Oracle® Communications MetaSolv Solution

Flow-through Packages Guide

Release 6.2.1

E48909-01

August 2014

Oracle Communications MetaSolv Solution Flow-through Packages Guide, Release 6.2.1

E48909-01

Contents

3 Activation API Data Schema

4 Planning For POTS Activation

5 Planning For DSL Activation

6 Viewing The POTS/DSL Flow-through Package

7 Troubleshooting

Glossary

Preface

This guide that contains the information and procedures you need to install and work with the flow-through packages provided by Oracle as an example of how to integrate Oracle Communications MetaSolv Solution with Oracle Communications ASAP for flow-through activation.

This guide includes some information on third-party software products used by the flow-through packages. However, this information is limited to information needed to install and work with the flow-through packages. If you need additional information on a third-party software application, consult the documentation provided by the product's manufacturer.

Audience

This guide is for you if you are a system architect, software engineer, or a systems integrator who is looking for information on how to integrate MetaSolv Solution and ASAP for a specific telecom service such as POTS or DSL. This guide assumes that you have a working knowledge of MetaSolv Solution, ASAP, Windows NT, UNIX, WebLogic Server, and Java.

Related Documents

For more information, see the following documents in Oracle Communications MetaSolv Solution 6.2.1 documentation set:

- *MSS Planning Guide*: Describes information you need to consider in planning your MetaSolv Solution environment prior to installation.
- *MSS Installation Guide*: Describes system requirements and installation procedures for installing MetaSolv Solution.
- *MSS System Administrator's Guide*: Describes post-installation tasks and administrative tasks such as maintaining user security.
- *MSS Operational Reports*: Provides an overview on using Operational Reports and Business Objects XI within MetaSolv Solution, and procedures for running reports, updating universes, and simple maintenance. It also includes a detailed description of the purpose and functionality of each report available in Operational Reports, as well as a Frequently Asked Questions (FAQ) section that addresses key questions regarding the software.
- *MSS Database Change Reference*: Provides information on the database changes for the MetaSolv Solution 6.2.1 release. Database changes for subsequent maintenance releases will be added to this guide as they are released.

- *MSS Network Grooming User's Guide*: Provides information about the MSS Network Grooming tool.
- *MSS Address Correction Utility User's Guide*: Provides information about the MSS Address Correction utility.
- *MSS Technology Module Guide*: Describes each of the MetaSolv Solution technology modules.
- *MSS Data Selection Tool How-to Guide*: Provides an overview of the Data Selection Tool, and procedures on how it used to migrate the product catalog, equipment specifications, and provisioning plans from one release of your environment to another.
- *MSS CORBA API Developer's Reference*: Describes how MetaSolv Solution APIs work, high-level information about each API, and instructions for using the APIs to perform specific tasks.
- *MSS Custom Extensions Developer's Reference*: Describes how to extend the MetaSolv Solution business logic with custom business logic through the use of custom extensions.
- *MSS XML API Developer's Reference*: Describes how to integrate MetaSolv Solution with other Oracle products, or with external applications, through the use of APIs.

For step-by-step instructions for tasks you perform in MetaSolv Solution, log into the application to see the online Help.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

1 Getting Started

This chapter introduces the Oracle Communications MetaSolv Solution flow-through package concept, and describes how a flow-through package fits into the overall architecture of MetaSolv Solution and Oracle Communications ASAP.

Assumptions

The samples provided in this document assume that you have MetaSolv Solution with the XML API Option and ASAP installed. It also assumes that you have Oracle WebLogic Server installed to support MetaSolv Solution and ASAP.

About the Flow-Through Package

A flow-through package includes:

- A sample integration that demonstrates how to integrate MetaSolv Solution and ASAP. The files include workflows that control events, order management, and data transformation.
- Documentation that provides best practice information on setup and configuration of both products.

The flow-through package demonstrates how to set up a MetaSolv Solution order (POTS or DSL) so that it can activated on the network automatically by ASAP. The files are intended as a learning tool to make integration easier whether you already know Oracle products or whether you are using an Oracle product for the first time.

The MetaSolv Solution installation provides a flow-through package that demonstrates POTS and DSL. Every MetaSolv Solution service that can be activated using ASAP has a representative example for integrating these two Oracle products.

Differences in business processes between companies mean that no single example integration will work in every case. The sample files should be used as an example of one way to integrate the products rather than a step-by-step process that must be followed exactly. The documentation clearly indicates each step in an example that Oracle considers a best practice.

What a Flow-Through Package Demonstrates

The following types of information are available through the sample code or this document:

- MetaSolv Solution setup information to collect and send the right data for ASAP activation

  - Product catalog setup in MetaSolv Solution
  - Setting up gateway events
  - Workaround solutions that capture data required for ASAP
- Workflows in WebLogic Workshop that create a sample XML API integration with ASAP
- WebLogic Server setup for communication between products
  - Foreign JMS server
  - Authentication credentials
  - WLI Message Broker channels
- Data mapping between MetaSolv Solution and ASAP

Process Used to Create the Flow-Through Packages

The following steps show the process used to create the sample integration between MetaSolv Solution and ASAP.

1. Plan the process.
   a. Modeling of ASAP CSDL commands and the specific parameters required to activate service for a specific type of order (for example, POTS) on a specific network element (for example, a Lucent 5ESS switch).

      For information on this process, see the ASAP documentation available on the Oracle software delivery website.

   b. Identify information in MetaSolv Solution that maps to the ASAP CSDL parameters.

   c. Complete a gap analysis to determine information required by ASAP for activation not currently collected by MetaSolv Solution.
2. Set up the ASAP environment.
   a. Install Oracle WebLogic Server 10.3.1

   b. Create and configure an ASAP WebLogic domain.

   c. Install and deploy ASAP.

   d. Deploy the ASAP cartridge responsible for activating service.
3. Set up the MetaSolv Solution environment.
   a. Install Oracle WebLogic Server 10.3.1 for MetaSolv Solution.

   b. Create and configure a MetaSolv Solution WebLogic domain.

      See *MetaSolv Solution Installation Guide* and *MetaSolv Solution System Administrator's Guide* for information on configuring a domain.

   c. Complete a standard installation and deployment of MetaSolv Solution with the XML API option.

      See the *MetaSolv Solution Installation Guide* for more information.

   d. Install the MetaSolv Solution flow-through package.

      See "Setting Up the Flow-Through Package" for more information.
4. Configure communication between MetaSolv Solution and ASAP.

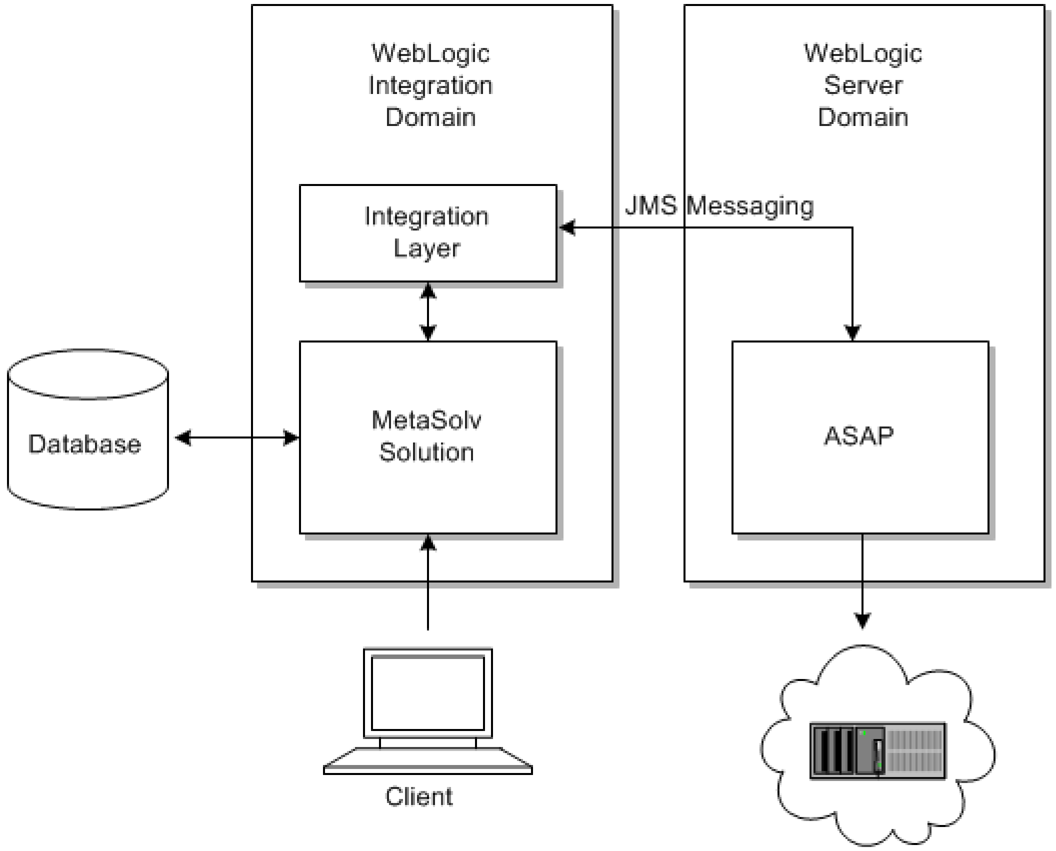
   a. Set up the Foreign JMS server in the MetaSolv Solution domain.
   b. Create authentication credentials.
   c. Create WebLogic Integration channels.
5. Set up the following areas in MetaSolv Solution to provide data required by ASAP for activation:
   a. Product specification
   b. Product catalog
   c. Provisioning plan
   d. Gateway event
   e. Equipment
   f. Custom attributes
   g. Special setup in the user interface required by ASAP because of vendor or other requirements.
6. Create integration workflows in WebLogic Workshop.
   a. Create .jpd files that define the steps in the integration process for XML API.
   b. Create .xq files that map MetaSolv Solution order data to ASAP data.
   c. Create .jcx files for communication.
7. Test integration workflows.

Implementation Architecture

Oracle recommends setting up MetaSolv Solution and ASAP in separate WebLogic domains located on separate machines.

Figure 1–1 shows the recommended hardware and software configuration.

Figure 1–1 Flow-Through Implementation



For complete information on installing MetaSolv Solution and ASAP, refer to the following documents:

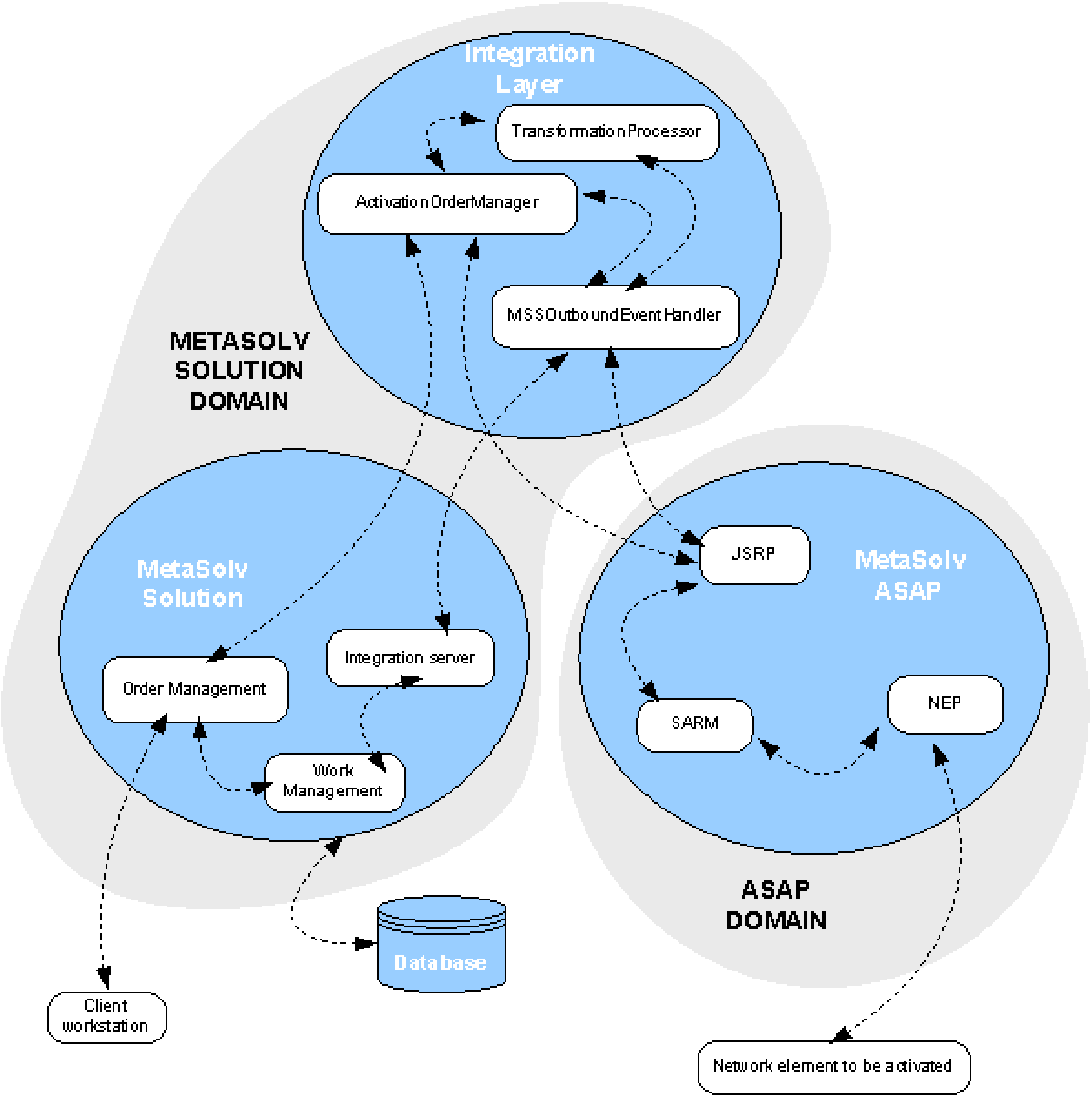
- *ASAP Installation Guide*
- *MetaSolv Solution Installation Guide*

These documents are available on the Oracle software delivery website.

Data Flow for the Sample Integrations

Figure 1–2 shows how order data flows from MetaSolv Solution to ASAP and back to MetaSolv Solution for the automatic activation of an order.

Figure 1–2 Data Flow for Flow-Through Activation



1. A CSR enters an order for service in Order Management in MetaSolv Solution and assigns a provisioning plan for the order.
2. The order is processed, and the provisioning plan tasks (which includes an activation task for turning up service) are set up in a Work Management queue.
3. The Activate task in Work Management is completed, which triggers an activation gateway event to the Integration server.
4. The Integration server routes the gateway event to the MSSOutboundEventHandler in the integration layer.
5. MSSOutBoundEventHandler detects the activation gateway event and sends the appropriate message to the ActivationOrderManager.

6. The ActivationOrderManager associates the appropriate workflow with the service requested on the order, and sends the request to the TransformationProcessor to transform it into the MetaSolv Solution format.
7. The TransformationProcessor returns the transformed request data to the ActivationOrderManager, which sends the request to MetaSolv Solution.
8. MetaSolv Solution responds by processing the request and sending the data back to the ActivationOrderManager through the Activation API.
9. The ActivationOrderManager sends the response to the TransformationProcessor where the data is transformed from MetaSolv Solution format to the MetaSolv Information Model (MIM format) and returned.
10. The ActivationOrderManager decomposes the exported MetaSolv Solution data and maps the relevant portions into the appropriate ASAP CSDL commands for the service being requested.

    The flow-through package that this document describes has a single MetaSolv Solution order to a single ASAP order. This was done to simplify the order for demonstration purposes. In everyday practice, there can be many ASAP activation orders resulting from a single MetaSolv Solution order for service.
11. The ActivationOrderManager sends the newly mapped ASAP CSDL request data to the TransformationProcessor where the data (which is in MIM format) is mapped into the XML format accepted by ASAP.

    Key sequences are assigned for the order to ensure that ASAP's return data is correctly matched to its original request.
12. The TransformationProcessor returns the activation request data to the ActivationOrderProcessor, which sends the data to the Java Service Request Processor (JSRP) module in ASAP.
13. In ASAP, the JSRP module validates the request and sends a message to the MSSOutboutEventProcessor that the request is good.
14. The message is relayed to the Integration server in MetaSolv Solution which updates the activation task in Work Management to Processing.

    A method is available that allows you to update the Events tab on the Work Management window as event messages are received from ASAP. The XSD for the method is **XmlOrderManagementEvents.xsd**.
15. The JSRP maps the order CSDL parameters to the appropriate CSDL commands and sends the order to the Service Activation Request Manager (SARM)
16. In the SARM, the order data from the JSRP that contains the CSDL commands and parameters is decomposed into lower-level ASDL commands that describe the exact tasks to be performed.
17. The SARM sends the order data (in ASDL commands) to the Network Element Processor (NEP).
18. ASAP executes the order, locates the network element, and activates service.
19. An event is sent from the ASAP JSRP to MSSOutboundEventProcessor indicating that the activation request was successful.
20. The event processor sends the event information to the Integration server.
21. The Integration server updates the activation task in Work Management to Complete.

22. The GUI is updated to show the status of the activation task in Work Management for the workstation.

Event Handling

Activation requires handling events from MetaSolv Solution and ASAP. A MetaSolv Solution gateway event triggers the activation process, and events from ASAP report progress back to MetaSolv Solution on activation results.

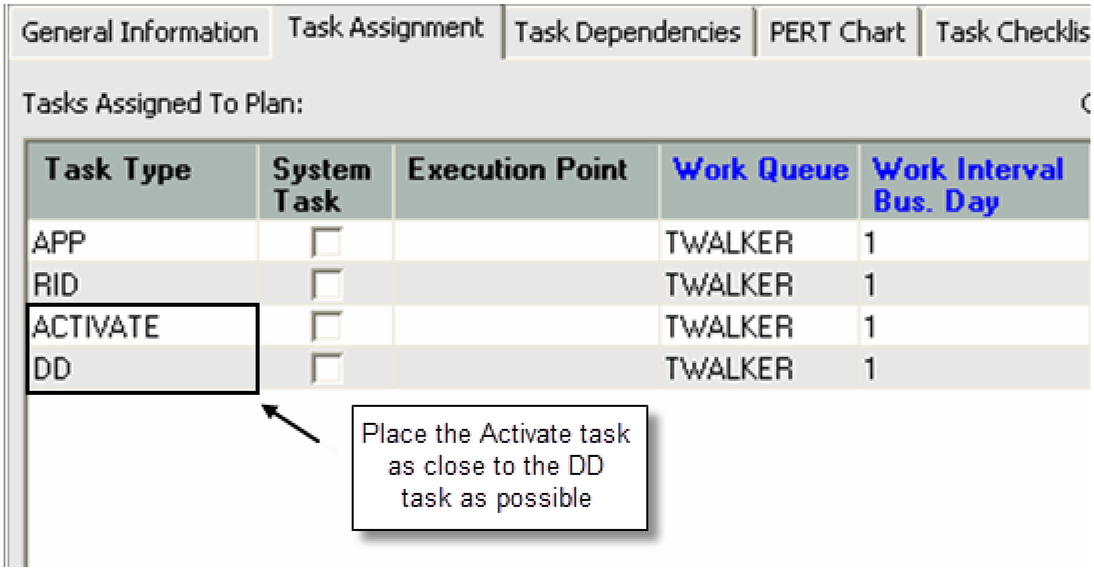
About MetaSolv Solution gateway events.

A MetaSolv Solution gateway event can be triggered at the order-level or at the service-item level. An order can have an order-level or service-level event associated with it, but it cannot have both. Almost all gateway events that trigger activation are fired at the order level. POTS is the only service that can have either order-level and service-item-level gateway events. Features for a line, such as call waiting or call forwarding would use a service-item-level gateway event. Hunt groups are not supported by service-item-level gateway events.

For example, POTS can require a service-item-level gateway event for each telephone number. Therefore, if an order requests activation of 10 telephone numbers, each number triggers a separate gateway event and data is sent to ASAP for each separate telephone number. DSL, on the other hand, requires order-level activation. A DSL order triggers a single order-level gateway event for activation.

Figure 1–3 shows an example of an Activate task.

Figure 1–3 Activate Task



Task Type	System Task	Execution Point	Work Queue	Work Interval Bus. Day
APP	□		TWALKER	1
RID	□		TWALKER	1
ACTIVATE	□		TWALKER	1
DD	□		TWALKER	1

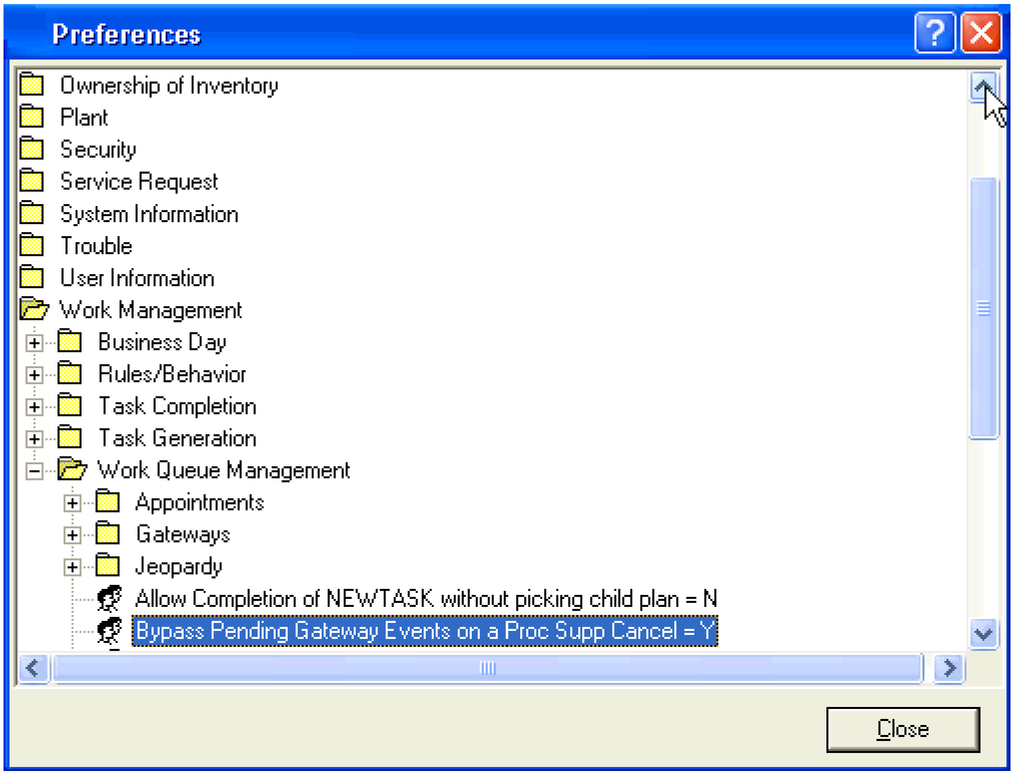
Place the Activate task as close to the DD task as possible

How MetaSolv Solution Handles Activation Events

This section describes the how MetaSolv Solution interfaces with the integration layer using the gateway event to report successes and failures on activation requests.

The descriptions that follow assume that the preference **Bypass Pending Gateway Events on a Proc Supp Cancel** is set to **Y.** This preference is located in the MetaSolv Solution Preferences under **Work Managemen**t, then **Work Queue Management**.

Figure 1–4 Setting a Preference for Gateway Events



Preferences

- Ownership of Inventory
- Plant
- Security
- Service Request
- System Information
- Trouble
- User Information
- Work Management
  - Business Day
  - Rules/Behavior
  - Task Completion
  - Task Generation
  - Work Queue Management
    - Appointments
    - Gateways
    - Jeopardy
    - Allow Completion of NEWTASK without picking child plan = N
    - Bypass Pending Gateway Events on a Proc Supp Cancel = Y

Successful Activation

1. An order is placed, provisioned, and is in work in Work Management.
2. The Activate task for an order is opened in the work queue of Work Management, and the gateway event status is set to Pending.
3. The system initiates the gateway event server, and the gateway event status is set to Sending, which indicates the system is looking for an integration layer to which it can pass a message.
4. The integration layer processes the order and determines the number of activation work orders that must be created.
5. The integration layer sends the following data to MetaSolv Solution:

```
Activation System Name = ASAP
Activation Service Key = Represents the identifier for the activation work
order
Gateway Event Id = from gateway event
Document Number = from gateway event
Task Number = from gateway event
Integration Status = In Progress
Gateway Event name = from gateway event
```

6. The integration layer updates the gateway event status to In Progress.
7. Once the activation system processes the work orders, the integration system updates the gateway event status and the integration status to Complete.
8. The DD task completes successfully.

Activation Completes But the Due Date Task Fails

In this case, the basic flow shown in Figure 1–3 is successful through step 7. In step 8, the Due Date task fails to complete and an error message is displayed.

1. The user creates a supp order to correct the original order.
2. The user manually bypasses the gateway event if the status for any of the activation work orders is already completed.
3. If the user has to make engineering changes to correct the due date problem, the gateway event can be reinitiated. The Integration layer handles this order as a change order.

Activation Error on Some work Orders

1. When the activation system fails on any of the activation work orders the gateway event status will be set to Error.
2. The integration layer sets the gateway event status to Error for the failed work order and passes an error message to MetaSolv Solution.

   The integration layer sets the error message to whatever message was returned by the activation system.
3. The integration layer sets the status to Complete for all successfully completed work orders.
4. In the work queue manager in MetaSolv Solution, the user can select the gateway event, right click and select the **Display Work Order Option** from the displayed selection menu.
5. The system displays the following information:
   - Activation work order number
   - Status (which is set to Error)
   - Error message
   - Date the error message was sent
6. The user has three options to correct the problem:
   - Supplement the order in MetaSolv Solution and manually re-initiate the gateway event. With this action, the process will start at basic flow Step 1.
   - Set the gateway event status to bypass and go the activation application's user interface to correct the issue.
   - Access ASAP and delete the work orders that were not successful, then reinitiate the gateway events.

Integration System Fails

1. When the integration system fails to create the activation work orders, only the gateway event status is updated.

   The event-status is updated to Error, and the error message indicates the time and source of the error. The error details are included in the logs.
2. The user must reinitiate the gateway event. This action starts the basic flow at Step 1.

Order is Supp Canceled

1. If the CSR initiates a supp cancel of a PSR order, the order is appended with a supplement type of 1 and the activity code for all service items is set to A.
2. The system checks the gateway event status and the gateway event preference (**Bypass Gateway Events on a Pending Supp Cancel)** to determine processing.
   - If gateway event status = Pending and the preference is set to **Yes**, the gateway event status is set to Bypass and the gateway event status is not sent to the integration layer.
   - If the gateway event status = Sending or In Progress, the system waits until the gateway event completes or fails.
     - If the status is set to Complete, the gateway event must be re-initiated. The integration layer interprets the supplement type passed with the order information to determine whether all service items on the order are canceled and the services need to be disconnected.
     - If the status of the gateway event is set to Error, the user must determine what caused the error. If some activation work orders are complete, the gateway event is re-initiated so the integration layer can determine that a Disconnect order needs to be created for the activation system A user can also access the activation application's interface manually and disconnect the work orders.

Note: To avoid the situation where an order is in progress, user's should place the gateway event as close as possible to the DD task. The user should backdate the Delay, Activate, and DD tasks. The Delay task keeps the Activate task from going live, which in turns triggers a gateway event.

Creating Custom Attributes in MetaSolv Solution

You may find it necessary to create custom attributes if ASAP requires data that MetaSolv Solution does not currently collect and a workaround for collecting the information is not readily available.

All custom attributes used to capture required or optional activation parameters must be associated with the appropriate building blocks (such as a DSL link or a DSLAM network element). In addition, custom attributes for each building block must be included in a layout that associates them with the Activation process point. This definition and association of custom attributes is performed in the MetaSolv Solution Utilities module.

Any label defined for a custom attribute must match the label expected by the integration layer. The generic custom attribute label can be overridden when the custom attribute is defined for the building block/process point. If the label is overridden, that is the value the API returns. Otherwise, the label from the generic custom attribute definition is returned.

Custom attributes are created using the MetaSolv Solution Utilities module. See the MetaSolv Solution online Help for more information on how to create custom attributes.

Mapping MetaSolv Solution Data to CSDL Command Parameters

ASAP has two levels of service modeling: CSDL commands and ASDL commands. Integrating to ASAP is always executed at the CSDL level. A CSDL command can map to many ASDL commands. ASDL represent low-level atomic actions on a particular network element. A CSDL command is an abstraction layer that allows service bundling and simplification.

CSDL and ASDL commands are typically similar for an ASAP cartridge. CSDL modeling is typically customer specific and occurs as part of the implementation of ASAP. The flow-through package describes integration to standard ASAP cartridges without customer-specific CSDL modeling. Therefore, the CSDL and ASDL commands are closely aligned for the flow-through package. If CSDL changes are required at customer implementation, mapping in the flow-through package can be changed accordingly.

MetaSolv Solution data is mapped to ASAP CSDL command parameters in the integration layer. To determine which ASAP commands are to be used, locate the ASAP cartridge guide for the equipment that will be activated. For example, if POTS service is to be activated on the Lucent 5ESS switch, locate the cartridge guide for that specific switch on the Oracle software delivery website.

Inside the guide, locate the CSDL command for the service activation that you want to accomplish. For example:

Add a POTS residential line:

```
C_LU-5ESS_16_ADD_POTS-RES-LINE
```

Add POTS call waiting:

```
C_LU-5ESS_16_ADD_POTS-CCW
```

The cartridge guide lists the required and optional parameters for the command. The required parameters must be mapped to MetaSolv Solution data coming from the Activation API.

Setting Up the Flow-Through Package

The following sections explain how to locate, set up, and view a flow-through package.

Where to Find the Flow-Through Package

The flow-through package is provided in the **MSS_XML_API_R6_*n*_*n*_b*nnn*.jar** file where:

R6_*n*_*n* is the release version

b*nnn* is the build version

For example: **MSS_XML_API_R6_2_0_b123.jar**

This jar file contains several files, including files that contain sample code. The file that contains the flow-through package sample code is **flow_through_package.jar**. This jar file contains code, libraries, and other files required for Oracle Workshop-based development for the flow-through package application.

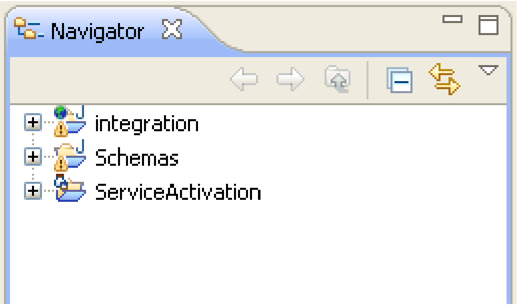
Setting Up on The Workstation

To set up the XML API flow-through package for viewing:

1. Create a local directory and name it **ServiceActivation_workspace**.
2. Locate the **MSS_XML_API_R6_*n_n*_b*nnn*.jar** file.
3. Open the jar file and locate the **flow_through_package.jar** file.
4. Open the jar file and locate the **ServiceActivation_workspace.jar** file.
5. Open the jar file, and extract the contents of the **ServiceActivation_workspace.jar** file into your local directory, **ServiceActivation_workspace**.
6. Open the Workshop IDE.

   The Workspace Launcher dialog box opens.
7. Populate the **Workspace** field by clicking **Browse** to navigate to your local directory, **ServiceActivation_workspace**.
8. Click **OK**.

   The following figure shows the directories that appear in the extracted workspace.



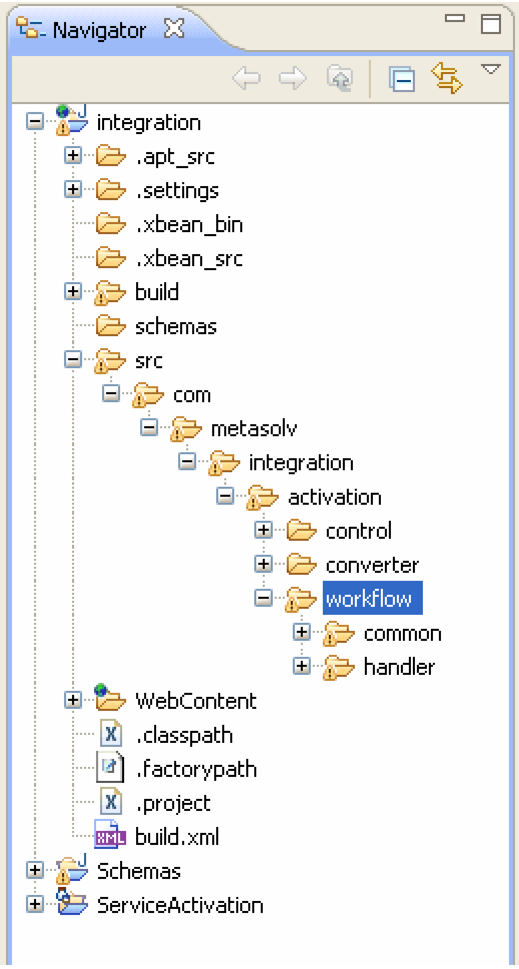
9. Build the application in Workshop.
10. Run the workflows or browse through the workflows to see how they are constructed.

Viewing a Flow-Through Package

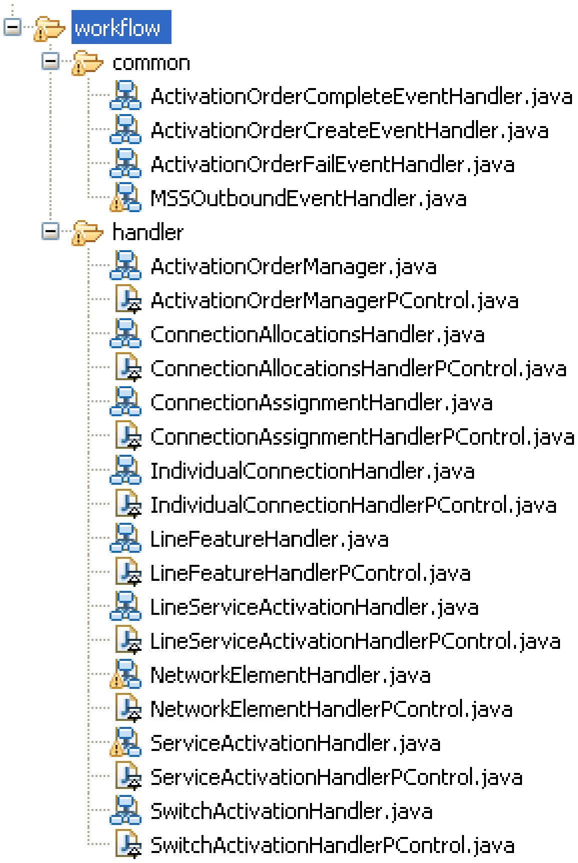
To view the samples in Workshop:

1. Expand the **integration** directory, and navigate to the **workflow** directory.

   The directory path is:
   integration/src/com/metasolv/integration/activation/workflow



2. Expand the **workflow** child directories, **common** and **handler**.



```
workflow
    common
        ActivationOrderCompleteEventHandler.java
        ActivationOrderCreateEventHandler.java
        ActivationOrderFailEventHandler.java
        MSSOutboundEventHandler.java
    handler
        ActivationOrderManager.java
        ActivationOrderManagerPControl.java
        ConnectionAllocationsHandler.java
        ConnectionAllocationsHandlerPControl.java
        ConnectionAssignmentHandler.java
        ConnectionAssignmentHandlerPControl.java
        IndividualConnectionHandler.java
        IndividualConnectionHandlerPControl.java
        LineFeatureHandler.java
        LineFeatureHandlerPControl.java
        LineServiceActivationHandler.java
        LineServiceActivationHandlerPControl.java
        NetworkElementHandler.java
        NetworkElementHandlerPControl.java
        ServiceActivationHandler.java
        ServiceActivationHandlerPControl.java
        SwitchActivationHandler.java
        SwitchActivationHandlerPControl.java
```

3. Double-click a .java file located in one of the **workflow** child directories to see the contents of the workflow in Workshop.

2

Setting Up Communication Between MetaSolv Solution And ASAP

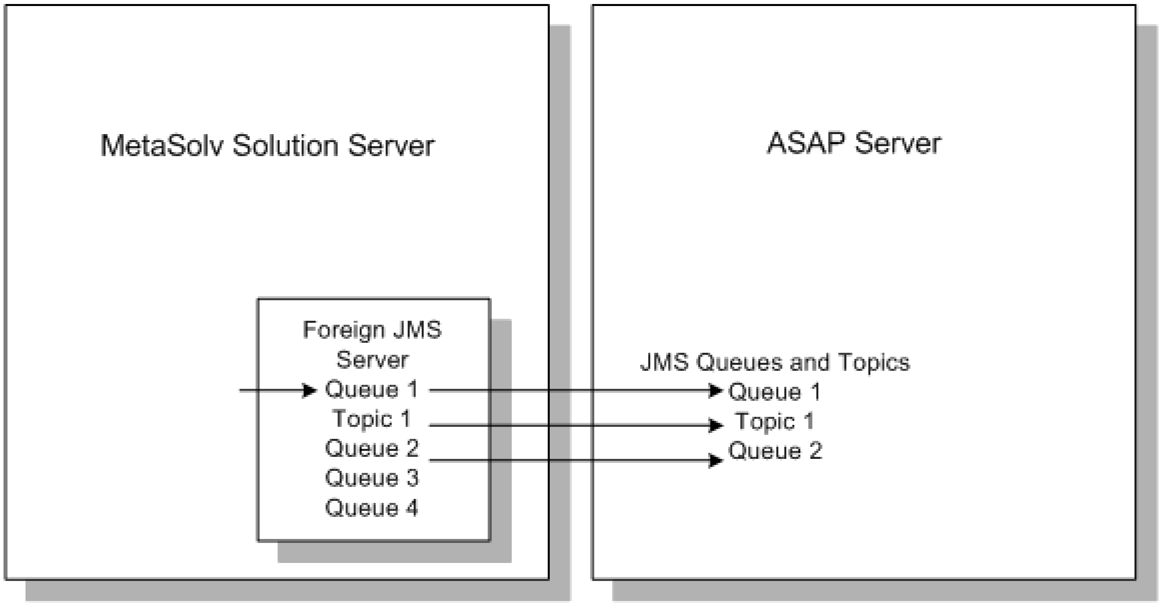
This chapter explains how communication was set up between Oracle Communications MetaSolv Solution and Oracle Communications ASAP for the sample integrations.

