer to that document whenever necessary.

SmarTeam – Gateway

The SmarTeam – Gateway product refers to a set of software modules used to set up and facilitate integration of SmarTeam object data in an ERP system. The modules are usually installed on servers separate from the server on which the SmarTeam database resides.

Software Modules

The following table shows the SmarTeam – Gateway modules, their server locations and if they are used for configuration or for real-time processing:

Software Module	Functionality	Location	When Used
SmGatewaySync COM object/ERP Sync Service	Gets and translates outbound binary SmObject into SmartGatewayBOM XML object and sends to Message transport	Gateway Server	Real-time
SmartERPSyncServer Connection (Gateway client) module	Works in parallel with the SmarTeam – Engine. Receives information on execution of each SmarTeam operation and writes Gateway events to the database	SmarTeam – Editor client	Real-time
Gateway Inbound Processor	Receives and translates inbound Smart Gateway XML object into SmObject and processes the object according to the inbound rules.	Gateway Server	Real-time
SmarTeam – Gateway Integration Manager	Configures inbound and outbound rules and Gateway object schema	Gateway Server	Configuration
BizTalk Adapter's Push Application COM object (called by AIC)	Transfers outbound SAP or Oracle XML data to SAP or Oracle	BizTalk Server	Real time
BizTalk Gateway Push Application COM Object (called by AIC)	Transfers inbound Gateway XML Object to Gateway Receive application	BizTalk Server	Real time

SmarTeam – Gateway Architecture

Object Data Flow

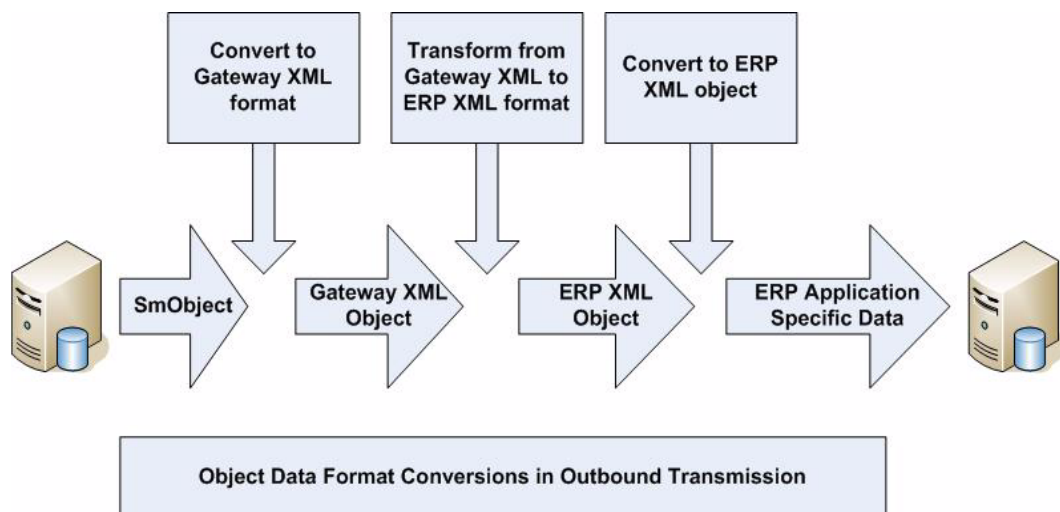
The SmarTeam – Gateway architecture insures the timely, reliable and flexible transmission of object data from SmarTeam to and from a remote ERP application within an EAI system.

Data flow from SmarTeam to the remote ERP application takes place in the following stages:

- 1 In the SmarTeam Gateway server, SmarTeam object data is converted to the Gateway standard XML format and forwarded by messaging to BizTalk.
- 2 In BizTalk, the Gateway standard XML format is transformed to a standard ERP application XML format and forwarded to the remote ERP application server.
- 3 At the remote ERP application server, the ERP application XML is converted to ERP application-specific data and transferred to the ERP application.

The data transmission is initiated by a SmarTeam operation event. Data flow from the remote ERP application to SmarTeam takes place in a similar way.

The following schematic shows the outbound data flow, where ERP denotes SAP, Oracle or other ERP application with which Gateway connects.



This architecture achieves the desired goals:

Reliable and timely data transmission: Event-driven transmission, asynchronous messaging, and the use of standard XML text insures reliable and timely data transmission.

Flexibility: The same Gateway XML object could be transmitted to another ERP application by simply using a different XML mapping. The data could also be sent to multiple ERP applications simultaneously.

Typical Integrations

Common SmarTeam – Gateway integrations are

- Integrating with a remote SmarTeam site
- Integrating with SAP
- Integrating with Oracle Applications

- Integration with proprietary middleware systems (not covered in this document)

Integration of Two SmarTeam Sites

In integrating and synchronizing a SmarTeam site with a remote one, there are two areas of Gateway operation:

- Gateway tools to configure object data transmission
- Gateway software modules to support actual transmission of object data, both sending and receiving

Configuring object data transmission

Configuring object data transmission is achieved in three configuration sessions using two tools:

- 1 Using the Gateway Integration Manager tool to configure the properties of outbound and inbound object data to and from the SmarTeam database, including setting the event that drives the data transmission and the XML schema for the objects involved.
- 2 Using the Gateway Integration Manager tool to configure the properties of outbound and inbound object data to and from the remote SmarTeam database, including setting the event that drives the data transmission and the XML schema for the objects involved.
- 3 Using Microsoft BizTalk 2004® / 2006® Server tools to set up the connection between the two XML formats. Note that if the local and remote SmarTeam databases have the same structure, no mapping is necessary.

Integrating SmarTeam with an ERP Application

In integrating SmarTeam with an ERP application such as SAP or Oracle Applications, there are two areas of Gateway operation:

- Using Gateway tools to configure object data transmission
- Operation of Gateway software modules to support real-time transmission of object data

Configuring object data transmission

Configuring object data transmission is achieved in three configuration sessions using three tools:

- 1 Using the Gateway Integration Manager tool to configure the properties of outbound and inbound object data to and from the SmarTeam database, including setting the event that drives the data transmission and the XML schema for the objects involved.
- 2 Using the Gateway Adapter manager application to configure the required XML schema for the ERP objects and to set the ERP operation to write the data to the ERP application.
- 3 Using Microsoft BizTalk 2004 / 2006 Server tools to set up the connection and mapping between the two XML schemas.

Understanding Event-Driven Data Transmission

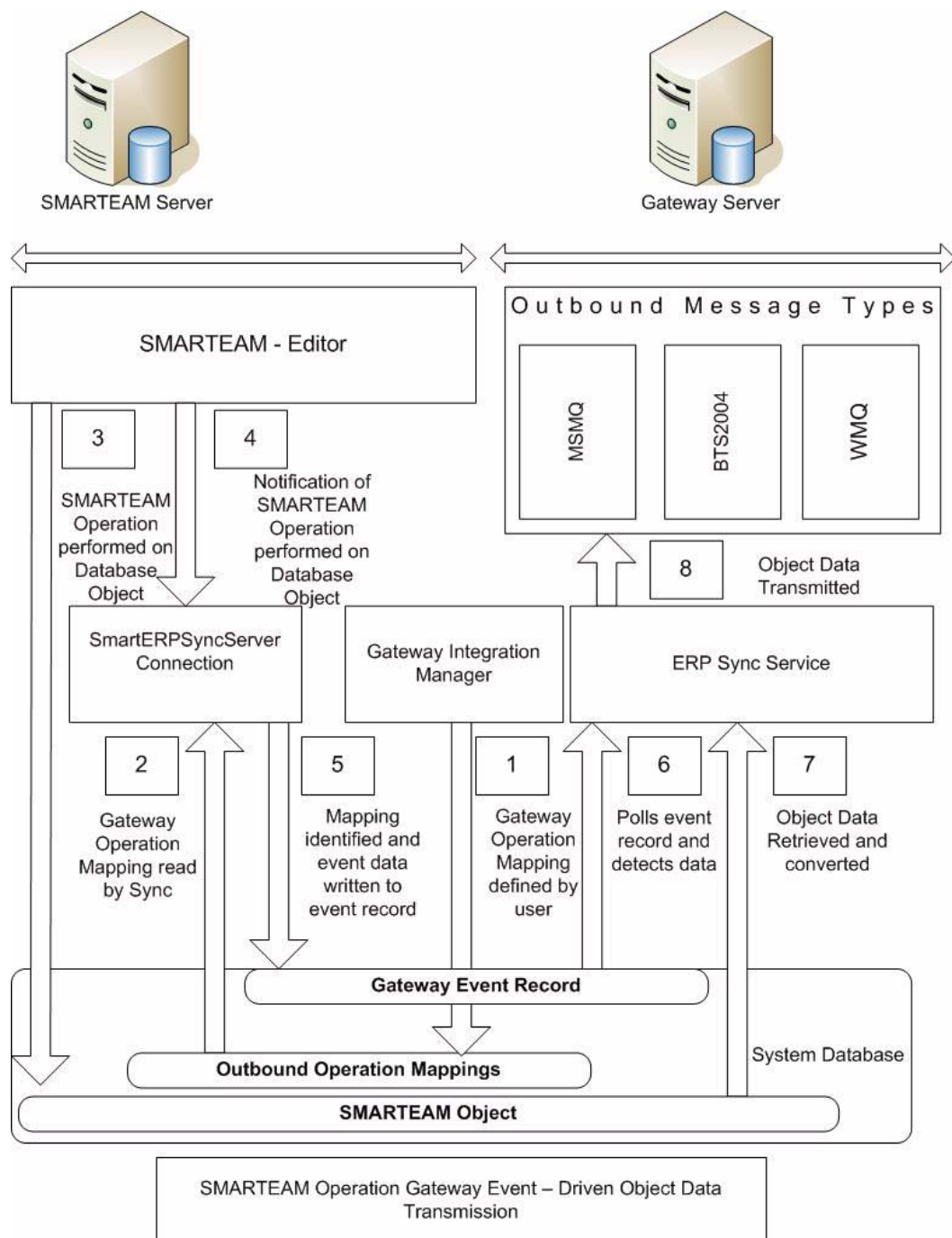
This section describes the mechanism that triggers data transfer.

There are two types of events that can trigger the ERP Sync Service to get SmarTeam object data: a standard SmarTeam operation event and a user-defined Gateway event.

Standard SmarTeam Operation Event

A standard SmarTeam operation event uses a standard SmarTeam operation to trigger the event. To setup this type of event, you use the Gateway Integration Manager to create a Gateway Operation Mapping, which specifies a SmarTeam operation and a SmarTeam class, for example, the Update operation and the Component class. This is the data required for the event. The event is triggered when the Update operation occurs on an object of the class, and causes the ERP Sync Service to get the updated object.

The mechanism is as follows: The Gateway Operation Mapping data is transferred to the SmartERPSyncServerConnection, a Gateway module that resides in the SmarTeam server. Each time the SmarTeamr performs an operation on an object in the SmarTeam database it informs the SmartERPSyncServerConnection module, which then checks whether the operation/object pair corresponds to one of the defined Gateway Operation Mappings. If it does, SmartERPSyncServerConnection writes the operation/object information in a special database “event” record that is continually polled by the Gateway ERP Sync Service. When Gateway ERP Sync Service detects that a Gateway Operation Mapping has been written there, it reads the operation/object information, and then gets the object data.



The write operation of SmartGatewaySync to the event database record can be viewed as setting the server event.

User-Defined Gateway Event

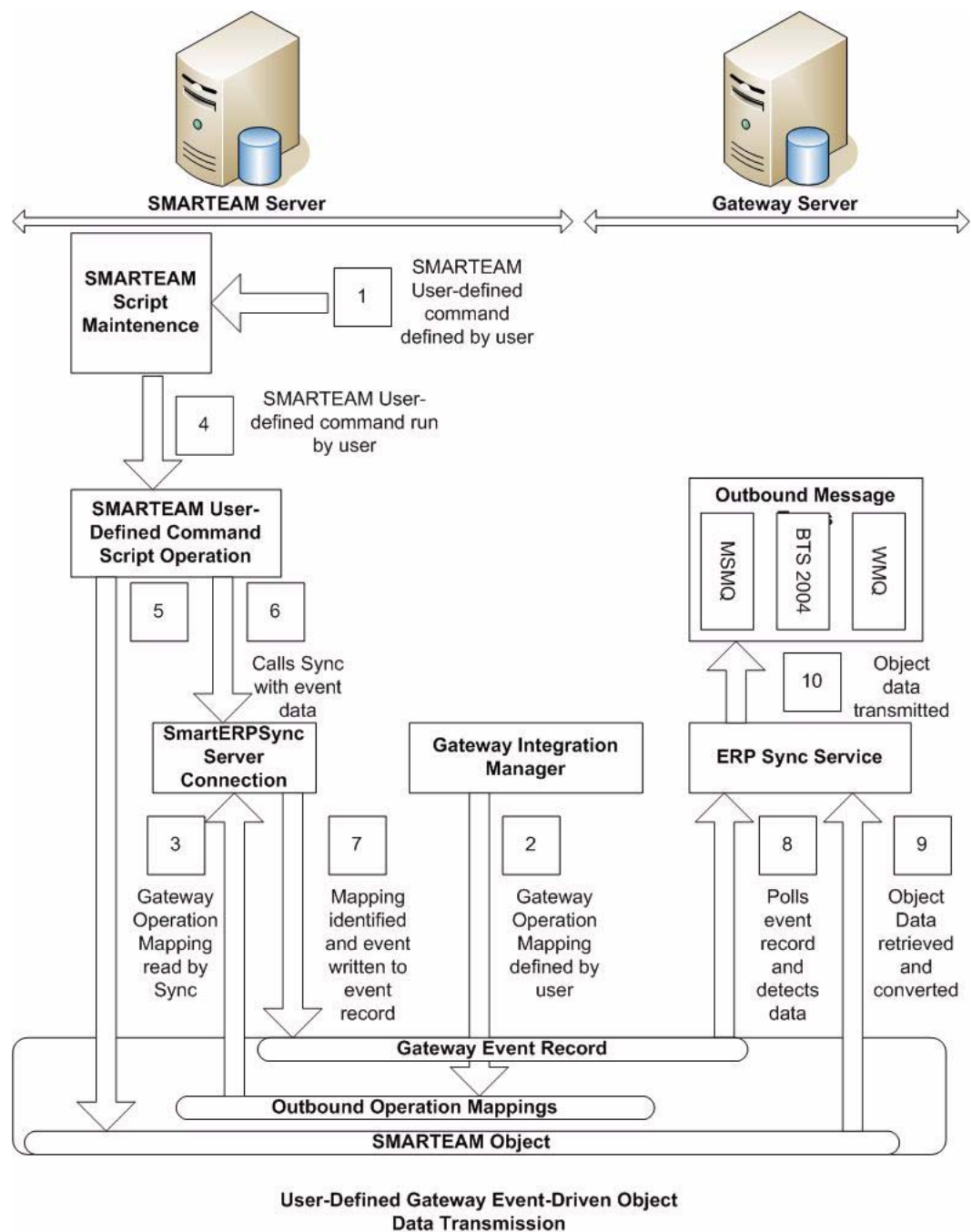
The *user-defined* Gateway event uses a SmarTeam user-defined command to trigger the event. It is similar to the SmarTeam operation Gateway event except that the SmartERPSyncServerConnection module is directed to write specific information to the event record by a call embedded in a script run by a SmarTeam user-defined command.

A user might want to use this type of event when using a user-defined command, which performs a custom operation on an object in the database before transmitting the object on Gateway. For example, the script could increment the value of an attribute (“Synchronized with ERP”) each time the object is sent. Another reason might be to enable the user to initiate Gateway transmission of an object without waiting for a specific SmarTeam operation.

This type of event is defined in the Gateway Integration Manager by specifying an event name and a SmarTeam class. The event name is typed in the user-defined command field and the class in the class name field. That same name/class information must be inserted into the script of the SmarTeam user-defined command. Specifically, it is inserted as parameters of a call `SyncConnection.ReportEventByCommand`, which writes the information to the event database record. As a result, when the user invokes the user-defined command, the event record is written.

The polling Gateway ERP Sync Service reads the event record and compares the event name with the event definition registered by the Gateway Integration Manager. If the name matches, the Gateway ERP Sync Service gets the object information.

Using this type of event assumes that the user has defined a special SmarTeam user-defined command (activated, for example, by a button), which runs the script. See section [Defining a SmarTeam User-Defined Command for a User-Defined Gateway Event](#) for information on how to create such a SmarTeam user-defined command.



Naming Conventions

In this guide, the following file and document naming conventions are used:

File type	File Format	Examples
Schema, Document Specification (in WEBDav)	[ApplicationName] [ObjectName] Specification	SmarTeam Component Specification,
BizTalk Document	[ApplicationName] [ObjectName] Document	SmarTeam Component Document,

Chapter 2: Preparing SmarTeam for Gateway Operations

Prior to using a SmarTeam database with SmarTeam – Gateway, the SmarTeam database needs to be prepared for that purpose. This chapter describes how to enable SmarTeam for Gateway operations.

Note: The description in this section applies to the standard SmarTeam SmDemo database. The description may vary for other databases.

The following database changes can be performed:

- [Modifying Data Structures](#)
- [Configuring the Database Connection for Gateway Operations](#)
- [Defining a SmarTeam User-Defined Command for a User-Defined Gateway Event](#)
- [Defining a Microsoft® Messaging Queue](#)

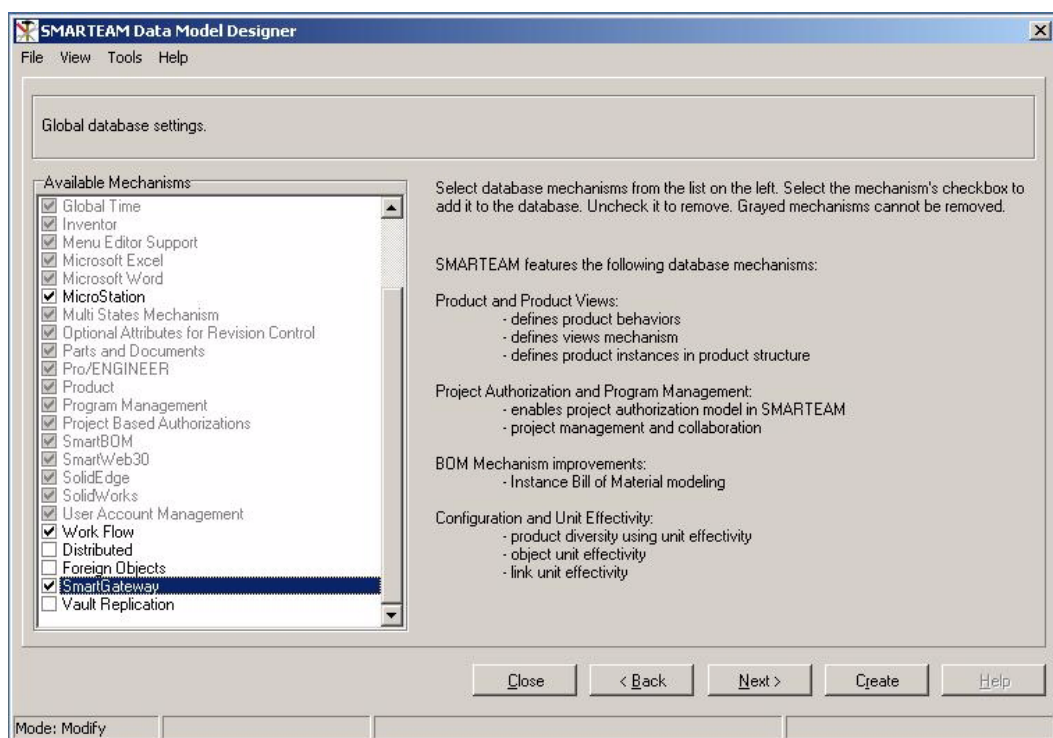
Modifying Data Structures

This section describes how to use the SmarTeam Data Model Designer to modify data structures in the SmarTeam database and includes the following topics:

- Adding SmarTeam – Gateway Behavior
- Adding ECO Class
- Component Class modification

Adding SmarTeam – Gateway Behavior

- 1 Select **File → Modify Database Structure**
- 2 Select the database from the database selection screen to which you want to add Gateway behavior
- 3 Click **OK**. The Wizard screen appears as follows.



- 4 On the Global database settings step of the Wizard, mark the check box on the left side of the SmartGateway item in the Available Mechanisms list.
- 5 If required, perform the procedure of the next section, Creating an ECO class at this point.
- 6 Click on the **Create** button
- 7 Answer Yes to all prompts

Creating an ECO Class

This section uses the SmarTeam Data Model Designer to create an ECO (Engineering Change Order) class in the SmDemo database. In addition, the Component class is modified to contain a reference to an ECO object. The procedure of this section can be performed as step 6 of the previous section.

There are two major steps:

- Creating the ECO class
- Modifying the class that uses the ECO – the Component class in our case.

Creating a ECO Class Procedure

- 1 Using the SmarTeam Data Model Designer, create a new Class, called “ECOs”.
- 2 Make sure that at least the two fields: CN_ID and TDM_DESCRIPTION are included in the class. The field CN_ID serves as the ECO identifier and the field TDM_DESCRIPTION serves as the ECO description text.
- 3 Define CN_ID as an index
- 4 Create a Link Class named “ECO Project Relation” where the first linked class is the class “Projects” and the second linked class is the class “ECOs”
- 5 Define the Display Name field of the ECOs as “ECO”

Modifying the Component Class

- 6 Add a CHANGE_NOTICE field to the Component class. It will represent the ECO Number (CN_ID field) of the ECO.
- 7 Add the CHANGE_NOTICE field to the Component Attributes Profile Card. It will identify the ECO that is related to this Component.

Using the ECO and Component Classes

The ECO class created in the previous section and the modified Component class are used in SmarTeam – Gateway as follows:

- 1 An ECO object is created in SmarTeam with a certain ECO number
- 2 The ECO object is sent to Oracle Applications via SmarTeam – Gateway
- 3 Oracle Applications creates an internal ECO object with same identifier and description as the ECO object it received from SmarTeam
- 4 A BOM is created in SmarTeam – Editor. The number of the ECO sent to Oracle Applications in step 2 is inserted in the CHANGE_NOTICE field of each BOM item
- 5 The BOM is sent to Oracle Applications

The Oracle Applications adapter reads the CHANGE_NOTICE field in the BOM items and identifies the ECO number as one that is has received previously. As a result, the BOM is transferred to the engineering department (instead of to manufacturing) and differences are saved under the ECO whose number appeared in the CHANGE_NOTICE field.

Configuring the Database Connection for Gateway Operations

This section describes how to enable the SmarTeam database to properly connect with SmarTeam – Gateway.

- 1 Open the SmarTeam Database Connection Manager
- 2 Select the database which was enabled for SmarTeam – Gateway in the previous section

- 3 Right click and select **Properties**
- 4 In the Servers box, mark the check box named **Available for use in SmarTeam Gateway**

Defining a SmarTeam User-Defined Command for a User-Defined Gateway Event

This section describes how to configure a SmarTeam user-defined command that sets a user-defined Gateway event. It is relevant only for users that need to use a user-defined Gateway event to initiate data transmission. See [User-Defined Gateway Event](#), for more information.

The major steps are:

- 1 Write the script that contains the function call `SyncConnection.ReportEventByCommand`
- 2 In Script Maintenance, create a SmarTeam user-defined command and attach the script to it.
- 3 Create a menu item or button to invoke the command.