Overview

Oracle recommends running MetaSolv Solution and ASAP on separate machines in separate domains. This setup dictates the use of a foreign JMS server to broker messages between the two applications. The foreign JMS server located on the MetaSolv Solution machine creates proxy topics and queues on the ASAP server.

Figure 2–1 shows the setup of the foreign JMS server.

Figure 2–1 Foreign JMS Server



Communication between ASAP and MetaSolv Solution requires three event generators. The names given here are the names used in the integration samples.

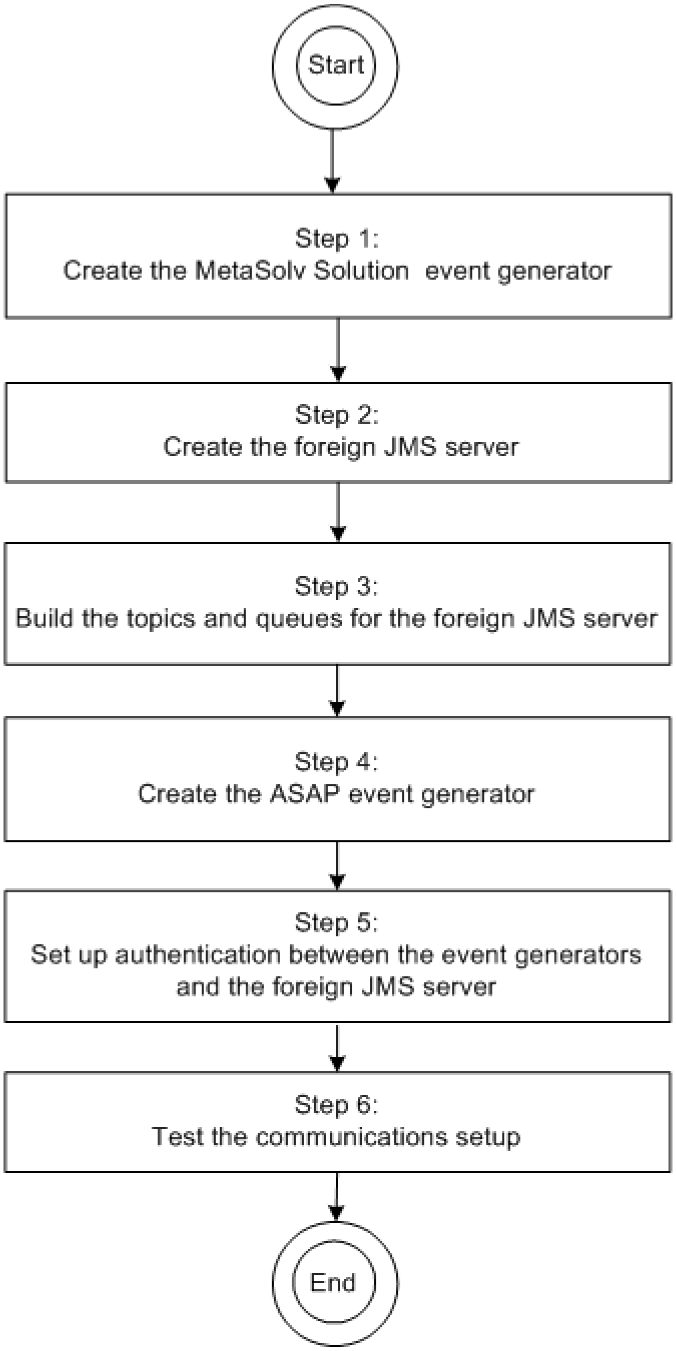
- **MSSOutboundEventGenerator -** This event generator listens on a specified channel for messages from MetaSolv Solution.

- **ASAPEventGenerator -** This event generator listens on a specified channel for messages from ASAP.
- **InternalOutboundGenerator -** This event generator is used internally by MetaSolv Solution. It is created manually after the installation of the core application with the XML API option. See *MetaSolv Solution Installation Guide* for more information.

Figure 2–2 shows the steps required for setting up event communications between the two applications. The steps assume the following:

- MetaSolv Solution with the XML API option has been installed and configured.
- ASAP has been installed and configured.
- The event generator, InternalOutboundGenerator, has been created according to instructions in the *MetaSolv Solution Installation Guide.*
- The host machines for MetaSolv Solution and ASAP have connectivity and can communicate. You can test this by pinging one machine from the other.

Figure 2–2 Basic Steps for Setting Up Communication Between MSS and ASAP


Start
Step 1:
Create the MetaSolv Solution event generator
Step 2:
Create the foreign JMS server
Step 3:
Build the topics and queues for the foreign JMS server
Step 4:
Create the ASAP event generator
Step 5:
Set up authentication between the event generators and the foreign JMS server
Step 6:
Test the communications setup
End

Accessing the WebLogic Consoles

The following sections describe how to complete each step in Figure 2–2. The steps are performed in either the WebLogic Integration Administration Console, or in the WebLogic Server Administration Console.

To open the WebLogic Integration Administration Console:

1. Enter the following in the browser:

```
http://host_admin:port/wliconsole
```

To open the WebLogic Server Administration Console:

1. Enter the following in the browser:

   ```
   http://host_admin:port/console
   ```

   where:

   host_admin is the name of the administration server

   port is the administration server port number

2. When the Login window appears for either console, enter your user name and password and click **Login**.

Step 1: Creating the MetaSolv Solution Event Generator

This procedure describes how to set up the event generator for the MetaSolv Solution application server.

To create the MetaSolv Solution event generator:

1. Copy the **create-jms-eg.py** file to the WebLogic domain directory.

   The **create-jms-eg.py file** is located in the *621home*/*m62server*/**appserver**/**utils**/ directory.

2. Open the command prompt, navigate to the *621home/m62domain*/**bin** directory, and then type the following:

   ```
   setDomainEnv.cmd
   ```

   The information about the domain directory is displayed.

3. Edit the **create-jms-eg.py** file and specify the server details, such as hostname, port, and so on.

4. At the command prompt, type the following:

   ```
   java weblogic.WLST create-jms-eg.py
   ```

   A new Event Generator is created in the WebLogic Integration Console.

To verify whether a new event generator is created:

1. Open the WebLogic Integration Administration Console.

2. In the navigation panel, under **System Configuration**, click **Event Generators**.

3. Under **JMS**, click **VIEW ALL** and ensure that MSSOutboundEventGenerator is created.

Step 2: Setting Up the Foreign JMS Server

A foreign JMS server must be set up on the MetaSolv Solution application server to communicate with the ASAP application server. It is important to make sure that the JNDI name of the connection factory, queues, and topics defined on the MetaSolv Solution application server are the same as those defined for the ASAP application server.

The following procedure explains how to set up the foreign JMS server.

To set up a foreign JMS server:

1. Open the WebLogic Server Administration Console.

2. On the home page, expand **Domain Configurations**, then **Services**, then **Messaging**, click the **JMS Modules** link.

   The JMS Modules page displays.

3. In the JMS Modules table, click **New**.

   The Create JMS System Module page displays.

4. In **Name**, enter a name such as `MyJMSModule`

   Note: The remaining fields are optional.

   ---

5. Click **Next**.
6. In the Servers table, select the target server on which the server will be deployed.
7. Click **Next**.
8. Select the **Would you like to add resources to this JMS system module?** check box, and click **Finish**.

   The Settings page for the JMS module that you just created displays.

9. In the Summary of Resources table, click **New**.

   The Create a New JMS System Module Resource displays.

10. Choose the **Foreign Server** option, and click **Next**.
11. In **Name**, enter a name such as `ASAP JMS Server`.
12. Click **Finish**.
13. In the Summary of Resources table, under the Name column, click the *ASAP JMS Server* link

    Where *ASAP JMS Server* is the name of the resource that you created in step 11.

    The Settings page for the resource that you just created displays.

14. Enter the following information:

    a. In **JNDI Initial Context Factory**, enter:

    ```
    weblogic.jndi.WLInitialContextFactory.
    ```

    b. In **JNDI Connection URL**, enter the URL for the ASAP administration server host and port. For example:

    ```
    t3://srvplsunxmlapi:7001
    ```

    Note: The remaining properties fields are optional.

    ---

    c. Select the **Default Targeting Enabled** check box.

15. Click **Save**.

Step 3: Building Topics and Queues for the Foreign JMS Server

You must build topics and queues for the foreign JMS server on the MetaSolv Solution server. Make sure that the names of the connection factory, queues, and topics defined for the MetaSolv Solution server are the same as those defined in ASAP. The remote

JNDI names for the MetaSolv Solution server must be the same values of the JNDI names defined for the ASAP server.

The local JNDI names for the Metasolv Solution server are defined as:

- ASAP.TopicConnectionFactory
- ASAP.MessageQueue
- ASAP.MessageResponse

  This JMS queue is local in MetaSolv Solution if both applications are using the same version of WebLogic. If ASAP is running on an older version of WebLogic, you must create this JMS queue on the ASAP server, then define it as a remote JMS queue on the MetaSolv Solution server.
- ASAP.XVEventTopic

To find the MetaSolv Solution remote JNDI name for a topic or queue:

1. Open the WebLogic Server Administration Console for the ASAP domain.
2. On the home page, expand **Domain Configurations**, then **Environment**, click the **Servers** link.

   The Summary of Servers page displays.
3. In the Servers table, under the **Name** column, click the link that represents your ASAP server.

   The Settings page for the server you selected displays, showing the Configuration tab > General tab.
4. On the General tab, click the **View JNDI Tree** link.

   In the left pane, the JNDI Tree Structure displays.
5. Drill down to the appropriate queue to see the path that contains the JNDI name that serves as the remote JNDI name on the MetaSolv Solution application server.

Adding a Topic Connection Factory

To add a topic connection factory:

1. Open the WebLogic Server Administration Console.
2. On the home page, expand **Domain Configurations**, then **Services,** then **Messaging**, click the **JMS Modules** link.

   The JMS Modules page displays.
3. In the JMS Modules table, under the Name column, click the *MyJMSServer* link

   Where *MyJMSServer* is the name of the JMS Module you created in "Step 2: Setting Up the Foreign JMS Server".

   The Settings page for the JMS module you selected displays.
4. In the Summary of Resources table, under the **Name** column, click the *ASAP JMS Server* link

   Where *ASAP JMS Server* the name of the resource you created in "Step 2: Setting Up the Foreign JMS Server".

   The Settings page for the resource you selected displays.
5. Click the **Connection Factories** tab.

6. In the Foreign Connection Factories table, click **New**.

   The Create a New Foreign JMS Connection Factory page displays.

7. Enter the following information:

   a. In **Name**, enter:

      ```
      ASAP.TopicConnectionFactory.
      ```

   b. In **Local JNDI Name**, enter:

      ```
      ASAP.TopicConnectionFactory.
      ```

   c. In **Remote JNDI Name**, enter:

      ```
      System.ASAP.ApplicationType.ServiceActivation.Application.1-0,4-7;ASAP.Comp
      .  TopicConnectionFactorySystem.
      ```

      Note: This name should match the JNDI name on the ASAP JMS.

8. Click **Save**.

Adding Queues for the Foreign JMS Server

This section explains how to add queues to foreign JMS servers.

Adding a Message Queue

To add a message queue:

1. Open the WebLogic Server Administration Console.

2. On the home page, expand **Domain Configurations**, then **Services**, then **Messaging**, click the **JMS Modules** link.

   The JMS Modules page displays.

3. In the JMS Modules table, under the Name column, click the *MyJMSServer* link

   Where *MyJMSServer* is the name of the JMS Module you created in "Step 2: Setting Up the Foreign JMS Server".

   The Settings page for the JMS module you selected displays.

4. In the Summary of Resources table, under the **Name** column, click the *ASAP JMS Server* link

   Where *ASAP JMS Server* is the name of the resource you created in "Step 2: Setting Up the Foreign JMS Server".

   The Settings page for the resource you selected displays.

5. Click the **Destinations** tab.

6. In the Foreign Destinations table, click **New**.

   The Create a New Foreign JMS Destination page displays.

7. Enter the following information:

   a. In **Name**, enter:

      ```
      ASAP.MessageQueue.
      ```

   b. In **Local JNDI Name**, enter:

```
ASAP.MessageQueue.
```

   c. In **Remote JNDI Name**, enter:

```
System.ASAP.ApplicationType.ServiceActivation.Application.1-0,4-7;ASAP.Comp
.MessageQueueSystem.
```

8. Click **Save**.

Adding a Message Response Queue

To add a message response queue:

1. Open the WebLogic Server Administration Console.
2. On the home page, expand **Domain Configurations**, then **Services**, then **Messaging**, and then click the **JMS Modules** link.

   The JMS Modules page displays.
3. In the JMS Modules table, under the Name column, click the *MyJMSServer* link

   Where *MyJMSServer* is the name of the JMS Module you created in"Step 2: Setting Up the Foreign JMS Server".

   The Settings page for the JMS module you selected displays.
4. In the Summary of Resources table, under the **Name** column, click the *ASAP JMS Server* link

   Where *ASAP JMS Server* is the name of the resource you created in "Step 2: Setting Up the Foreign JMS Server".

   The Settings page for the resource you selected displays.
5. Click the **Destinations** tab.
6. In the Foreign Destinations table, click **New**.

   The Create a New Foreign JMS Destination page displays.
7. Enter the following information:

   a. In **Name**, enter:

```
ASAP.MessageResponse.
```

   b. In **Local JNDI Name**, enter:

```
ASAP.MessageResponse.
```

   c. In **Remote JNDI Name**, enter:

```
System.ASAP.ApplicationType.ServiceActivation.Application.1-0,4-7;ASAP.Comp
. MessageResponse.
```

8. Click **Save**.

Adding an Event Topic

To add an event topic:

1. Open the WebLogic Server Administration Console.
2. On the home page, expand **Domain Configurations**, then **Services**, then **Messaging**, and then click the **JMS Modules** link.

   The JMS Modules page displays.

3. In the JMS Modules table, under the Name column, click the *MyJMSServer* link.

   Where *MyJMSServer* is the name of the JMS Module you created in "Step 2: Setting Up the Foreign JMS Server".

   The Settings page for the JMS module you selected displays.

4. In the Summary of Resources table, under the **Name** column, click the *ASAP JMS Server* link

   Where *ASAP JMS Server* is the name of the resource you created in "Step 2: Setting Up the Foreign JMS Server".

   The Settings page for the resource you selected displays.

5. Click the **Destinations** tab.

6. In the Foreign Destinations table, click **New**.

   The Create a New Foreign JMS Destination page displays.

7. Enter the following information:

   a. In **Name**, enter:

      ```
      ASAP.XVTEventTopic.
      ```

   b. In **Local JNDI Name**, enter:

      ```
      ASAP.XVTEventTopic.
      ```

   c. In **Remote JNDI Name**, enter:

      ```
      System.ASAP.ApplicationType.ServiceActivation.Application.1-0,4-7;ASAP.Comp
      . ASAP.XVTEventTopic.
      ```

8. Click **Save**.

Step 4: Creating the ASAP Event Generator

This procedure describes how to set up the event generator for the ASAP application server. It is the same procedure described in "Step 1: Creating the MetaSolv Solution Event Generator", but you are entering information that is specific to the ASAP application server, not the MetaSolv Solution application server.

To create the ASAP event generator:

1. Open the WebLogic Integration Administration Console.

2. In the navigation panel, under **System Configuration**, click **Event Generators**.

3. In the **Event Generators** navigation panel, under **JMS**, click the **Create New** link.

   The Create a New JMS Event Generator window displays.

4. Enter the following information:

   a. In **Generator Name**, enter:

      ```
      ASAPEventGenerator.
      ```

   b. In **Destination Type**, select **foreign_jms_destination.**

   c. In **Destination JNDI Name**, select **System.ASAP.ApplicationType.ServiceActivation.Application.1-0,4-7;ASAP. Comp. XVTEventTopic**.

Note: The JNDI name that appears in this field is dependent on how ASAP is installed and can vary from the name given here.

d. In **JMS Connection Factory JNDI Name**, select the default value of **weblogic.jws.jms.QueueConnectionFactory**.

e. Leave the optional **Message Selector** field blank.

f. In **Default Rule Channel**, select **/ServiceActivation/ASAPOutboundEventChannel (xml).**

Note: The **/ServiceActivation/ASAPOutboundEventChannel (xml)** channel is a default from the flow-through packages jar file.

5. Click **Submit**.

Step 5: Setting Up Authentication Credentials

Authentication credentials involve a common password that each server uses to identify and validate communications with the other. The same credential must be set up in the same way in both domains.

To set up authentication credentials:

1. Open the WebLogic Server Administration Console for the MetaSolv Solution domain.
2. On the home page, expand **Domain Configurations**, then **Domain**, and then click the **Domain** link.

   The Settings for your domain displays.
3. Click the **Security** tab.
4. Scroll down, and click the **Advanced** link.
5. In the right pane, select the **Configuration** tab, then select the **Advanced** tab.
6. In **Credential**, enter a password.
7. In **Confirm Credential**, reenter the same password.
8. Click **Save**.
9. Open the WebLogic Server Administration Console for the ASAP domain.
10. Repeat step 2 to step 8, entering the same credentials for the ASAP domain that you entered for the MetaSolv Solution domain.

Note: Both the MetaSolv Solution and the ASAP domains must have the same password for the domains to communicate. This password is used to authenticate communications between the domains.

Step 6: Testing the Communication Setup

You can test your communication setup by triggering a gateway event from MetaSolv Solution.

To see if the communication setup is working, look for the following changes when the gateway event is triggered:

- In MetaSolv Solution, the status of the task that initiated the gateway event changes.
- In ASAP, an order is created.

The basic steps in the communication process are:

1. A gateway event is triggered from MetaSolv Solution.
2. The gateway event causes data to be pulled from MetaSolv Solution for the creation of an order in ASAP.
3. The data is mapped to ASAP formats in the integration layer and sent to ASAP where the order is created.
4. When the order is completed in ASAP, an event is generated to update the event status in MetaSolv Solution to one of the following: Completed or Failed.

You can find message counts in the following locations:

- WebLogic Server, JMS monitoring
- WebLogic Integration Server, Event generators

3

Activation API Data Schema

This chapter defines the attributes of the data returned when the Activation API makes a call to Oracle Communications MetaSolv Solution to retrieve activation data.

MetaSolvServiceActivationType Attribute Hierarchy

The element returned by the Activation XML API is metaSolvServiceActivationValue, which is defined in the schema **XmlMetaSolvServiceEntities.xsd**. The element is defined as type MetaSolvServiceActivationType. The following list is a hierarchical representation of the attributes contained within MetaSolvServiceActivationType. See "Schema Type and Attribute Descriptions" for the descriptions of the types and attributes within MetaSolvServiceActivationType.

```
orderDetails (OrderDetailsType)
    companyName
    firstName
    lastName
    orderKey (OrderKeyType)
        primaryKey
    serviceCategory
    purchaseOrderNumber
    desiredDueDate
    suppType
    orderType

switchActivation (Array of SwitchActivationType)
    switchLocationNetworkCode
    lineServices (Array of LineServiceType)
```

Note: See "ServiceType" for more information.

```
        lineEquipmentNumber
        networkElementIdentifier
        networkElementName
        originatingEquipmentType
        isHuntGroupMember
        telephoneNumber (TelephoneNumberType)
            telephoneNumber
            activityCd
        relatedTelephoneNumbers
            fromTelephoneNumber
            toTelephoneNumber
            associationType
            intercept
```

```
                interceptId
                option
                name
                endDate
        picInformation
            pic
            picTypeCd
            freezePic
            partyName
            activityCd
        lineFeatures (Array of ServiceType)
```

Note: See "ActivationCustomAttributeType" for more information.

```
    systemOptions (Array of ServiceType)
```

Note: See "ServiceType" for more information.

```
    huntGroups (Array of HuntGroupType)
```

Note: See "ServiceType" for more information.

```
        groupName
        groupNumber
        groupType
        huntGroupMember (Array of HuntGroupMemberType)
            memberNumber
            memberHuntTo
            activityCd

networkElements (Array of NetworkElementType)
```

Note: See "ServiceType" for more information.

```
    status
    networkElementIdentifier
    networkElementName
    networkElementType
    managementIPAddress
    elementCustomAttributes (Array of ActivationCustomAttributeType)
```

Note: See "ActivationCustomAttributeType" for more information.

```
    individualConnections (Array of IndividualConnectionType)
        connectionType
        connectionAssignments (Array of ConnectionAssignmentType)
```

Note: See "ServiceType" for more information.

```
            connectionDesignKey (connectionDesignKey)
            designKey
            connectionIdentifier
            connectionStatus
            ipAddress
            macAddress
            physicalNodeAddress
            logicalPort
            shelfEquipmentName
            shelfSlotNumber
            shelfEquipmentKey
            portEquipmentKey
            portAddressSequenceNumber
            channels  (Array of int)
            portAddressStatus
            connectionCustomAttributes  (Array of ActivationCustomAttributeType)
```

Note: See "ActivationCustomAttributeType" for more information.

```
            connectionType
            dueDate
            locationName
        connectionAllocations  (Array of ConnectionAllocationType)
```

Note: See "ServiceType" for more information.

```
            connectionDesignKey (connectionDesignKey)
                designKey
            connectionIdentifier
            connectionStatus
            ipAddress
            macAddress
            physicalNodeAddress
            logicalPort
            shelfEquipmentName
            shelfSlotNumber
            shelfEquipmentKey
            portEquipmentKey
            portAddressSequenceNumber
            channels  (Array of int)
            portAddressStatus
            connectionCustomAttributes  (Array of ActivationCustomAttributeType)
```

Note: See "ActivationCustomAttributeType" for more information.

```
            connectionType
            relationshipCustomAttributes  (Array of ActivationCustomAttributeType)
```

Note: See "ActivationCustomAttributeType" for more information.

```
networkSystems (Array of NetworkSystemType)
    networkSystemKey (NetworkSystemKey)
        networkSystemKey
    networkNameShort
    networkNameLong
    networkDescription
    status
    dateCreated
    dateInService
    providerSystemId
    customerSystemId
    hardSoftAssignment
    systemCustomAttributes  (Array of ActivationCustomAttributeType)
```

Note: See "ActivationCustomAttributeType" for more information.

```
    sameChannelAssignmentIndicator
    inherentlyProtIndicator
    networkElements  (Array of NetworkElementType)
```

Note: Attributes are the same as networkElements presented previously.

```
ServiceType
    serviceKey
        primaryKey
    itemType
    itemAlias
    provisioningName
    activityCD
    valueLabels  (Array of ValueLabelType)
        label
        value
        activityCD

ActivationCustomAttributeType
    caUsageKey
    label
    customAttributeValues  (Array of ActivationCustomAttributeValueType)
        value
        unitOfMeasure
```

Schema Type and Attribute Descriptions

This section describes the types and attributes contained in MetaSolvServiceActivationType. The following containers are described:

- MetaSolvServiceActivationType
- OrderDetailsType
- SwitchActivationType
- LineServiceActivationType
- TelelphoneNumberType
- RelatedTelephoneNumberType
- InterceptType
- PicInfoType
- HuntGroupType
- HuntGroupMemberType

- NetworkElementType
- IndividualConnectionType
- ConnectionAssignmentType
- ConnectionAllocationType
- NetworkSystemType
- ServiceType
- ValueLabelType
- ActivationCustomAttributeType
- ActivationCustomAttributeValueType

MetaSolvServiceActivationType

This container represents information used to activate POTS service, a network system, or a connection within a network system. This structure supports multiple technologies such as DSL, DSL with POTS, VoDSL (Voice over DSL), ATM (Asynchronous Transfer Mode) and Frame Relay, VoATM (Voice over ATM), and MPLS (Multiple Protocol Label Switching).

Table 3–1 describes the attributes or types in the MetaSolvActivationType container.

Table 3–1 MetaSolvServiceActivationType

Attribute or Type	Description
orderDetails	Type: OrderInformationType This information represents information about the order that the activation applies to.
switchActivations	Array of type SwitchActivationType. Represents a collection of switch-related services on an order, such as POTS. The collection will contain a SwitchActivationType instance for every switch location associated with the ordered services.
networkElements	Array of type NetworkElementType. Represents a collection of network elements which meet any of the following criteria: ■ The network element is an item on the order. ■ A connection on the order is assigned to the network element. ■ A connection on the order is allocated to a connection that is assigned to the network element. For example, for a DSL service order that includes a physical connection and one virtual connection, this collection would include a NetworkElementType instance for the DSLAM that the physical connection is assigned to. In addition, it would include an instance for every network element in the path of the virtual connection, such as an ATM Switch and an IP Router. Network elements are included in this collection only if the network element or related connection is ordered separately from a network system. For ordered network systems, such as a VPN network, the network system is represented as a NetworkSystemType (see the networkSystems attribute) that contains network elements and connections.

Table 3–1 (Cont.) MetaSolvServiceActivationType

Attribute or Type	Description
networkSystems	Array of type NetworkSystemType. Represents a collection of ordered network systems such as a virtual private network (VPN). The data includes information about the network system and all network elements and connections contained within the network system.

OrderDetailsType

This container represents information about the activation order.

Table 3–2 describes the attributes or types in the OrderDetailsType container.

Table 3–2 OrderDetailsType

Attribute or Type	Description
companyName	Company name of the customer.
firstName	First name of the customer.
lastName	Last name of the customer.
orderKey	Type.attribute: OrderKeyType.primaryKey Identifier of the specific order that resulted in the activation.
serviceCategory	A code that categorizes bill cycles, customer accounts, and products for PSRs. When placing a service request for a customer account, you can order only those products that have the same service category. Service categories are maintained by each organization. Examples of service categories include business, residential, and government. Valid values: User defined.
purchaseOrderNumber	Customer's purchase order number.
desiredDueDate	The date by which an order is to be completed. This is represented as a string with the format: *yyyy-MM-dd'T'HH:mm:ss.*
supplementType	Reason an order is being supplemented, or changed. Only applicable for supplemented orders. Valid values: Cancel, New Due Date, Correction.
orderType	Identifies the type of order. Currently, the only order type supported for activation is *SO* (a PSR order).

SwitchActivationType

This container represents information about a switch and services to be activated on the switch such as a line service.

Table 3–3 describes the attributes or types in the SwitchActivationType container.

Table 3–3 SwitchActivationType

Attribute or Type	Description
switchLocationNetworkCode	The network location code of the switch.

Table 3–3 (Cont.) SwitchActivationType

Attribute or Type	Description
lineServices	Array of Type: LineServiceType Represents a collection of line service data for a switch.
systemOptions	Array of Type: ServiceType Represents a collection of system options for a switch ordered at the same level as the line service items.
huntGroups	Array of Type: HuntGroupType Represents a collection of hunting groups to be configured on the switch. A hunting group determines how an incoming call may hunt (search) for an available line when the one dialed is not available.

LineServiceActivationType

This container represents information about a line service.

Table 3–4 describes the attributes or types in the LineServiceActivationType container.

Table 3–4 LineServiceType

Attribute or Type	Description
lineEquipmentNumber	The port activated on the switch. This value maps to the node address of the port in MetaSolv Solution. The node address is populated on the equipment specification for the equipment containing the assigned port. This value must be present to enable activation.
networkElementIdentifier	The identifier of the network element. This value is entered in the NEID field on the properties window for the network element.
networkElementName	The name of the network element.
originatingEquipmentType	Identifies the format of the line equipment number (LEN). This value is entered in the Mounting Type field on the equipment specification for the shelf equipment.
isHuntGroupMember	Identifies whether or not the line is a member of a hunt group on the order.
telephoneNumber	Type: TelephoneNumberType Represents data for the telephone number of the line service.
relatedTelephoneNumbers	Array of RelatedTelephoneNumberType Represents a collection of telephone numbers related to the telephone number of the line service.
picInformation	Array of PicInfoType Represents a collection of Primary Interexchange Carrier (PIC) information for the line service. The PIC is a customer's preferred provider for long distance. Customers can choose PICs for different types of calls, such as interlata/boundary calls and intralata/boundary calls. The PICs you can select are entered through Access Customer in Order Management on the navigation bar. They are referred to as CICs instead of PICs in Order Management.
lineFeatures	Array of ServiceType. Represents a collection of features on the line, such as call waiting.

Table 3–4 (Cont.) LineServiceType

Attribute or Type	Description
elementCustomAttributes	Array of ActivationCustomAttributeType Represents a collection of custom attributes for the switch network element. Custom attributes can be defined to capture any activation data associated with a network element not otherwise available through the application. For a switch network element this might include the manufacturer, technology and software version of the switch. This collection includes an instance for all custom attributes defined for the element regardless of whether or not the custom attribute is populated with a value. Populated values represent the current value of the custom attribute. The custom attribute information does not include an activity code and will not include both the old and the new value in the event of a change.

TelelphoneNumberType

This container represents information about the telephone number for a line service.

Table 3–5 describes the attributes or types in the TelephoneNumberType container.

Table 3–5 TelephoneNumberType

Attribute or Type	Description
telephoneNumber	The telephone number for the line service including any special characters used for formatting. The special characters can be used to delimit the components of telephone number. For example, the dashes of a North American telephone number could be used to separate the number into NPA, NXX and Line Range.

activityCd	The activity to be performed on this telephone number as specified by the order.
	Valid values:
	iactn = New
	iactc = Change
	iactd = Disconnect
	iacti = In Service
	iacts = Suspend
	iactr = Restore
	iacta = Cancel

RelatedTelephoneNumberType

This container represents information about telephone numbers related to the telephone number of a line service.

Table 3–6 describes the attributes or types in the RelatedTelephoneNumberType container.

Table 3–6 RelatedTelephoneNumberType

Attribute or Type	Description
fromTelephoneNumber	The telephone number of the line service.

Table 3–6 (Cont.) RelatedTelephoneNumberType

Attribute or Type	Description
toTelephoneNumber	The telephone number to which the line service telephone number is related.

associationType	Identifies the type of association between the telephone numbers.
	Valid values:
	LT = Alternate
	RFC = Remote Call Forward

intercept	Type: InterceptType Represents information about an intercept message, which is set up following a line's removal from service to alert incoming callers of the change.

InterceptType

This container represents information about an intercept message, which is set up following a line's removal from service to alert incoming callers of the change.

Table 3–7 describes the attributes or types in the InterceptType container.

Table 3–7 InterceptType

Attribute or Type	Description
interceptId	Identifier for the intercept.
option	The type of transfer requested by the end user. Valid values: plit = The new telephone number is given. Standard = More than one telephone number is given. None =
name	Name associated with the telephone number to which calls are to be referred.
endDate	The date at which the intercept is no longer effective. This is represented as a string with the format: *yyyy-MM-dd'T'HH:mm:ss*.

PicInfoType

This structure represents information on the Primary Interexchange Carrier (PIC) for a given voice service.

Table 3–8 describes the attributes or types in the PicInfoType container.

Table 3–8 PicInfoType

Attribute or Type	Description
pic	A Carrier Identification Code (CIC) that represents an end user's toll provider. When designated for this purpose, these codes are call primary interexchange codes (PICs).

Table 3–8 (Cont.) PicInfoType

Attribute or Type	Description

picTypeCd	The type of PIC.
	Valid values:
	LPIC = IntraLATA
	PIC = InterLATA
	INTNL = International

freezePic	Indicates the customer requested the freeze option on the PIC, meaning the end user is specifying that the PIC can only be changed when authorized by the end user. This attribute will not be populated by the Activation API.
partyName	The provider who is responsible to the customer using the PIC for toll calling.

activityCd	The activity to be performed on this PIC as specified by the order.
	Valid values:
	iactn = New
	iactc = Change
	iactd = Disconnect
	iacti = In Service

HuntGroupType

This structure represents information about a hunting group, which determines how an incoming call may search for an available line when the one dialed is not available.

Table 3–9 describes the attributes or types in the HuntGroupType structure.

Table 3–9 HuntGroupType

Attribute or Type	Description
Include ServiceType Section	Include the ServiceType section below.
groupName	Descriptive name for the hunting group.
groupNumber	A numeric identifier for the hunting group.
groupType	Describes the type of hunting group, indicating its behavior. Valid values: User defined.
huntGroupMember	Array of HuntGroupMemberType Represents a collection of hunting group members that belong to the hunting group. The telephone number for each member will be included as a line service in the lineServices collection under SwitchActivationType.

HuntGroupMemberType

This structure represents information about a member of a Hunting group used to route incoming calls when the original number dial is not available to pick up.

Table 3–10 describes the attributes or types in the HuntGroupMemberType structure.

Table 3–10 HuntGroupMemberType

Attribute or Type	Description
memberNumber	The member number that is the original or from number.
memberHuntTo	The member the call will be routed to if this member cannot be reached.

activity	The activity to be performed on this hunt group member as specified by the order.
	Valid values:
	iactn = New
	iactc = Change
	iactd = Disconnect
	iacti = In Service

NetworkElementType

This container represents information about a network element, any connections assigned to a port(s) on the network element, and any virtual connections allocated to those connection assignments.

Table 3–11 describes the attributes or types in the NetworkElementType container.

Table 3–11 NetworkElementType

Attribute or Type	Description
Include ServiceType Section	Include the ServiceType section below. The ServiceType attributes are populated only if the network element is an ordered item. For example, in the case of an ordered virtual connection, the network elements that make up the path of the virtual connection will be included because they are associated to the ordered connection, but the ServiceKey attributes for those network elements will not be populated.
networkElementIdentifier	The identifier of the network element. To enable activation, this identifier must match the identifier in the Network Element Manager.
status	The status of the network element. Valid values: Pending In Service Pending Disconnect Disconnected Cancelled
networkElementName	Descriptive name for the network element.
networkElementType	Indicates the type of network element such as DSL Multiplexer or ATM Switch. Valid values: User defined.
managementIPAddress	The IP address of the network element host box.

Table 3–11 (Cont.) NetworkElementType

Attribute or Type	Description
individualConnections	Array of IndividualConnectionType Represents a collection of connections connected to the network element. Each instance represents information about an individual connection assigned to a port(s) on the network element and any virtual connections allocated to that connection assignment.
elementCustomAttributes	Array of ActivationCustomAttributeType Represents a collection of custom data about the network element. Custom attributes can be defined to capture any activation data associated with a network element not otherwise available through the application. For a network element this might include a profile name or the manufacturer, technology and software version. This collection will include an instance for all custom attributes defined for the element regardless of whether or not the custom attribute is populated with a value. Populated values represent the current value of the custom attribute. The custom attribute information does not include an activity code and will not include both the old and the new value in the event of a change.

IndividualConnectionType

This container represents information about an individual connection assigned to a port(s) on a network element and any virtual connections allocated to that connection assignment.

Table 3–12 describes the attributes of types in the IndividualConnectionType container.

Table 3–12 IndividualConnectionType

Attribute or Type	Description
connectionType	The connection type of the physical connection represented by connectionAssignments. Valid values: User defined.
connectionAssignments	Array of ConnectionAssignmentType Represents a collection of connection assignments to ports on a network element. Each connectionAssignment instance represents information about a different port assignment on the same physical connection. If the order involves the change of a connection from one port to another, a connectionAssignment occurrence will be included for both the port to be disconnected and the new port. The old port can be identified by a portAddressStatus = *Pending Disconnect*, while the new port will have a portAddressStatus = *Pending*.
connectionAllocations	Array of ConnectionAllocationType Represents a collection of virtual connections that are allocated to the physical connection represented by the collection of connectionAssignments.

ConnectionAssignmentType

This container represents information about the assignment of a physical connection to a particular port on a network element.

Table 3–13 describes the attributes or types in the ConnectionAssignmentType container.

Table 3–13 ConnectionAssignmentType

Attribute or Type	Description
ServiceType	Include the ServiceType attributes (See Table 3–16). The attributes are populated only if the connection is an item on the order.
connectionDesignKey	The MetaSolv Solution internal numeric identifier for the connection.
connectionIdentifier	Identifier for the connection defined by the service provider.
connectionStatus	Current status of the connection. Valid values: Pending Assigned In Progress CLR Issued DLR Issued In Service Pending Disconnect Disconnected Problem Cancelled
ipAddress	IP address for the port to which the assignment is made.
macAddress	MAC address for the port to which the assignment is made.
physicalNodeAddress	This represents the port address to which the connection has been assigned. If populated for a virtual connection (ConnectionAllocationType), this represents a virtual port.
logicalPort	A user-defined identifier for a group of channel positions.
shelfSlotNumber	Describes the slot number for the network element. This data is captured in the Mounting Position field of the Equipment Specification for the shelf equipment.
shelfEquipmentKey	The MetaSolv Solution internal numeric identifier for the shelf equipment that contains the equipment that contains the port.
portEquipmentKey	The MetaSolv Solution internal numeric identifier for the card that contains the port.
portAddressSequenceNumber	The MetaSolv Solution internal sequence number used in conjunction with the portEquipmentKey to identify the port.
shelfEquipmentName	The Equipment Name of the shelf equipment. If a shelf number is required for activation, it is recommended that it is stored in this field.
channels (Array)	Represents a group of channel positions that comprise a logical port to which a physical connection is allocated.