Write the Script

The required script can be any script that includes specifying a SmarTeam object and class for transmission by Gateway. That info should appear in the second and third parameters of

```
SyncConnection.ReportEventByCommand "SendUpdate", ClassID, ObjectID
```

In the first parameter, you use the same event name that you type in the “user-defined command” field of the Gateway Operation Mapping dialog of the Gateway Integration Manager. The event name can be any string as long as you are consistent.

The following script is an example that can be copied and used.

```
Function SendUpdate(ApplHndl As Long,Sstr As String,FirstPar As Long,SecondPar As Long,ThirdPar As Long ) As Integer
Dim SmSession As ISmSession
Dim SyncConnection As Object
Dim ClassID As Integer
Dim ObjectID As Long
Dim SmRecordList As Object
Dim GUIServices As Object
Dim SelectedObject As Object

CONV_RecListToCOMRecordList FirstPar, SmRecordList

' Create SmarterERPSyncServerConnection Service
Set SmSession = SCREXT_ObjectForInterface(ApplHndl)
Set SyncConnection =
SmSession.GetService("SmarterERPSyncServerConnection.SmERPSyncServerConnection")

If (SmRecordList.ValueAsInteger("CLASS_ID", 0) > 0) And
(SmRecordList.ValueAsInteger("OBJECT_ID", 0) > 0) Then
    ' Get ST selected object's ClassID and ObjectID
    ClassId = SmRecordList.ValueAsInteger("CLASS_ID", 0)
    ObjectId = SmRecordList.ValueAsInteger("OBJECT_ID", 0)
Else
    Set GUIServices = SmSession.GetService("SmGUISrv.SmCommonGUI")

    If (Not GUIServices Is Nothing) And (Not
GUIServices.ActiveViewWindow Is Nothing) And (Not
```

```
GUIServices.ActiveViewWindow.SmView Is Nothing) And (Not
GUIServices.ActiveViewWindow.SmView.Selected Is Nothing) Then
    Set SelectedObject =
GUIServices.ActiveViewWindow.SmView.Selected.Objects(0)

    If (Not SelectedObject Is Nothing) Then
        ' get class id and object id
        ClassId = SelectedObject.ClassId
        ObjectID = SelectedObject.ObjectID
    Else
        MsgBox "Exception"
    End If
Else
    MsgBox "Exception"
End If
End If

MsgBox "Update Sent (" & ClassID & ", " & ObjectID & ")"

' Report a user-defined command to the server, using the Sync Connection.
' The method is After (1) Command ("My ERP Event") with ClassID and
ObjectID.
SyncConnection.ReportEventByCommand "SendUpdate", ClassID, ObjectID
' here you can change the event name SendUpdate to any other name
' making sure you use that same name in the "user-defined command"
' field of the Gateway Operation Mapping dialog

MyERPEvt=Err_None ' the function should return a value

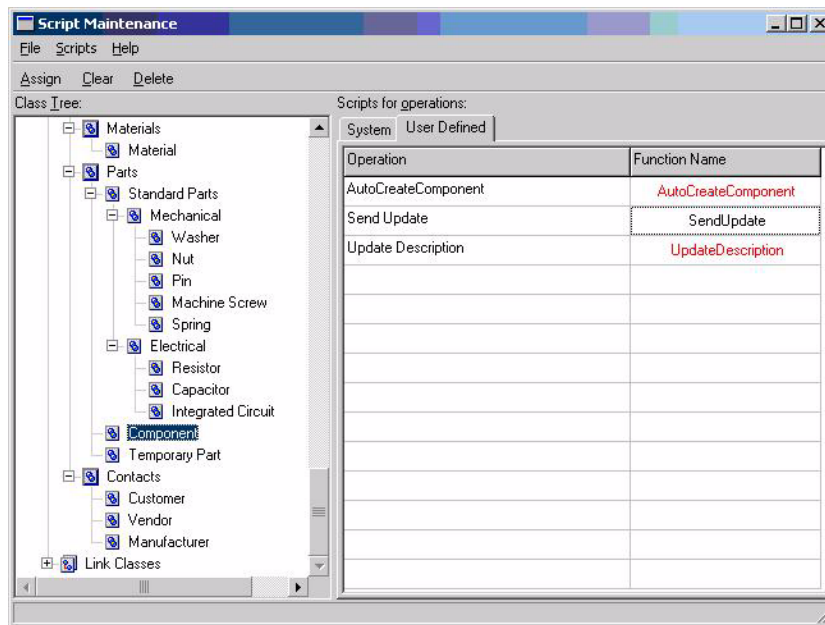
End Function
```

Create a SmarTeam User-Defined Command

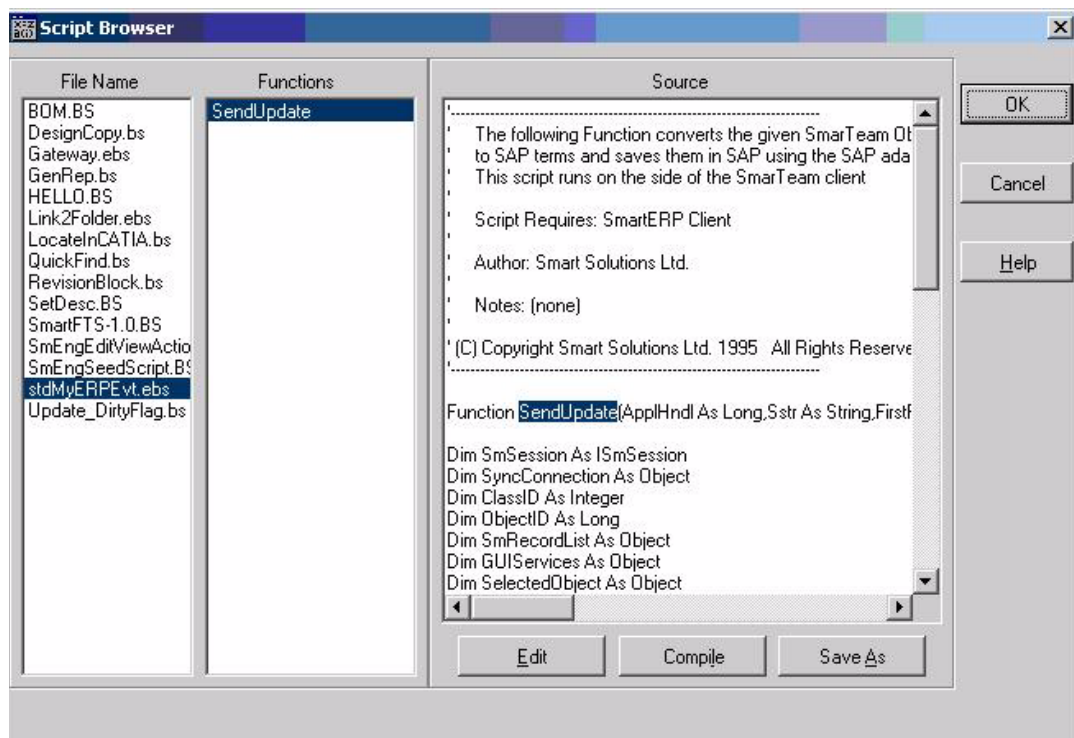
This section uses the script of the previous section to create a SmarTeam user-defined command. Refer to the section User-Defined Operations, Chapter 10 of the SmarTeam – Editor Administration Guide for more details on how to create a SmarTeam user-defined command.

Note: The class you choose for the user-defined command must be the same as the class you type in the Operation Mapping screen in the Gateway Integration Manager.

- 1 Copy the text above and create a file named stdMyERPEvt.ebs located in the \$SmarTeam\script directory.
- 2 Open SmarTeam Administrative Tools/SmartBasic Script Maintenance



- 3 In the Class Tree view, select the **Component** Class. This is the class that will be used in the scenario in section [Defining an Outbound Gateway Operation Mapping](#).
- 4 Select the **User Defined** tab in the **Scripts for operations** area
- 5 In the Operation column, type a name for the operation, for example, Send Update. This is the event name that will be used in the scenario in section [Defining an Outbound Gateway Operation Mapping](#).
- 6 Double click on function name column. The script browser dialog appears.



- 7 In the File Name column, select the script **stdMyERPEvt.ebs**
- 8 Click **Compile**

- 9 Select the **SendUpdate** function in the Functions column,
- 10 Click **Ok**. The Script Maintenance appears as above. The selected function name appears as follows:



- 11 Close Script Maintenance
- 12 Create a button named, for example “Send Update”, to invoke this command in SmarTeam. Refer to SmarTeam Administration Guide for details.

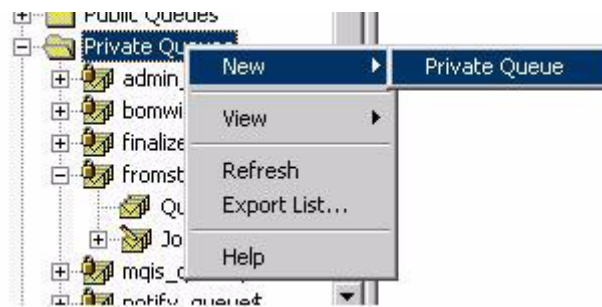
Defining a Microsoft® Messaging Queue

This section shows how to define an MSMQ to work with SmarTeam – Gateway: Please refer to Microsoft Windows documentation for more details.

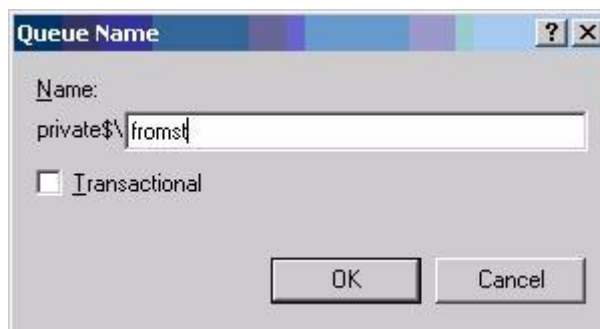
- 1 Navigate to:
Control Panel → Administrative Tools → Computer Management → Services and Applications → Message Queueing → Private Queues



- 2 Right click **Private Queues** and select **New → Private Queue**.



- 3 Make up a name for the queue and type it in the **Name** field.



- 4 Click **Ok**

Chapter 3: Configuring SmarTeam – Gateway

This chapter describes in detail how to configure SmarTeam – Gateway for SmarTeam object data transmission. This is a basic configuration procedure that will be referred to in the scenarios described in subsequent chapters.

SmarTeam – Gateway configuration defines the behavior of SmarTeam object data for a selected SmarTeam object as it leaves and enters the SmarTeam database. It includes the following topics:

- Object specification (schema) configuration – Defining a Gateway object specification (schema) for the selected SmarTeam object class
- Outbound data transmission configuration – Defining a Outbound Gateway Operation Mapping for the selected SmarTeam object class
- Inbound data transmission configuration – Defining allowed SmarTeam functionality for the Inbound SmarTeam object class
- Setting up the Inbound API for user-defined functionality.

Scenarios

This chapter contains four specific data transmission scenarios to illustrate SmarTeam – Gateway configuration:

- 1** Outbound data transmission of a SmarTeam Component object driven by a SmarTeam Update operation event
- 2** Outbound data transmission of a SmarTeam object Assembly driven by a SmarTeam user-defined operation event
- 3** Inbound data transmission of a SmarTeam Component object
- 4** Using SmarTeam – Gateway API for Inbound Connectivity

Scenario 1: Outbound SmarTeam Component Object Data Transmission

This section describes in detail how to configure SmarTeam Gateway for outbound data transmission of a SmarTeam Component object to a MSMQ message destination where the transmission is initiated by a SmarTeam Update operation event.

The configuration includes the major steps.

- Defining an Outbound Gateway Operation Mapping
- Defining a Gateway Object Specification (Schema)

Defining an Outbound Gateway Operation Mapping

This section describes how to specify a Gateway Operation Mapping for the Gateway object, including:

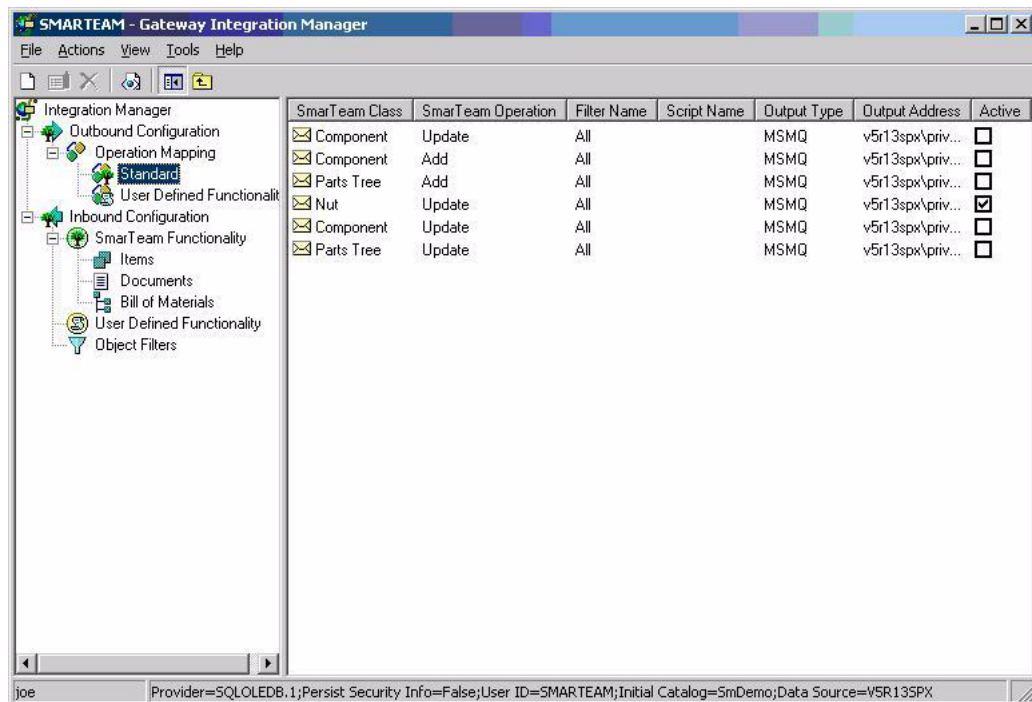
- Setting the Update Component operation event that initiates the SmarTeam Component data transfer from SmarTeam.
- Specifying which object link data will be sent together with the object
- Setting the Gateway XML object message destination

Procedure

- 1 Launch the SmarTeam – Gateway Integration Manager from the Start Menu, as shown below. The SmarTeam – Gateway Integration Manager – main dialog appears.



- 2 The SmarTeam – Gateway Integration Manager – main dialog appears, as below.



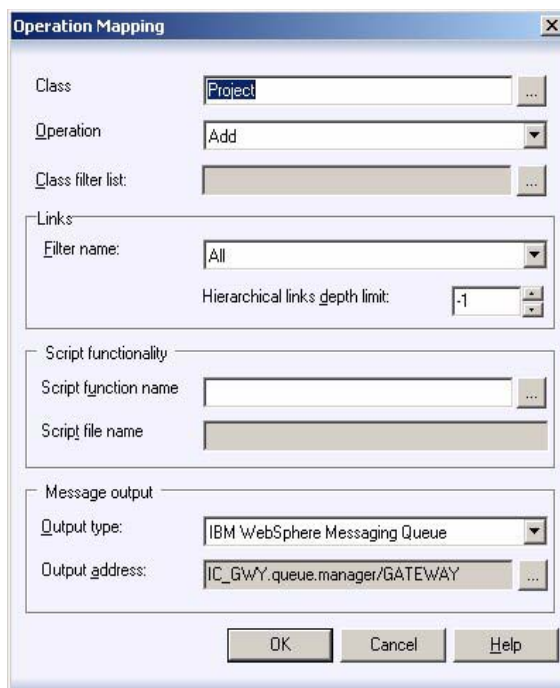
The rows of the right pane of the dialog show the existing Gateway Operation Mappings, characterized by the fields:

Gateway Operation Mapping Field	Description
SmarTeam Class	Class of SmarTeam object to be transmitted
SmarTeam Operation	The SmarTeam operation that initiates the object data transfer
Filter Name	Selects additional link objects to be sent along with the SmarTeam class ("All" denotes all link types)
Output Type	Destination message queue type
Output Address	Destination message queue address

For example, the Operation Mapping in the first row of the figure indicates that when an SmarTeam Update operation is performed on a SmarTeam Component object, that object, together with all of its links, will be transformed into a Gateway XML object and transmitted to the MSMQ queue at the address listed. The operation mapping in the second row indicates that the same thing will occur when a SmarTeam Add operation is performed on a SmarTeam Component object.

The Gateway Operation Mappings need to be different in at least one field and more than one can be active.

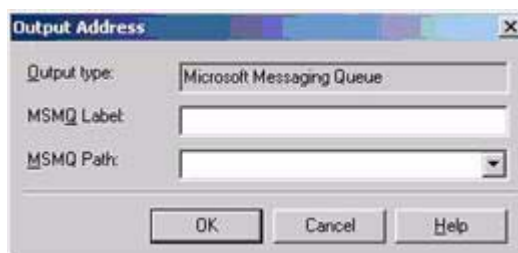
- 3 On the tree on the left pane, select **Operation Mapping/Standard** as shown in the figure above.
- 4 With the cursor on the right pane, right click and select **Add**. The Operation Mapping Dialog appears, as below.



- 5 In the Class name field, click on the ellipses (...) and select **Component**
- 6 In the **SmarTeam Operation** field, select **Update** from the dropdown box
- 7 Set **Hierarchical links depth limit** to **None**. This setting ensures that only the isolated Component object is sent – and not an assembly.
- 8 In the **Output type** dropdown box, select Microsoft Messaging Queue (MSMQ).



- 9 In the Output address field, click on the ellipsis (...). You get the MSMQ Output Address dialog, as below.



- a In the **MSMQ Label** field, assign a label for your messages (for example, SmarTeam Message)
 - b The **MSMQ Path** dropdown displays a set of previously-defined queues; select from the combo box the one you want to use as a transport queue for your operation mapping.
 - c Click **Ok** to return to the Operation Mapping dialog
- 10 Leave all other details as default.

11 Click **Ok**

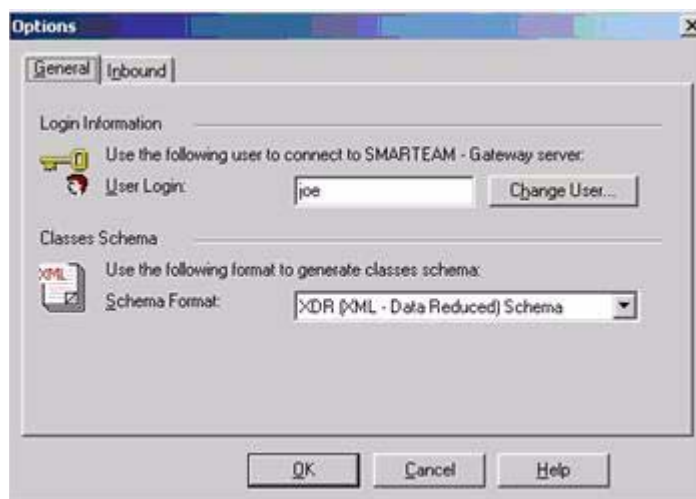
12 The new Gateway Operation Mapping appears on the right pane of the Gateway Integration Manager screen.

Defining a Gateway Object Specification

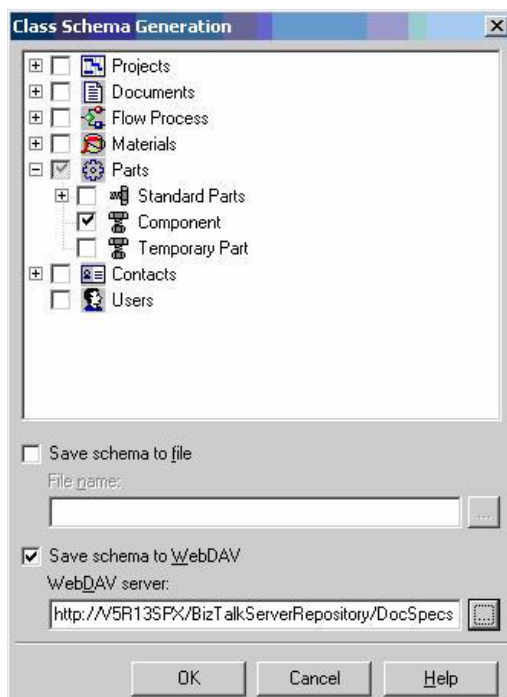
This section describes how to use the SmarTeam – Gateway Integration Manager to create the Gateway specification (schema) for the Smart Gateway Component XML object that is sent from the Gateway server. The specification is tailored to the specific class being sent – the Component class – and has a standard format that is recognized by BizTalk.

Procedure

- 1 If not already launched, launch the SmarTeam – Gateway Integration Manager from the Start menu, as in the previous section.
- 2 From the *Tools* menu of the SmarTeam – Gateway Integration Manager, select **Options**, select the **General** Tab and, in the field **Schema Format**, make sure that the selected type of the schema is **XDR (XML Data-Reduced) Schema**, as shown below, leave all other options in all tabs as default, and click **Ok**.



- 3 From the *Tools* menu of the SmarTeam – Gateway Integration Manager, select **Generate SmarTeam Schema**. The *Class Schema Generation* dialog is displayed, showing the SmarTeam class tree, as below.



- 4 Select the **Component** class, as shown (the **Parts** class is selected automatically as the parent class).

Note: You must check the same class – Component – you used to define the Gateway Operation Mapping in the previous section. You can also check additional classes; the schema can define several classes at once.

- 5 You must save the new schema to WebDAV in order for it to be available to BizTalk. To save the schema to the WebDAV repository, select **Save Schema to WebDAV** and specify the name and location of the WebDAV server in the text box. Click to browse to the server location. Name the file “SmarTeam Component Specification” (see section [Naming Conventions](#)).
- 6 Click **Ok**.

Initiating and Verifying Data Transmission

This section describes how to test the configuration described in the previous sections:

- 1 Perform Update operation on a Component object in SmarTeam – Editor
- 2 Check the MSMQ to see if the message with the label you provided in step 9i in section [Defining an Outbound Gateway Operation Mapping](#) has arrived.
- 3 Purge all test messages from the message queue so that they will not be picked up by the BizTalk server in subsequent scenarios.

Scenario 2: Outbound SmarTeam Assembly Data Transmission

This section describes in detail how to configure SmarTeam – Gateway for outbound data transmission of a SmarTeam object Assembly to a MSMQ message destination where the transmission is driven by a user-defined Gateway event.

The configuration includes the major steps:

Defining an Outbound Gateway Operation Mapping

Defining a Gateway Object Specification

IMPORTANT! This section assumes that a SmarTeam user-defined command has been created that can trigger the user-defined Gateway event which will be defined here. See [User-Defined Gateway Event](#) for general information about user-defined Gateway events and see [Defining a SmarTeam User-Defined Command for a User-Defined Gateway Event](#) for details about how to define such an event.

SmarTeam Assembly

A SmarTeam Assembly is a tree of objects whose classes inherit from the same Superclass. The root object of the assembly represents the assembly in the Gateway configuration process. In our scenario, the root object belongs to the Component class and represents an assembly of Component objects. SmarTeam - Gateway transmits the root object together with the remaining tree to the depth specified in the “Hierarchical links depth limit” field. A SmarTeam BOM is equivalent to a SmarTeam Assembly.