Table 3–13 (Cont.) ConnectionAssignmentType

Attribute or Type	Description
portAddressStatus	The status of the port. Valid values: Pending In Service Pending Disconnect
connectionCustomAttributes	Array of ActivationCustomAttributeType Represents a collection of custom data about the connection. Custom attributes can be defined to capture any activation data associated with the connection not otherwise available through the application. MetaSolv Solution provides some custom attributes that are attributes of a connection, such as Broadband Service Category, In Service Upstream Bitrate and In Service Downstream Bitrate. Note that this collection includes an instance for all custom attributes defined for the connection regardless of whether or not the custom attribute is populated with a value. Populated values represent the current value of the custom attribute. The custom attribute information does not include an activity code and will not include both the old and the new value in the event of a change.
connectionType	The type of connection such as DSL Link or Assignable Bandwidth Link. Valid values: User defined.
dueDate	The date the connection is due to be activated. This is represented as a string with the format: *yyyy-MM-dd'T'HH:mm:ss*.
locationName	The name of the location where the equipment is located. This is expressed as a Common Language Location Identifier (CLLI) or Coded Location.

ConnectionAllocationType

Represents information about a connection that is allocated to a connection assignment. For example, this could represent a virtual connection allocated to a physical connection to provide DSL service.

Table 3–14 describes the attributes or types in ConnectionAllocationType.

Table 3–14 ConnectionAllocationType

Attribute or Type	Description
ServiceType	Include the ServiceType attributes. See Table 3–16 for more information.
connectionIdentifier	The identifier for the connection used by an Exchange Carrier.

Table 3–14 (Cont.) ConnectionAllocationType

Attribute or Type	Description
connectionStatus	The current status of the connection. Valid values: Pending Assigned In Progress CLR Issued DLR Issued In Service Pending Disconnect Disconnected Problem Cancelled
ipAddress	The IP address for the port to which the assignment is made.
macAddress	The MAC address for the port to which the assignment is made.
physicalNodeAddress	This represents the Port address to which the connection has been assign.
logicalPort	A user-defined identifier for a group of channel positions.
shelfSlotNumber	Describes the slot number for the network element. This data is captured in the Mounting Position field of the Equipment Specification for the shelf equipment.
shelfEquipmentKey	The MetaSolv Solution internal numeric identifier for the shelf equipment that contains the equipment that contains the port.
portEquipmentKey	The MetaSolv Solution internal numeric identifier for the equipment that contains the port.
portAddressSequenceNumber	The MetaSolv Solution internal sequence number used in conjunction with the portEquipmentKey to identify the port.
shelfEquipmentName	The equipment name of the shelf equipment. If shelf number is required for activation, it is recommended that it is stored in this field.
channels	Array of ints Represents a collection of assigned enabled port addresses (EPAs) on the logical port.
portAddressStatus	Status of the port address. Valid values: Unassigned ending In Service Pending Disconnect Reserved Reserved Capacity

Table 3–14 (Cont.) ConnectionAllocationType

Attribute or Type	Description
connectionCustomAttributes	Array of ActivationCustomAttributeType Represents a collection of custom data about the virtual connection. Custom attributes can be defined to capture any activation data associated with the connection not otherwise available through the application. MetaSolv Solution provides some custom attributes that are attributes of a connection, such as Broadband Service Category, In Service Upstream Bitrate and In Service Downstream Bitrate. Note that this collection will include an instance for all custom attributes defined for the connection regardless of whether or not the custom attribute is populated with a value. Populated values represent the current value of the custom attribute. The custom attribute information does not include an activity code and will not include both the old and the new value in the event of a change.
connectionType	The type of connection such as DSL Link or Assignable Bandwidth Link. Valid values: User defined.
relationshipCustomAttributes	Array of ActivationCustomAttributeType Represents a collection of custom data about the relationship between the connection represented by the ConnectionAssignment and the connection represented by this ConnectionAllocation. For example, this could include the Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) for a virtual connection allocated to a DSL Link connection. Note that this collection will include an instance for all custom attributes defined for the relationship regardless of whether or not the custom attribute is populated with a value. Populated values represent the current value of the custom attribute. The custom attribute information does not include an activity code and will not include both the old and the new value in the event of a change.

NetworkSystemType

Represents information about a network system.

Table 3–15 describes the attributes or types in NetworkSystemType.

Table 3–15 NetworkSystemType

Attribute or Type	Description
ServiceType	Include the ServiceType attributes. See Table 3–16 for more information.
networkSystemKey	Type.attribute: NetworkSystemKey.networkSystemKey Represents the unique numeric identifier for the network system.
networkNameShort	Short descriptive name for the network system
networkNameLong	Longer descriptive name for the network system
networkDescription	Description of the network system

Table 3–15 (Cont.) NetworkSystemType

Attribute or Type	Description
status	The current status of the network system. Valid values: Pending Assigned In Progress CLR Issued DLR Issued In Service Pending Disconnect Disconnected Problem Cancelled
dateCreated	Date when the network system was created. This is represented as a string with the format *yyyy-MM-dd'T'HH:mm:ss.*
dateInService	Date on which the network system was placed in service. This is represented as a string with the format *yyyy-MM-dd'T'HH:mm:ss.*
providerSystemId	A number or text string used by the service provider to identify the network system
customerSystemId	A number or text string used by a customer to identify the network system
hardSoftAssignment	Specifies whether assignments are made to core components within a network system or not. Valid values: 1 = H (hard) - Assignments are made to core components. 2 = S (soft) - Assignments are not made to core components, but to edge components that provide entry and exit from the network.
networkElements	Array of NetworkElementType Represents a collection of network elements contained within the network system. Information about the connections between the network elements is contained in each network element the connection is assigned to.
systemCustomAttributes	Array of ActivationCustomAttributeType Represents a collection of custom data about the network system. Custom attributes can be defined to capture any activation data associated with the network system not otherwise available through the application. Note that this collection includes an instance for all custom attributes defined for the network system regardless of whether or not the custom attribute is populated with a value. Populated values represent the current value of the custom attribute. The custom attribute information does not include an activity code and will not include both the old and the new value in the event of a change.

Table 3–15 (Cont.) NetworkSystemType

Attribute or Type	Description
sameChannelAssignmentIndicator	Indicates if the network requires the same channel assignment (for example, wavelength) on each facility segment when assigning to multiple segments through a network. The indicator will be applicable only to networks based on the Optical template. It will be used by the OPA - Optical Provisioning Assistant and auto-assignment to enforce equivalent channel assignments along a given assignment route. Valid values: Y = Use the same channel assignment (for example, wavelength) for a connection through the system. N = Do not use the same channel assignment (for example, wavelength) for a connection through the system. The circuit/connection may be assigned to different channels (for example, wavelengths) on each segment. NULL = Applies only to Optical Network Systems. Value will be Null when the Network System type is not *Optical*.
inherentlyProtIndicator	Indicates if a second set of assignments is necessary to protect a circuit when provisioning a connection to an optical network.

ServiceType

This container represents generic information about the service. This type is currently only used in MetaSolvServiceActivationType. It is included as part of other activation types representing services that may be activated including NetworkSystemType, NetworkElementType, ConnectionAssignment, ConnectionAllocation, LineServiceType and HuntGroupType. It is also used to represent system options and line features for switch activation. The attributes in this type are populated only if the service item is included on the order.

Table 3–16 describes the attributes or types in the ServiceType container.

Table 3–16 ServiceType

Attribute or Type	Description
serviceKey	Type.attribute: MetaSolvServiceKey.servicePrimaryKey The unique numeric identifier for the service item.
itemType	The MetaSolv Solution defined code for the type of the service. Item types are combined in building product specifications. These specifications serve as templates for creating a service provider’s product catalog. For example, the item type code for the virtual connection in a DSL service would be CONNECTOR. Under the virtual connection in the product catalog could be an IP Address service, which would have the item type code IPADDR.
itemAlias	The name used by the service provider to describe the service item as it applies to a product catalog item.

Table 3–16 (Cont.) ServiceType

Attribute or Type	Description
provisioningName	This name is required in MetaSolv Solution when the Switch Provisioning Indicator or the Server Provisioning Indicator is set on a product specification item or product specification value label. The provisioning name can be used to differentiate between services items that have the same item type. For example, a line service can contain multiple features such as call waiting or call forwarding. The features are represented as a collection of ServiceTypes under the LineServiceType, each with item type = OPTION. To differentiate between the call waiting and call forwarding features, the provisioning name on the two product specifications could be populated with different values such as CCW an CFW. ItemAlias is not recommended to be used for this purpose because it is defined for each product catalog item rather than at the product specification level.
activityCD	The activity to be performed on the service item as specified by the order. Valid values: iactn - New iactc - Change iactd - Disconnect iacts - Suspend iactr - Restore iacta - Cancel If a service item that is *In Service* is placed on an order and is not disconnected, suspended, restored or cancelled, the activity code will be *iactc* (Change) even if the properties or design of the service item were not modified.
valueLabels	Array of ValueLabelType This represents a collection of value and label combinations for attributes defined by the service provider in the product specification.

ValueLabelType

This container represents information about a value and label combination for an attribute defined by the service provider in a product specification.

Table 3–17 describes the attributes or types in the ValueLabelType container.

Table 3–17 ValueLabelType

Attribute or Type	Description
label	The label of the service attribute
value	The value of the service attribute

Table 3–17 (Cont.) ValueLabelType

Attribute or Type	Description
activityCD	The activity to be performed on the attribute as specified by the order. Valid values: actn - New iactc - Change iactd - Disconnect iacti - In Service iacta - Cancel

ActivationCustomAttributeType

This container represents information about a value and label combination for a custom attribute. Custom attributes can be defined by the service provider and associated with network systems, network elements, physical connections, virtual connections and the relationship between physical and virtual connections. Custom attributes can be used to capture activation data for these items not otherwise available through the application.

Table 3–18 describes the attributes or types in the ActivationCustomAttributeType container.

Table 3–18 ActivationCustomAttributeType

Attribute or Type	Description
caUsageId	The MetaSolv Solution internal numeric key that identifies a custom attribute usage. As a part of defining custom attributes, a custom attribute is first defined generically and then associated with a *building block*, such as a type of network system, network element, connection or connection relationship. When that association is made, a custom attribute usage is created with a caUsageId. For example, Broadband Service Category is generically defined as a custom attribute. When Broadband Service Category is associated to the connection type DSL Link, a CA usage is created for that association.
label	The label of the custom attribute. A label is assigned to CA in its generic definition. That label can be overridden when the generic CA is associated to a building block and a process point (such as Activation). This is the label value defined for the building block and Activation process point.
customAttributeValues	Array of ActivationCustomAttributeValueType This represents a collection of values for a custom attribute. A custom attribute can be defined to allow either a single value or multiple values.

ActivationCustomAttributeValueType

This container represents information about a value for a custom attribute.

Table 3–19 describes the attributes or types in the ActivationCustomAttributeValueType container.

Table 3–19 ActivationCustomAttributeValueType

Attribute or Type	Description
value	The value of the custom attribute. Values can be defined
unitOfMeasure	The unit of measure associated with the value. Valid values: User defined.

4

Planning For POTS Activation

This chapter describes how the POTS flow-through package was planned and the resulting setup in Oracle Communications MetaSolv Solution. The package shows a MetaSolv Solution/ASAP integration that allows a POTS order entered in MetaSolv Solution to be activated automatically by ASAP. The package describes an order placed through the Product Service Request (PSR) module. Any grooming or engineering activity outside of an order or from a non-PSR order such as an engineering work order (EWO) or internal service request (ISR) is not shown.

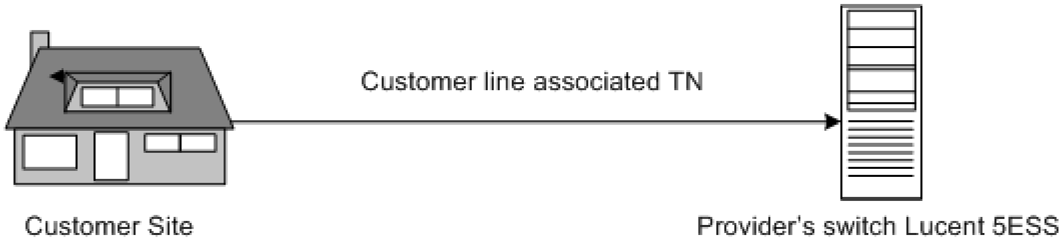
The switch used for POTS activation in the package is the Lucent 5ESS. For mapping and planning purposes, the following document was used: *Lucent 5ESS International Cartridge Guide* (FCS Edition). You can locate this document on the Oracle software delivery Web site.

A POTS order consists of an electrical path (line) between a service provider switch and a subscriber. The line is associated with an individual telephone number that can be used for incoming and outgoing telephone calls.

The line itself is an item type and the order can contain other item types for features such as call waiting, call forwarding, and so on.

Figure 4–1 shows basic POTS service.

Figure 4–1 POTS Service



Basic Data Flow for a New POTS Service

This section describes the basic data flow from the time an order is entered into MetaSolv Solution until the order is activated by ASAP.

1. An order for POTS service is entered in MetaSolv Solution.
2. A provisioning plan is selected that includes an Activate task associated with an order-level gateway event for activation.
3. In the MetaSolv Solution Work Queue Manager, the order-level gateway event is initiated on the Activate task.

This results in an event containing the order key being passed to the to the integration layer.

4. The event is received by the integration layer, triggering a call to the Activation XML API to get activation information for the order.
5. The activation data returned by the Activation XML API to the integration layer is used to build and pass an order to ASAP containing CSDL command parameters to activate the POTS service.
6. ASAP returns a create order event with the ASAP order key, and the integration layer stores the ASAP order key in the gateway event.
7. ASAP activates the service and returns confirmation to the integration layer.

   The integration layer updates the status of the gateway event in MetaSolv Solution to Complete.
8. The Due Date task in MetaSolv Solution is completed and the ordered POTS line and any order features are placed in In Service status.

Alternate Data Flows for POTS Service

This section describes the alternate data flows that differ from the basic data flow described in "Basic Data Flow for a New POTS Service".

Changing POTS Service

A service change occurs when you issue a change order against an in-service POTS service or service feature. The activation data includes the same information as described for the basic flow with any changes. The service and associated feature copied to the order will have an activityCd value of iactc (Change).

When a service or feature is copied to a change order, the service will have an activityCd value of iactc (Change) even if no modifications are made on the order or in design. Information about the service or feature that is represented with its own activityCd will have an activityCd value of iacti (In-Service) unless the information is changed (activityCd = iactc) or deleted (activityCd = iactd) on the order. Such information includes:

- Telephone number
- Value labels
- PIC information

Note: If the port associated with an existing line service is to be changed, it must be handled in the way the following procedure describes for the activation data to include information for both the old port and the new port.

To change the port on a line service:

1. Copy the existing line service to a change order.
2. Disconnect the existing line service on the order.
3. Add a new line service to the order and associate the same telephone number to the new line service.

4. Complete the RID task by disconnecting the port for the connection of the existing line and assigning a new port to the connection for the new line.

This ensures that activation data is exported for two separate line services. The Activation API does not currently export the network element identifier and line equipment number (LEN) for both the old port and the new port unless they are represented as separate line services. If the port is changed on an existing line service without disconnecting and adding a new line, the activation data includes the existing line service without values populated for the lineEquipmentNumber and networkElementIdentifier attributes.

Disconnecting POTS Service

A service disconnect occurs when you issue a disconnect action against the service or feature on an order. The disconnected service and/or feature will have an activityCd value of iactd (Disconnect).

A line can be disconnected with an intercept message. An intercept message, once activated on the switch, plays a message that gives the new phone number where the owner of a pending disconnect number can be reached. The activation data includes the pending disconnect number and its associated intercept as an instance of the relatedTelephoneNumbers attribute on the line service.

In the following example, the relatedTelephoneNumbers data is part of the line service for disconnected telephone number 972-908-0046. The new number given in the intercept message is 214-529-3143.

```
lineServices (One instance of lineServiceActivationType)
…
relatedTelephoneNumbers (One instance of RelatedTelephoneNumberType)
fromTelephoneNumber = 972-908-0046
toTelephoneNumber = 214-529-3143
associationType = Intercept
interceptTypeInfo = (InterceptType)
interceptID = 1
option = Standard
name = Bill Brumbaugh
endDate = 1900-01-01T00:00:00
```

Adding Hunt Group Service

A hunt group determines how an incoming call can hunt (search) for an available line when the one dialed is not available. A hunt group consists of multiple hunt group member telephone numbers with one lead telephone number. Before a telephone number can be added to a hunt group, a line must be added to the order, and the telephone number must be assigned to the line. A hunt sequence is configured which defines what number each telephone number hunts to.

The activation data for hunt groups is included in the huntGroups attribute of the SwitchActivationType. The following example shows a HuntGroupType instance.

```
switchActivations (One instance of SwitchActivationType)
…
huntGroups (One instance of HuntGroupType)
serviceKey (MetaSolvServiceKey)
servicePrimaryKey = 2978468
itemType = SYSOPTION
itemAlias = Global Hunt
provisioningName = Hunt
valueLabels = null
```

```
activityCd = iactn
groupName = tmhunt
groupNumber = 1112
groupType = DNH - Sequential
members = (Three instances of HuntGroupMemberType)
[1]   memberNumber = 972-908-0026
[1]   memberHuntTo = 972-908-0027
[1]   activityCd = iactn
[2]   memberNumber = 972-908-0027
[2]   memberHuntTo = 972-908-0028
[2]   activityCd = iactn
[3]   memberNumber = 972-908-0028
[3]   memberHuntTo = 972-908-0026
[3]   activityCd = iactn
```

The members attribute represents the set of hunt group members. Each hunt group member is represented by a HuntGroupMemberType instance that defines the member telephone number and the number it hunts to. The lead number in a hunt group is always the first hunt group member in the list. Both the HuntGroupType and the HuntGroupMemberType have an activityCd attribute. In this example, the value of each activityCd is iactn (new).

The activation data will have a line service (LineServiceActivationType) instance with a matching telephone number for each member of the hunt group. The LineServiceActivationType instance will contain the network element identifier and other line specific activation data required to activate the hunt group member.

Line services that are also hunt group members may need to be processed with the hunt group and separate from other line services when CSDL commands are created in the integration layer. Line services that are also hunt group members can be identified by the isHuntGroupMember attribute on the LineServiceActivationType instance. If the telephone number for a line service is included as a hunt group member on any hunt group instance in the activation data, the isHuntGroupMember attribute on the line service is set to true.

Changing Hunt Group Members

You can change the order of hunt group members, add a new member, or delete a member. A line for a telephone number must be added before the telephone number can be added as a hunt member. MetaSolv Solution has a validation that forces the deletion of a hunt member if its telephone number is disconnected.

If a hunt group is copied to an order, all of the hunt group members will be included in the exported activation data. The activityCd attribute of the HuntGroupType instance will be iactc (Change). The activityCd attribute of the HuntGroupMemberType instances reflect whether the hunt group member is added (iactn), deleted (iactd), or not changed (iacti).

Note: The activityCd iactc (change) does not apply to a hunt group member. If the hunt sequence is changed, an existing hunt member is deleted and a new one is added.

Suspending POTS Service

You can suspend a customer's entire account or specific service items for non-payment through a PSR order. If a customer's account is suspended, all of the customer's services are copied to the order and suspended. Individual services and features on a line service can be copied to an order and suspended.

The activation data includes all of the services and features on the order. Any suspended items have an activityCd of iacts (Suspend).

Restoring or Disconnecting Suspended POTS Service

When an order is created to suspend a customer's service items through a PSR order, the provisioning plan should include a Suspend task, which is placed after the Due Date task. The Suspend task appears in the work queue after the Due Date task is completed and the suspend grace period has passed.

When a Suspend task is selected, the user is given the option to restore or disconnect the suspended items. The system automatically generates an order that includes the items to be restored or disconnected. The new order is then completed, and a provisioning plan is selected that includes an Activate task to initiate activation.

The exported activation data includes the restored or disconnected service items with an activityCd of iactr (Restore) or iactd (Disconnect).

Moving Line Service

When an order is created to move a POTS line from one service location to another, both service locations get placed on the order. The in-service POTS line is associated with the old service location. The move activity copies the POTS line with all its attributes, including its telephone number, if possible, to the new service location.

In the work queue, the old POTS line must be disconnected from the port address, and the new POTS line must be assigned to a port address.

The exported activation data includes the disconnected service item with an activityCd of iactd (Disconnect), which represents the service item at the old service location, and the service item with the activityCd of iactn (New), which represents the service item at the new service Location.

Best Practices for Setting Up MetaSolv Solution Data

This section details the actions taken in MetaSolv Solution to ensure the appropriate POTS activation information is available for ASAP in the flow-through package detailed in "Viewing The POTS/DSL Flow-through Package".

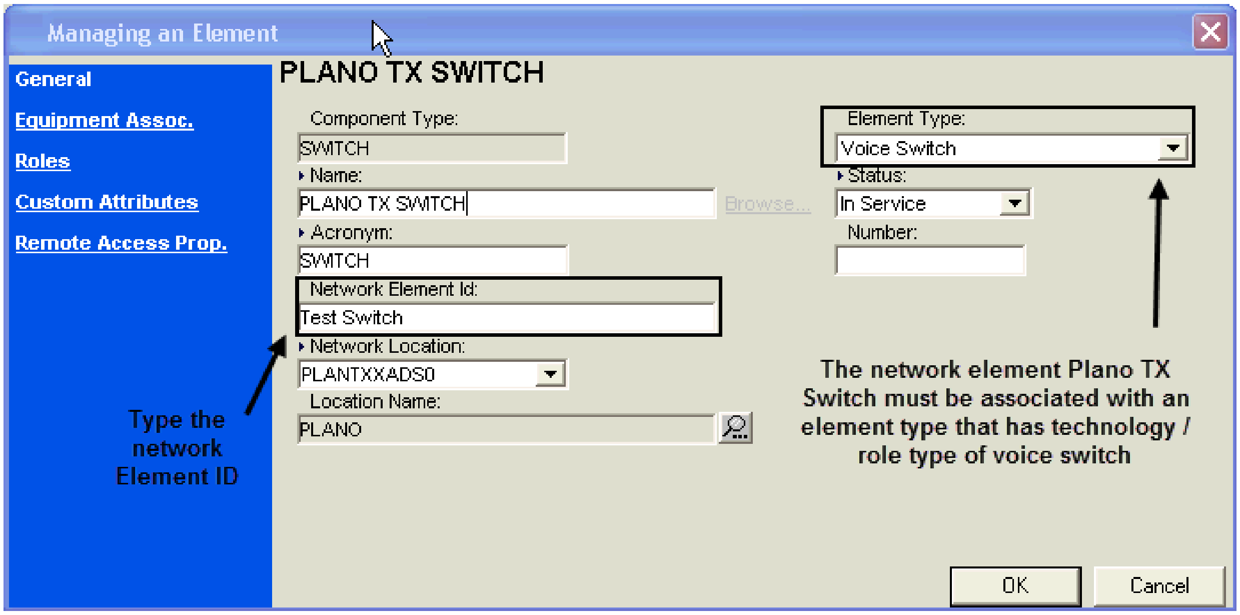
Network Elements/Equipment

The following list describes the associations and changes made to network elements or equipment to collect POTS data for activation:

- The switch used for POTS activation must be associated with a network element that is associated with a network element type with a technology/role of Voice/Switch. In the following figure, the element type name is Voice Switch. In the definition of Voice Switch, the technology is Voice, and the role is Switch.

  Figure 4–2 shows a screenshot of setting up a switch as a Network Element.

Figure 4–2 Setting Up a Switch as a Network Element



PLANO TX SWITCH

Component Type: SWITCH

Name: PLANO TX SWITCH

Acronym: SWITCH

Network Element Id: Test Switch

Network Location: PLANTXXADS0

Location Name: PLANO

Element Type: Voice Switch

Status: In Service

Number:

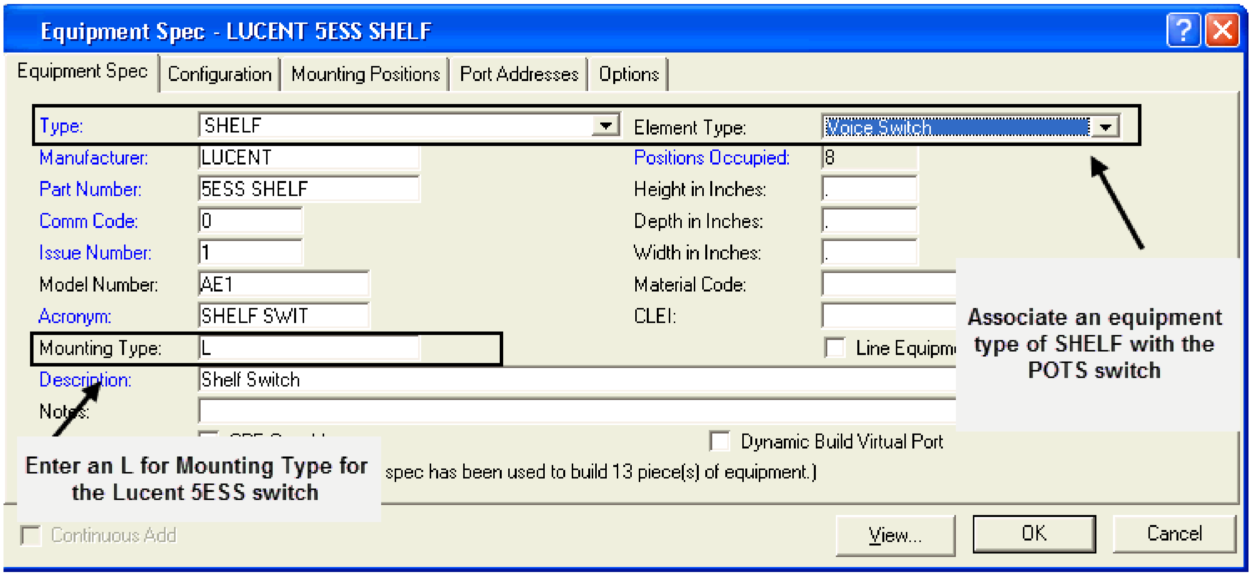
Type the network Element ID

The network element Plano TX Switch must be associated with an element type that has technology / role type of voice switch

- The equipment specification type of shelf must be associated with a network element that is associated with a network element type with a technology/role of Voice/Switch.

  Figure 4–3 shows a screenshot of associating an equipment type of shelf with the equipment specification.

Figure 4–3 Associating an Equipment Type of Shelf With the Equipment Spec



Type: SHELF

Manufacturer: LUCENT

Part Number: 5ESS SHELF

Comm Code: 0

Issue Number: 1

Model Number: AE1

Acronym: SHELF SWIT

Mounting Type: L

Description: Shelf Switch

Notes:

Element Type: Voice Switch

Positions Occupied: 8

Height in Inches:

Depth in Inches:

Width in Inches:

Material Code:

CLEI:

□ Line Equipm

□ Dynamic Build Virtual Port

spec has been used to build 13 piece(s) of equipment.)

□ Continuous Add

Associate an equipment type of SHELF with the POTS switch

Enter an L for Mounting Type for the Lucent 5ESS switch

- The Lucent 5ESS switch requires the originating equipment type. The information is required only for this switch and is used by ASAP to determine the correct format of the LEN. This value is L. It is stored in the Mounting Type field of the equipment specification.
- Every affected network location for a POTS activation must be identified as a network element. That means each piece of equipment used in a POTS activation

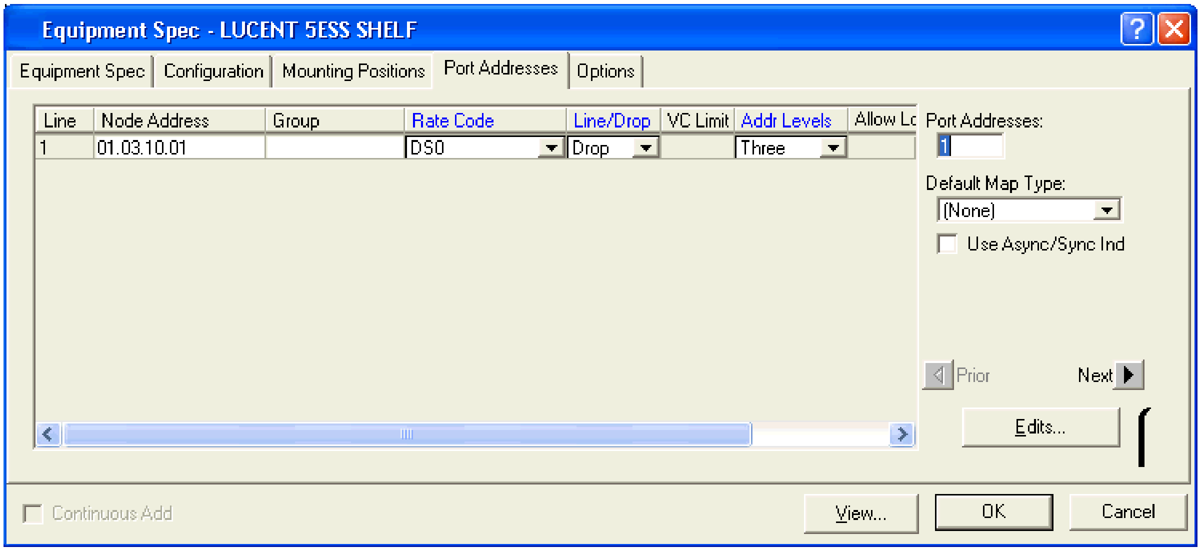
must be defined either individually or in an equipment grouping as a network element.

If network elements are not defined, the ASAP ID routing method must be employed, which involves using the MetaSolv Solution network location ID and mapping it against a network element ID maintained in ASAP.

- Node address must be populated for the assigned port. Node address is populated on the equipment specification for the equipment containing the port. This field maps to the lineEquipmentNumber in the LineServiceActivationType in the XML schema.

  Figure 4–4 shows a node address requirement for the assigned port.

Figure 4–4 Node Address Requirement for the Assigned Port



Line	Node Address	Group	Rate Code	Line/Drop	VC Limit	Addr Levels	Allow Lc
1	01.03.10.01		DS0	Drop		Three	

Port Addresses: 1

Default Map Type: (None)

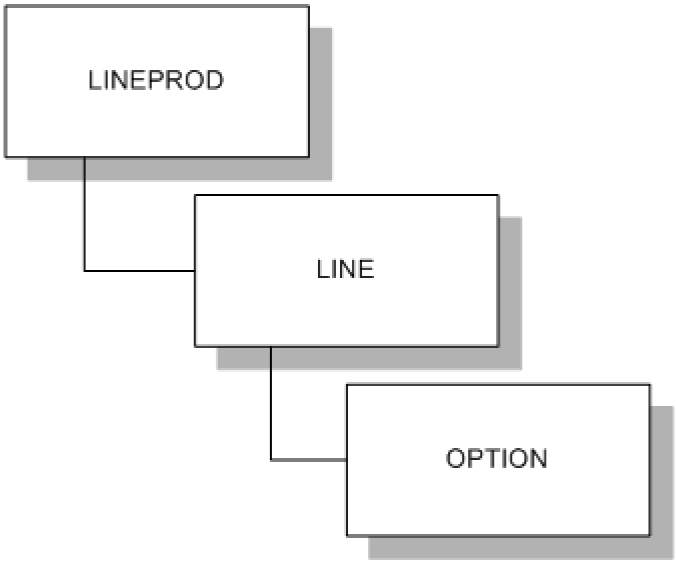
□ Use Async/Sync Ind

□ Continuous Add

Product Specification

Figure 4–5 shows the recommended hierarchy for a POTS product specification.

Figure 4–5 POTS Product Specification Hierarchy



The following list describes the types in the hierarchy:

LINEPROD: This item type represents the grouping product, but this product is not configured as a circuit or activated on a switch.

LINE: This item type represents the electrical path between a phone company central office and a subscriber with an individual phone number that can be used for incoming and out going phone calls. In MetaSolv Solution, a LINE can be associated with zero, one, or many telephone numbers. Basic POTS setup includes one telephone number. Product specification for POTS

For activation, the **Switch Provision** indicator (see Figure 4–5) should be set to **Yes** when the specification is created. This prompts the user creating an order to select a provisioning name.

OPTION: This item type represents the features that can be ordered with a line, such as call waiting, call forwarding, call blocking, and so on that the phone company or the switch is capable or supporting for the line.

> **Note:** If the service provider's equipment inventory includes multiple vendors, this means a provisioning product for each vendor must be set up.

For POTS, a way is needed to differentiate between the options (features) being ordered for POTS service. The **Provision Name** field on the product specification is used for this purpose.

Table 4–1 lists the provision name values used in the flow-through package for each level of the POTS product specification hierarchy.

Table 4–1 Provision Name Value

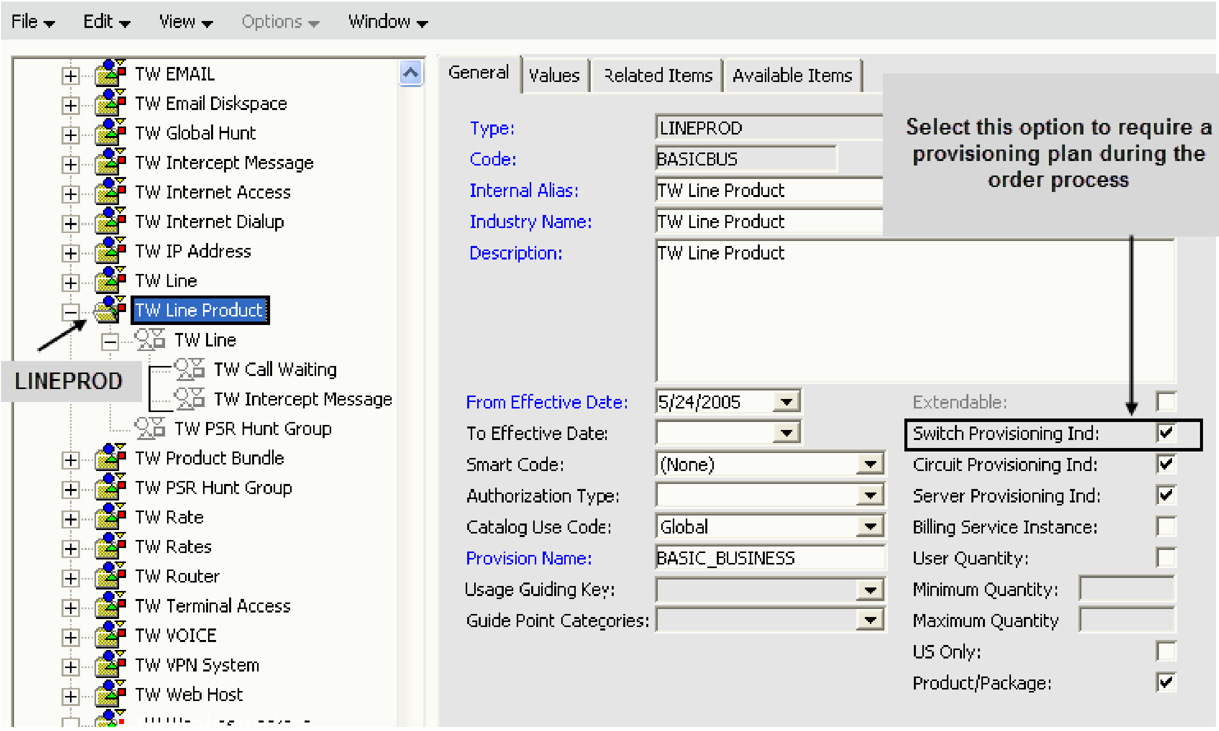
Hierarchy level	Provision Name value
LINEPROD	Provision Name=Basic_Business
LINE type	Provision Name=POTS

Table 4–1 (Cont.) Provision Name Value

Hierarchy level	**Provision Name value**
OPTION item type	Extract the name value from the CSDL command that turns up service. For example, the Lucent 5ESS ASAP CSDL command for call waiting is C_LU-5ESS_16_ADD_POTS_CCW. The part of the command that identifies the call waiting option, *CCW*, is what was used in the flow-through package as a value for the **Provision Name** field. Other OPTION item types are identified in the same manner.

Figure 4–6 shows the POTS LINEPROD product specification.