Defining an Outbound Gateway Operation Mapping

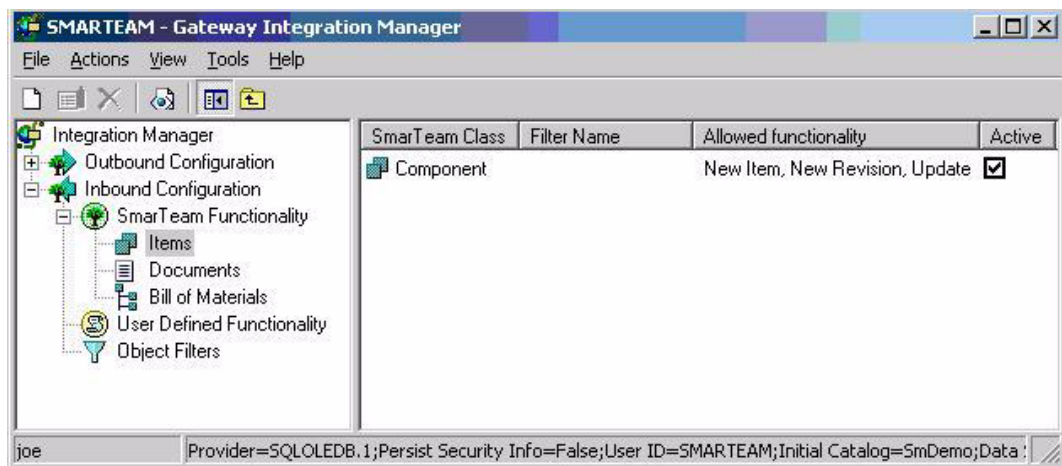
This section describes how to specify a Gateway Operation Mapping for the Gateway object, including:

- 1 Specifying the user-defined Gateway event that initiates the SmarTeam Assembly data transfer from SmarTeam.
- 2 Specifying the depth of the tree to be transmitted
- 3 Setting the Gateway XML object message destination

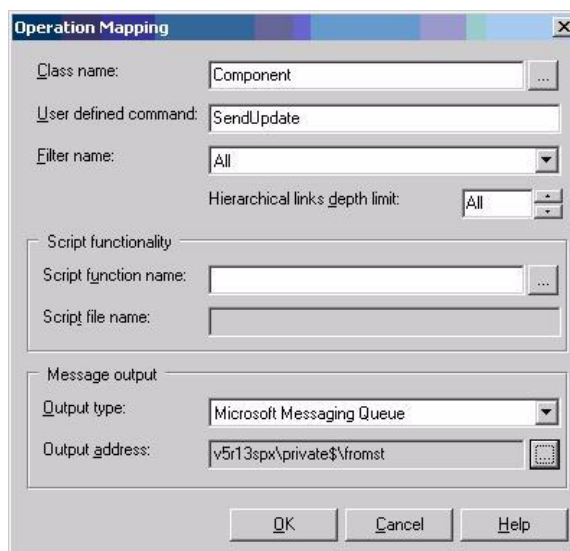
Procedure

- 1 If not already launched, launch the SmarTeam – Gateway Integration Manager from the Start Menu, as in the beginning of the previous scenario.

The SmarTeam – Gateway Integration Manager – main dialog appears, as below.



- 2 On the tree on the left pane, select **Operation Mapping/User Defined Functionality** as shown.
- 3 With the cursor on the right pane, right click and select **Add**. The Operation Mapping Dialog appears, as below. This format of the Operation Mapping dialog is suited for user-defined Gateway events.



- 4 In the Class name field, click on the ellipses (...) and select **Component**. (Component is the root class of the assembly)
- 5 In the **User-defined command** field, type **SendUpdate**.

Note: The string that you type for the User-defined command must be identical to the first parameter of the call to the function `SyncConnection.ReportEventByCommand`, embedded in the script run by the SmarTeam user-defined command that triggers the event. In particular:

- 6 `SyncConnection.ReportEventByCommand "SendUpdate", ClassID, ObjectID`

Note that any other string could have been used instead of `SendUpdate` as long as it is used consistently.

- 7 See *Preparing SmarTeam for Gateway Operations*, section *Defining a SmarTeam User-Defined Command for a User-Defined Gateway Event*, for more information.

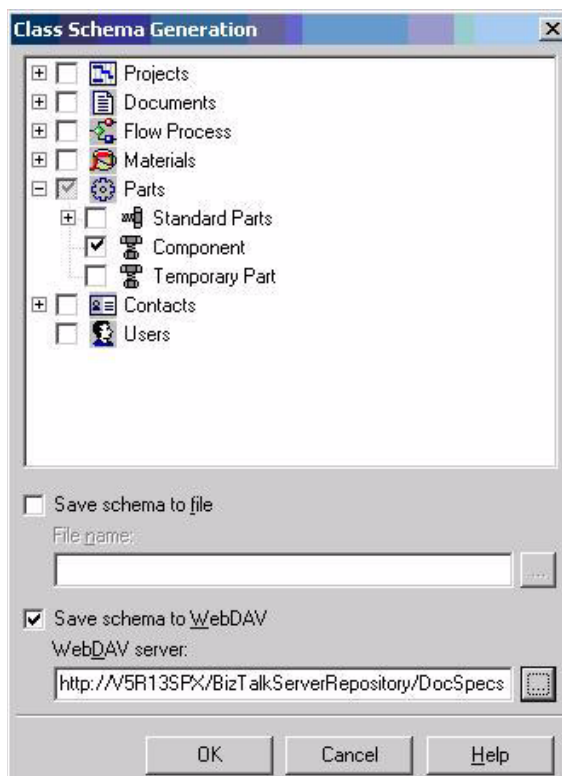
- 8 Set hierarchical links depth to **All**. This setting ensures that SmarTeam – Gateway transmits the entire assembly (tree).
- 9 Leave all other details as default.
- 10 Click **Ok**
- 11 The new Gateway Operation Mapping appears on the right pane of the Gateway Integration Manager screen.

Defining a Gateway Object Specification

This section describes how to use the SmarTeam – Gateway Integration Manager to create the Gateway specification (schema) for the Smart Gateway Assembly XML object that is sent from the Gateway server. The schema is tailored to the specific assembly root class being sent and has a standard format that is recognized by BizTalk.

Procedure

- 1 If not already launched, launch the SmarTeam – Gateway Integration Manager from the Start menu, as in the previous section.
- 2 From the *Tools* menu of the SmarTeam – Gateway Integration Manager, select **Generate SmarTeam Schema**. The *Class Schema Generation* dialog is displayed, showing the SmarTeam class tree, as below.



- 3 Select the **Component** class, as shown

Notes: You must check the same class you used to define the Gateway Operation Mapping in the previous section. You can also check additional classes; the schema can define several classes at once.

If the assembly were composed of objects of more than one class of the same super class you would need to check all the classes.

- 4 You must save the new schema to WebDAV in order for it to be available to BizTalk. To save the schema to the WebDAV repository, select **Save Schema to WebDAV** and specify the name and location of the WebDAV server in the text box. Click to browse to the server location. Name the file “SmarTeam Component Specification”.
- 5 Click **Ok**.

Initiating and Verifying Data Transmission

Refer to [Chapter 3](#), section [Initiating and Verifying Data Transmission](#) for details.

Scenario 3: Inbound SmarTeam Component Object Data Transmission

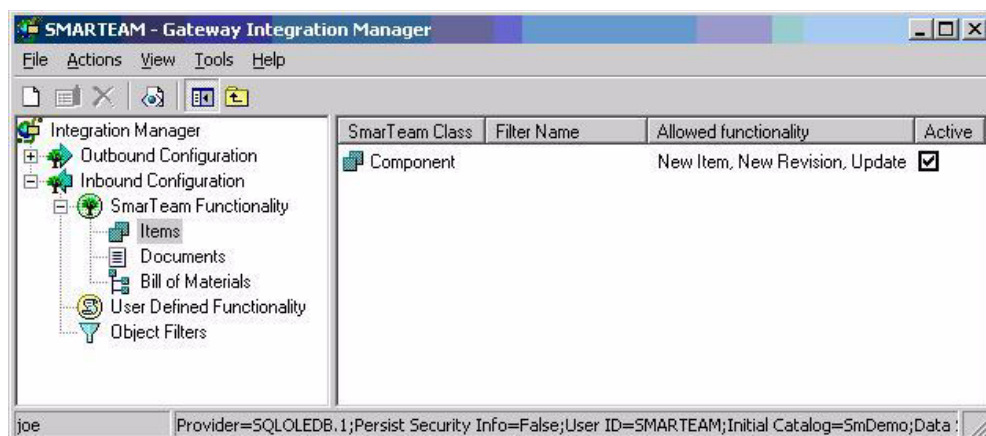
This section describes how to specify SmarTeam functionality for an inbound Gateway object, including:

- 1 Allowed SmarTeam functionality of the Component object in the SmarTeam database.
- 2 Filters to be applied to the Component object.

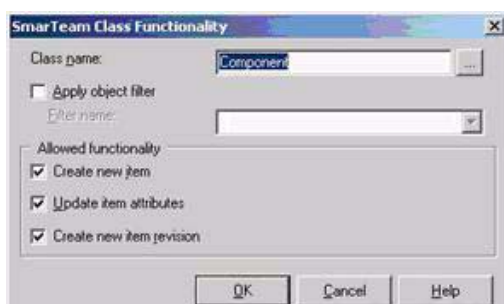
Defining Inbound SmarTeam Functionality

Procedure

- 1 If not already launched, launch the SmarTeam – Gateway Integration Manager from the Start Menu, as in the previous section.
- 2 The SmarTeam – Gateway Integration Manager – main dialog appears, as below.



- 3 On the tree on the left pane select **Inbound Configuration/SmarTeam Functionality/Items**, as shown.
- 4 With the cursor on the right pane, right click and select **Add**. The SmarTeam Class Functionality dialog appears, as below.



The settings on this screen allow you to prevent the Gateway server from writing the received Class object data in the SmarTeam database. If all the Allowed functionality check boxes are checked, the Gateway server will attempt to create a new item, update item attributes or create a new item revision, according to the requirements of the received object. If a check box is not checked, the Gateway server will be prevented from performing that action. The Allowed functionality is described in the following table:

Allowed SmarTeam Functionality of Inbound Object	Description
Create new item	Allow creation of new SmarTeam object from inbound data
Update item attributes	Allow update of existing SmarTeam object from inbound data
Create new item revision	Allow creation of new revision of an existing SmarTeam object from inbound data

- 5 In the Class name field, click on the ellipses (...) and select **Component**.
- 6 Check all check boxes in **Allowed functionality** area. This means that the SmarTeam database will allow all the functionality items listed in the table above.
- 7 Leave all other details as default.
- 8 Click **Ok**
- 9 The new SmarTeam Functionality appears on the right pane of the Gateway Integration Manager screen.

Initiating and Verifying Data Transmission

Refer to [Chapter 3](#), section [Initiating and Verifying Data Transmission](#) for details.

Scenario 4: Using SmarTeam – Gateway API for Inbound Connectivity

You can add functionality to SmarTeam Gateway Inbound processing by using a script activated by the Gateway inbound script hook. For example, you can create and maintain a log file for inbound processing.

Note: Script operations on the SmarTeam database are not restricted by the settings in the SmarTeam Class Functionality screen.

This section discusses:

- How to write the script
- How to activate the script.

Writing a Script

The script calling sequence

```
Function PullTestScript(Session As SmApplic.SmSession, SmObject As
SmApplic.ISmObject) As Integer
has two input parameters:
```

- *Session* is the current SmarTeam session owned by the Gateway server.
- *SmObject* is the inbound Gateway object after its translation into a SmarTeam SmObject. This SmObject is disconnected (not in SmarTeam database).

Sample Script

The following script produces a comparison file that shows the attributes of the inbound object and the attributes of the existing object in the SmarTeam database. It also produces a log file. If the inbound object has a non-zero Object Id, it is assumed in the script that an SmObject with that Object Id exists in the database. In that case, in the script you can either update the object's attributes or create a new revision of the object or not update the database at all. If the inbound object has a zero or negative Object Id, it is assumed that the SmObject does not exist in the database. In this case, you have the option of creating a new object in the database.

The example VB project for this is located at:

CD 2\SmarTeam – Gateway\program files\SmarTeam\SDK\Samples\SmartGateway

The following is the content of the SmGatewayPullDemoScript.cls from the SmGatewayPullDemoScript Microsoft Visual Basic project.

```
Option Explicit
Const OUT_FILE = "C:\Temp\PullDemoScriptOutput.txt"
Const LOG_FILE = "C:\Temp\PullDemoScriptOutput_Log.txt"

Function PullTestScript(Session As SmApplic.SmSession, SmObject
As SmApplic.ISmObject) As Integer
    On Error GoTo PullTestScript_Err

    Dim ObjData As SmRecList.SmRecord 'To hold inbound data
```

```
' Check the validity of the input parameters
If (SmObject Is Nothing) Or (Session Is Nothing) Then
    RaiseError "Invalid input parameters"
End If

' Write records to the log file
Whitetail "Pull Demo Script execution started"
WriteToLog "=====",
OUT_FILE

' If this object exists in SmarTeam, write its database
' attributes and its inbound attributes to the comparison
file OUT_FILE.
If SmObject.ObjectId > 0 Then

    ' Copy inbound object data to a temporary SmRecord object
    Set ObjData = New SmRecList.SmRecord
    ObjData.CopyEx SmObject.Data

    ' Retrieve the attributes of the inbound object from the
    database,
    'overwriting the inbound attributes
    SmObject.Retrieve

    ' Write the database object attributes to the comparison
    file, OUT_FILE
    SmObject.Data.PrintToFile "Existing Object attributes",
OUT_FILE
    ' Write the inbound object attributes to the comparison file
    ObjData.PrintToFile "Inbound Object attributes", OUT_FILE
Else
    ' No ObjectId exists. This is a new object; write its
    attributes to
    ' the comparison file.
    SmObject.Data.PrintToFile "New Object", OUT_FILE
End If

PullTestScript = 0

Exit Function
PullTestScript_Err:
    WriteToLog Err.Description
    PullTestScript = 1
End Function
```

```

Private Sub RaiseError(ByVal Description As String, Optional
    ByVal Number As Integer = 513)
    Err.Raise vbObjectError + Number, "SmartGateway Pull Demo
    Script", Description
End Sub

Private Sub WriteToLog(ByVal Message As String, Optional ByVal
    LogFile = LOG_FILE)
    Open LogFile For Append As #1
        Write #1,
        Write #1, Now
        Write #1, Message
        Write #1,
    Close #1
End Sub

```

Setting the Script Hook

The Gateway inbound script hook is a SmarTeam server side hook. A server side hook is set up in two basic steps:

- Setting the Script name to be activated by SmarTeam Gateway by creating a dummy script in the [Scripts] directory and assigning it in Script Maintenance.
- Creating a COM object to actually execute the script.

For more information see the manual Customizing Using Server-Side Hooks for Server-Based Applications.

These steps are detailed as follows:

- 1 In the [Scripts] directory, create a new script file containing only one dummy function as follows:

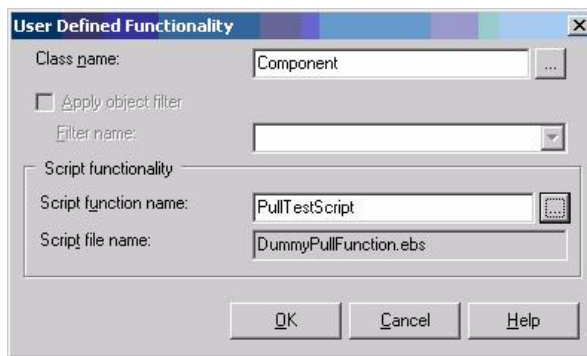
```

Function (Session As SmApplic.SmSession, SmObject As
    SmApplic.ISmObject) As Integer
    PullTestScript = Err_None
End Function

```

Name the script file, for example, DummyPullFunction.ebs.

- 2 In the SmarTeam – Gateway Integration Manager, expand the Integration Manager tree to: **Integration Manager\Inbound Configuration**.
- 3 Right click **User Defined Functionality** and then select **Add**. The User Defined Functionality screen appears.
- 4 Click on the Script function name browser (...). The Script Browser screen appears.
- 5 In the User-Defined tab, in **FileName** column, select DummyPullFunction.ebs, in the **Source** area click **Compile**, in the **Functions** column select PullTestScript and click **Ok**. The User Defined Functionality screen reappears, showing the selected script function name and script file name.



- 6 In the **Class name** field, click the button and select the class for which you want the script hook to apply – in our case the Component class.
- 7 Click **OK**. The defined inbound rule is displayed in the Display and Work area of the Integration Manager.
- 8 Compile the SmGatewayPullDemoScript project using Microsoft Visual Basic 6.0.
- 9 Follow the steps in the manual Customizing Using Server-Side Hooks for Server-Based Applications to register the compiled module as a script hook.

Chapter 4: Integrating with a Remote SmarTeam Site

This chapter describes how to integrate with a remote site for several different data transmission scenarios.

- 1 Send a SmarTeam Component object from a local SmarTeam application to a remote SmarTeam Application where the transmission is driven by a SmarTeam Update Component operation event
- 2 Send a SmarTeam Assembly object from a local SmarTeam application to a remote SmarTeam Application where the transmission is driven by a SmarTeam user-defined operation event

These scenarios include the Gateway configuration scenarios in the previous chapter, as described below in each case.

Integration Architecture

This section shows a full example of the architecture of integrating SmarTeam with a remote SmarTeam site. The example shows the two areas of Gateway operation:

- Gateway tools to configure object data transmission
- Gateway software modules to support actual transmission of object data

Configuring object data transmission

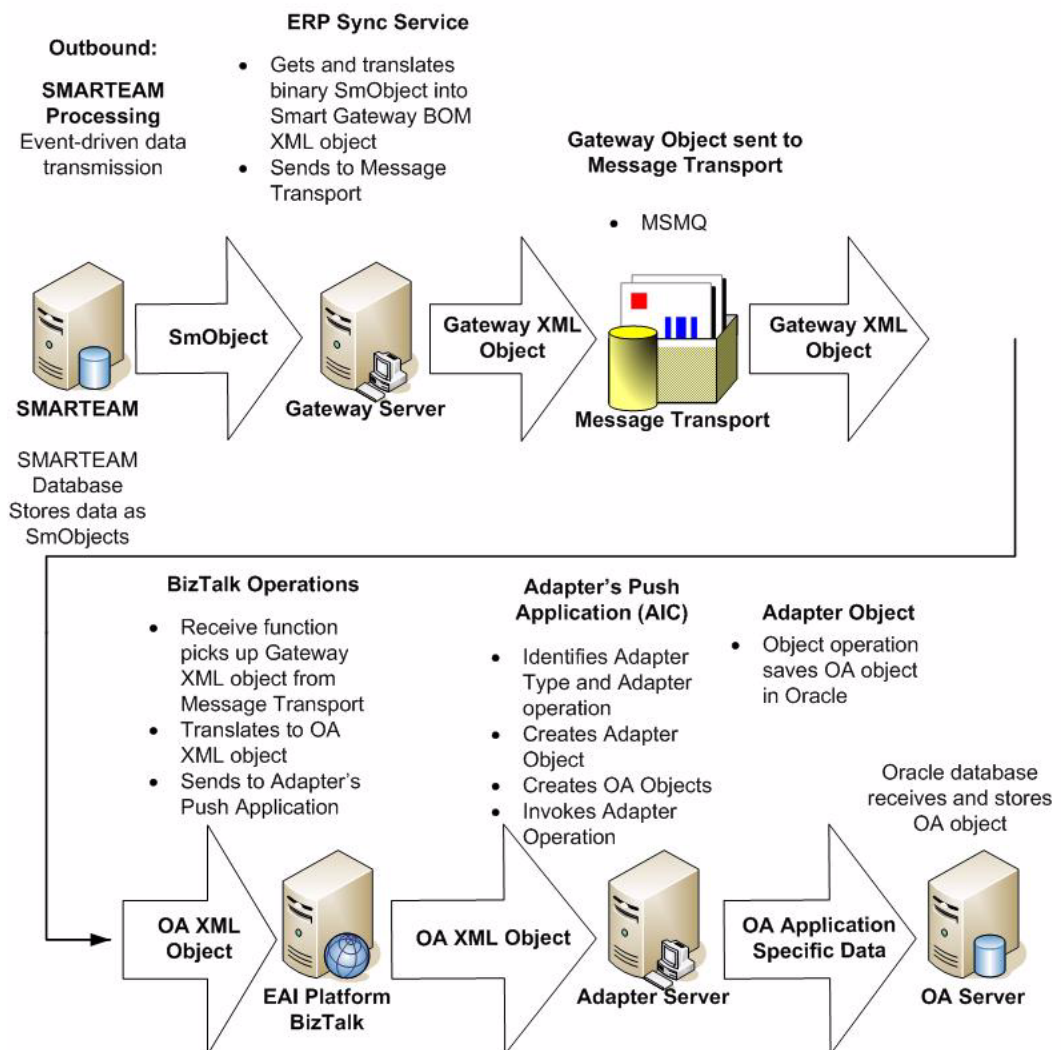
Configuring object data transmission is achieved in three configuration sessions using three tools:

- 1 Using the Gateway Integration Manager tool to configure the properties of outbound and inbound object data to and from the SmarTeam database, including setting the event that drives the data transmission and the schema for the object data XML format.
- 2 Using the Gateway Integration Manager tool to configure the properties of outbound and inbound object data to and from the remote SmarTeam database, including setting the event that drives the data transmission and the schema for the object data XML format.
- 3 Using BizTalk tools to set up the connection between the two XML formats. Note that if the local and remote SmarTeam databases have the same structure, no mapping is necessary.

Operation of Gateway software modules

The two figures below show outbound and inbound data flow for Gateway object transmission respectively.

At each stage the operation of the Gateway software modules is described.



Outbound Gateway Run-Time Operation for SMARTEAM/OA Object Data Transmission

Scenario 1: Sending a SmarTeam Component to a Remote SmarTeam Site

This section describes how to configure the ERP system for data transmission to a remote SmarTeam site.

Configuring this scenario proceeds in three stages:

- 1** Using the Gateway Integration Manager tool to configure outbound and inbound object data to and from the SmarTeam database.
- 2** Using the Gateway Integration Manager tool to configure outbound and inbound object data to and from the remote SmarTeam database.
- 3** Using BizTalk tools to set up the connection between the two XML formats. For SmarTeam to SmarTeam transmission no mapping is necessary.

Configuring Source SmarTeam – Gateway for Outbound Flow

To configure the source SmarTeam – Gateway for outbound flow follow the procedures in [Chapter 3, Scenario 1: Outbound SmarTeam Component Object Data Transmission](#)

Configuring Remote SmarTeam – Gateway for Inbound Flow

To configure the remote SmarTeam – Gateway for inbound flow follow the procedure in [Chapter 3, Scenario 3: Inbound SmarTeam Component Object Data Transmission](#).