Figure 4–6 LINEPROD Product Specification



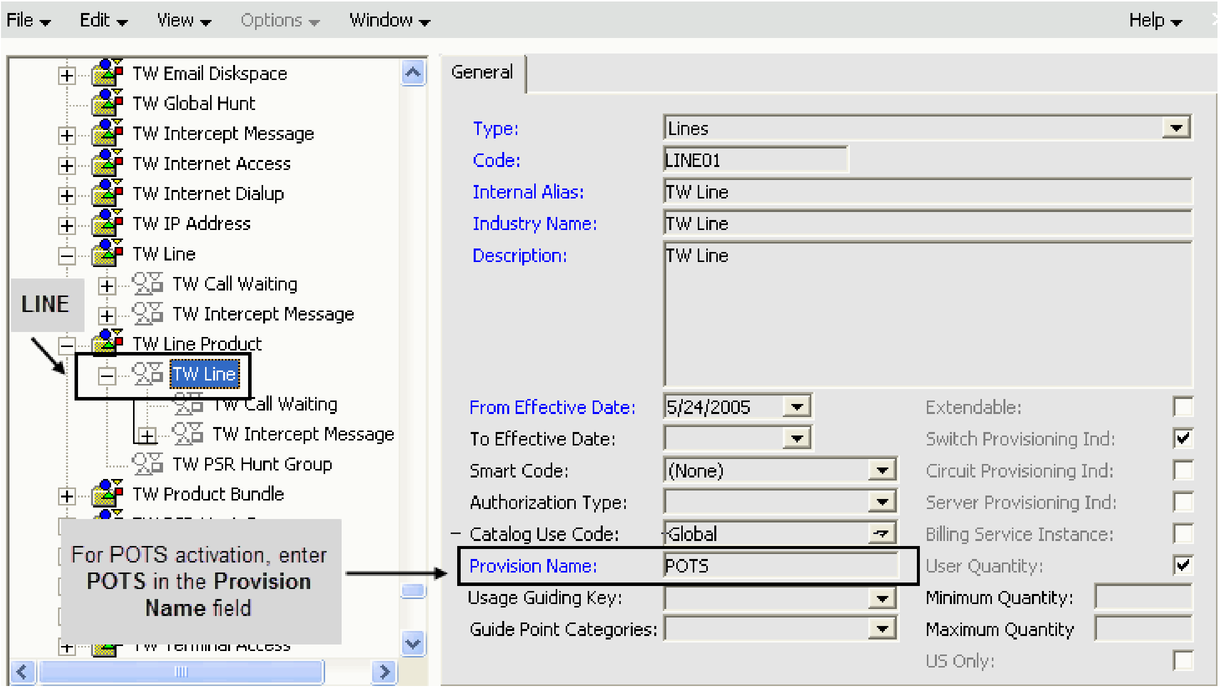
Select this option to require a provisioning plan during the order process

Type: LINEPROD
Code: BASICBUS
Internal Alias: TW Line Product
Industry Name: TW Line Product
Description: TW Line Product
From Effective Date: 5/24/2005
To Effective Date:
Smart Code: (None)
Authorization Type:
Catalog Use Code: Global
Provision Name: BASIC_BUSINESS
Usage Guiding Key:
Guide Point Categories:

Extendable: □
Switch Provisioning Ind: ☑
Circuit Provisioning Ind: ☑
Server Provisioning Ind: ☑
Billing Service Instance: □
User Quantity: □
Minimum Quantity:
Maximum Quantity
US Only: □
Product/Package: ☑

Figure 4–7 shows the line product specification. Note that the **Provision Name** field has a value of POTS.

Figure 4–7 Line Item Type Product Specification

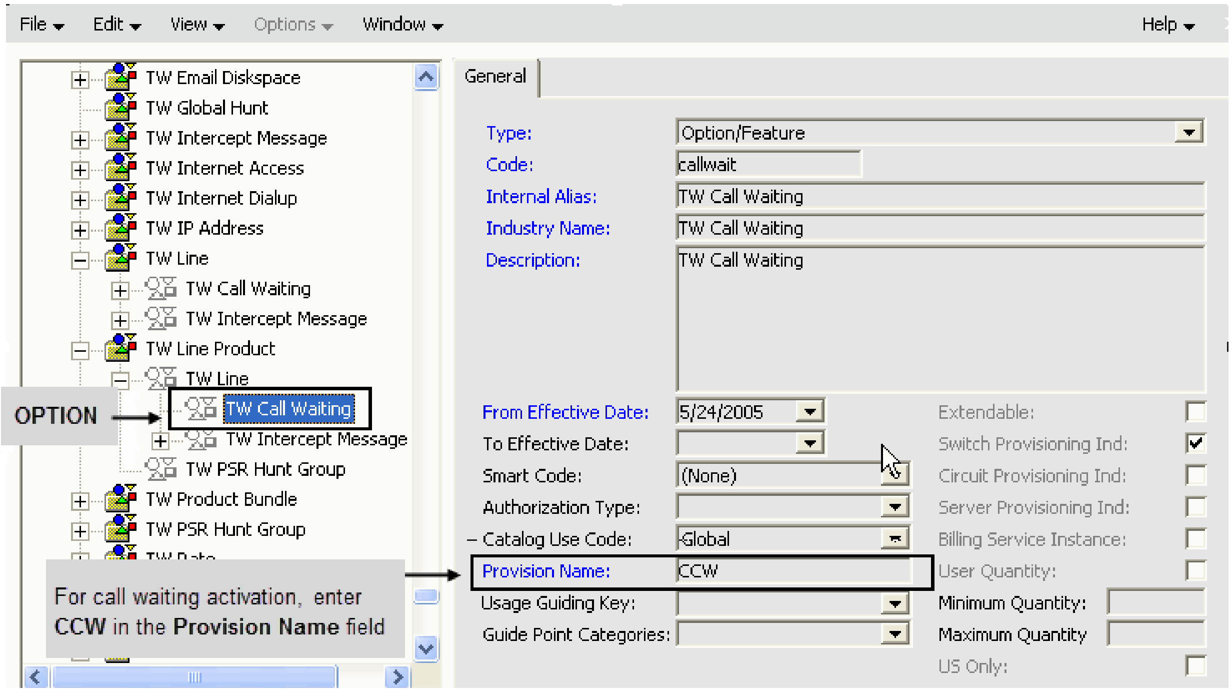


Type: Lines
Code: LINE01
Internal Alias: TW Line
Industry Name: TW Line
Description: TW Line
From Effective Date: 5/24/2005
To Effective Date:
Smart Code: (None)
Authorization Type:
Catalog Use Code: Global
Provision Name: POTS
Usage Guiding Key:
Guide Point Categories:
Extendable: □
Switch Provisioning Ind: ☑
Circuit Provisioning Ind: □
Server Provisioning Ind: □
Billing Service Instance: □
User Quantity: ☑
Minimum Quantity:
Maximum Quantity
US Only: □

For POTS activation, enter **POTS** in the **Provision Name** field

Figure 4–8 shows the option item type specification for call waiting. Note that the **Provision Name** field has a value of **CCW**.

Figure 4–8 Option Item Type Product Specification



Type: Option/Feature
Code: callwait
Internal Alias: TW Call Waiting
Industry Name: TW Call Waiting
Description: TW Call Waiting
From Effective Date: 5/24/2005
To Effective Date:
Smart Code: (None)
Authorization Type:
Catalog Use Code: Global
Provision Name: CCW
Usage Guiding Key:
Guide Point Categories:
Extendable: □
Switch Provisioning Ind: ☑
Circuit Provisioning Ind: □
Server Provisioning Ind: □
Billing Service Instance: □
User Quantity: □
Minimum Quantity:
Maximum Quantity
US Only: □

For call waiting activation, enter **CCW** in the **Provision Name** field

Table 4–2 lists the optional parameters that can be set up in the product specification as value labels for the Lucent 5ESS switch.

Table 4–2 Lucent 5ESS Switch Optional Parameters

Optional parameters	Type
Activate/de-activate on a per line basis the calling identify delivery on call waiting special features	LINE
Carrier access code dialing permissions	LINE
Indication whether or not terminating calls for this TN will be intercepted and send to regular intercept	LINE
Line Class Code	LINE
Multi frequency ringing individual type	LINE
All call privacy indicator	LINE
Deny or suspend termination treatment	LINE
Call waiting Deluxe Alert type	OPTION (CALL WAITING)
Long distance distinctive ring type	OPTION (CALL WAITING)
Usage sensitive Conference calling	OPTION (CALL WAITING)
Activate/de-activate on a per line basis the calling identify delivery on call waiting special features	LINE

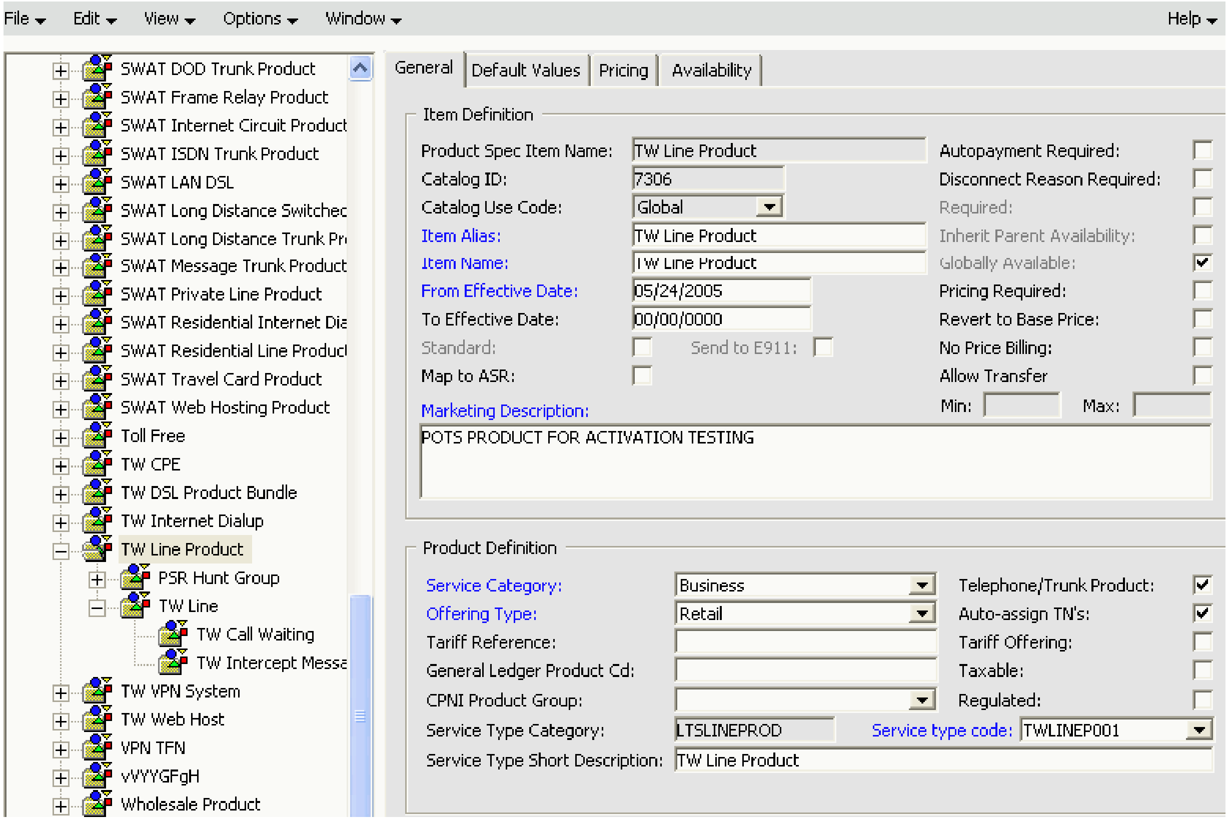
The set of optional parameters is different for each network element that service is activated against. For example, the optional parameters would be different for a Nortel or Cisco switch. The ASAP parameters TN and PIC are captured on the order.

Do not use MetaSolv Solution EWOs to affect mass port changes on telephone number circuits. None of the associated features (options) are sent along with the port activations. To solve this problem, an additional work flow must be set up outside of MetaSolv Solution to affect the change of all of the associated features. Because features are not inventoried with circuits, they are not picked up in an EWO or ISR order.

Product Catalog Entry For POTS

Figure 4–9 shows a product catalog entry built from the sample product specifications. See "Product Specification" for more information.

Figure 4–9 Sample Product Catalog Entry For POTS



Item Definition

Product Spec Item Name: TW Line Product
Catalog ID: 7306
Catalog Use Code: Global
Item Alias: TW Line Product
Item Name: TW Line Product
From Effective Date: 05/24/2005
To Effective Date: 00/00/0000
Standard: ☐ Send to E911: ☐
Map to ASR: ☐
Autopayment Required: ☐
Disconnect Reason Required: ☐
Required: ☐
Inherit Parent Availability: ☐
Globally Available: ☑
Pricing Required: ☐
Revert to Base Price: ☐
No Price Billing: ☐
Allow Transfer ☐
Min: Max:
Marketing Description:
POTS PRODUCT FOR ACTIVATION TESTING

Product Definition

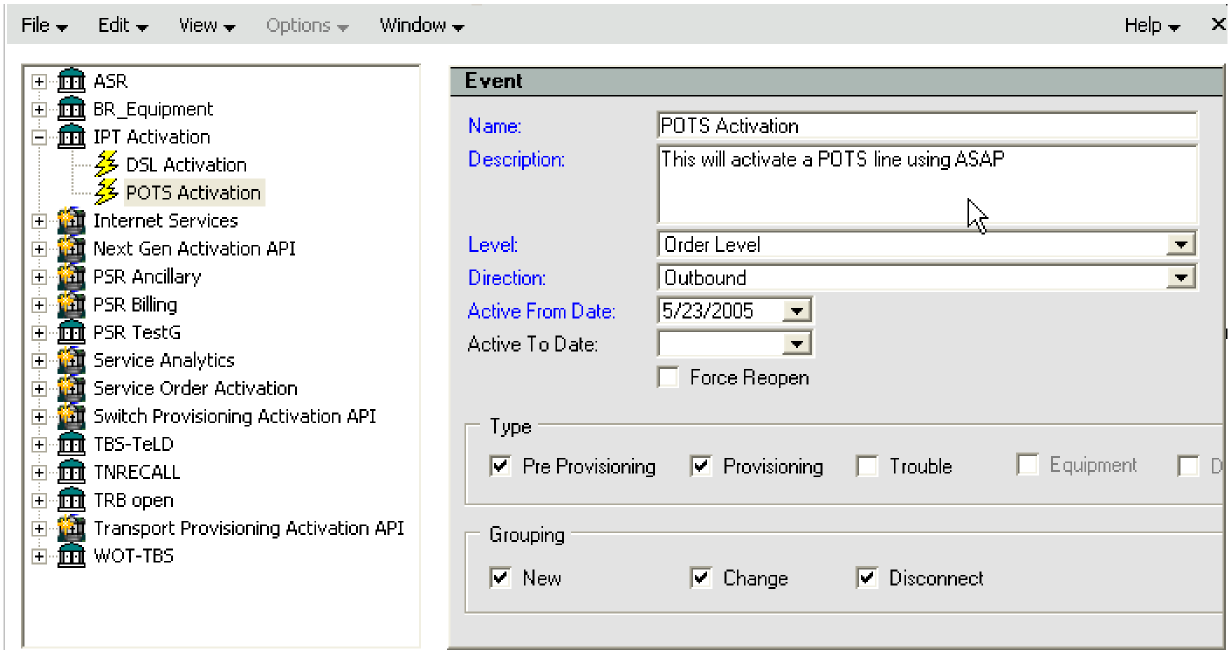
Service Category: Business
Offering Type: Retail
Tariff Reference:
General Ledger Product Cd:
CPNI Product Group:
Service Type Category: LTSLINEPROD
Service type code: TWLINEP001
Service Type Short Description: TW Line Product
Telephone/Trunk Product: ☑
Auto-assign TN's: ☑
Tariff Offering: ☐
Taxable: ☐
Regulated: ☐

Gateway event

You must create a gateway event for POTS activation.and associate it with a task on the provisioning plan for POTS activation.

Figure 4–10 shows the gateway event for POTS activation.

Figure 4–10 POTS Activation Gateway Event



- ASR
- BR_Equipment
- IPT Activation
  - DSL Activation
  - POTS Activation
- Internet Services
- Next Gen Activation API
- PSR Ancillary
- PSR Billing
- PSR TestG
- Service Analytics
- Service Order Activation
- Switch Provisioning Activation API
- TBS-TeLD
- TNRECALL
- TRB open
- Transport Provisioning Activation API
- WOT-TBS

Event

Name: POTS Activation

Description: This will activate a POTS line using ASAP

Level: Order Level

Direction: Outbound

Active From Date: 5/23/2005

Active To Date:

□ Force Reopen

Type

☑ Pre Provisioning ☑ Provisioning □ Trouble □ Equipment □ D

Grouping

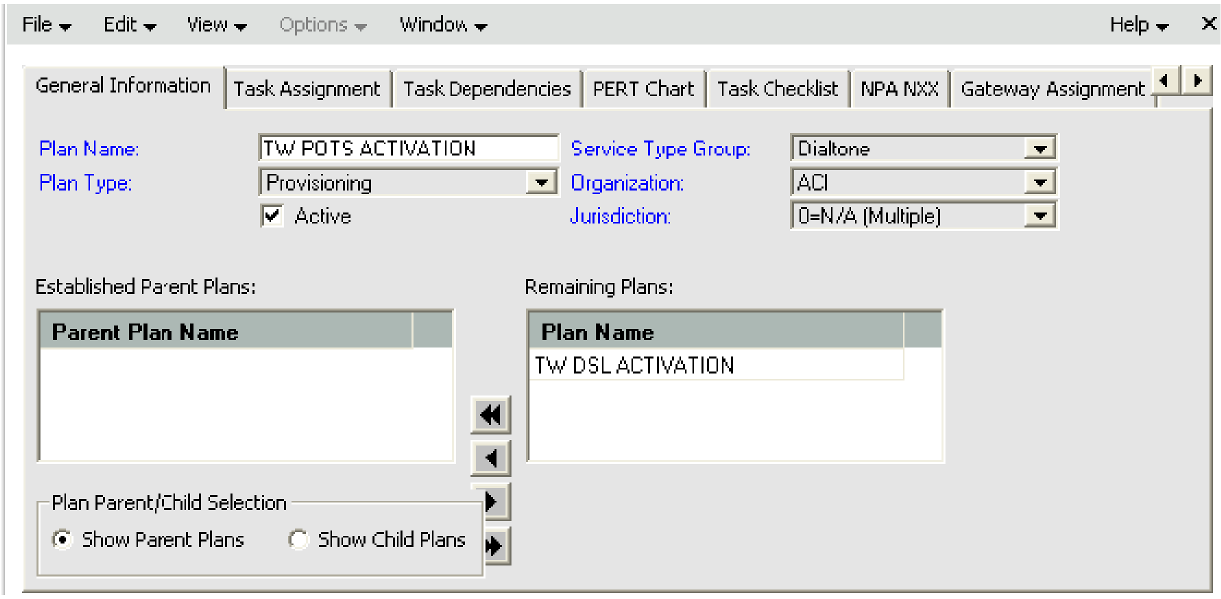
☑ New ☑ Change ☑ Disconnect

Provisioning Plan

You must create or select a provisioning plan and associate it with a gateway event set up for POTS activation.

Figure 4–11 shows the provisioning plan used for the POTS flow-through package.

Figure 4–11 Provisioning Plan Setup



Plan Name: TW POTS ACTIVATION

Plan Type: Provisioning

☑ Active

Service Type Group: Dialtone

Organization: ACI

Jurisdiction: 0=N/A (Multiple)

Established Parent Plans:

Parent Plan Name

Remaining Plans:

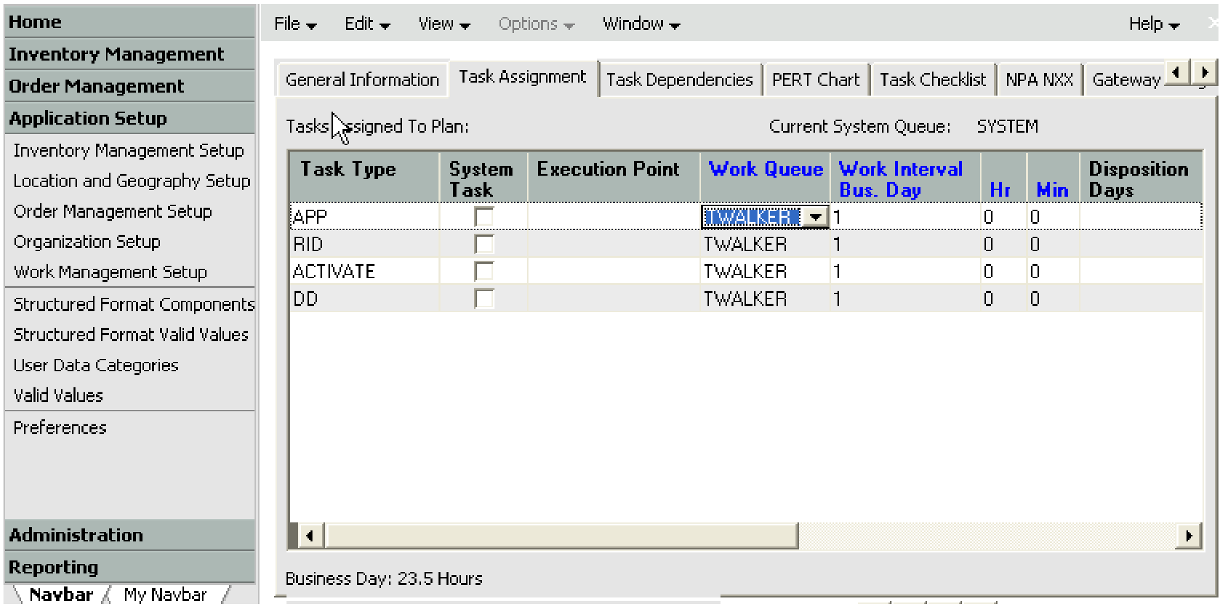
Plan Name
TW DSL ACTIVATION

Plan Parent/Child Selection

● Show Parent Plans ○ Show Child Plans

Figure 4–12 shows the POTS tasks that must be completed in Work Management for service to be activated.

Figure 4–12 Defining Tasks for the Plan



Tasks Assigned To Plan:

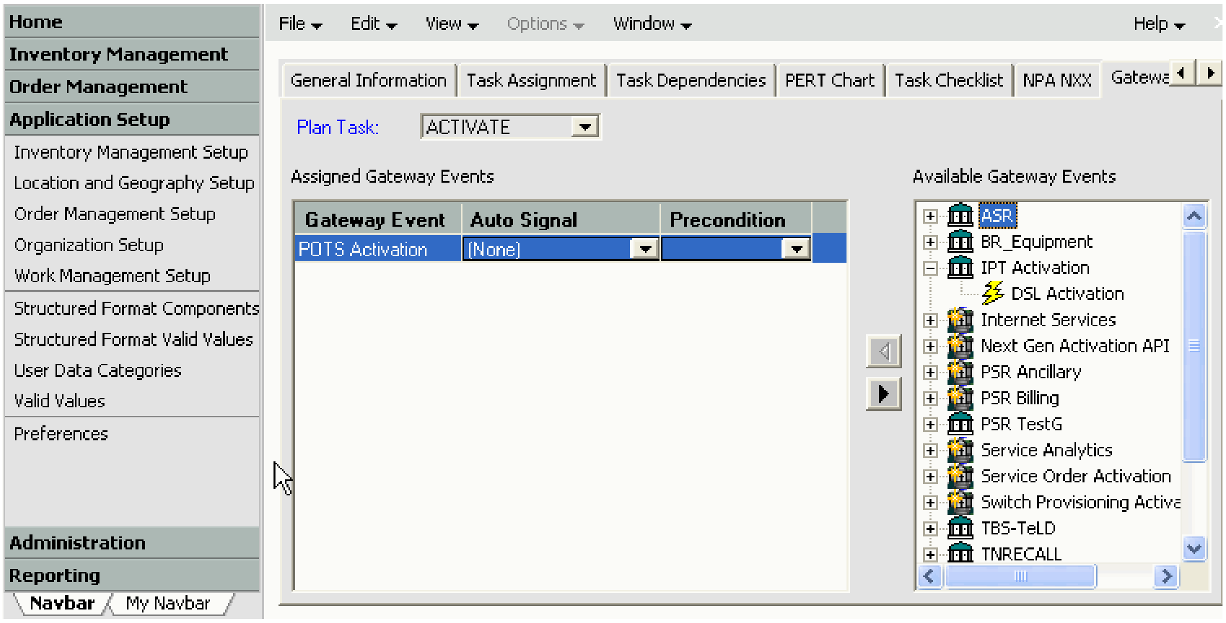
Current System Queue: SYSTEM

Task Type	System Task	Execution Point	Work Queue	Work Interval Bus. Day	Hr	Min	Disposition Days
APP	☐		TWALKER	1	0	0	
RID	☐		TWALKER	1	0	0	
ACTIVATE	☐		TWALKER	1	0	0	
DD	☐		TWALKER	1	0	0	

Business Day: 23.5 Hours

Figure 4–13 shows the gateway event associated with the Activate task in the provisioning plan.

Figure 4–13 Associating a POTS Gateway Event With the Activate Task



Plan Task: ACTIVATE

Assigned Gateway Events

Gateway Event	Auto Signal	Precondition
POTS Activation	(None)	

Available Gateway Events

- ASR
- BR_Equipment
- IPT Activation
  - DSL Activation
- Internet Services
- Next Gen Activation API
- PSR Ancillary
- PSR Billing
- PSR TestG
- Service Analytics
- Service Order Activation
- Switch Provisioning Activa
- TBS-TeLD
- TNRECALL

Custom Attributes

Any parameter required by the ASAP cartridge but not yet defined in MetaSolv Solution must be defined as a custom attribute. Custom attributes can be defined for network elements, connections, and the relationships between the physical and virtual connection.

Custom attributes are set up in the MetaSolv Solution Utilities module. See the MetaSolv Solution online Help for information on setting up a custom attribute.

For the Lucent products, ASAP requires the manufacturer, technology (switch, DSLAM, and so on) and version. This information is not captured by MetaSolv Solution. Custom attributes were created for these three items in the MetaSolv Solution Utilities module and associated with the element. *** 'Custom Attributes for the Lucent Stinger DSLAM Network Element' on page 23 *** shows custom attributes for the Lucent Stringer DSLAM. The figure shows three separate fields for the information on the Lucent equipment, but you could as an alternative capture the information in a single field if you concatenate the information (for example, LU_DSLAM_9-6-1).

Mapping POTS Data

Table 4–3 lists the ASAP commands required for POTS activation on a Lucent 5ESS switch.

Table 4–3 Activation Commands for the Lucent 5ESS Switch

ASAP Command	Activation tasks
C_LU-5ESS_16_ADD_POTS-RES-LINE	Adds a POTS residential line.
C_LU-5ESS_16_ADD_POTS-CCW	Adds POTS call waiting.
C_LU-5ESS_16_CHG_POTS-CCW	Changes POTS call waiting.
C_LU-5ESS_16_DEL_POTS-CCW	Deletes POTS call waiting.
C_LU-5ESS_16_DEL_POTS-RES-LINE	Deletes POTS line.
C_LU-5ESS_16_SET_POTS-PIC	Sets the PIC.
C_LU-5ESS_16_REMOVE_POTS-PIC	Removes the PIC.
C_LU-5ESS_16_QRY_POTS-LINE	Queries the POTS line.

The commands are different for each switch used for activation. ASAP provides a cartridge guide containing the appropriate commands for each switch that it supports for activation. Consult the cartridge guide for the switch you plan to activate to see a listing of the commands and their parameters.

Mapping Data for the Switch

Table 4–4 maps the parameters of the CSDL command to add a POTS residential line (C_LU-5ESS_16_ADD_POTS-RES-LINE) for the Lucent 5ESS switch with the corresponding field in the classes returned by the MetaSolv Solution Activation API.

Table 4–4 Mapping CSDL Parameters for C_LU-5ESS_16_ADD_POTS-RES-LINE to MetaSolv Solution Fields

Parameter	Description	Req/Opt	Default	MSS Activation API Mapping
CHNGPTY	Party position of telephone number	R	I	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
CIDCW	Activate/deactivate on a per line basis the calling identity delivery on call waiting special feature	O	N	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.

Table 4–4 (Cont.) Mapping CSDL Parameters for C_LU-5ESS_16_ADD_POTS-RES-LINE to MetaSolv Solution Fields

Parameter	Description	Req/Opt	Default	MSS Activation API Mapping
CIDIAL	Carrier access code dialing permissions	O	-	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
ICP	Indicates whether or not terminating calls to this TN will be intercepted and send to regular intercept	O	N	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
LCC	Line class code	R	1FR	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
LEN	Line equipment number	R	-	SwitchActivationType.lineService s. lineEquipmentNumber
MFRI	Multi frequency ringing individual type	O	N	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
NE_ID_ LU-5ESS	Host NE identifier	R	-	SwitchActivationType.lineService s. networkElementIdentifier
OETYPE	Originating equipment type	R	-	SwitchActivationType.lineService s. originatingEquipmentType
PIC	Pre-subscribed inter-LATA carrier identification number	O	-	SwitchActivationType.lineService s. picInformation.pic where picTypeCode = Interlata
PLIT	Prohibit automatic line insulation test	O	Y	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
PRIVACY	All call privacy indicator	O	N	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
PSLIM	Prohibit subscriber line and instrument test	O	Y	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
PTC	Pre-subscribed inter-LATA toll carrier identification number	O	-	
RAX	Rate center the switch is located in	O	-	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.

Table 4–4 (Cont.) Mapping CSDL Parameters for C_LU-5ESS_16_ADD_POTS-RES-LINE to MetaSolv Solution Fields

Parameter	Description	Req/Opt	Default	MSS Activation API Mapping
SUSO	Deny or suspend origination treatment	O	N	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
SUST	Deny or suspend termination treatment	O	N	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.
TN	Telephone Number	R	-	SwitchActivationType.lineServices. telephoneNumber.telephoneNumber
TTC	Touch tone calling	O	Y	Not captured in MetaSolv Solution. Accept the default, or create a custom attribute on the network element or a value/label on the product catalog item.

Table 4–5 maps the parameters of the CSDL command to add call waiting (C_LU-5ESS_16_ADD_POTS-CCW) for the Lucent 5ESS switch with the corresponding field in the XML types returned by the MetaSolv Solution Activation API.

Table 4–5 Mapping CSDL Parameters for C_LU-5ESS_16_ADD_POTS-CCW to MetaSolv Solution Fields

Parameter	Description	Req/Opt	Default	MSS Activation API Mapping
MCLI	Host NE identifier	R	-	SwitchActivationType.lineServices. networkElementIdentifier
TN	Telephone Number	R	-	SwitchActivationType.lineServices. telephoneNumber.telephoneNumber

Mapping Example

This section presents a simple example following the basic flow described in "Basic Data Flow for a New POTS Service". The basic flow describes an order for new POTS service. The example explains how to navigate through the activation data returned by the Activation XML API to find the values necessary to populate the parameters for the following CSDL command:

```
C_LU-5ESS_16_ADD_POTS-RES-LINE.
```

Service Provider's Switch Equipment and Network Element

To provide POTS service to customers, a service provider builds out a network system containing switch network elements and associates the switch equipment to the network elements. The network elements are identified as switches because they are associated with a network element type that has a technology/role of Voice/Switch. Each switch network element is assigned a network element identifier that matches the element identifier in ASAP.

In this example, the switch technology is a Lucent 5ESS. The network element for this example is named PLANO TX SWITCH and has a network element identifier of 5ESS_ 1.

The following custom attribute is set up for the PLANO TX SWITCH network element and given the value specified. This custom attribute is used to identify the correct ASAP cartridge to use in creating a CSDL command for a new POTS line.

Cartridge = LU-5ESS 16

The values for the line equipment number (LEN) for each port are captured in MetaSolv Solution on the equipment specification for the cards contained in the switch equipment. The value entered as the node address for a port maps to the LEN parameter on the CSDL.

Because this example uses the Lucent 5ESS switch, the originating equipment type is a required parameter on the CSDL to add a new POTS line. This parameter identifies the format of the LEN. The value for the originating equipment type is captured in MetaSolv Solution in the Mounting Type field on the equipment specification of the switch shelf equipment (see Figure 4–2, "Setting Up a Switch as a Network Element").

Customer POTS Line Order

A new order is created for a POTS line with a Call Waiting feature. The switch network location PLANTXXADS0 is selected, and telephone number 972-881-0000 is assigned. The following values are entered for the Value Labels associated with the line service:

- RCC = 1RF
- RAX = 1

The following PIC values are entered on the order:

- International = 6123
- Intralata = 5500
- Interlata = 113

A provisioning plan is selected that includes:

- A RID task to design the connection for the telephone number
- An Activate task to initiate ASAP activation
- A Due Date task to complete the order and place the line in service

Connection Design

In completing the RID task, the connection for the line service is created and assigned to a port. The port is on a card that is contained in a shelf that is associated to the network element PLANO TX SWITCH.

Activation Data

Once the RID task is complete, the activation process is initiated from the Activate task in the MetaSolv Solution Work Queue Manager. This results in a gateway event containing the order key being passed to the service provider's MetaSolv Solution to ASAP integration layer. The event is received by the integration layer triggering a call to the Activation XML API to get activation information for the order. This data is used by the integration layer to build and pass an order to ASAP containing command parameters to activate the line service and call waiting feature.

This section describes how to access the data values for the example described above from the activation data and map them to the parameters on the CSDLs to add a new POTS line and call waiting.

The XML complex type containing the activation data is MetaSolvServiceActivationType. This type contains data for all services on the order that can be activated.

Table 4–6 describes the attributes or types in the MetaSolvServiceActivationType XML complex data type.

Table 4–6 MetaSolvServiceActivationType

Attribute or Type	Description
orderDetails	Type: OrderInformationType This information represents information about the order that the activation applies to.
switchActivations	Array of SwitchActivationType Represents a collection of switch related services on the order such as POTS. The collection contains a SwitchActivationType instance for every switch location associated with the ordered services.
networkElements	Array of NetworkElementType Represents a collection of network elements which meet any of the following criteria: ▪ The network element is an item on the order. ▪ A connection on the order is assigned to the network element. ▪ A connection on the order is allocated to a connection that is assigned to the network element.
networkSystems	Array of NetworkSystemType Represents a collection of ordered network systems such as a virtual private network (VPN). The data includes information about the network system and all network elements and connections contained within the network system.

In the example, the orderDetails attribute is an instance of type OrderInformationType and is populated in the following manner:

```
companyName =
firstName = Peter
lastName = Goodman
orderNumber = 1375517
serviceCategory = Residential
purchaseOrderNumber =
desiredDueDate = 2005-07-06T00:00:00
supplementType =
rderType = SO
```

The following comments describe values relevant to this example:

- **orderNumber** 1375517 is the key to the order in MetaSolv Solution.
- **serviceCategory** identifies the customer account as Residential. This attribute indicates tell whether a CSDL for a residential line or a business line should be passed to ASAP.

- **desiredDueDate** 2005-07-06T00:00:00 is the date the service is to be activated. All date attributes returned by the Activation API are in the format *yyyy-MM-dd'T'HH:mm:ss*.
- **orderType** identifies the MetaSolv Solution order type as SO, which is a PSR order. Nothing in MetaSolv Solution prevents another type of order such as an Engineering Work Order (EWO) from being assigned a task in its provisioning plan that initiates an activation gateway event. Currently, the Activation API should only be used for PSR orders, so Oracle recommends using this attribute to filter out other order types.

In the example, the switchActivation attribute contains a single instance of type SwitchActivationType. If another service on the order was associated with a different switch location, there would be more than one SwitchActivationType instance populated. The SwitchActivationType for the example would be populated as:

```
switchNetworkLocationCode = PLANTXXADS0
lineServices (One instance of type LineServiceActivationType)
serviceKey (MetaSolvServiceKey)
servicePrimaryKey = 5040033
itemType = LINE
itemAlias = TW Line
provisioningName = POTS
valueLabels (Two instances of ValueLabelType)
[1]   label = LCC
[1]   value = 1FB
[1]   activityCd = iactn
[2]   label = RAX
[2]   value = 1
[2]   activityCd = iactn
activityCd = iactn
lineEquipmentNumber = 15
networkElementIdentifier = 5ESS_1
networkElementName = PLANO TX SWITCH
originatingEquipmentType = L
isHuntGroupMember = false
telephoneNumber (TelephoneNumberType)
telephoneNumber = 972-881-0000
activityCd = iactn
picInformation (Three instances of PicInfoType)
[1]   pic = 6123
[1]   picTypeCd = International
[1]   freezePic =
[1]   partyName = AICN
[1]   activityCd = iactn
[2]   pic = 5500
[2]   picTypeCd = Intralata
[2]   freezePic =
[2]   partyName = IN REGION
[2]   activityCd = iactn
[3]   pic = 113
[3]   picTypeCd = Interlata
[3]   freezePic =
[3]   partyName = ACI
[3]   activityCd = iactn
lineFeatures (One instance of ServiceType)
serviceKey (MetaSolvServiceKey)
servicePrimaryKey = 5040421
itemType = OPTION
itemAlias = Call Waiting
```

```
provisioningName = CCW
valueLabels =
activityCd = iactn
relatedTelephoneNumbers =
elementCustomAttributes = (One instance of ActivationCustomAttributeType)
caUsageId = 101041
label = Cartridge
caValues = (One instance of ActivationCustomAttributeValueType)
value = LU-5ESS 16
unitOfMeasure =
systemOptions =
huntGroups =
```

The bolded attributes map to required parameters on the CSDL command C_LU-5ESS_16_ADD_POTS-RES-LINE (see "Mapping POTS Data" for parameter mappings). Other attributes worth noting include:

- **elementCustomAttributes** contains the custom attribute label Cartridge with a value of LU-5ESS 16. This is the custom attribute associated with the switch network element used to identify the applicable ASAP cartridge.
- **activityCd** is an attribute on the line service or line feature data, which are ordered services. The value iactn indicates that the services are new. The same attribute is on telephoneNumber, picInformation, and valueLabels, which are attributes of the line service.
- **telephoneNumber** is the formatted value including any characters that delimit the components of the telephone number. If necessary, the telephone number may be parsed based on the delimiting character to get the individual components, such as NPA, NXX and line range.

The CSDL parameters populated with the activation data values include:

```
NE_ID_LU-5ESS  =  5ESS_1
LEN  =  15
OETYPE  =  L
TN  =  972-881-0000
LCC  =  1FB
RAX  =  1
```

In this example, the line service contains a single lineFeature. All line features have itemType of OPTION; therefore, the provisioningName value CCW is used to identify the specific feature. The line feature has an activityCd value of iactn (New). A feature can also have valueLabels containing activation data. In this example, there are no value labels.

```
lineFeatures (One instance of ServiceType)
serviceKey (MetaSolvServiceKey)
servicePrimaryKey = 5040421
itemType = OPTION
itemAlias = Call Waiting
provisioningName = CCW
valueLabels =
activityCd = iactn
```

The CDSL command C_LU-5ESS_16_ADD_POTS-CCW is created to add the call waiting feature using the networkElementId and telephoneNumber values:

```
MCLI  =  5ESS_1
TN  =  972-881-0000
```

5

Planning For DSL Activation

This chapter describes how the DSL flow-through package was planned and shows the resulting setup in Oracle Communications MetaSolv Solution. The package shows a MetaSolv Solution/ASAP integration that allows a DSL order entered in MetaSolv Solution to be activated automatically by ASAP. The flow-through package assumes the used of a Lucent Stinger DSLAM. The ASAP document used in the integration effort is: the *Lucent Stinger and MRT DSLAM Cartridge Guide* (First Edition).

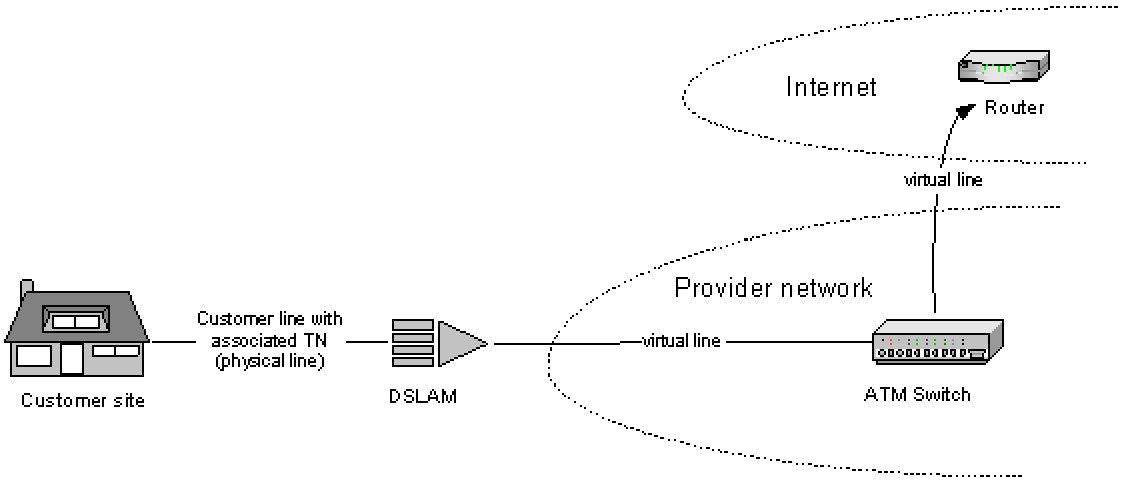
The DSL flow-through package describes the data required to activate DSL service after resources have been allocated by MetaSolv Solution. MetaSolv Solution 6.0.5 does not include virtual path functionality, which means that the internet connection or virtual connection cannot be preprovisioned.

DSL service is represented by two service items based on a physical connection and a virtual connection. The service items are based on the Connectors product specification and represent the connections between two network components.

DSL service provisioning involves both the assignment of the equipment resources and the allocation of the virtual channel assignment. Physical connections are the connections that have port assignments. Virtual connections are those allocated on one or more physical connections and represent a logical link between endpoints.

Figure 5–1 shows a basic DSL service setup.

Figure 5–1 Basic DSL Service

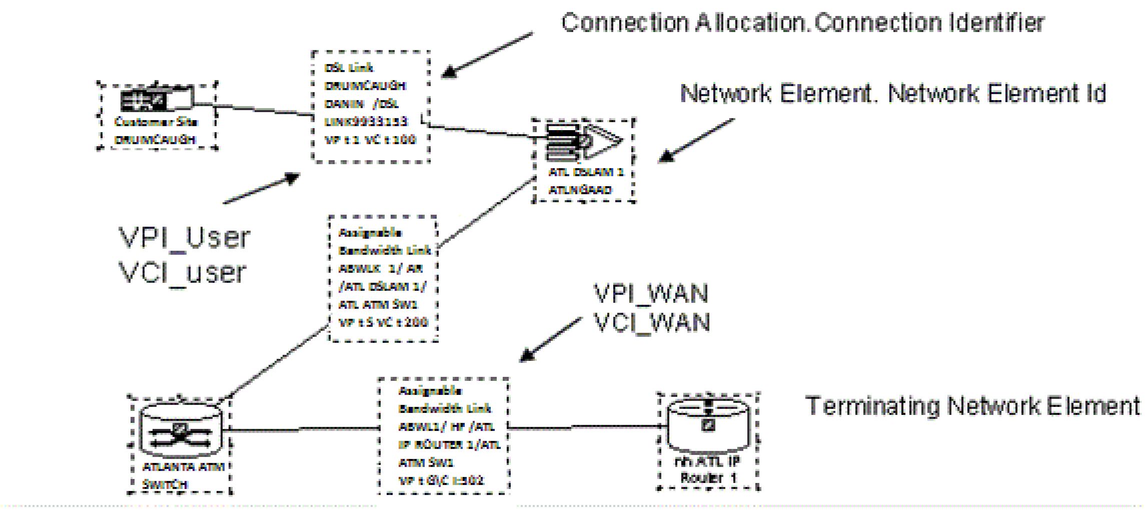


MetaSolv Solution exports the latest issue number for all connections on an order (based on a status of issued on the design lines) when the export request is received.

When a virtual connection is configured in MetaSolv Solution, you must specify its terminating network element.

Figure 5–2 shows a virtual connection design.

Figure 5–2 Virtual Connection Design



Basic Data Flow for a New DSL Service

This section describes the basic data flow from the time an order is entered into MetaSolv Solution until the order is activated by ASAP.

1. An order for DSL service is entered in MetaSolv Solution.

   The order consists of a new physical connection of type DSL Link and a new virtual connection of type Internet Access.

2. A provisioning plan is selected that includes an activation task that is associated with an order-level gateway event for activation.

3. The DSL link connection is designed in MetaSolv Solution as a physical connection linking the customer site to a DSL Access Multiplexer (DSLAM) which is a network element in the provider's network.

   The connection is assigned to an available port on the DSLAM.

4. The internet access connection is designed in MetaSolv Solution as a virtual connection beginning at the customer site, running through the DSLAM to an ATM switch and terminating at an IP Router.

   The ATM switch and IP router are both network elements in the providers network. The physical connection between the DSLAM and the ATM switch is of type Assignable Bandwidth Link.

   As part of the design, the virtual channel assignments (Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI)) are specified for the allocation of the virtual connection to a physical connection in the following situations:

- On the user side of the DSLAM (the DSL link connection from the customer site to the DSLAM)
- On the wide area network (WAN) side of the DSLAM (the Assignable Bandwidth Link connection from the DSLAM to the ATM Switch)

5. In the MetaSolv Solution Work Queue Manager, the order-level gateway event is initiated on the Activate task.

   This results in an event containing the order key being passed to MetaSolv Solution and to the ASAP integration layer.

6. The event is received by the integration layer, triggering a call to the Activation XML API to get activation information for the order.
7. The activation data is used in the integration layer to build and pass an order to ASAP containing command parameters to activate the DSL service.

   See section "Mapping DSL Data" for information on mapping ASAP command parameters to the activation data returned from the Activation API.

8. ASAP returns a create order event with the ASAP order key, and the integration layer stores the ASAP order key in the gateway event.
9. ASAP activates the service and returns confirmation to the integration layer.

   The integration layer updates the status of the gateway event in MetaSolv Solution to Complete.

10. The Due Date task in MetaSolv Solution is completed and the ordered DSL link and internet access connections are placed in In Service status.

Alternate Data Flows for DSL Service

This section describes the alternate data flows that differ from the basic data flow described in "Basic Data Flow for a New DSL Service".

Changing a Custom Attribute on the DSL Link Connection

This flow involves changing custom attributes on the order for a connection. Typically, one or more custom attribute values is added, changed, or deleted on the DSL Link connection in the Connection Design window.

1. An order is created and an existing in-service DSL Link connection is copied to the order.
2. A provisioning plan is selected that includes an Activate task associated with an order-level gateway event.
3. One or more custom attribute values is added, changed, or deleted on the DSL Link connection design.
4. Resume the Basic Flow at step 5.

Note: This change can also occur on the Internet Access virtual connection. This connection also has custom attributes.

Activation Data Result

The activation data exported by the Activation API includes information about the same set of network elements and connections as described for the basic flow. All items in the following list apply to connection information under the DSLAM network element:

- The ConnectionAssignmentType instance representing the in-service DSL Link connection will have the field activityCd = iactc (Change), indicating that the connection was included on the order. An existing service that is included on an order will have the field activityCd = iactc, even if no properties of the service have been modified.
- The ConnectionAssignmentType instance representing the DSL Link connection will have the field **connectionCustomAttributes** populated with a collection of ActivationCustomAttributeType instances representing all of the custom attributes for the DSL Link connection, including custom attributes that are not populated with values. This collection reflects the current state of the custom attribute values, including any changes that were applied. There is no activity indicator associated with each individual custom attribute to identify which custom attribute values were added or changed.

> **Note:** Because the change order included the DSL Link connection and not the Internet Access connection, the ConnectionAssignmentType instance representing the Internet Access connection will have a blank **activityCd** field. In addition, the following attributes on the ConnectionAssignmentType instance will not be populated if the connection is not included on the order:
>
> - serviceKey.primaryKey
> - itemType
> - itemAlias
> - provisioningName
> - valueLabels

Changing Value/Labels on the DSL Link Connection

This flow involves changing label and value information on the order for a connection. Typically, custom attributes are used in place of value/labels to collect activation data for connections.

Alternate Flow Steps

1. An order is created and an existing in-service DSL Link connection is copied to the order.
2. One or more value/labels for the connection is added, changed, or deleted on the order (on the Values tab when the connection is selected on the order).
3. A provisioning plan is selected that includes an Activate task associated with an order-level gateway event.
4. Resume the basic flow at step 5.

Activation Data Result

The activation data exported by the Activation API includes information about the same set of network elements and connections as described for the basic flow. All of the items in the following list apply to connection information under the DSLAM network element:

- The ConnectionAssignmentType instance representing the port for the DSL Link connection will have the field activityCd = iactc, indicating that the connection was included in the change order. An existing service that is included on an order will have the field activityCd = iactc, even if no properties of the service have been modified.
- The ConnectionAssignmentType instance representing the port for the DSL Link connection will have the field **valueLabels** populated with a collection of ValueLabelType instances representing all the value/labels for the DSL Link connection, including those with values that are not populated. Possible values for the **activityCd** field on the ValueLabelType instance include:
  - iactc: the value has changed.
  - iactd: the value has been disconnected.
  - iacti: the value has not been changed (the value is In Service).

Changing the Port Address on the DSL Link Connection

This flow involves changing port address for a connection on the order. Typically, the port for the DSL Link connection is changed to another port address.

Alternate Flow Steps

1. An order is created and an existing in-service DSL Link connection is copied to the order.
2. A provisioning plan is selected that includes an Activate task associated with an order-level gateway event.
3. The port for the DSL Link connection is changed to another port address.
4. Resume the basic flow at step 5.

Activation Data Result

The activation data exported by the Activation API includes information about the same set of network elements and connections as described for the basic flow. All items in the following list apply to connection information under the DSLAM network element:

- The IndividualConnectionType instance representing the DSL Link connection will contain two ConnectionAssignmentType instances representing the original and the new port:
  - The ConnectionAssignmentType instance representing the original port will have the field portAddressStatus = Pending Disconnect.
  - The ConnectionAssignmentType instance representing the new port will have the field portAddressStatus = Pending.
- Both ConnectionAssignmentType instances will have the field activityCd = iactc (Change).

Changing Location on DSL Service

This flow involves changing port address for a connection on the order. Typically, a new DSL Link connection and Internet Access connection are added to the order.

Alternate Flow Steps

1. An order is created and an existing in-service DSL Link connection and Internet Access virtual connection are copied to the order.
2. The existing DSL Link connection and Internet Access connection are disconnected on the order.
3. A new DSL Link connection and Internet Access connection are added to the order.
4. Resume the basic flow at step 2.

Activation Data Result

The activation data exported by the Activation API will include information about the network elements in the path of the disconnected Internet Access connection and the new Internet Access connection.

All items in the following list apply to connection information under the DSLAM network element to which the disconnected DSL Link connection is assigned:

- The ConnectionAssignmentType instance representing the port for the disconnected DSL Link connection will have the field activityCd = iactd (Disconnect).
- The ConnectionAllocationType instance representing the disconnected Internet Access connection will have the field activityCd = iactd (Disconnect).
- The activation data for the new DSL Link connection and new Internet Access connection will include information about the same set of network elements and connections as described for the basic flow. Both the ConnectionAssignmentType instance representing the port for the new DSL Link connection and the ConnectionAllocationType instance representing the new Internet Access connection will have the field activityCd = iactn (New).

Disconnecting DSL Service

This flow involves disconnecting the DSL service on the order. Typically, the existing DSL Link connection and Internet Access connection are disconnected on the order.

Alternate Flow Steps

1. An order is created and an existing in-service DSL Link connection and Internet Access virtual connection are copied to the order.
2. The existing DSL Link connection and Internet Access connection are disconnected on the order.
3. A provisioning plan is selected that includes an Activate task associated with an order-level gateway event.
4. Resume the basic flow at step 5.

> **Note:** You can select to disconnect just the virtual connection or the whole DSL product, including both the physical and virtual connection. If the physical connection is disconnected, the virtual connection must be disconnected with it.

Activation Data Result

The activation data exported by the Activation API will include information about the same set of network elements and connections as described for the basic flow. All items in the following list apply to connection information under the DSLAM network element:

- The ConnectionAssignmentType instance representing the port for the disconnected DSL Link connection will have the field activityCd = iactd (Disconnect).
- The ConnectionAllocationType instance representing the disconnected Internet Access connection will have the field activityCd = iactd (Disconnect).

Suspending DSL Service

This flow involves suspending the DSL service on the order. Typically, the existing DSL Link connection and Internet Access connection are suspended on the order.

Alternate Flow Steps

1. An order is created and an existing in-service DSL Link connection and Internet Access virtual connection are copied to the order.
2. The existing DSL Link connection and Internet Access connection are suspended on the order.
3. A provisioning plan is selected that includes an Activate task associated with an order-level gateway event.

   In addition, the provisioning plan includes a Suspend task that is placed after the Due Date task.

   > **Note:** The purpose of the Suspend task is to restore or disconnect the suspended service items after the suspend grace period has passed. In this alternate flow, the Activate task initiates the gateway event to suspend the connections in ASAP. The Suspend task is used in the Restore DSL Service alternate flow to restore the suspended items.

4. Resume the basic flow at step 5.

Activation Data Result

The activation data exported by the Activation API will include information about the same set of network elements and connections as described for the basic flow. All items in the following list apply to connection information under the DSLAM network element:

- The ConnectionAssignmentType instance representing the port for the DSL Link connection will have the field activityCd = iacts (Suspend).
- The ConnectionAllocationType instance representing the Internet Access connection will have the field activityCd = iacts (Suspend).

Restoring DSL Service

This flow involves restoring the DSL service on the order. Typically, the existing DSL Link connection and Internet Access connection are restored on the order.

Preconditions

- DSL service was previously ordered and placed in-service, including a DSL Link physical connection and an Internet Access virtual connection.
- A new order was created on which the DSL Link connection and the Internet Access connection were suspended. A provisioning plan was selected that includes a Suspend task, which is placed after the Due Date task.

> **Note:** The purpose of the Suspend task is to restore or disconnect the suspended service items. The Suspend task will appear in Work Queue after the Due Date task is completed and the suspend grace period has passed.

Alternate Flow Steps

1. The suspend grace period expires and the Suspend task appears in the MetaSolv Solution Work Queue Manager.

   The Suspend task is selected, and the user is given the option to restore or disconnect the suspended products. The user elects to restore the DSL Link connection and the Internet Access connection. The system creates a new order with the items to be restored.

2. The new order with the items to restore is finished, and a provisioning plan is selected that includes an Activate task associated with an order-level gateway event.
3. Resume the basic flow at step 5.

Activation Data Result

The activation data exported by the Activation API includes information about the same set of network elements and connections as described for the basic flow. All items in the following list apply to connection information under the DSLAM network element:

- The ConnectionAssignmentType instance representing the port for the DSL Link connection will have the field activityCd = iactr (Restore).
- The ConnectionAllocationType instance representing the Internet Access connection will have the field activityCd = iactr (Restore).

Supplementing to Correct the Due Date or Make Modifications

This flow involves supplementing an order to correct the due date or make any modifications on the order.

Preconditions

- An order has been created for a new DSL Link physical connection and Internet Access virtual connection.The provisioning plan selected for the order includes an Activate task.
- The Activate task is completed, and an order to activate the service is created in ASAP. The order is pending in ASAP and has not yet been completed.

Alternate Flow Steps

1. Before the ASAP order is completed, the original order in MetaSolv Solution is supplemented to change the due date or make some other modification.
2. The supplement order is finished, and the completed tasks in the provisioning plan are reopened.
3. In the MetaSolv Solution Work Queue Manager, the order-level gateway event is initiated on the Activate task. This results in an event containing the order key being passed to the service provider's MetaSolv Solution/ASAP integration layer.
4. The event is received by the integration layer triggering a call to the Activation XML API to get activation information for the order. The activation data includes a suppType attribute in the OrderDetailsType container indicating that the supplement type is Correction or New Due Date.
5. The integration layer determines whether the original ASAP order has been completed. If not, the original ASAP order is canceled, and a new ASAP order is created from the activation data from the supplement order in MetaSolv Solution.

> **Note:** If the original ASAP order has already been completed, the activation must be manually reversed and corrected.

6. ASAP returns a create order event with the ASAP order key, and the integration layer stores the ASAP order key in the gateway event.
7. ASAP activates the service and returns confirmation to the integration layer. The integration layer updates the status of the gateway event in MetaSolv Solution to Complete.
8. The Due Date task in MetaSolv Solution is completed.

Activation Data Result

The activation data exported by the Activation API includes the same information that was exported for the original order plus changes from the supplement. The OrderDetailsType container includes a suppType attribute with a value of Correction or New Due Date.

> **Note:** To minimize the chance of a supplement order in MetaSolv Solution occurring after the ASAP order has been created, the Activate task should be set as a predecessor to the Due Date task, with no time interval between the tasks.

Supplementing to Cancel an Order

This flow involves supplementing the original order to cancel the order.

Preconditions

- An order has been created for a new DSL Link physical connection and Internet Access virtual connection. The provisioning plan selected for the order includes an Activate task.
- The Activate task is completed, and an order to activate the service is created in ASAP. The order is pending in ASAP and has not yet been completed.

Alternate Flow Steps

1. Before the ASAP order is completed, the original order in MetaSolv Solution is supplemented to cancel the order.
2. The supplement order is finished, and the completed tasks in the provisioning plan are reopened.
3. In the MetaSolv Solution Work Queue Manager, the order-level gateway event is initiated on the Activate task.

   This results in an event containing the order key being passed to the service provider's MetaSolv Solution/ASAP integration layer.
4. The event is received by the integration layer triggering a call to the Activation XML API to get activation information for the order. The activation data includes a suppType attribute in the OrderDetailsType container indicating that the supplement type is Cancel.
5. The integration layer determines whether the original ASAP order has been completed.

   If not, the integration layer sends a command to ASAP to cancel the original order.

   > **Note:** If the original ASAP order has already been completed, the activation must be manually reversed.
6. ASAP cancels the order and returns confirmation to the integration layer.

   The integration layer updates the status of the gateway event in MetaSolv Solution to complete.
7. The Due Date task in MetaSolv Solution is completed.

Activation Data Result

The activation data exported by the Activation API will include the same information as was exported for the original order plus and changes from the supplement. The activation data will include a suppType attribute in the OrderDetailsType container with a value of Cancel. The containers for all product items on the order will have the attribute activityCd = iacta (Cancel).

> **Note:** To minimize the chance of a supplement order in MetaSolv Solution occurring after the ASAP order has been created, the Activate task should be set as a predecessor to the Due Date task, with no time interval between the tasks.

Best Practices for Setting Up MetaSolv Solution Data

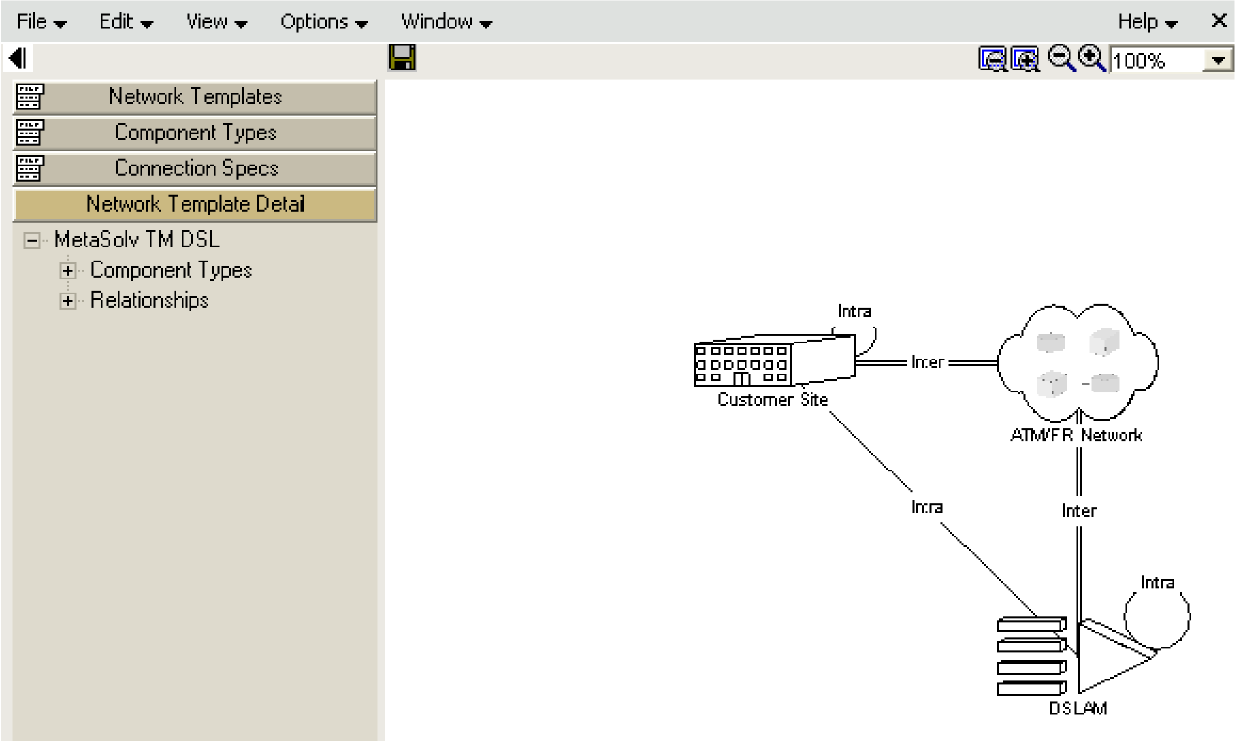
This section details the actions taken in MetaSolv Solution to ensure the appropriate POTS activation information is available for ASAP in the flow-through package described in "Viewing The POTS/DSL Flow-through Package".

DSL Template

You must create or select a network system. This requires a network template. You can select the predefined MetaSolv Solution DSL template or you can create a new DSL template.

Figure 5–3 shows the DSL network template.

Figure 5–3 DSL Network Template

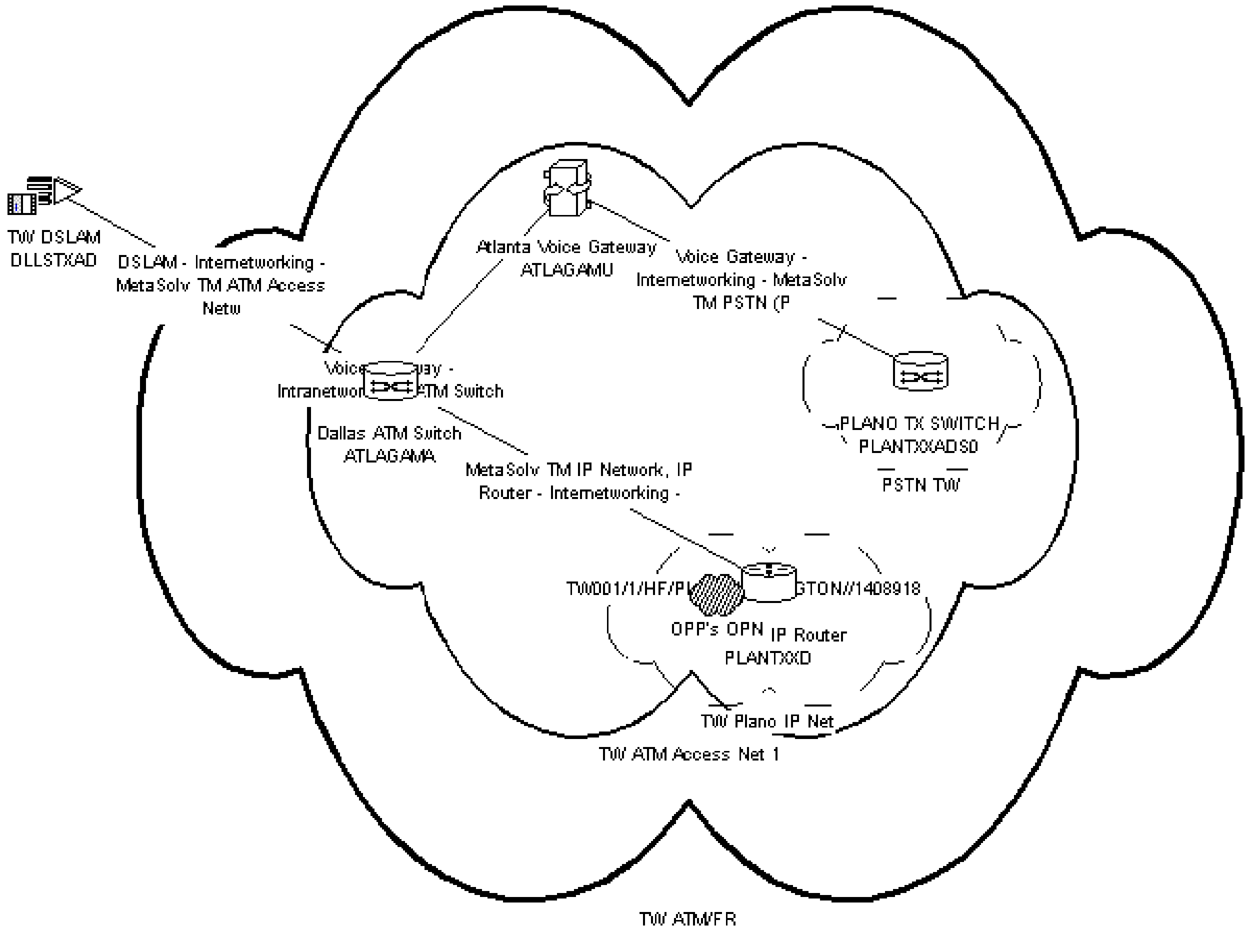


Internal DSL Network System

You must provision your DSL order over an internal network built using a DSL template.

Figure 5–4 shows the DSL network system.

Figure 5–4 DSL Network System



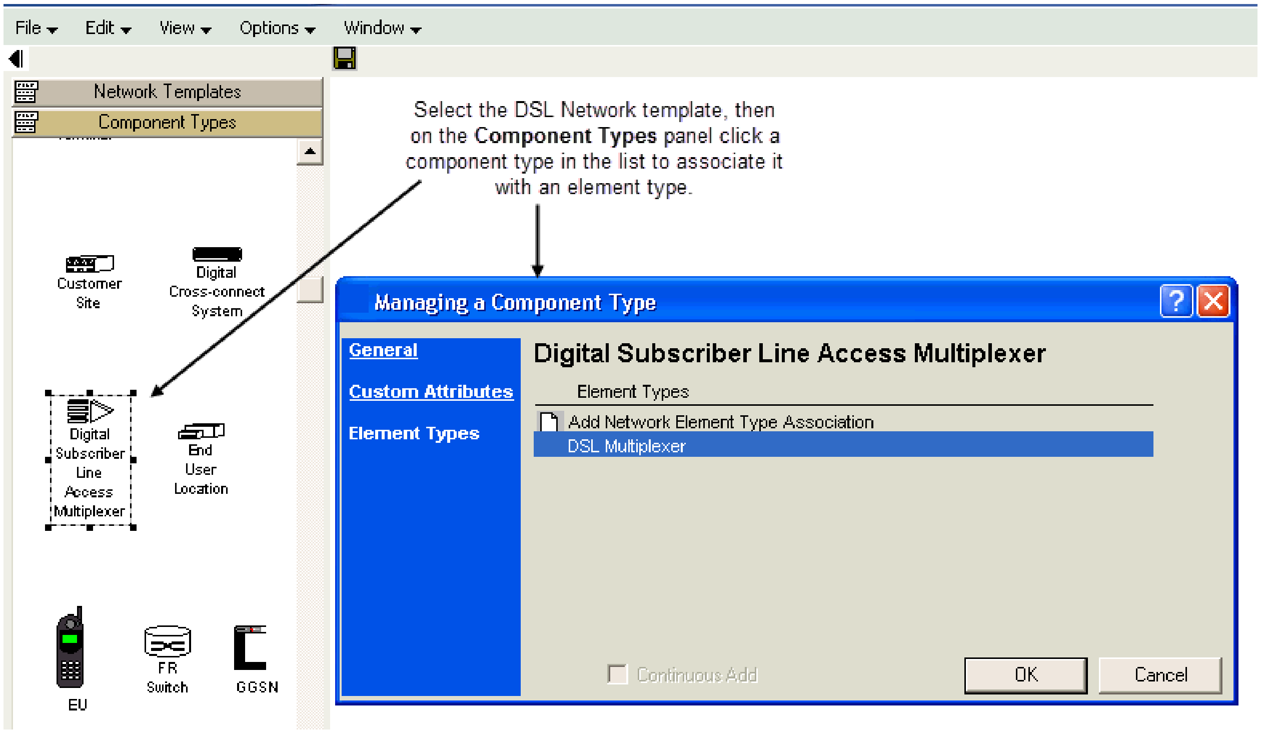
As you can see, the internal network system provides a DSLAM with a connection to a switch. From the switch, a virtual connection can be routed anywhere in the network. Breaking this information into manageable parts for mapping from MetaSolv Solution to ASAP is covered later in this chapter.

Network Elements/Equipment

- All network component types representing equipment that can be activated must be associated with a network element type.

  For example, Figure 5–5 shows the association in Network Templates between the network component type and network element type for a DSLAM.

Figure 5–5 Associating Component Types and Element Types



Select the DSL Network template, then on the **Component Types** panel click a component type in the list to associate it with an element type.

Digital Subscriber Line Access Multiplexer

Element Types

Add Network Element Type Association

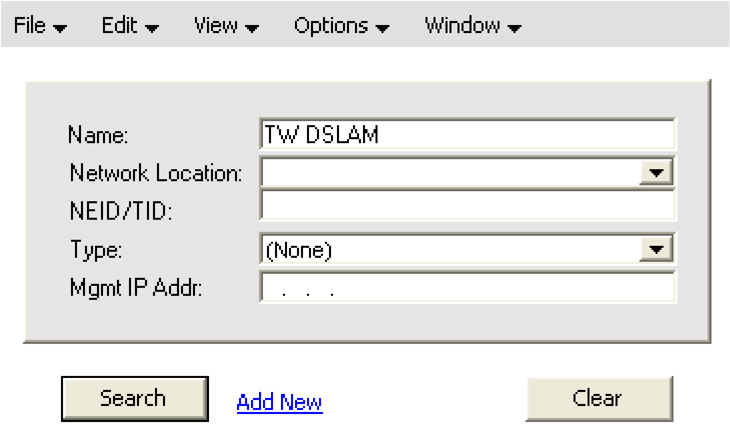
DSL Multiplexer

- The DSLAM network element and any other elements that the virtual connection will pass through must be associated with a network system. A network element is associated automatically when it is added to the a network system on the network canvas.

To check whether a specific network element is part of a network system:

1. On the Navigation Bar, click Inventory Management, then click the **Network Elements** link.

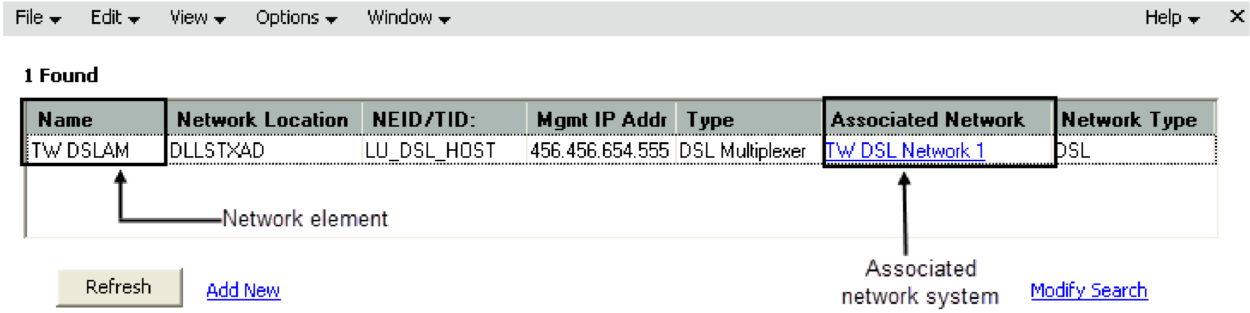
   The Network Element Search window appears.



Name:	TW DSLAM
Network Location:	
NEID/TID:	
Type:	(None)
Mgmt IP Addr:	. . .

2. Type the network element name and click **Search**.

   The Network Element Search Results window displays a list of networks with which the element is associated.

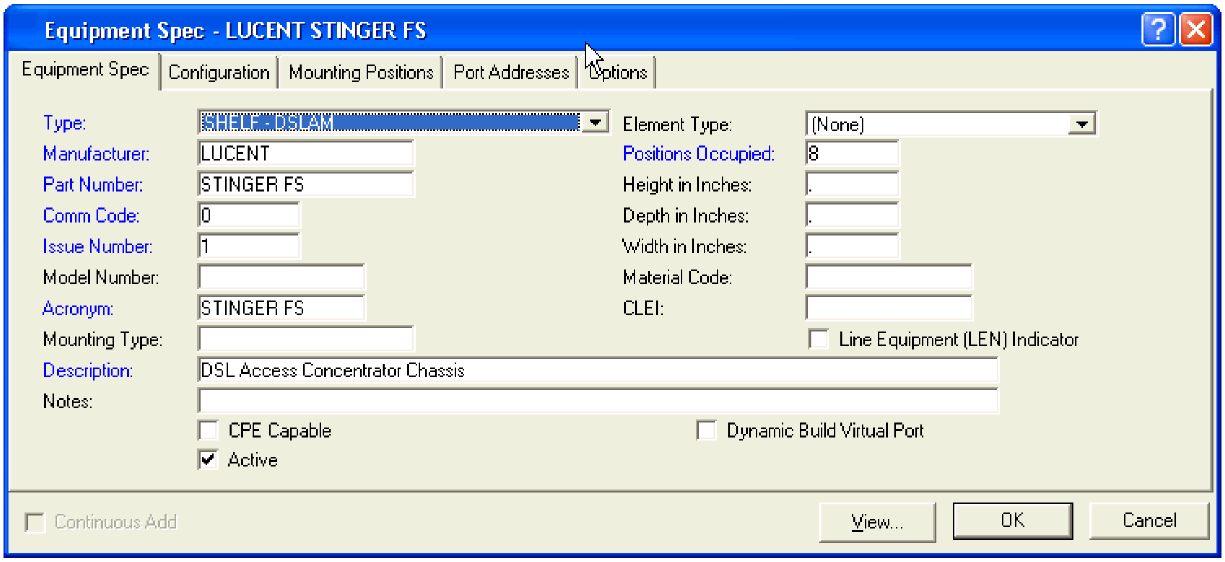


1 Found

Name	Network Location	NEID/TID:	Mgmt IP Addr	Type	Associated Network	Network Type
TW DSLAM	DLLSTXAD	LU_DSL_HOST	456.456.654.555	DSL Multiplexer	TW DSL Network 1	DSL

- The DSLAM must be defined within MetaSolv Solution as both an equipment specification and as a network element. Figure 5–6 shows the DSLAM equipment specification.