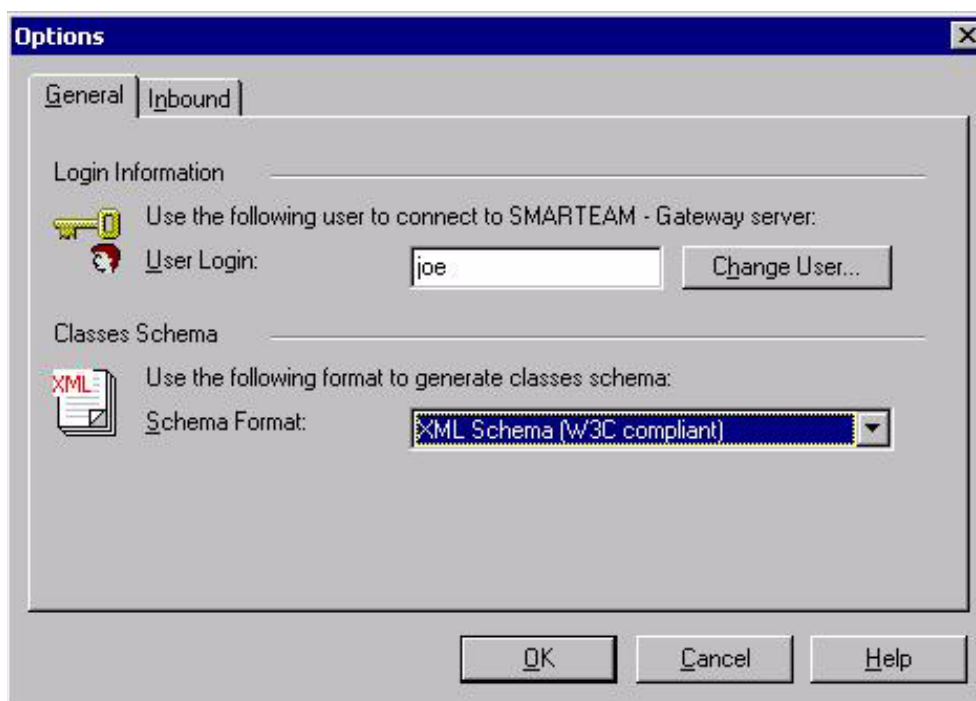
Chapter 5: Integrating with BizTalk 2004/2006

This chapter describes how to use Microsoft BizTalk 2004/2006 as middleware in the ERI integration of SmarTeam with an ERP system, for example, Oracle Applications.

Note: The example shown in chapter 6 is based on Microsoft Visual Studio .Net 2003.

Creating Schemas

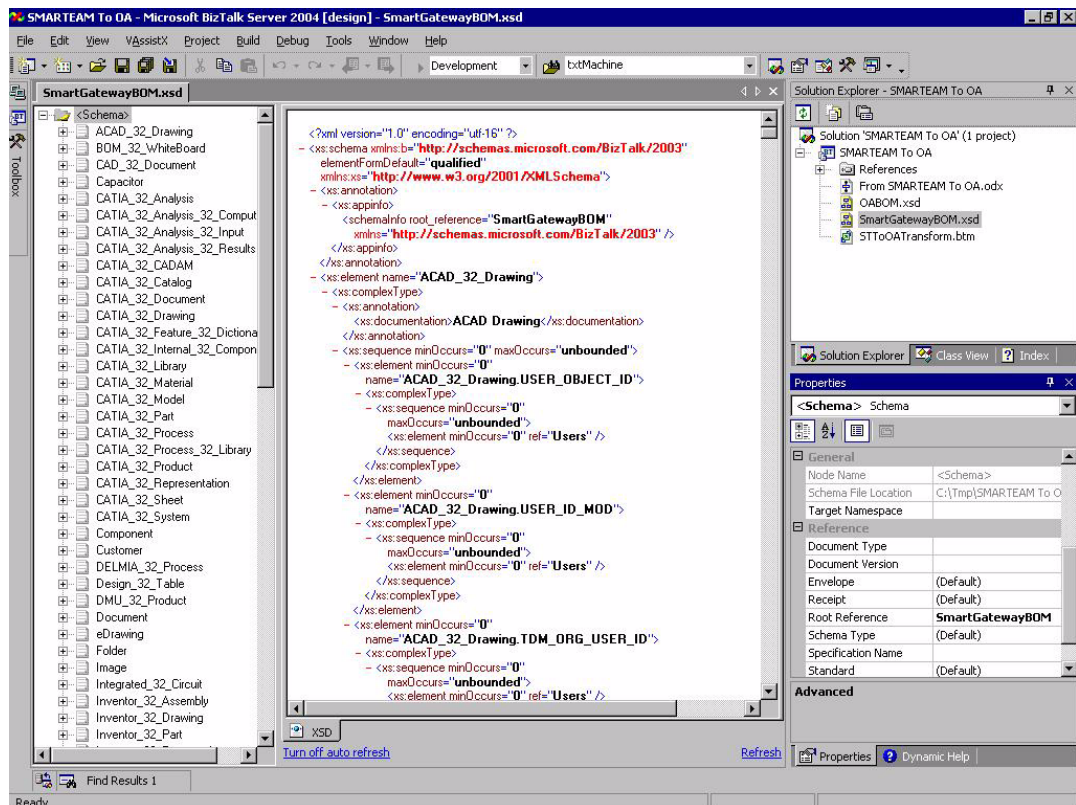
1. Create outbound mapping for Component class as defined in [Chapter 3, Defining an Outbound Gateway Operation Mapping](#) except for the following differences:
 - In the Output type dropdown box, select **SmarTeam – Gateway Remoting Service**
 - In properties of the Remoting Service, provide the network name or IP of the BizTalk host machine.
2. Create a Gateway object specification (schema) for the Component class as defined in [Chapter 3, Defining a Gateway Object Specification](#) except for the following differences:
 - From the Tools menu of the SmarTeam – Gateway Integration Manager, select **Options**, select the **General** tab and, in the field Schema Format, make sure that the selected type of the schema is **XML schema (W3C-compliant)**, as shown below, leave all other options in all tabs as default, and click **OK**.



- Store the schema as a file in the designated folder in order for it to be available to BizTalk. To save the schema to the designated folder, select **Save Schema to File** and specify the name and location of the schema file in the text box. Name the file **SMARTGatewayBOM.XSD**.

Load Schemas into a Microsoft Visual Studio Project

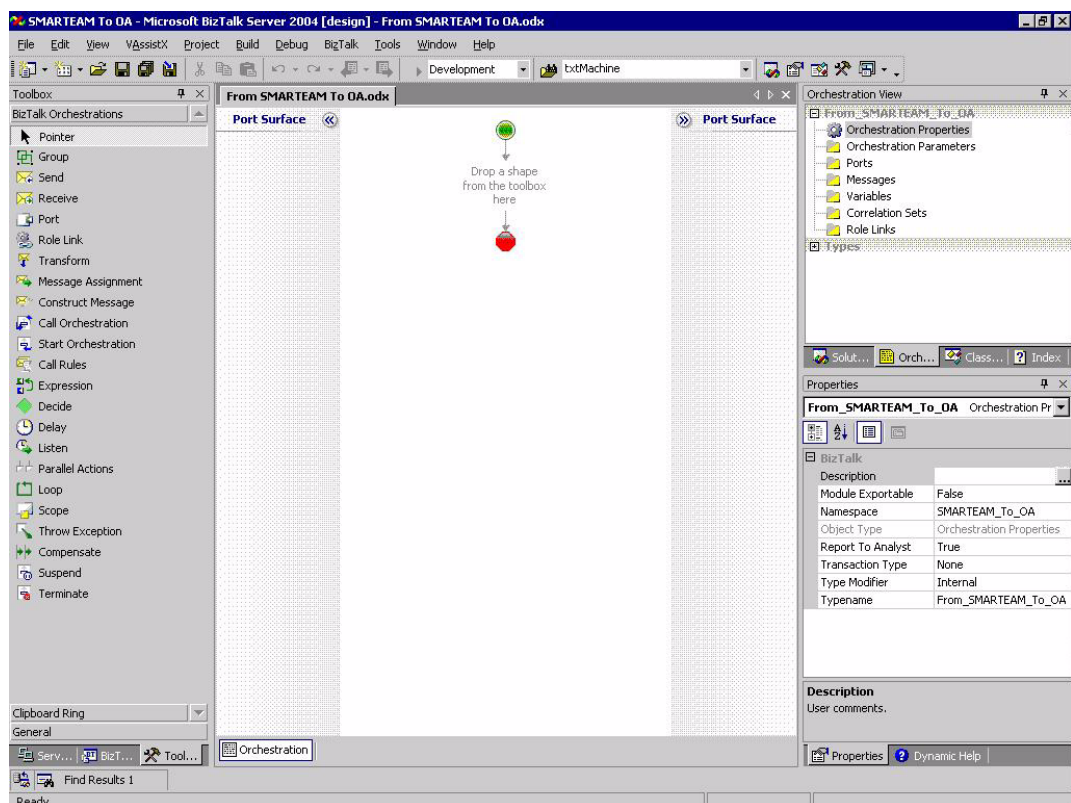
1. Open Microsoft Visual Studio .NET (prerequisite to BizTalk installation).
2. Browse to **File/New/Project**.
3. In the Project Types tree, select **BizTalk Projects** node and choose the **Empty BizTalk Server Project** project in the Templates area.
4. Provide a name, "SmarTeam To OA", location (designated directory) and solution name, "SmarTeam To OA", for the project.
5. Open the Solution Explorer window and right click on the project, select **Add/Add Existing Item**.
6. Browse to the designated folder in which you saved the schema files, select the file **SMARTGatewayBOM.XSD** and click **OK**. SMARTGatewayBOM.XSD appears in the Project Tree.
7. Double click on **SMARTGatewayBOM.XSD**. The schema editor window appears showing the schema contents and schema hierarchy.
8. Select the **<Schema>** node, right click and select **Properties**. The Properties window appears.
9. Set the Root Reference property to **SartGatewayBOM**. The Schema Editor appears with the above entries as below.



10. Save the Schema.
11. Repeat steps 5 – 10 for the schema OABom.XSD (choose **Oracle__Applications**).

Orchestration

1. Right click on **Project in the Solution Explorer** window and select **Add/Add New Item**
2. In the Templates area, select the **BizTalk Orchestration** icon and click **Ok**.
3. In the Name field, provide the name for the Orchestration, **From SmarTeam to OA.odx** and click **Ok**. The Orchestration Designer window appears, as below.



Messages

1. In the Orchestration View window, right click on the **Messages node** and select **New Message**.
2. In the Properties window, provide an identifier, "STMessage" and Message Type (Using the dropdown menu, select **schemas/SmarTeam_To_OA.SmartGatewayBOM**) for the message.
3. In the Orchestration View window, right click on the Messages node and select **New Message**.
4. In the Properties window, provide an identifier (OAMessage) and Message Type (Using the dropdown menu, select **schemas/SmarTeam_To_OA.OABOM**) for the message.

Receive Node

1. From the ToolBox drag and drop the Receive item to the Orchestration Flow window.

2. In the Properties window, set **Activate** property to **true**, set **Message** property to **STMessage** and set **Name** to **ReceiveFromST**.

Receive Port

1. From the ToolBox drag and drop the Port item to the Left Port Surface. The Port Configuration Wizard appears.
2. Provide a name for the Port: **STPort** . Click **Next**.
3. In the Port Type selection screen, choose **Creating a New Port Type** and call the Port Type **STPortType**. Leave all other options as default. Click **Next**.
4. In the Port Binding dialog, leave all options as default (they will be set later). Click **Next**, then **Finish**.
5. In the STPort Icon, highlight **Operation_1** and in the Properties window set Identifier to **ReceiveFromSTOp**.
6. From the STPort Icon, drag the port connector to connect to the ReceiveFromST receive node.

Send Node

1. From the ToolBox, drag and drop the Send item into the Orchestration Flow window.
2. In the Properties window, set Message property to **OAMessage** and set Name to **SendToOA**.