Figure 5–6 DSLAM Equipment Specification



Equipment Spec - LUCENT STINGER FS

Type:	SHELF - DSLAM	Element Type:	(None)
Manufacturer:	LUCENT	Positions Occupied:	8
Part Number:	STINGER FS	Height in Inches:	.
Comm Code:	0	Depth in Inches:	.
Issue Number:	1	Width in Inches:	.
Model Number:		Material Code:	
Acronym:	STINGER FS	CLEI:	
Mounting Type:			□ Line Equipment (LEN) Indicator
Description:	DSL Access Concentrator Chassis		
Notes:			

□ CPE Capable

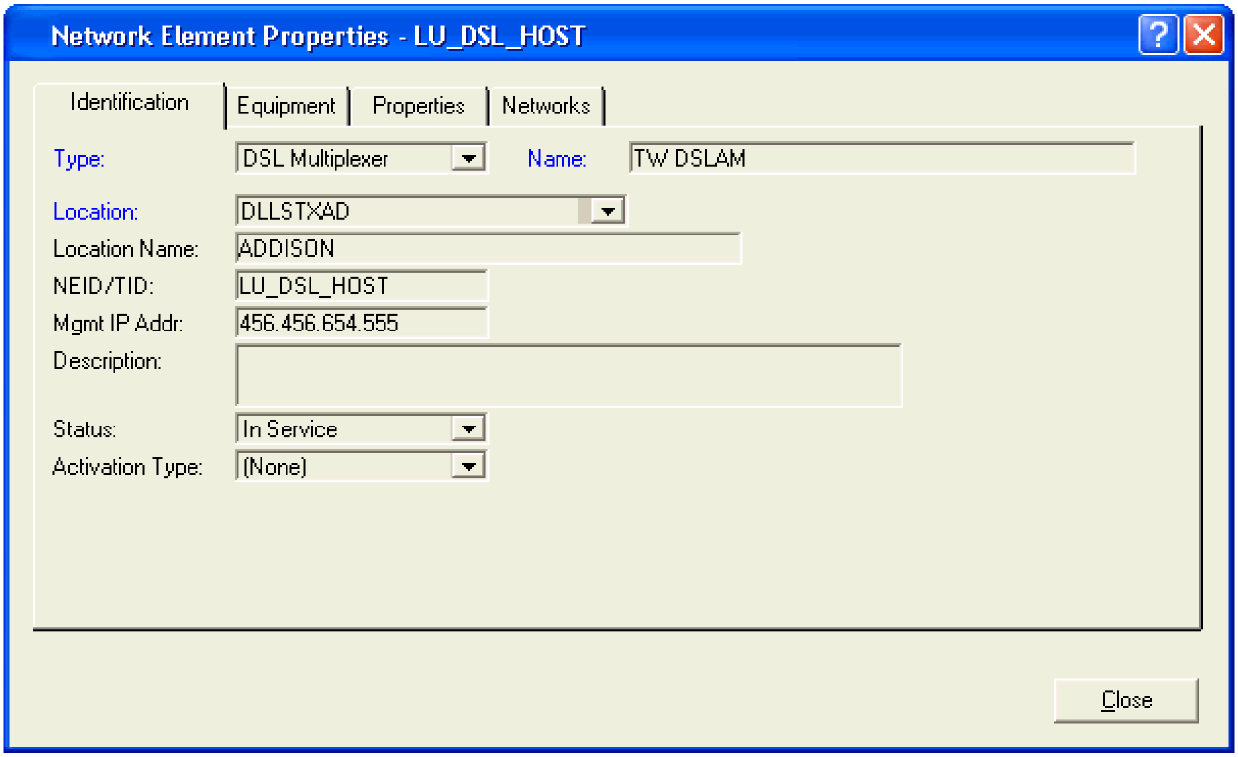
□ Dynamic Build Virtual Port

☑ Active

□ Continuous Add

Figure 5–7 shows the Network Element Properties window, which displays the fields populated for the network element for DSLAM.

Figure 5–7 Network Element for the DSLAM



Network Element Properties - LU_DSL_HOST

Type: DSL Multiplexer
Name: TW DSLAM
Location: DLLSTXAD
Location Name: ADDISON
NEID/TID: LU_DSL_HOST
Mgmt IP Addr: 456.456.654.555
Description:
Status: In Service
Activation Type: (None)

For some DSLAM models, it may be necessary to define activation custom attributes at the network-element level. The need for custom attributes is determined by the parameters ASAP needs for a specific model.

- ConnectionAssignmentType data has three fields required for ASAP that exist but are not required in MetaSolv Solution. If these fields do not appear in your data, you can check the locations shown in Table 5–1 in MetaSolv Solution to make sure the fields are being populated.

Table 5–1 MetaSolv Solution and ASAP Attributes and their Location in the MetaSolv Solution Application

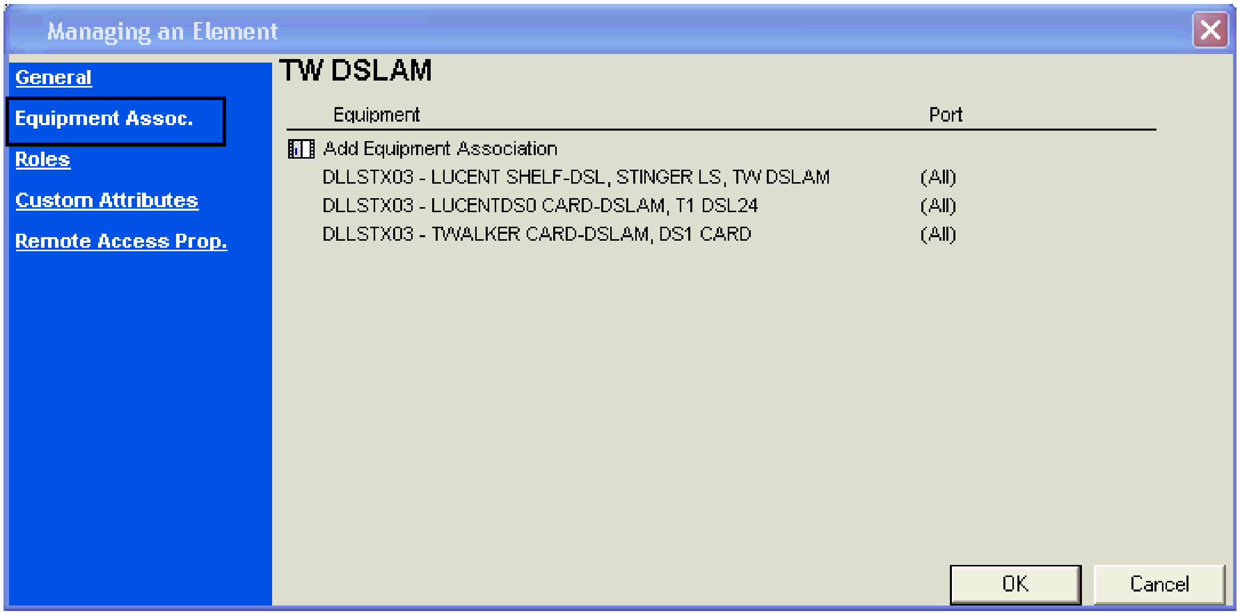
MetaSolv Solution attribute	ASAP attribute	Location in MetaSolv Solution
shelfEquipmentName	SHELF	**Name** field on the installed equipment instance for the shelf.
shelfSlotNumber	SLOT	**Mounting Position** field on the Mounting Position tab on the equipment specification for the shelf.
physicalNodeAddress	PORT	**Node Address** field on the Node Addresses tab of the equipment specification for the card containing the assigned port.

Figure 5–8 shows a DSL network with a DSLAM. The DSLAM network element contains all of the values shown in the table that must be populated in MetaSolv Solution.

- shelfEquipmentName

Double-click the **equipment** (network element) on the Network Systems canvas to display the Managing an Element window which contains properties for the element. Click the **Equipment Association** link on the left to see all equipment associated with the network element TW DSLAM.The following figure shows the properties window for a DSLAM.

Figure 5–8 Equipment Associations for the DSLAM



Managing an Element

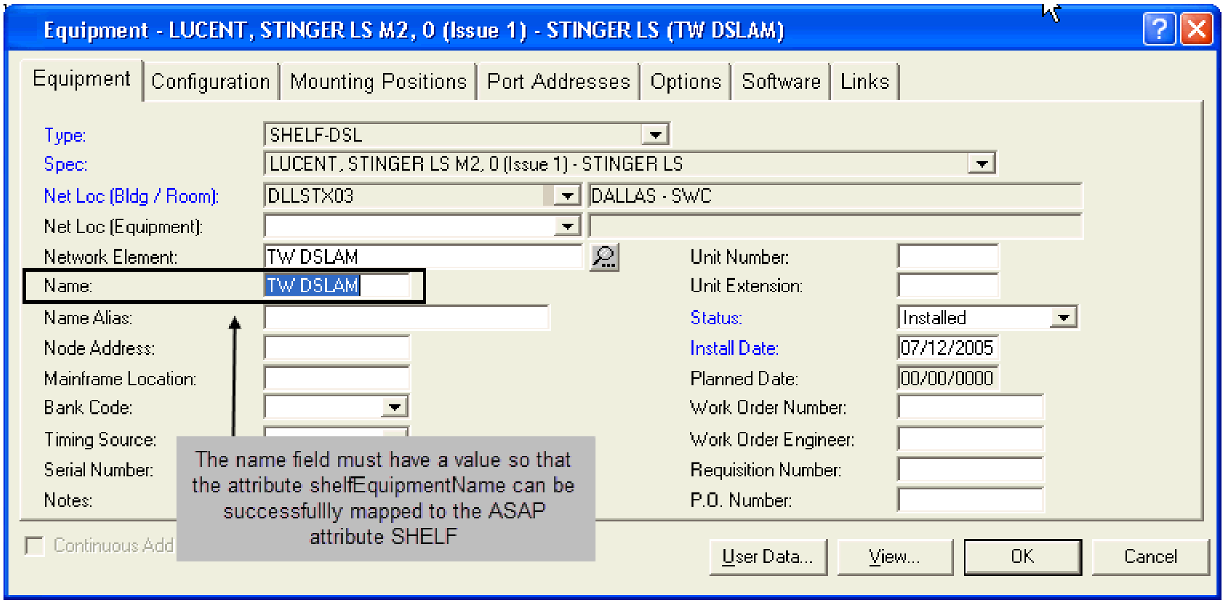
General
Equipment Assoc.
Roles
Custom Attributes
Remote Access Prop.

TW DSLAM

Equipment	Port
Add Equipment Association	
DLLSTX03 - LUCENT SHELF-DSL, STINGER LS, TW DSLAM	(All)
DLLSTX03 - LUCENTDS0 CARD-DSLAM, T1 DSL24	(All)
DLLSTX03 - TWALKER CARD-DSLAM, DS1 CARD	(All)

To see the **Name** field for shelfEquipmentName, double-click the shelf in the equipment list. The window shown in Figure 5–9 appears.

Figure 5–9 Name (shelfEquipmentName) on the Equipment Window



Equipment - LUCENT, STINGER LS M2, 0 (Issue 1) - STINGER LS (TW DSLAM)

Type: SHELF-DSL
Spec: LUCENT, STINGER LS M2, 0 (Issue 1) - STINGER LS
Net Loc (Bldg / Room): DLLSTX03 DALLAS - SWC
Net Loc (Equipment):
Network Element: TW DSLAM
Name: TW DSLAM
Name Alias:
Node Address:
Mainframe Location:
Bank Code:
Timing Source:
Serial Number:
Notes:
Unit Number:
Unit Extension:
Status: Installed
Install Date: 07/12/2005
Planned Date: 00/00/0000
Work Order Number:
Work Order Engineer:
Requisition Number:
P.O. Number:

□ Continuous Add

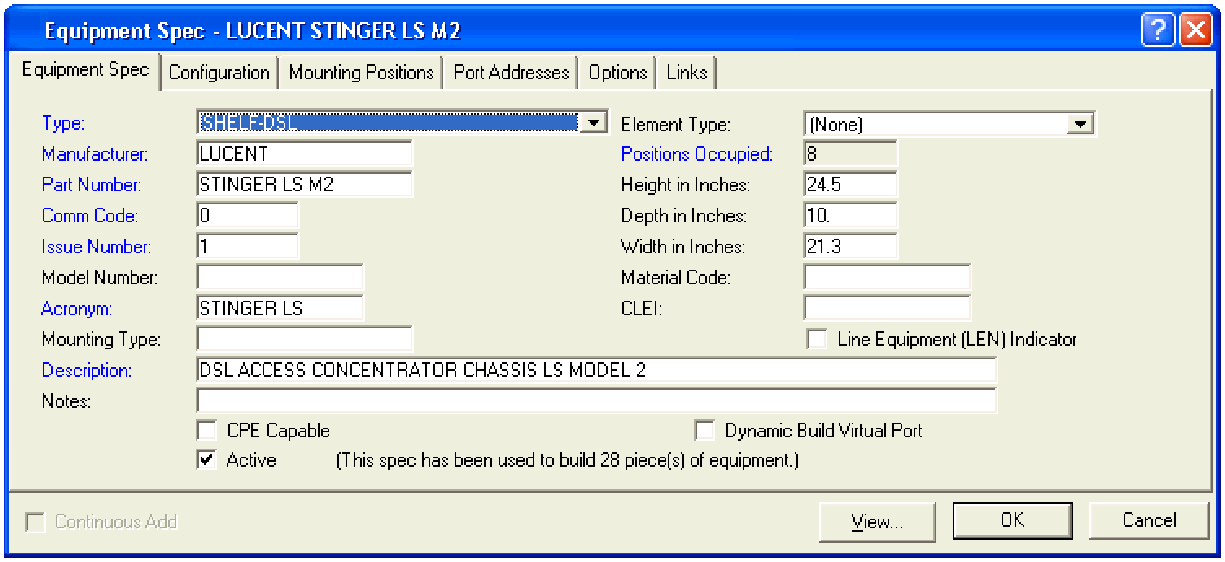
The name field must have a value so that the attribute shelfEquipmentName can be successfullly mapped to the ASAP attribute SHELF

- shelfSlotNumber

To locate the field for the MetaSolv Solution attribute value (shelfSlotNumber), locate the equipment specification for the shelf.

Figure 5–10 shows the equipment specification for the shelf.

Figure 5–10 Equipment Specification for the Shelf



Equipment Spec - LUCENT STINGER LS M2

Type: SHELF-DSL

Element Type: (None)

Manufacturer: LUCENT

Positions Occupied: 8

Part Number: STINGER LS M2

Height in Inches: 24.5

Comm Code: 0

Depth in Inches: 10.

Issue Number: 1

Width in Inches: 21.3

Model Number:

Material Code:

Acronym: STINGER LS

CLEI:

Mounting Type:

□ Line Equipment (LEN) Indicator

Description: DSL ACCESS CONCENTRATOR CHASSIS LS MODEL 2

Notes:

□ CPE Capable

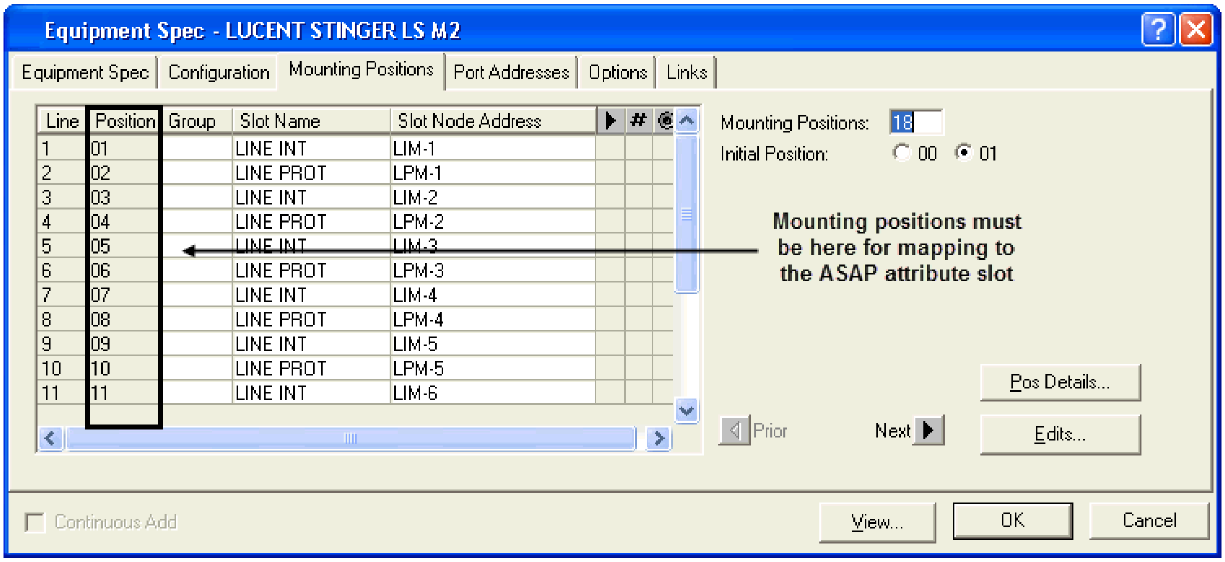
□ Dynamic Build Virtual Port

☑ Active (This spec has been used to build 28 piece(s) of equipment.)

□ Continuous Add

Click the **Mounting Positions** tab to see if the mounting position for the shelf is included as shown in Figure 5–11.

Figure 5–11 Mounting Positions for the DSL Shelf on the Equipment Specification



Equipment Spec - LUCENT STINGER LS M2

Line	Position	Group	Slot Name	Slot Node Address
1	01		LINE INT	LIM-1
2	02		LINE PROT	LPM-1
3	03		LINE INT	LIM-2
4	04		LINE PROT	LPM-2
5	05		LINE INT	LIM-3
6	06		LINE PROT	LPM-3
7	07		LINE INT	LIM-4
8	08		LINE PROT	LPM-4
9	09		LINE INT	LIM-5
10	10		LINE PROT	LPM-5
11	11		LINE INT	LIM-6

Mounting Positions: 18

Initial Position: ○ 00 ● 01

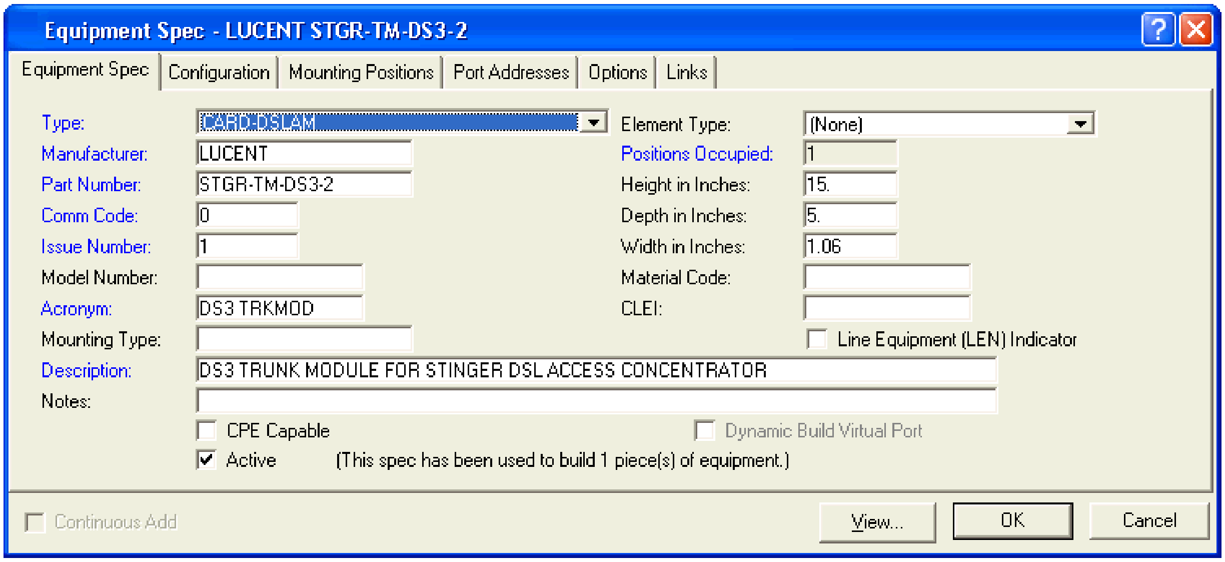
Mounting positions must be here for mapping to the ASAP attribute slot

□ Continuous Add

- physicalNodeAddress

  To check the field for the MetaSolv Solution attribute physicalNodeAddress, locate the equipment specification for the card that contains the assigned port. Figure 5–12 shows the equipment specification for the DSL card used for TW DSLAM.

Figure 5–12 Equipment Specification for the Card That Contains the Port



Equipment Spec - LUCENT STGR-TM-DS3-2

Type:	CARD-DSLAM	Element Type:	(None)
Manufacturer:	LUCENT	Positions Occupied:	1
Part Number:	STGR-TM-DS3-2	Height in Inches:	15.
Comm Code:	0	Depth in Inches:	5.
Issue Number:	1	Width in Inches:	1.06
Model Number:		Material Code:	
Acronym:	DS3 TRKMOD	CLEI:	
Mounting Type:		□ Line Equipment (LEN) Indicator	
Description:	DS3 TRUNK MODULE FOR STINGER DSL ACCESS CONCENTRATOR		
Notes:			

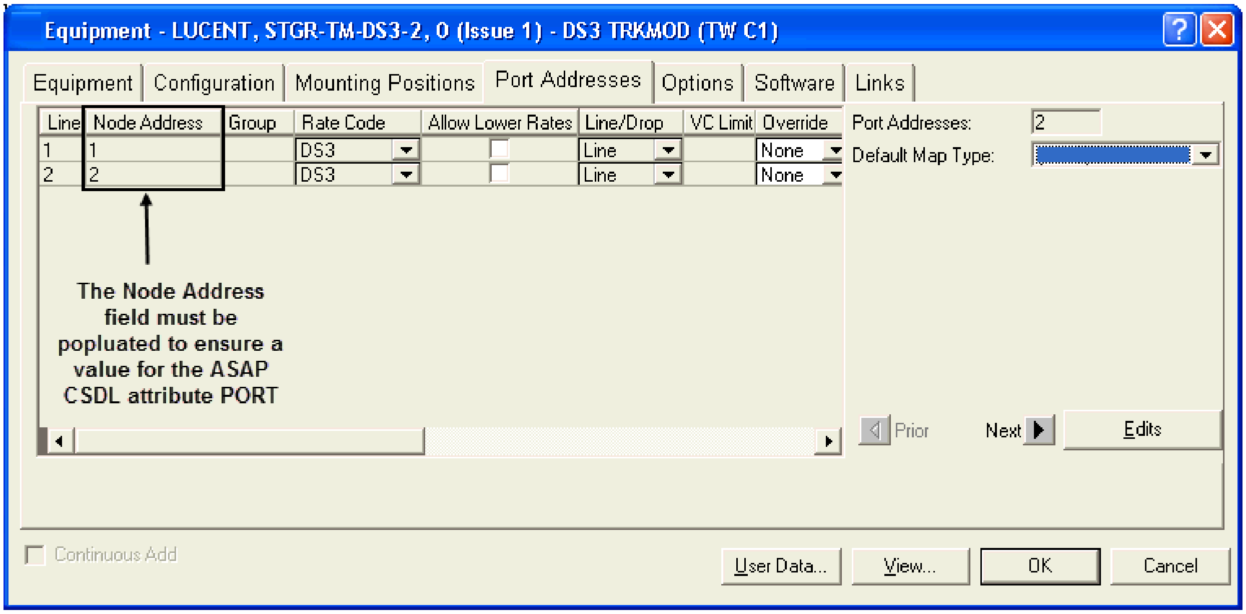
□ CPE Capable □ Dynamic Build Virtual Port

☑ Active (This spec has been used to build 1 piece(s) of equipment.)

□ Continuous Add

On the Port Addresses tab of the equipment specification, make sure a value appears in the **Node Address** field as shown in Figure 5–13.

Figure 5–13 Node Address Field on the Card That Contains the Port



Equipment - LUCENT, STGR-TM-DS3-2, 0 (Issue 1) - DS3 TRKMOD (TW C1)

Line	Node Address	Group	Rate Code	Allow Lower Rates	Line/Drop	VC Limit	Override
1	1		DS3	□	Line		None
2	2		DS3	□	Line		None

Port Addresses: 2

Default Map Type:

The Node Address field must be popluated to ensure a value for the ASAP CSDL attribute PORT

□ Continuous Add

Product Specification

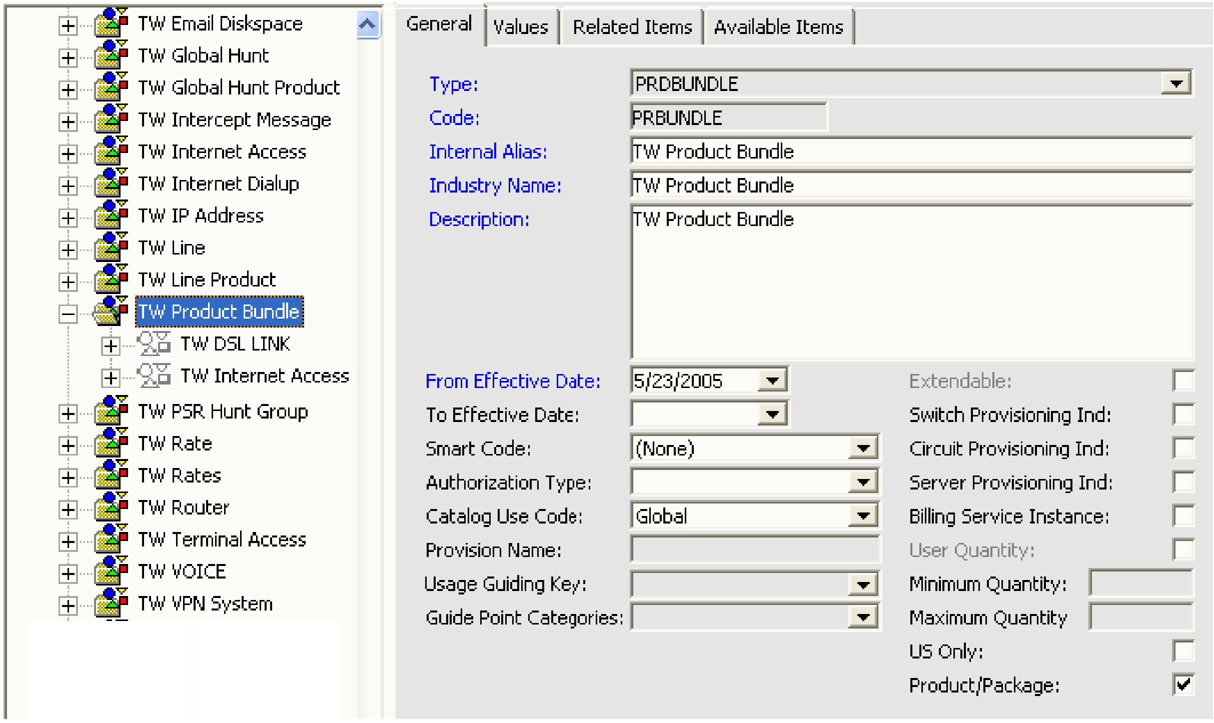
The following item types are required for DSL:

- Product bundle
- Network connector

For typical DSL service, product specifications are associated with the following connection specifications: DSL link (physical) and Internet access (virtual). These two connection specifications are included in the MetaSolv Solution DSL template with a set of predefined activation parameters.

Figure 5–14 shows the product bundle specification.

Figure 5–14 DSL Product Specification



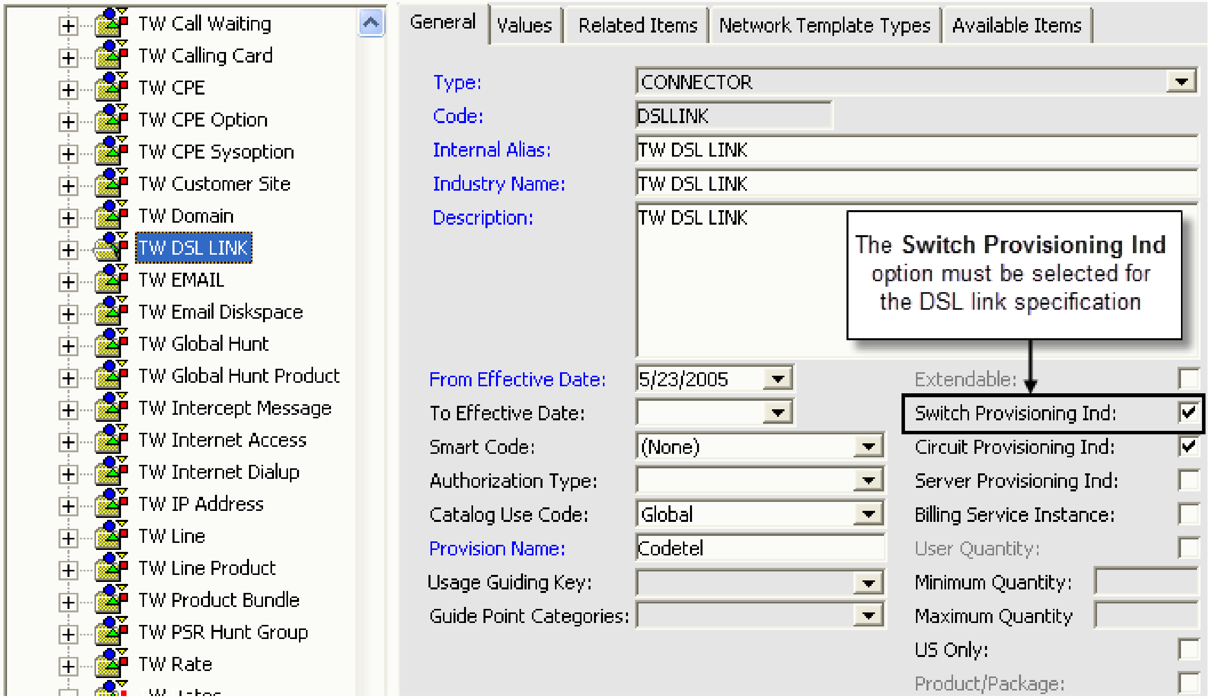
- TW Email Diskspace
- TW Global Hunt
- TW Global Hunt Product
- TW Intercept Message
- TW Internet Access
- TW Internet Dialup
- TW IP Address
- TW Line
- TW Line Product
- TW Product Bundle
  - TW DSL LINK
  - TW Internet Access
- TW PSR Hunt Group
- TW Rate
- TW Rates
- TW Router
- TW Terminal Access
- TW VOICE
- TW VPN System

Type:	PRDBUNDLE
Code:	PRBUNDLE
Internal Alias:	TW Product Bundle
Industry Name:	TW Product Bundle
Description:	TW Product Bundle
From Effective Date:	5/23/2005
To Effective Date:	
Smart Code:	(None)
Authorization Type:	
Catalog Use Code:	Global
Provision Name:	
Usage Guiding Key:	
Guide Point Categories:	

Extendable:	□
Switch Provisioning Ind:	□
Circuit Provisioning Ind:	□
Server Provisioning Ind:	□
Billing Service Instance:	□
User Quantity:	□
Minimum Quantity:	
Maximum Quantity	
US Only:	□
Product/Package:	☑

Figure 5–15 shows the DSL link connection product specification. You must select the **Switch Provisioning Indicator** check box for the DSL link product specification.

Figure 5–15 DSL Link Product Specification



- TW Call Waiting
- TW Calling Card
- TW CPE
- TW CPE Option
- TW CPE Sysoption
- TW Customer Site
- TW Domain
- TW DSL LINK
- TW EMAIL
- TW Email Diskspace
- TW Global Hunt
- TW Global Hunt Product
- TW Intercept Message
- TW Internet Access
- TW Internet Dialup
- TW IP Address
- TW Line
- TW Line Product
- TW Product Bundle
- TW PSR Hunt Group
- TW Rate

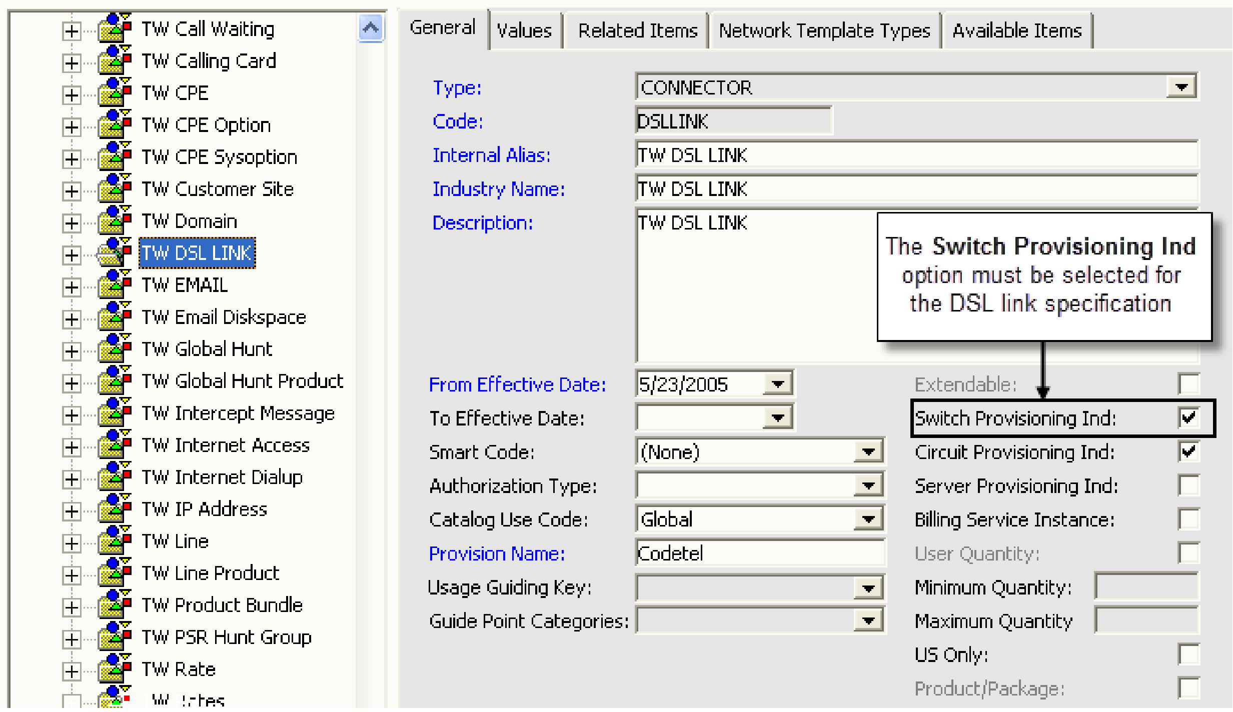
Type:	CONNECTOR
Code:	DSLLINK
Internal Alias:	TW DSL LINK
Industry Name:	TW DSL LINK
Description:	TW DSL LINK
From Effective Date:	5/23/2005
To Effective Date:	
Smart Code:	(None)
Authorization Type:	
Catalog Use Code:	Global
Provision Name:	Codetel
Usage Guiding Key:	
Guide Point Categories:	

Extendable:	□
Switch Provisioning Ind:	☑
Circuit Provisioning Ind:	☑
Server Provisioning Ind:	□
Billing Service Instance:	□
User Quantity:	□
Minimum Quantity:	
Maximum Quantity	
US Only:	□
Product/Package:	□

The Switch Provisioning Ind option must be selected for the DSL link specification

Figure 5–16 shows the Internet access connection product specification.

Figure 5–16 Internet Access Product Specifications



- TW Call Waiting
- TW Calling Card
- TW CPE
- TW CPE Option
- TW CPE Sysoption
- TW Customer Site
- TW Domain
- TW DSL LINK
- TW EMAIL
- TW Email Diskspace
- TW Global Hunt
- TW Global Hunt Product
- TW Intercept Message
- TW Internet Access
- TW Internet Dialup
- TW IP Address
- TW Line
- TW Line Product
- TW Product Bundle
- TW PSR Hunt Group
- TW Rate

Type: CONNECTOR

Code: DSLLINK

Internal Alias: TW DSL LINK

Industry Name: TW DSL LINK

Description: TW DSL LINK

From Effective Date: 5/23/2005

To Effective Date:

Smart Code: (None)

Authorization Type:

Catalog Use Code: Global

Provision Name: Codetel

Usage Guiding Key:

Guide Point Categories:

Extendable: ☐

Switch Provisioning Ind: ☑

Circuit Provisioning Ind: ☑

Server Provisioning Ind: ☐

Billing Service Instance: ☐

User Quantity: ☐

Minimum Quantity:

Maximum Quantity

US Only: ☐

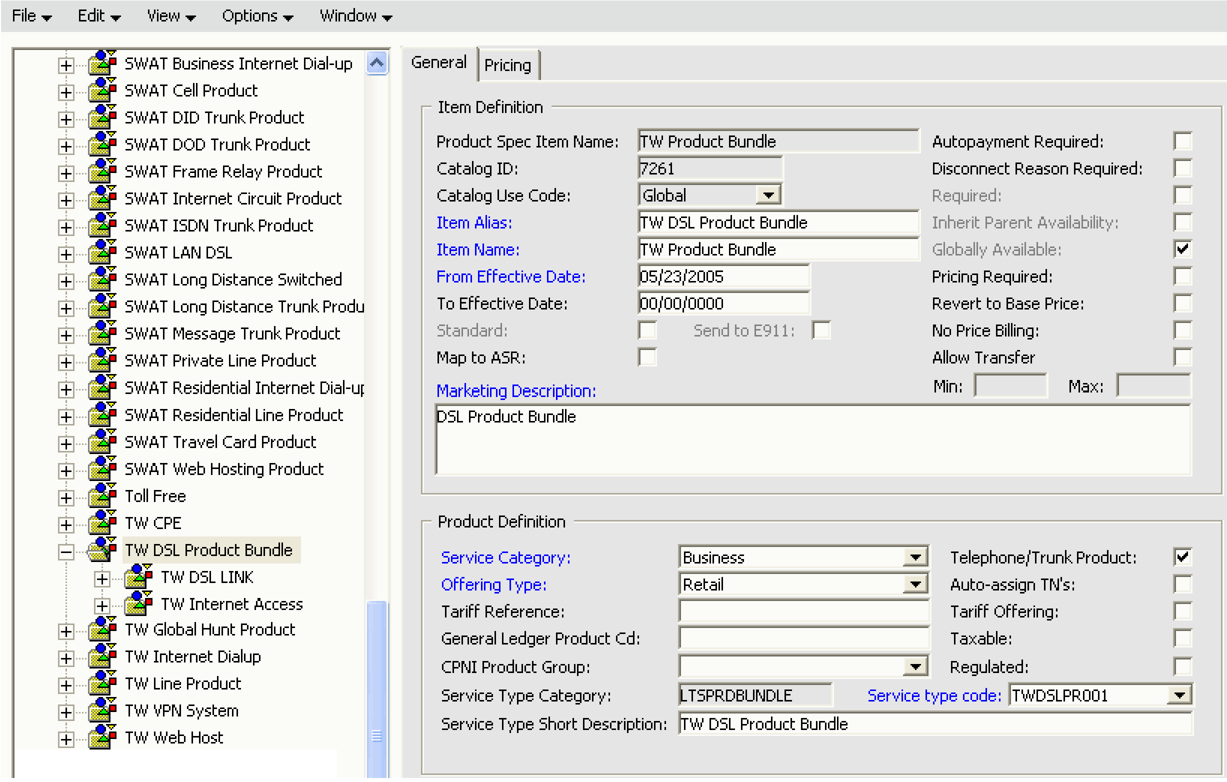
Product/Package: ☐

The **Switch Provisioning Ind** option must be selected for the DSL link specification

Product Catalog

The DSL product catalog entry must be associated with the DSL product specification. Figure 5–17 shows a product catalog entry built using the product specification shown in Figure 5–14.

Figure 5–17 DSL Product Catalog Entry



Item Definition

Product Spec Item Name: TW Product Bundle
Catalog ID: 7261
Catalog Use Code: Global
Item Alias: TW DSL Product Bundle
Item Name: TW Product Bundle
From Effective Date: 05/23/2005
To Effective Date: 00/00/0000
Standard: ☐ Send to E911: ☐
Map to ASR: ☐
Marketing Description:
DSL Product Bundle

Autopayment Required: ☐
Disconnect Reason Required: ☐
Required: ☐
Inherit Parent Availability: ☐
Globally Available: ☑
Pricing Required: ☐
Revert to Base Price: ☐
No Price Billing: ☐
Allow Transfer ☐
Min: Max:

Product Definition

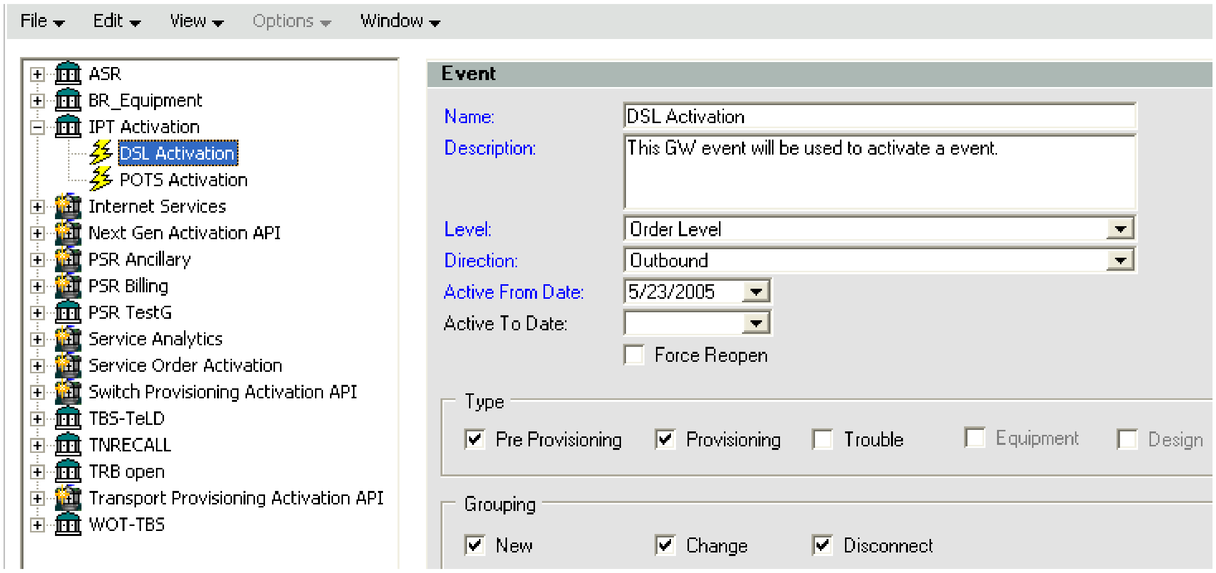
Service Category: Business
Offering Type: Retail
Tariff Reference:
General Ledger Product Cd:
CPNI Product Group:
Service Type Category: LTSPRDBUNDLE
Service Type Short Description: TW DSL Product Bundle

Telephone/Trunk Product: ☑
Auto-assign TN's: ☐
Tariff Offering: ☐
Taxable: ☐
Regulated: ☐
Service type code: TWDSLPR001

Gateway Event

You must create a gateway event and associate it with a task on the provisioning plan for DSL activation as shown in Figure 5–18 .

Figure 5–18 DSL Activation Gateway Event



Event

Name: DSL Activation
Description: This GW event will be used to activate a event.
Level: Order Level
Direction: Outbound
Active From Date: 5/23/2005
Active To Date:
☐ Force Reopen

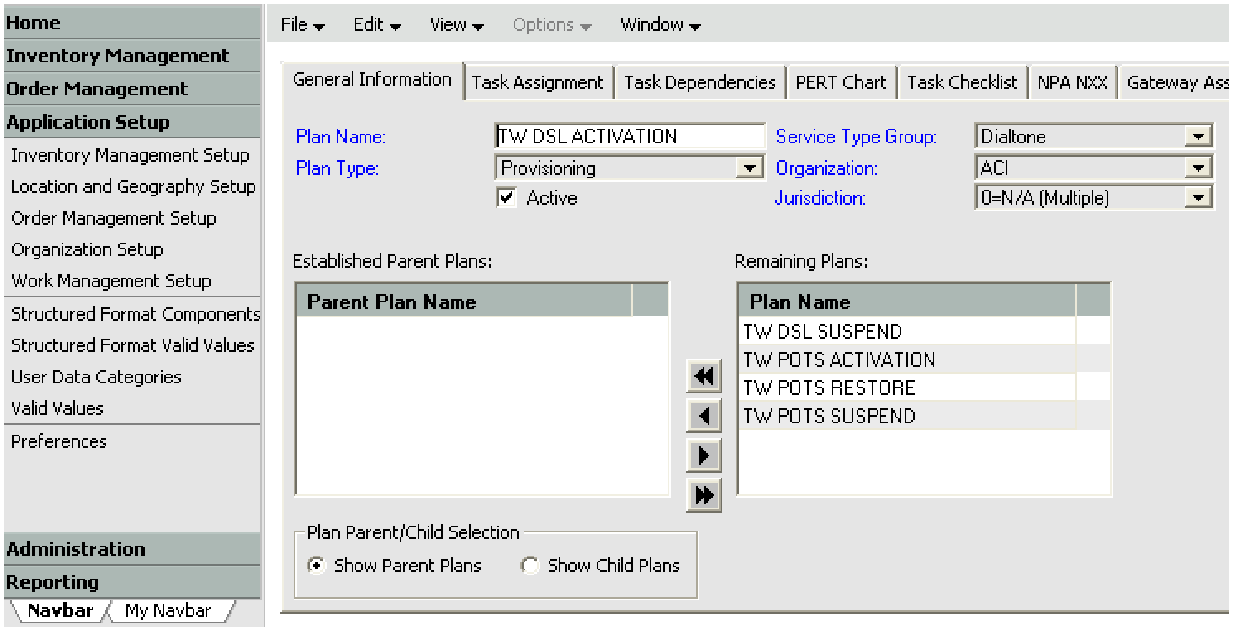
Type
☑ Pre Provisioning ☑ Provisioning ☐ Trouble ☐ Equipment ☐ Design

Grouping
☑ New ☑ Change ☑ Disconnect

Provisioning Plan

You must create or select a provisioning plan and associate it with a gateway event set up for DSL activation as shown in Figure 5–19.

Figure 5–19 DSL Provisioning Plan



Plan Name: TW DSL ACTIVATION

Plan Type: Provisioning

☑ Active

Service Type Group: Dialtone

Organization: ACI

Jurisdiction: 0=N/A (Multiple)

Established Parent Plans:

Parent Plan Name

Remaining Plans:

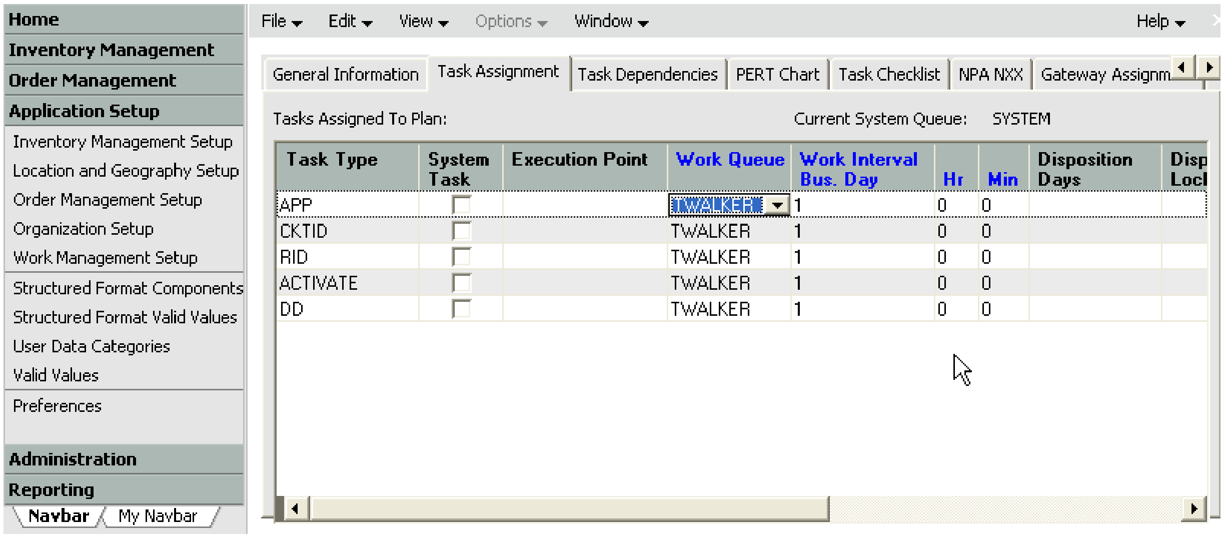
Plan Name
TW DSL SUSPEND
TW POTS ACTIVATION
TW POTS RESTORE
TW POTS SUSPEND

Plan Parent/Child Selection

● Show Parent Plans ○ Show Child Plans

Figure 5–20 shows the tasks assigned to the provisioning plan.

Figure 5–20 Tasks Assigned to the Provisioning Plan



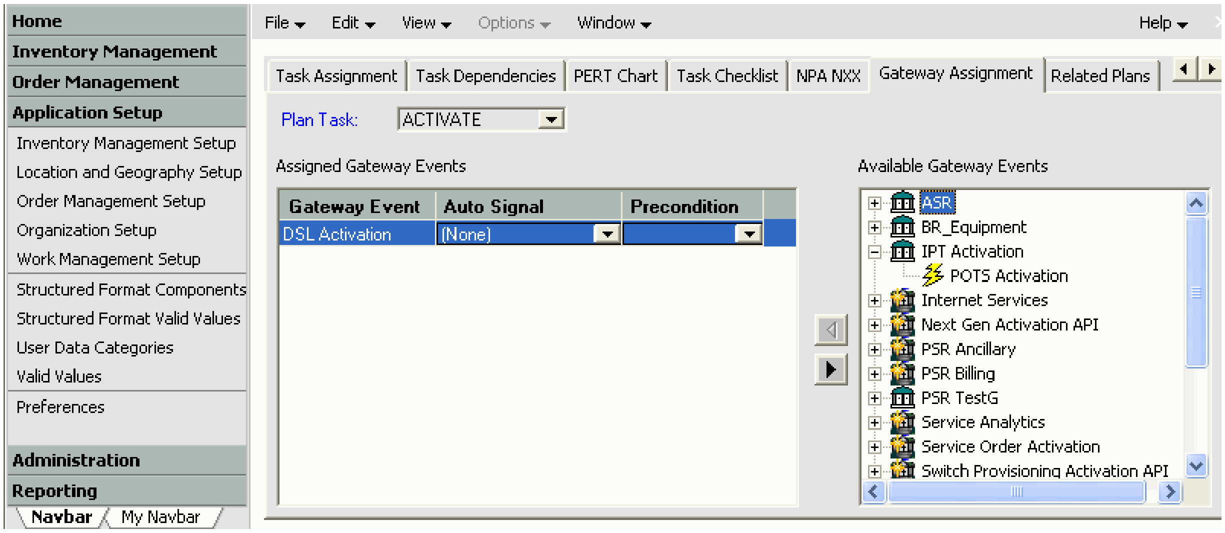
Tasks Assigned To Plan:

Current System Queue: SYSTEM

Task Type	System Task	Execution Point	Work Queue	Work Interval Bus. Day	Hr	Min	Disposition Days	Disp Locl
APP	□		TWALKER	1	0	0		
CKTID	□		TWALKER	1	0	0		
RID	□		TWALKER	1	0	0		
ACTIVATE	□		TWALKER	1	0	0		
DD	□		TWALKER	1	0	0		

Figure 5–21 shows the association of the gateway event with the Activate task.

Figure 5–21 Associating the Task with a Gateway Event



Plan Task: ACTIVATE

Gateway Event	Auto Signal	Precondition
DSL Activation	(None)	

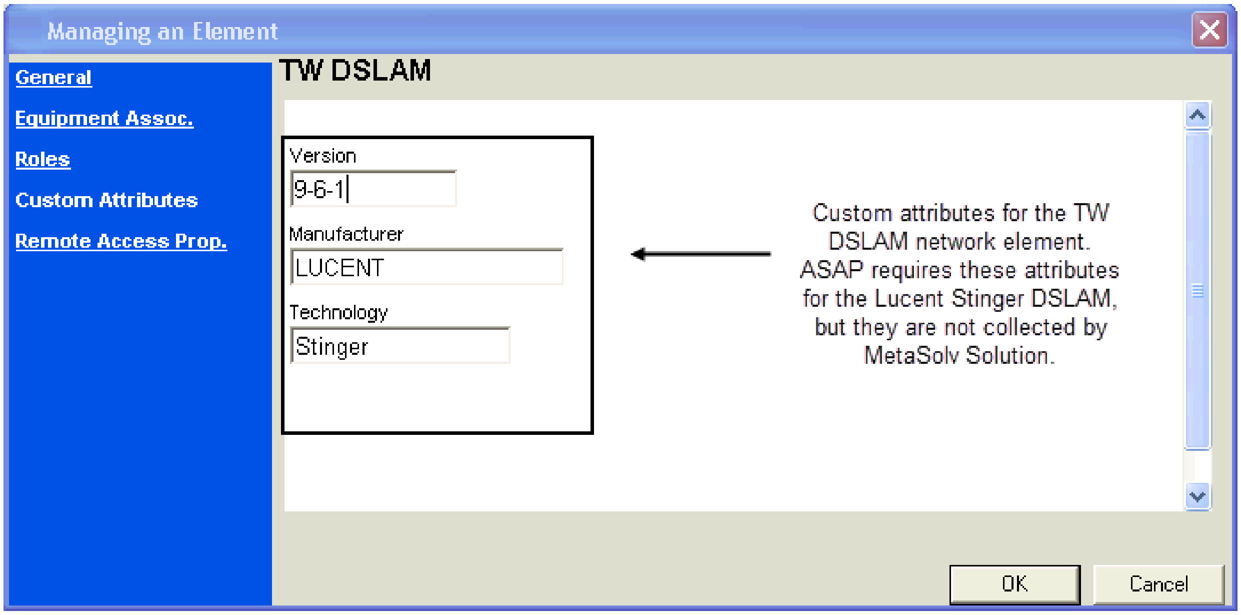
Custom Attributes

The specific custom attributes identified to collect data for the DSL flow-through package for the Lucent Stinger DSLAM include: Manufacturer, Technology, and Version.

These attributes identify the correct ASAP commands to use for activation.

Figure 5–22 shows the custom attributes, which were associated with the TW DSLAM network element.

Figure 5–22 Custom Attributes for the Lucent Stinger DSLAM Network Element



Version: 9-6-1

Manufacturer: LUCENT

Technology: Stinger

Custom attributes for the TW DSLAM network element. ASAP requires these attributes for the Lucent Stinger DSLAM, but they are not collected by MetaSolv Solution.

Figure 5–22 shows three fields. However, as an alternative, you can create a single field that carries the values for all three if you concatenate the information (for example, LU_DSLAM_9-6-1).

Mapping DSL Data

This section demonstrates how to map the parameters for two CSDL commands for the Lucent Stinger DSLAM to the corresponding attributes in MetaSolvServiceActivationType, the complex type returned by the Activation XML API. The CSDL parameters used in this example include:

- C_LU_DSLAM_9-6-1_ADD_ADSL-PORT

  The CSDL command to activate a physical connection on the Lucent Stinger DSLAM.

- C_LU_DSLAM_9-6-1_ADD_ATM_CRS_VC

  The CSDL command to activate a virtual connection on the Lucent Stinger DSLAM.

The mappings sometimes includes qualifying information.This is needed to identify the correct instance when multiple instances of a type may be present. For example, in the first table the mapping for the parameter NE_ID_LU_LUC-DSLAM includes the following qualifier: where networkElements.networkElementType = DSL Multiplexer. Because multiple network elements are expected, the qualifier is added to identify the correct one. The values used for qualifiers in the mappings in this section are examples. Actual values must be verified for each implementation.

See "Mapping Example" for an example of how to navigate through the data in MetaSolvServiceActivationType to find the CSDL parameter values.

CSDL mapping for the physical connection

Table 5–2 provides the mapping for C_LU_DSLAM_9-6-1_ADD_ADSL-PORT, the CSDL to activate a physical connection on the Lucent Stinger DSLAM.

Table 5–2 Mapping for C_LU_DSLAM_9-6-1_ADD_ADSL-PORT

CSDL Parameter	Req/ Opt	MetaSolvServiceActivationTypeMapping
NE_ID_LU_LUC-DSLAM	R	networkElements.networkElementIdentifier where: networkElements.networkElementType = DSL Multiplexer
SHELF	R	networkElements.individualConnections. connectionAssignments.shelfEquipmentName where: networkElements. individualConnections.connectionType = DSL Link
SLOT	R	networkElements.individualConnections. connectionAssignments.shelfSlotNumber where: networkElements. individualConnections.connectionType = DSL Link
PORT	R	networkElements.individualConnections. connectionAssignments.physicalNodeAddress where: networkElements. individualConnections.connectionType = DSL Link

Table 5–2 (Cont.) Mapping for C_LU_DSLAM_9-6-1_ADD_ADSL-PORT

CSDL Parameter	Req/ Opt	MetaSolvServiceActivationTypeMapping
MAX_BITRATE_UP	O	networkElements.individualConnections. connectionAssignments. connectionCustomAttributes. customerAttributeValues.value and …customerAttributeValues.unitOfMeasure where: networkElements.individualConnections. connectionAssignments. connectionCustomAttributes.label = In Service Upstream Bit Rate and networkElements.individualConnections.connectionType = DSL Link
MAX_BITRATE_DOWN	O	networkElements.individualConnections. connectionAssignments. connectionCustomAttributes. customerAttributeValues.value and …customerAttributeValues.unitOfMeasure where: networkElements.individualConnections. connectionAssignments. connectionCustomAttributes.label = In Service Downstream Bit Rate and networkElements.individualConnections.connectionType = DSL Link
MIN_BITRATE_UP	O	Default or create additional custom attribute values
MIN_BITRATE_DOWN	O	Default or create additional custom attribute values
DMT_LINE_CODE	O	Default or create additional custom attribute values
UPSTREAM_START_BIN	O	Default or create additional custom attribute values
UPSTREAM_END_BIN	O	Default or create additional custom attribute values
DOWNSTREAM_START_BIN	O	Default or create additional custom attribute values
BIT_SWAPPING	O	Default or create additional custom attribute values
LINE_LATENCY_UP	O	Default or create additional custom attribute values
LINE_LATENCY_DOWN	O	Default or create additional custom attribute values

CSDL Mapping for a Virtual Connection

Table 5–3 provides the mapping for the CSDL to activate a virtual connection on the Lucent Stinger DSLAM: C_LU_DSLAM_9-6-1_ADD_ATM_CRS_VC.

Table 5–3 Mapping for C_LU_DSLAM_9-6-1_ADD_ATM_CRS_VC

CSDL Parameter	Req/ Opt	MetaSolvServiceActivationTypeMapping
NE_ID_LU_LUC-DSLAM	R	networkElements.networkElementIdentifier where: networkElements.networkElementType = DSL Multiplexer
CIRCUIT_ID	R	networkElements.individualConnections. connectionAllocations.connectionIdentifier
VPI_USER	R	networkElements.individualConnections. connectionAllocations. relationshipCustomAttributes. customAttributeValues.value where: relationshipCustomAttributes.label = VPI and networkElements.individualConnections.connectionType = DSL Link
VCI_USER	R	networkElements.individualConnections. connectionAllocations. relationshipCustomAttributes. customerAttributeValues.value where: relationshipCustomAttributes.label = VCI and networkElements.individualConnections. connectionType = DSL Link
NAILED_GROUP_LIM	R	Calculated Value
VPI_WAN	R	networkElements.individualConnections. connectionAllocations. relationshipCustomAttributes. customerAttributeValues.value where: relationshipCustomAttributes.label = VPI and networkElements.individualConnections. connectionType = Assignable Bandwidth Link
VCI_WAN	R	networkElements.individualConnections. connectionAllocations. relationshipCustomAttributes. customerAttributeValues.value where: relationshipCustomAttributes.label = VCI and networkElements.individualConnections. connectionType = Assignable Bandwidth Link
NAILED_GROUP_TM	R	Calculated Value
QOS_UPSTREAM_ CONTACT	R	Default Value
QOS_DOWNSTREAM_ CONTACT	R	Default Value

Mapping Example

This section explains how to navigate through the activation data returned by the Activation XML API to find the values necessary to populate the CSDL parameters listed in the previous section.

Service Provider's Internal Network

To provide DSL service, a service provider builds an internal network containing equipment, network elements, and connections that support the service. In this example we assume that the service provider has a network containing the following:

Network Elements

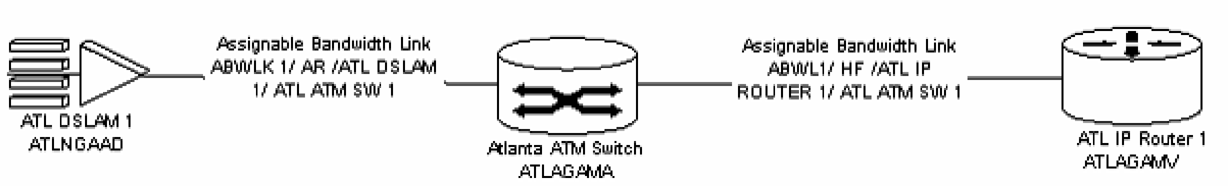
- DSL Access Multiplexer (DSLAM) located at ATLNGAAD named ATL DSLAM 1
- ATM Switch located at ATLAGAMA named Atlanta ATM Switch
- IP Router located at ATLAGAMV named ATL IP Router 1

Physical Connections

- Assignable Bandwidth Link connection linking the DSLAM to the ATM switch
- Assignable Bandwidth Link connection linking the ATM switch to the IP Router

Figure 5–23 shows the setup of a service provider internal network as it appears in the MetaSolv Solution network system.

Figure 5–23 Service Provider Internal Network



Custom Attributes

These are set up in the MetaSolv Solution Utility and entered for the ATL DSLAM 1 network element. The custom attributes are used to determine which CSDL commands to use for the network element.

- Manufacturer = Lucent

  This identifies the manufacturer of the equipment on the network element.

- Technology = DSLAM

  This identifies the equipment and technology.

- Version = 9-3-170

  This identifies the software version on the network element.

DSL Order

The service provider enters an order from a customer for DSL service. Based on the service provider's product catalog, the DSL order consists of the following items:

- A physical connection of type DSL Link that connects the customer site to the service provider's DSLAM
- A virtual connection of type Internet Access

DSL Link Connection Design

The service provider designs the ordered DSL link connection. This physical connection is designed to run from the customer site to the service provider's DSLAM network element ATL DSLAM 1 located at ATLNGAAD. The identifier for the connection is:

```
DSL LINK 1/HF / ACS OKCITY/ATLNAAD DSLAM
```

The DSLAM equipment is a Lucent Stinger. The details of the connection assignment to the DSLAM equipment are as follows:

- Shelf = 1234
- Slot = 01
- Port = LIM-1-2

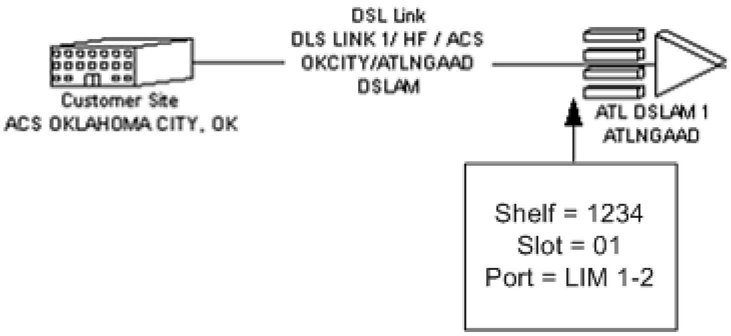
The following values are examples of custom attributes a service provider can define for the DSL link connection:

- In Service Upstream Bit Rate = 56 K
- In Service Downstream Bit Rate = 56 K

You can define any parameter regarding the DSL link connection that is not captured in MetaSolv Solution as a custom attribute.

Figure 5–24 shows the DSL link connection as it appears in the MetaSolv Solution network system.

Figure 5–24 DSL Link Connection



Internet Access Connection Design

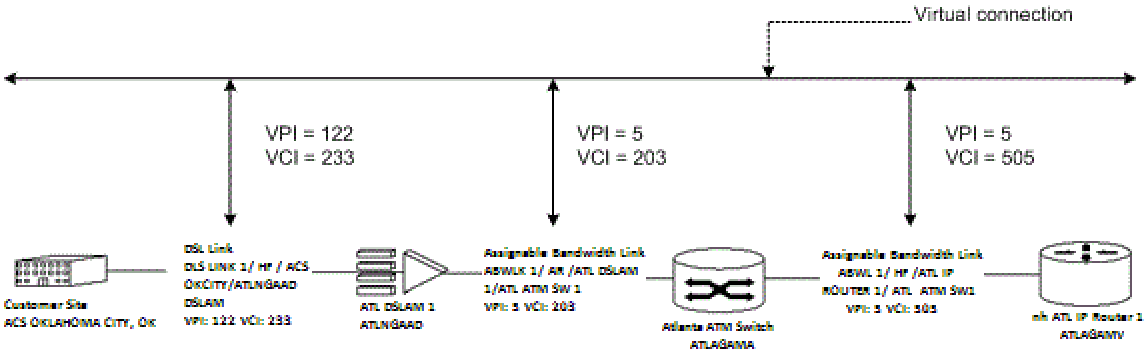
The service provider next designs the ordered internet access connection. This virtual connection is designed to run over the previously designed DSL link connection which terminates at the ATL DSLAM 1 network element. From there, the internet access connection runs over the service provider's internal network from ATL DSLAM 1 through the Atlanta ATM Switch and terminates at the ATL IP Router 1. The connection identifier for the internet access connection in this example is:

```
INTERNET CONN 1/ HF / ACS OKCITY/ATLAGAMV IP ROUTER
```

The virtual connection is allocated to each physical connection in the path. A virtual path identifier (VPI) and virtual channel identifier (VCI) are assigned to each combination of virtual connection and physical connection.

Figure 5–25 shows the virtual connection and VPI/VCI values for each physical connection to which the virtual connection is allocated.

Figure 5–25 Internet Access Connection



Mapping the Example Activation Data

Once the connection designs are complete, the activation process is initiated from the Activate task in the MetaSolv Solution Work Queue Manager. This results in a gateway event containing the order key being passed to the service provider’s MetaSolv Solution/ASAP integration layer. The event is received by the integration layer, triggering a call to the Activation XML API to get activation data for the order. The data is used by the integration layer to build and pass an order to ASAP containing command parameters to activate the DSL service.

This section describes how to access the data values for the example described above from the activation data and map them to the CSDL parameters for the Lucent Stinger.

The XML complex type containing the activation data is MetaSolvServiceActivationType. This type contains data for all services to be activated on an order, and at its highest level contains the following information:

- **orderDetails** (OrderInformationType)

  This container represents information about the activation order.

- **switchActivations** (an array of SwitchActivationType)

  This container represents a collection of switch-related services on the order, such as POTS. The collection contains a SwitchActivationType instance for every switch location associated with the ordered services.

- **networkElements** (an array of NetworkElementType)

  This container represents a collection of network elements that meet any of the following criteria:

  – The network element is an item on the order.

  – A connection on the order is assigned to the network element.

  – A connection on the order is allocated to a connection assigned to the network element.

- **networkSystems** (an array of NetworkSystemType)

  This container represents a collection of ordered network systems such as a virtual private network (VPN). The data describes the network system and all network elements and connections contained within the network system.

For this example, details for every attribute or complex type contained in MetaSolvServiceActivationType are not provided. Only information relevant to the example described in previous sections are covered. For more information about the structure and attributes in MetaSolvServiceActivationType, see the XML schema **XmlMetaSolvServiceEntities.xsd**.

Step 1: Identify the DSLAM Network Element

In this example, no switch-related services or network systems are being ordered, so switchActivations and networkSystems will not be populated under MetaSolvServiceActivationType.

The networkElements collection will be populated with a NetworkElementType for each network element in the path of the virtual connection. That includes the following network elements:

- ATL DSLAM 1
- Atlanta ATM Switch
- ATL IP Router 1

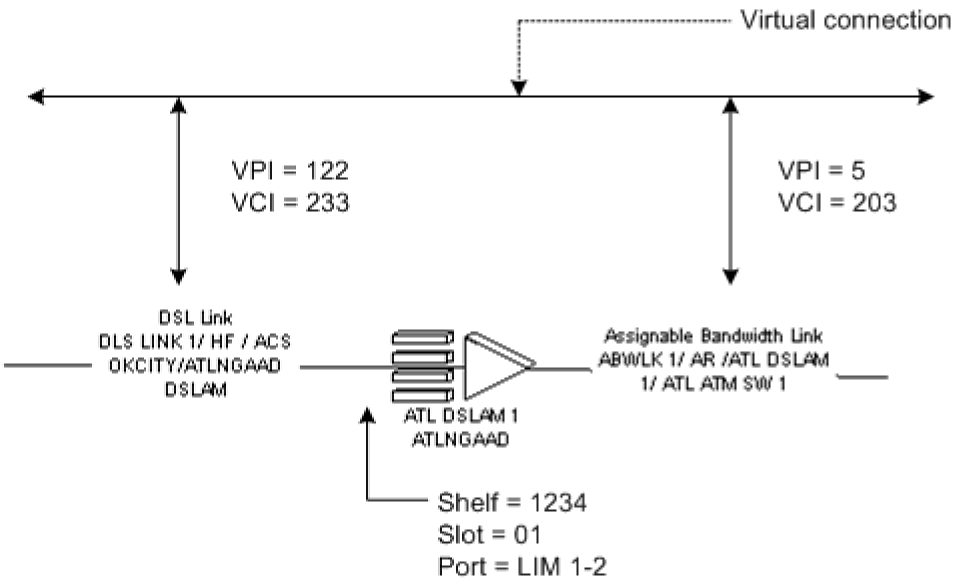
The data contained within a NetworkElementType includes information about the network element as well as information about the physical connections assigned to the network element and the virtual connections that are allocated to those physical connections.

For example, the NetworkElementType for ATL DSLAM 1 will contain information about:

- The DSLAM network element
- The virtual connection
- The DSL link connection on the user side of the DSLAM, including VPI and VCI values for the relationship between the DSL link connection and the virtual connection.
- The assignable bandwidth link connection on the wide area network (WAN) side of the DSLAM, including VPI and VCI values for the relationship between the assignable bandwidth link connection and the virtual connection.

Figure 5–26 shows the information contained in NetworkElementType for ATL DSLAM 1.

Figure 5–26 Information Contained in NetworkElementType for ATL DSLAM 1



This pattern is repeated for each network element. In this example, all of the data needed to populate the CSDL parameters is contained in the NetworkElementType for ATL DSLAM 1. Therefore, the NetworkElementType data for the Atlanta ATM switch and the ATL IP router 1 can be ignored.

The following attributes of NetworkElementType are relevant to the example:

- networkElementType = DSL Multiplexer

  Network element types are defined by a service provider in MetaSolv Solution, which means the possible values depend on the implementation. This attribute is used to differentiate between the network elements in the collection.

- networkElementIdentifier = ATL DSLAM 1

  This attribute uniquely identifies the network element. It maps to CSDL parameter NE_ID_LU_LUC-DSLAM in the CSDLs for both the physical and the virtual connection.

- activityCd = “”

  This attribute is the activity the order specifies to be performed on an item. In this example activityCd is blank because the DSLAM is not being ordered. Ordered items include the DSL link and internet access connections. The containers representing these connections will have a populated activityCd. These containers are described in the following section.

- elementCustomAttributes

  This is a collection of ActivationCustomAttributeType instances representing custom attributes for the network element. The relevant custom attributes for this example identify manufacturer, technology, and version of the network element. These attributes are needed to build the correct CSDL commands for the DSLAM:

  – ActivationCustomAttributeType.label = Manufacturer

    ActivationCustomAttributeType.customAttributeValues.value = Lucent

  – ActivationCustomAttributeType.label = Technology

    ActivationCustomAttributeType.customAttributeValues.value = DSLAM

- ActivationCustomAttributeType.label = Version

  ActivationCustomAttributeType.customAttributeValues.value = 9-3-170

- individualConnections

  This is a collection of IndividualConnectionType instances representing connections assigned to the network element. Only connections that are on the order or are included in the path of an ordered virtual connection are included. More detail about what information is contained in each IndividualConnectionType is presented in the following section.

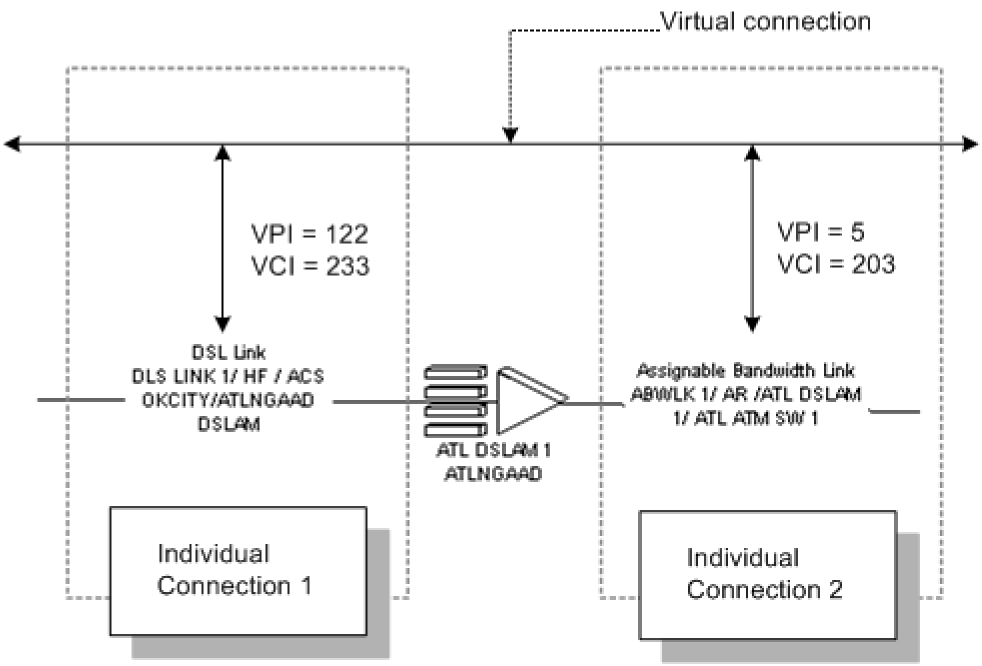
Step 2: Get data for the DSL link connection

The previous step demonstrated how to find the instance of NetworkElementType that represents the DSLAM and identify the relevant attributes used to populate some of the CSDL command parameters. This step explains how to use the instances of IndividualConnectionType to get data for the DSL link connection.

In this example, there are two instances of IndividualConnectionType under the DSLAM network element. The two instances represent information about the DSL link connection going into the DSLAM and the assignable bandwidth link connection coming out of the

DSLAM. In addition, each IndividualConnectionType instance contains information about the virtual connection and the allocation of the virtual connection to the physical connection, as shown in Figure 5–27.

Figure 5–27 IndividualConnectionType instances (2) Under Network Element ATL DSLAM1



In the case where more than one DSL link connection is ordered, there will be more than two instances of IndividualConnectionType. For example, if an order contains two DSL Link connections, and each is assigned to a port on ATL DSLAM 1, there will be three instances of IndividualConnectionType: two for the ordered DSL link connections assigned to the DSLAM, and one for the Assignable Bandwidth link coming out of the DSLAM.

The instance of IndividualConnectionType representing the DSL link connection contains the following attributes:

- connectionType = DSL Link

  This is the connection type of the physical connection represented by the IndividualConnectionType. In this example, this attribute is used to distinguish between the IndividualConnectionType instance that represents the DSL link connection and the one that represents the connection of type assignable bandwidth link.

- connectionAssignments

  This is a collection of ConnectionAssignmentType instances, each representing a separate port on the DSLAM to which the DSL link connection is connected.

  The example has one ConnectionAssignmentType instance with the following values that map to parameters on the CSDL C_LU_DSLAM_9-6-1_ADD_ADSL-PORT command.

  - shelfEquipmentName = 1234

    This attribute maps to the SHELF parameter.

  - shelfSlotNumber = 01

    This attribute maps to the SLOT parameter.

  - physicalNodeAddress = LIM-1-2

    This attribute maps to the PORT parameter

  - activityCd = iactn

    This attribute is the activity to be performed on an item as specified by the order. In this example, the activityCd = iactn, meaning new. This means the CSDL command applied will be an add of the connection assignment as opposed to a change (activityCd = iactc) or a disconnect (activityCd = iactd).