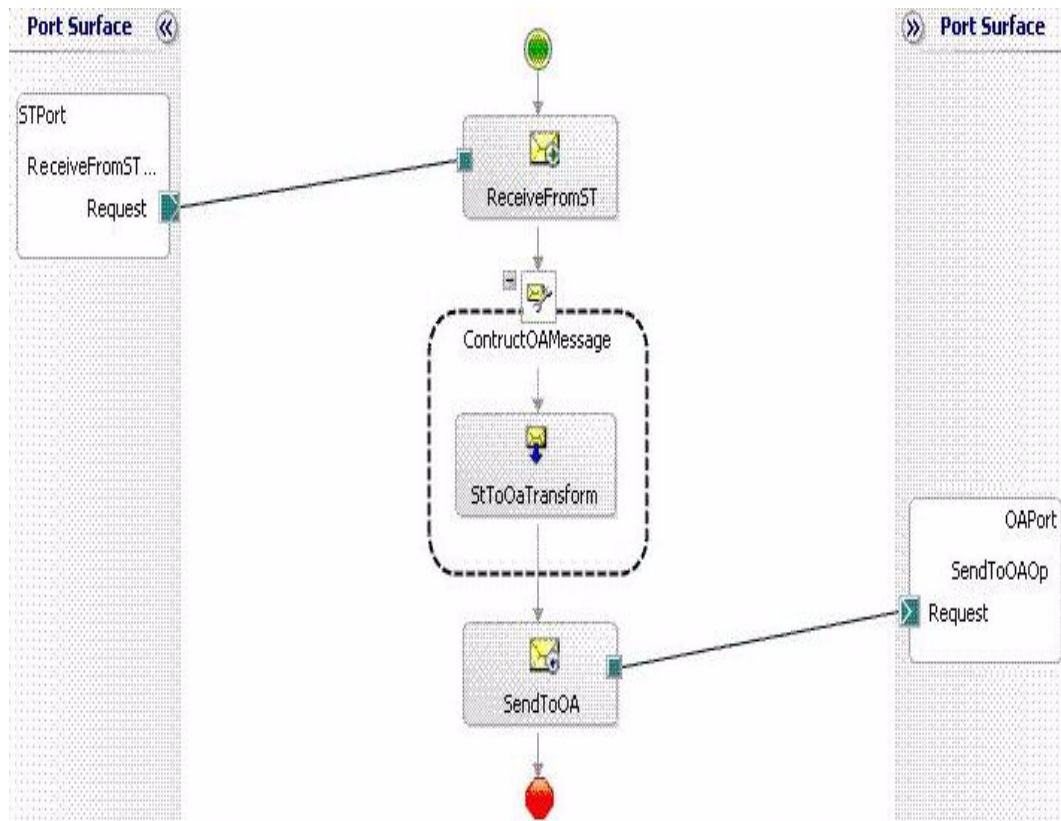
Send Port

1. From the ToolBox drag and drop the Port item to the right Port Surface. The Port Configuration Wizard appears.
2. Provide a name for the Port: **OAPort** . Click **Next**.
3. In the Port Type selection screen, choose **Creating a New Port Type** and call the Port Type **OAPortType**. Leave all other options as default. Click **Next**.
4. In the Port Binding screen, in Port direction of communication, select **I'll always always be sending messages on the port**. Click **Next**, then **Finish**.
5. In the OAPort Icon, highlight **Operation_1** and in the Properties window set Identifier to **SendToOAOp**.
6. From the OAPort Icon, drag the port connector to connect to the SendToOA receive node.

Mapping

1. From ToolBox drag and drop Transform Item between the orchestration nodes ReceiveFromST and SendToOA.
2. In the Properties window, click on the **Ellipses** button in front of the Input Messages property. The Transform Configuration dialog appears.
3. In the Transform Configuration dialog, set the Fully Qualified Map Name to **SmarTeam_To_OA.STToOATransform**.
4. Set Transform Source to **STMessage** and Transform Destination to **OAMessage**, Make sure the option **When I click OK, launch the Biztalk Mapper** is checked and click **OK**. The BizTalk Mapper appears.

5. Go back to the Orchestration Design window and select the node **Transform_1** and in Properties window set Name to **StToOATransform**
6. Select the node **Construct Message** and in the Properties window, set the name to **ConstructOAMessage**. The Orchestration should now appear as below.



Set Strong Name Key

1. Execute **Start/ Programs/Microsoft Visual Studio.NET 2003/Visual Studio.NET Tools/Visual Studio .NET 2003 Common Prompt**.
2. Change current directory to the project's directory and from there execute the following command:

```
sn -k SmarTeam_To_OA.snk
```

3. Make sure that the output contains the following text: **Key pair written to SmarTeam_To_OA.snk**. In case the output contains any error records, check the syntax of the command. The strong name key is now located in the project's directory.
4. Open the project properties by clicking **Project/SmarTeam To OA Properties**. The SmarTeam To OA Properties Pages window appears.
5. Select the **Assembly** node on the left pane and set the Assembly Key File property to the path to the previously created file **SmarTeam_To_OA.snk**.

Build and Deploy Solution

1. Build the solution by selecting the menu item Build/Build Solution. The solution build progress messages are displayed. Make sure that the "Build Complete" line has 0 errors and the summary section reports: "Build: 1 succeeded, 0 failed".
2. In case the build fails, analyze the output and recheck the previous steps.
3. Go to the **Build** menu and click **Deploy SmarTeam To OA**.
4. Make sure the output contains the following information: **Deploy: 1 succeeded, 0 failed, 0 skipped**. In case of deployment failure, analyze the output and recheck the previous steps.

Create Send Port

1. Open **BizTalk Explorer** window and expand the tree of the BizTalk Server instance, where Orchestration was deployed
2. Right click on the **Send Ports** node and click on **Add Send Port...** and the Create New Send Port dialog appears.
3. Select **Static One-Way** Port from the combo box and click **OK**. The Static One-Way Send Port Properties - Configurations window appears.
4. Set the Name field to **SendToGatewayAdapter**.
5. In the General properties area, set the Transport Type to **SmarTeam**.
6. Select **Address (URL)** property and click on the **Ellipsis** button (...). The SmarTeam Transport Properties window appears.
7. Set the Messaging Delivery Destination to **SmarTeam – Gateway Adapter**. The Port transport properties now appear as below:

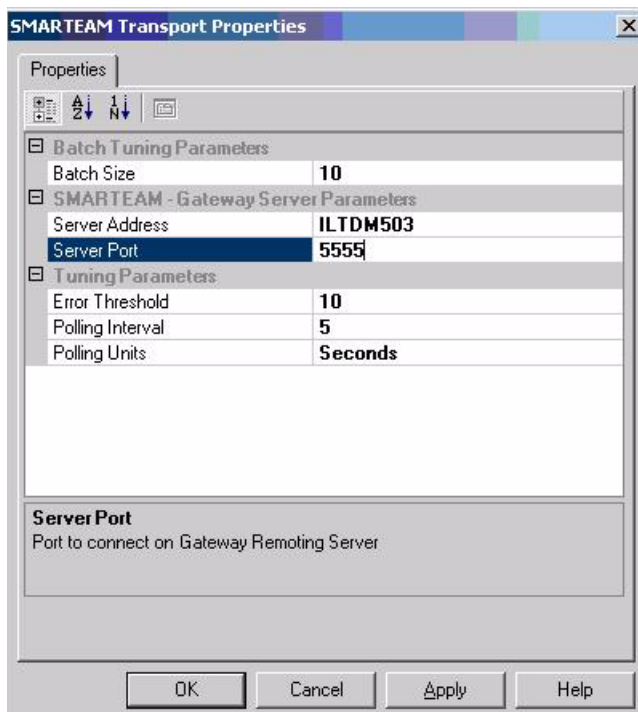


8. Click **OK**

9. In the Static One-Way Send Port Properties - Configurations window, in the left pane, select **Configurations/Send/General** and in the right pane, set the Send Pipeline property to **Microsoft.BizTalk.DefaultPipelines.XMLTransmit** and click **OK**.

Create Receive Port

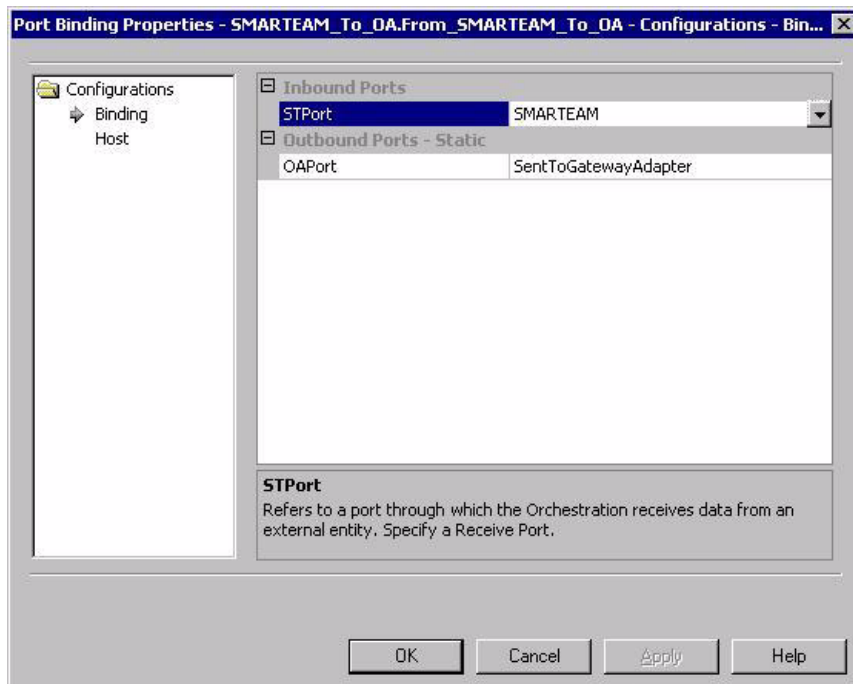
1. Right click on the **Receive Ports** node and click on **Add Receive Port...** and the Create New Receive Port dialog appears.
2. Select **One-Way Port** from the combo box and click **OK**. The One-Way Receive Port Properties - Configurations - General window appears.
3. Set the Name field to **SmarTeam** and click **OK**.
4. In the BizTalk Explorer window, select **Receive Ports/SmarTeam** and right click on **Receive Locations** and select **Add Receive Location...** The Receive Location Properties - Configurations - General window appears
5. Set the Name to **SmarTeamRemotingService**.
6. In the General properties area, set the Transport Type to **SmarTeam**
7. Select **Address (URL)** property and click on the **Ellipsis** button (...). The SmarTeam Transport Properties window appears.
8. Set the Server Address to **Network Name** or **IP** of the machine where SmarTeam Gateway server was installed
9. Set the Server Port property to the value to which the SmarTeam – Gateway Server port is set. The Port transport properties now appear as below:



10. Click **OK**.
11. The Receive Location Properties - Configurations - General window. Set Receive Handler property to **BizTalkServerApplication** and ReceivePipeline to **Microsoft.BizTalk.DefaultPipelines.XMLReceive** and click **OK**

Final Installation Procedures

1. In the BizTalk Explorer window, expand the Orchestrations node and select the Orchestration created in the previous steps: **SmarTeam_To_OA.From_SmarTeam_To_OA**. Right click and select **Bind ...**. The Port Binding Properties dialog appears.
2. Select **Configurations/Binding** in the left pane and in the right pane set the STPort property to **SmarTeam** and OAPort to **SendToGatewayAdapter**.



3. Select Configurations/Host in the left pane and in the right pane set the Host property to BizTalkServerApplication (the Host property value depends on your installation and may differ for each particular case.) Click **OK**.
4. In the BizTalk Explorer window left pane, right click on **SmarTeam_To_OA.From_SmarTeam_To_OA** and select **Enlist**.
5. Right click on **SmarTeam_To_OA.From_SmarTeam_To_OA** and select **Start...**. The BizTalk Explorer - Express Start dialog appears.
6. Leave all default preferences unchanged and click **OK**.

The BizTalk server is now configured with your Orchestration to receive and transmit messages from SmarTeam to the Oracle Application.