- connectionCustomAttributes

  This is a collection of ActivationCustomAttributeType instances representing custom attributes for the DSL link connection.

  The relevant custom attributes for this example are:

  - ActivationCustomAttributeType.label = In Service Upstream Bit Rate

    ActivationCustomAttributeType.customAttributeValues.value = 56

    ActivationCustomAttributeType.customAttributeValues.unitOfMeasure = K

    This attribute maps to the MAX_BITRATE_UP parameter

  - ActivationCustomAttributeType.label = In Service Downstream Bit Rate

    ActivationCustomAttributeType.customAttributeValues.value = 56

    ActivationCustomAttributeType.customAttributeValues.unitOfMeasure = K

    This attribute maps to the MAX_BITRATE_DOWN parameter

- connectionAllocations

  This is a collection of ConnectionAllocationType instances representing virtual connections that are allocated to the DSL link connection. In this example, there is only one virtual connection on the order, so there is only one instance of ConnectionAllocationType. The data from this instance is used to populate the CSDL for the internet access connection. How the data maps to the CSDL parameters is explained in the following section.

Step 3: Get Data for the Internet Access Connection

This section explains how to locate values to populate parameters for the following command which is the CSDL command to activate a virtual connection on the Lucent Stinger DSLAM:

```
C_LU_DSLAM_9-6-1_ADD_ATM_CRS_VC
```

As noted earlier, the following parameter:

```
 NE_ID_LU_LUC-DSLAM
```

is an identifier of the DSLAM network element and is populated with the value from NetworkElementType.networkElementIdentifier.

Note that the instance of IndividualConnectionType representing the DSL link connection contains a connectionAllocations attribute. In the example, connectionAllocations contains a ConnectionAllocationType instance for the internet access connection on the order. The following attributes on ConnectionAllocationType apply to this example:

- connectionIdentifier = INTERNET CONN 1/ HF / ACS OKCITY/ATLAGAMV IP ROUTER

  This is the connection identifier for the Internet Access connection. It maps to the CIRCUIT_ID parameter in the CSDL.

- relationshipCustomAttributes

  This is a collection of ActivationCustomAttributeType instances representing custom attributes for the relationship between the internet access connection and the DSL link connection. If there are any parameters specific to the Internet access connection that are not otherwise captured in MetaSolv Solution, they can be defined as custom attributes and included in this collection. The relevant custom attributes for this example are:

  – ActivationCustomAttributeType.label = VPI

    ActivationCustomAttributeType.customAttributeValues.value = 122

    This attribute maps to the VPI_USER parameter

  – ActivationCustomAttributeType.label = VCI

    ActivationCustomAttributeType.customAttributeValues.value = 233

    This attribute maps to the VCI_USER parameter

    ConnectionAllocationType also contains a connectionCustomAttributes attribute. Unlike relationshipCustomAttributes, which are custom attributes for the relationship between the physical and virtual connections, this collection represents custom attributes for just the virtual connection.

    In addition to the VPI and VCI for the user side of the DSLAM, the CSDL requires the VPI and VCI for the wide area network (WAN) side of the DSLAM. This information is contained in the IndividualConnectionType instance that represents the Assignable Bandwidth Link connection on the opposite side of the DSLAM (see Individual Connection 2 in Figure 5). In our example, this IndividualConnectionType instance is identified by the connectionType attribute = Assignable Bandwidth Link.

That IndividualConnectionType instance will have a connectionAllocations attribute which contains a ConnectionAllocationType instance for the Internet Access connection. This instance will be identical to the instance we identified on the user side of the DSLAM, with the exception of the relationshipCustomAttributes attribute. On the WAN side, this will include custom attributes for the relationship between the Assignable Bandwidth Link connection and the Internet Access connection. The relevant custom attributes for this example are:

- ActivationCustomAttributeType.label = VPI

  ActivationCustomAttributeType.customAttributeValues.value = 5

  This attribute maps to the VPI_WAN parameter

- ActivationCustomAttributeType.label = VCI

  ActivationCustomAttributeType.customAttributeValues.value = 203

  This attribute maps to the VCI_WAN parameter

6

Viewing The POTS/DSL Flow-through Package

This chapter describes the POTS/DSL flow-through package. The description gives a high-level description of the data flow between the workflows in the package. As you read the sections in this chapter, open the workflows in Workshop and view the contents.

What's In the Package

The package contains workflows that can be used for both POTS and DSL activation.

For the POTS portion of the package, activation set up for the following activities:

- Residential line: Add, Change, Disconnect
- Line feature (call waiting): Add, Change, Delete

The switch used for POTS is the Lucent 5ESS. This switch determines the ASAP cartridge and CSDL commands used.

For DSL, activation is set up for adding DSL service for a user. Voice is not included in the DSL example. The DSLAM used for DSL is the Lucent Stinger. This equipment determines the ASAP cartridge and CSDL commands used.

- **Event management workflows**

  Table 6–1 lists the workflows that manage messages between Oracle Communications MetaSolv Solution, the integration layer, and Oracle Communications ASAP.

  Table 6–1 Event Management Workflows

Workflow	Description
ActivationOrderCreateEventHandler ActivationOrderCompleteEventHandler AcitvationOrderFailEventHandle	These workflows subscribe to ASAP outbound events and update MSS with the status and external keys.
MSSOutboundEventHandler	This workflow subscribes to the MetaSolv Solution JMS queue and processes the activation.

- **Order management workflows**

  The following workflows manage the activation order by requesting data from MetaSolv Solution, decomposing the activation data it receives in response, and constructing an activation order for ASAP based on CSDL commands:

  ActivationOrderManager

ConnectionAllocationsHandler

ConnectionAssignmentHandler

IndividualConnectionHandler

LineFeatureHandler

LineServiceActivationHandler

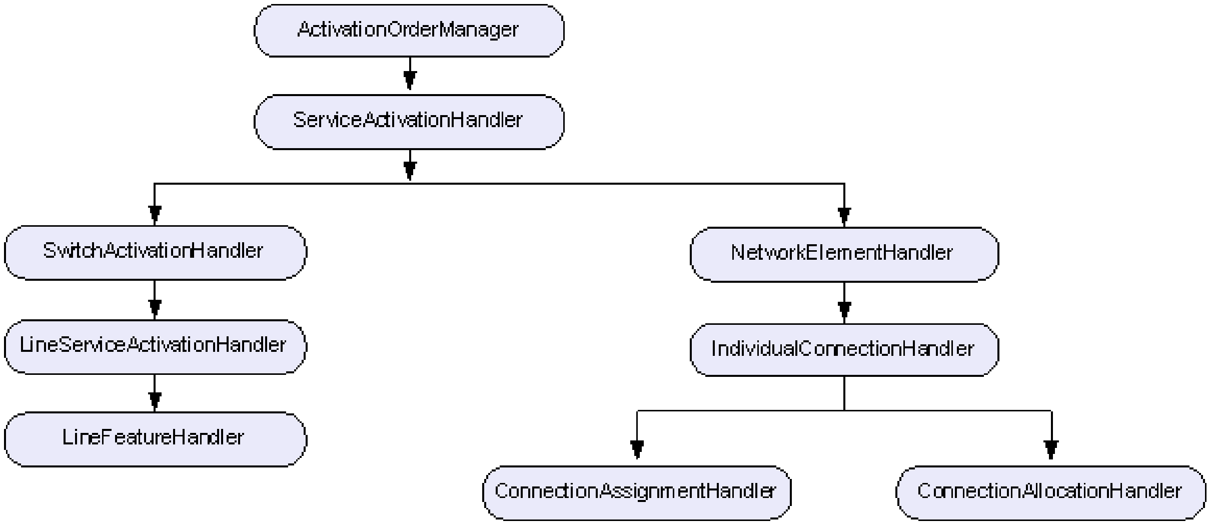
NetworkElementHandler

ServiceActivationHandler

SwitchActivationHandler

The order management workflows are structured in the hierarchy shown in Figure 6–1.

Figure 6–1 Order Management Workflow Hierarchy


ActivationOrderManager
ServiceActivationHandler
SwitchActivationHandler
NetworkElementHandler
LineServiceActivationHandler
IndividualConnectionHandler
LineFeatureHandler
ConnectionAssignmentHandler
ConnectionAllocationHandler

- **XQuery Transformation Files**

  The following transformation files use XQuery mappings to transform MetaSolv Solution and ASAP data into the proper XML formats as the data moves between the applications and the integration layer during the activation process.

  – ActivationRequestConverter (common XQuery workflows for requests)

  makeASAPOrderFromServiceActivation

  makeCreateOrderByValueRequestFromASAPOrder

  makeGetActivationRequestFromMSSEvent

  makeServiceActivationFromGetActivationResponse

  – ActivationResponseConverter (common XQuery workflows for responses)

  makeEventStatusChangeFromCreateOrderException

  makeEventStatusChangeFromOrderCompleteEvent

  makeEventStatusChangeFromOrderCreateEvent

  makeEventStatusChangeFromOrderFailEvent

- LuStingerActivationRequestConverter (DSL workflows specific to the Lucent Stinger DSLAM)

  makeAddDSLPortFromConnectionAssignment

  makeAddVCFromConnectionAllocation

  makeWanConnAllocArrayFromIndividualConnection

- LU5ESSActivationRequestConverter (POTS workflows specific to the Lucent 5ESS switch)

  makeAddPotsCallWaitingFromActivationData

  makeAddPotsResLineFromActivationData

  makeChangePotsCallWaitingFromActivationData

  makeDeletePotsCallWaitingFromActivationData

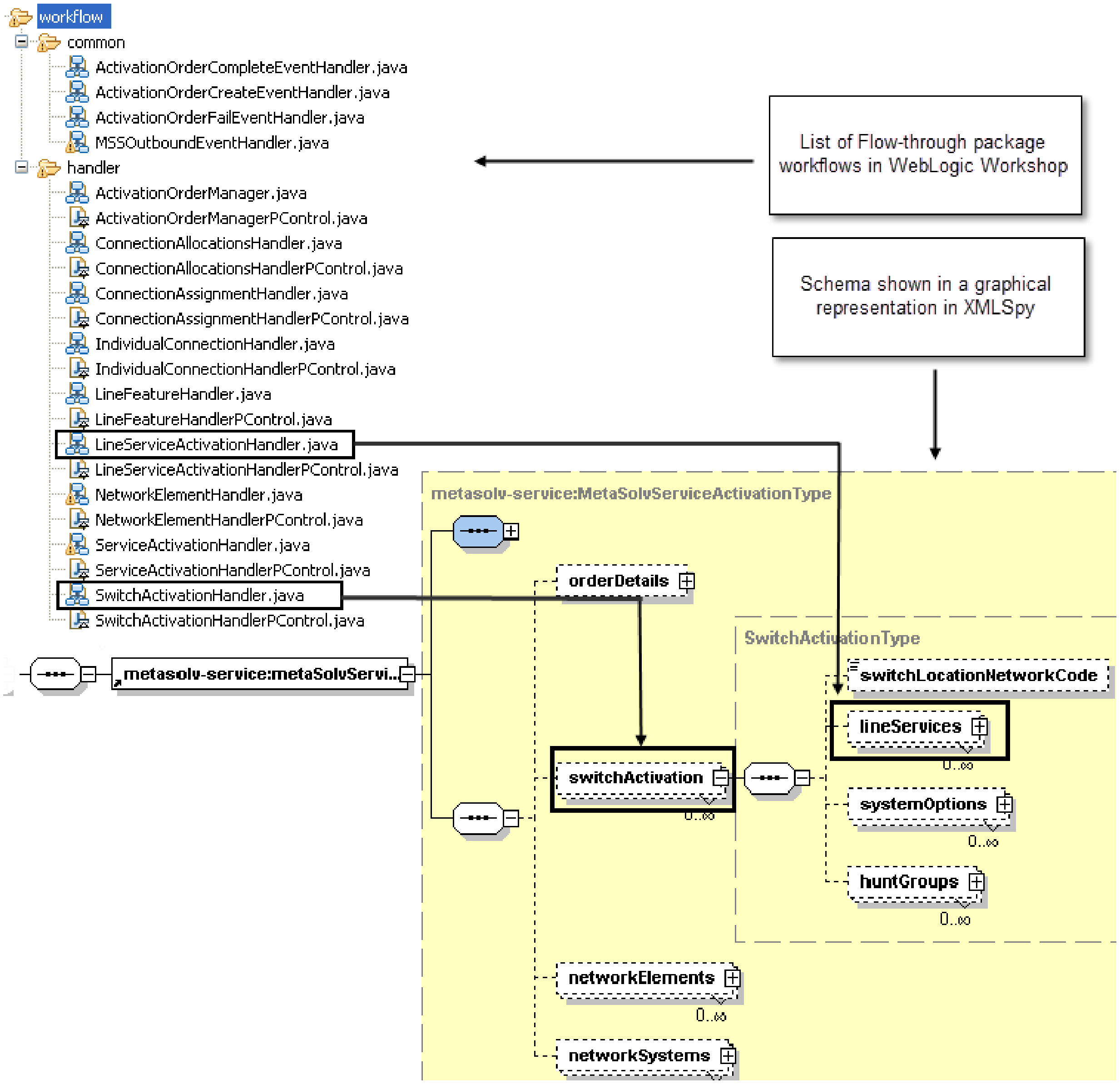
  makeDeletePotsLineFromActivationData

Naming Conventions

The flow-through package uses the same names for the workflows that appear in the schema as container names with a few exceptions.

Figure 6–2 shows the switchActivation portion of the schema for the getActivationDataByKeyResponse. The figure also shows the schema displayed graphically in XMLSpy. The switchActivation name in the schema is SwitchActivationHandler in the workflows, and the lineServices name from the schema is LineServiceActivationHandler in the workflows.

Figure 6–2 Schema Showing Switch Activation Container


workflow
common
ActivationOrderCompleteEventHandler.java
ActivationOrderCreateEventHandler.java
ActivationOrderFailEventHandler.java
MSSOutboundEventHandler.java
handler
ActivationOrderManager.java
ActivationOrderManagerPControl.java
ConnectionAllocationsHandler.java
ConnectionAllocationsHandlerPControl.java
ConnectionAssignmentHandler.java
ConnectionAssignmentHandlerPControl.java
IndividualConnectionHandler.java
IndividualConnectionHandlerPControl.java
LineFeatureHandler.java
LineFeatureHandlerPControl.java
LineServiceActivationHandler.java
LineServiceActivationHandlerPControl.java
NetworkElementHandler.java
NetworkElementHandlerPControl.java
ServiceActivationHandler.java
ServiceActivationHandlerPControl.java
SwitchActivationHandler.java
SwitchActivationHandlerPControl.java
List of Flow-through package workflows in WebLogic Workshop
Schema shown in a graphical representation in XMLSpy
metasolv-service:metaSolvServi...
metasolv-service:MetaSolvServiceActivationType
orderDetails
SwitchActivationType
switchLocationNetworkCode
lineServices
0..∞
switchActivation
0..∞
systemOptions
0..∞
huntGroups
0..∞
networkElements
0..∞
networkSystems

How the Workflows for POTS and DSL Handle Data

The following section describes how the workflows handle data and events for activation. The entire activation process is triggered when the Activate task in MetaSolv Solution is completed and an event is sent to the MSSOutboundEvent processor in the integration layer.

Figure 6–3 shows how the data flow in the integration layer is controlled by processors.

Figure 6–3 Data Flow in the Integration Layer is Controlled by Processors

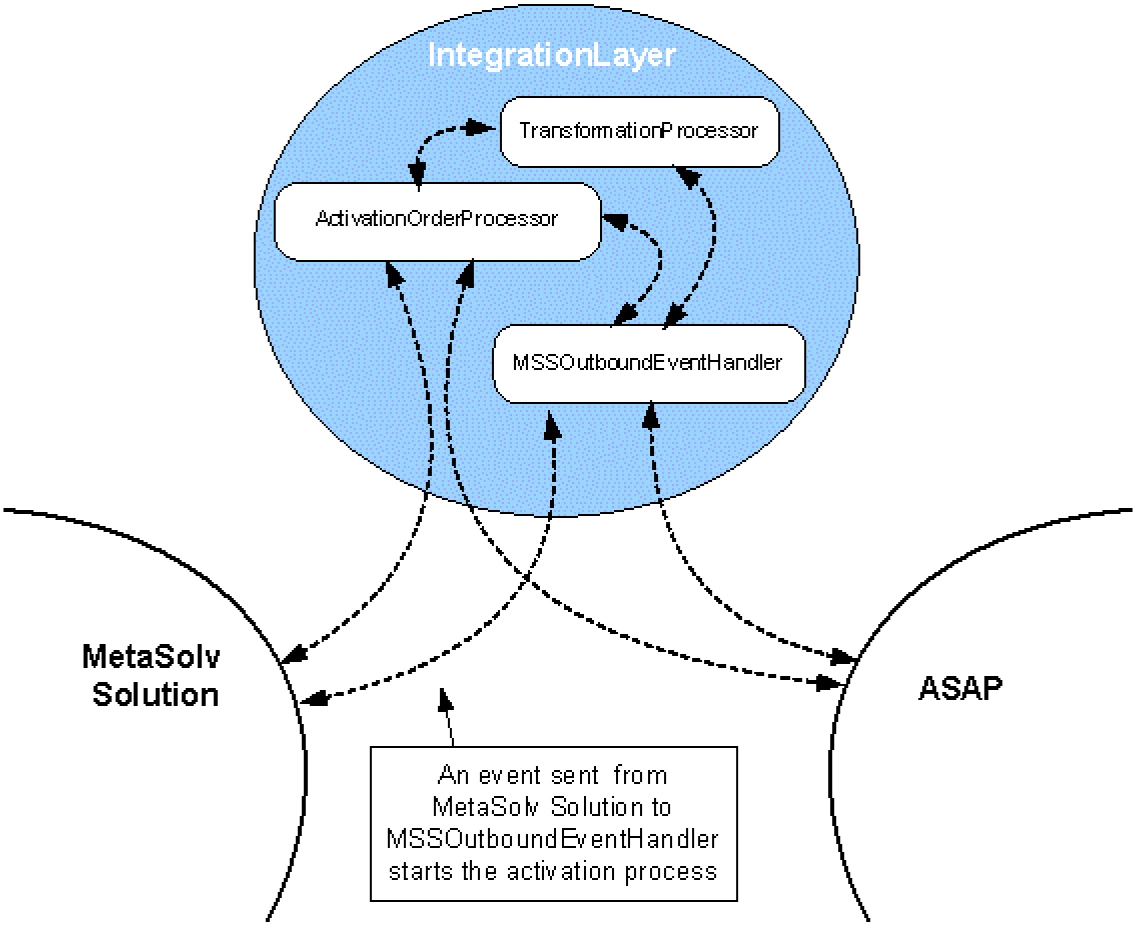
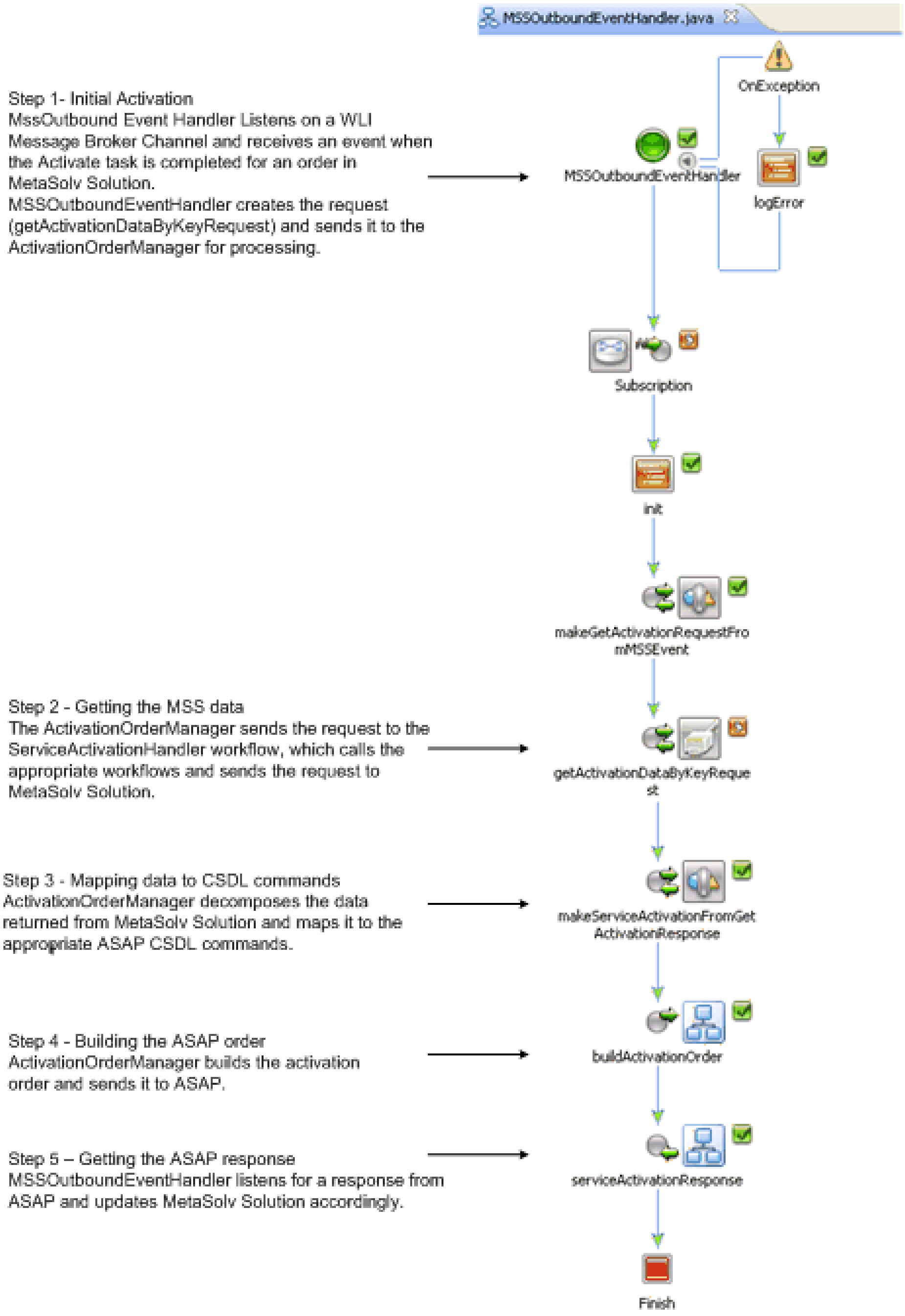


Figure 6–4 shows the MSSOutboundEventHandler workflow. This workflow is initially called when an event is detected. The remaining workflows for order management and transformation management are called from this workflow.

Figure 6–4 MSSOutboundEventHandler Workflow


MSSOutboundEventHandler.java
OnException
MSSOutboundEventHandler
logError
Subscription
init
makeGetActivationRequestFromMSSEvent
getActivationDataByKeyRequest
makeServiceActivationFromGetActivationResponse
buildActivationOrder
serviceActivationResponse
Finish

Step 1- Initial Activation
MssOutbound Event Handler Listens on a WLI Message Broker Channel and receives an event when the Activate task is completed for an order in MetaSolv Solution.
MSSOutboundEventHandler creates the request (getActivationDataByKeyRequest) and sends it to the ActivationOrderManager for processing.

Step 2 - Getting the MSS data
The ActivationOrderManager sends the request to the ServiceActivationHandler workflow, which calls the appropriate workflows and sends the request to MetaSolv Solution.

Step 3 - Mapping data to CSDL commands
ActivationOrderManager decomposes the data returned from MetaSolv Solution and maps it to the appropriate ASAP CSDL commands.

Step 4 - Building the ASAP order
ActivationOrderManager builds the activation order and sends it to ASAP.

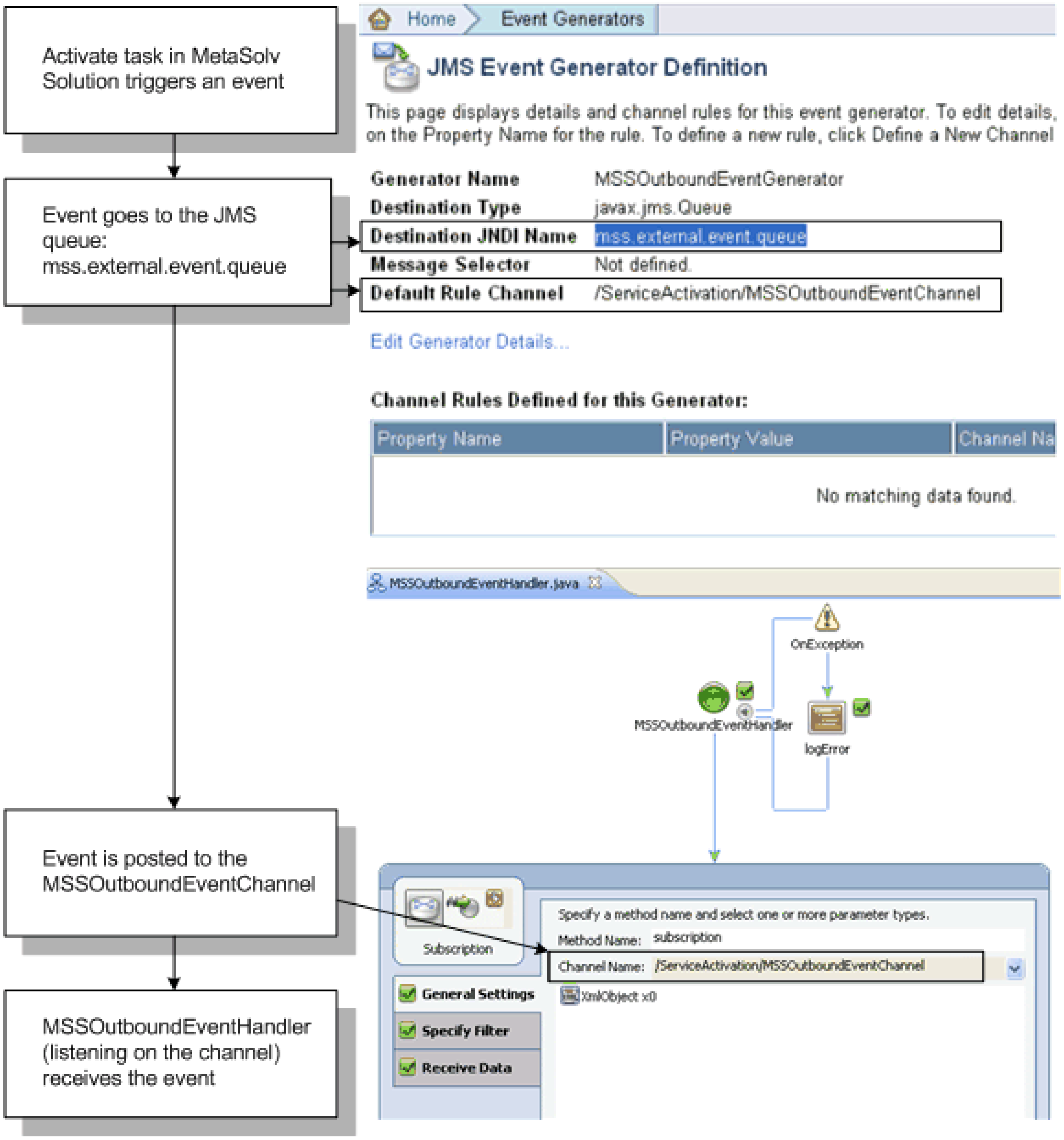
Step 5 – Getting the ASAP response
MSSOutboundEventHandler listens for a response from ASAP and updates MetaSolv Solution accordingly.

Step 1: Initial Activation Event

Figure 6–5 shows how event messages were set up in WebLogic for the flow-through package:

- WebLogic Integration setup of the JMS queue for outbound MetaSolv Solution activation events
- Identification of the channel name that MSSOutboundEventHandler subscribes to in the workflow in WebLogic Workshop

Figure 6–5 Event Setup in the WebLogic Integration Console and in Workshop



Activate task in MetaSolv Solution triggers an event
Event goes to the JMS queue: mss.external.event.queue
Event is posted to the MSSOutboundEventChannel
MSSOutboundEventHandler (listening on the channel) receives the event
Home
Event Generators
MSSOutboundEventHandler.java
OnException
MSSOutboundEventHandler
logError
Subscription
General Settings
Specify Filter
Receive Data

JMS Event Generator Definition

This page displays details and channel rules for this event generator. To edit details, on the Property Name for the rule. To define a new rule, click Define a New Channel

Generator Name	MSSOutboundEventGenerator
Destination Type	javax.jms.Queue
Destination JNDI Name	mss.external.event.queue
Message Selector	Not defined.
Default Rule Channel	/ServiceActivation/MSSOutboundEventChannel

Edit Generator Details...

Channel Rules Defined for this Generator:

Property Name	Property Value	Channel Na
	No matching data found.	

Specify a method name and select one or more parameter types.

Method Name: subscription

Channel Name: /ServiceActivation/MSSOutboundEventChannel

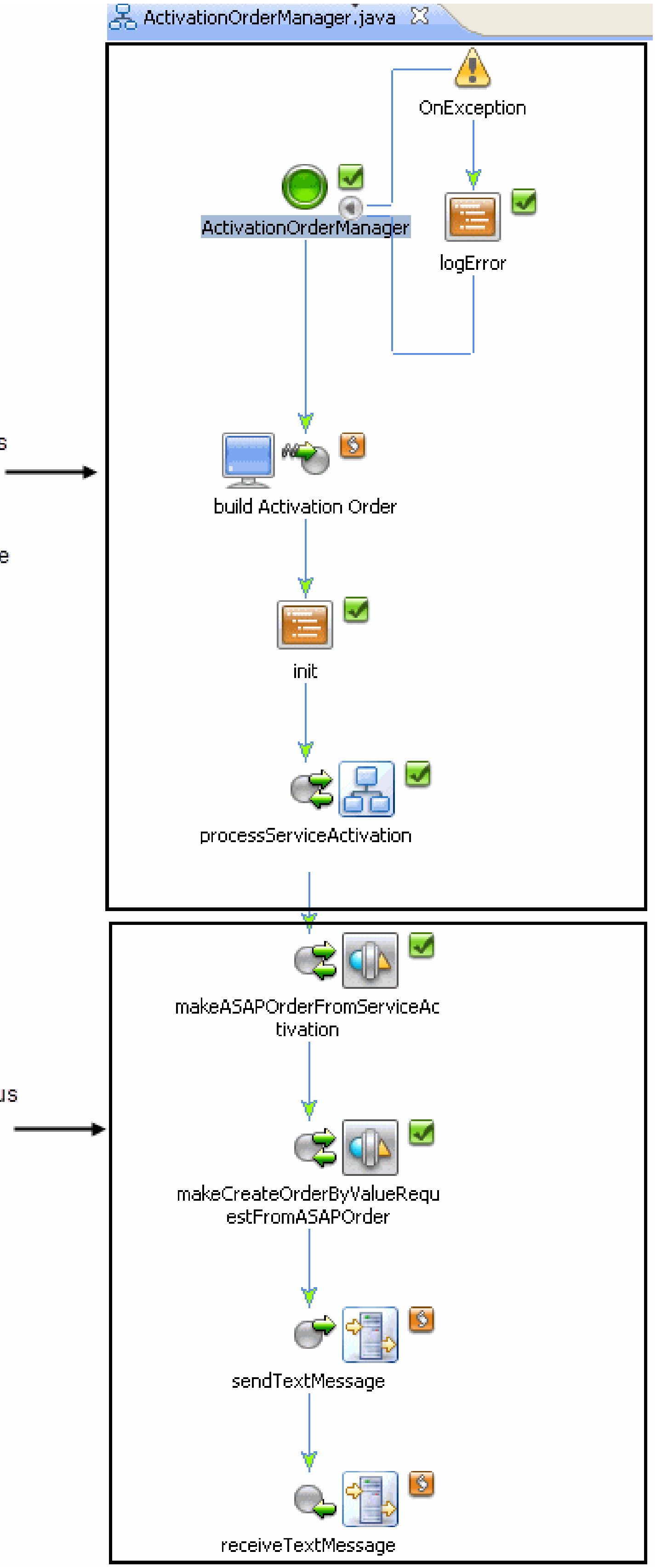
XmlObject x0

Step 2: Getting MetaSolv Solution Data

MSSOutboundEventHandler sends the request to ActivationOrderManager which handles getting activation data from MetaSolv Solution, decomposing the data, mapping the data to CSDL commands and building the ASAP order.

Figure 6–6 shows the ActivationOrderManager workflow.

Figure 6–6 ActivationOrderManager Workflow



This part of the workflow handles:

- Getting activation data from MetaSolv Solution
- Building the required CSDL commands using the MetaSolv Solution data

The CSDL commands built during this part of the workflow are used to build the ASAP order in the next part of the workflow.

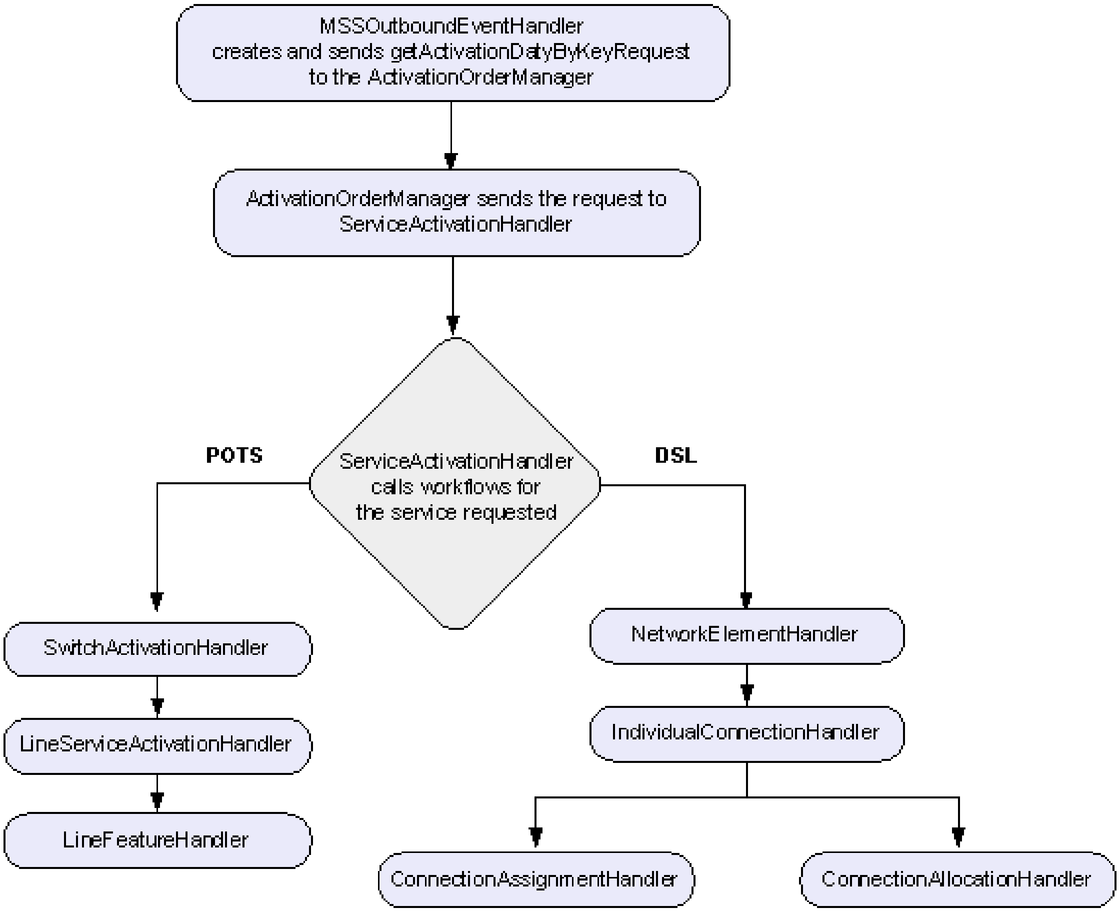
This part of the workflow handles:

- Building the order from the CSDL commands constructed in the previous part of the workflow
- Sending the order to ASAP

ServiceActivationHandler (a workflow called by ActivationOrderManager) determines the workflows required by a request and makes the call for data to MetaSolv Solution.

Figure 6–7 shows how POTS and DSL activation data requests are managed by ServiceActivationHandler.

Figure 6–7 ServiceActivationHandler Managing POTS and DSL Activation


MSSOutboundEventHandler creates and sends getActivationDatyByKeyRequest to the ActivationOrderManager
ActivationOrderManager sends the request to ServiceActivationHandler
POTS
ServiceActivationHandler calls workflows for the service requested
DSL
SwitchActivationHandler
LineServiceActivationHandler
LineFeatureHandler
NetworkElementHandler
IndividualConnectionHandler
ConnectionAssignmentHandler
ConnectionAllocationHandler

For DSL, ServiceActivationHandler calls NetworkElementHandler. SwitchActivationHandler is not involved in the request for DSL data because there is no switch involved directly in DSL activation. The physical line for a DSL activation is connected to a DSLAM. All lines from the DSLAM to the Internet are virtual lines. The properties required for activation in the case of DSL can be captured through network elements in the network system over which service is provisioned.

Step 3: Mapping the Data to CSDL Commands

When the request data is returned from MetaSolv Solution (through the method getActivationDataByKeyResponse), ActivationOrderManager maps the data to the appropriate ASAP CSDL command. For POTS, the XQuery workflows that map CSDL commands for the Lucent 5ESS switch for the flow-through package are shown in the following figures. The POTS portion of the flow-though package was designed to add or delete a residential line.

Figure 6–8 shows the XQuery workflows that map CSDL commands for adding and deleting a line.

Figure 6–8 Mapping CSDL Commands for Adding or Deleting a POTS Residential Line

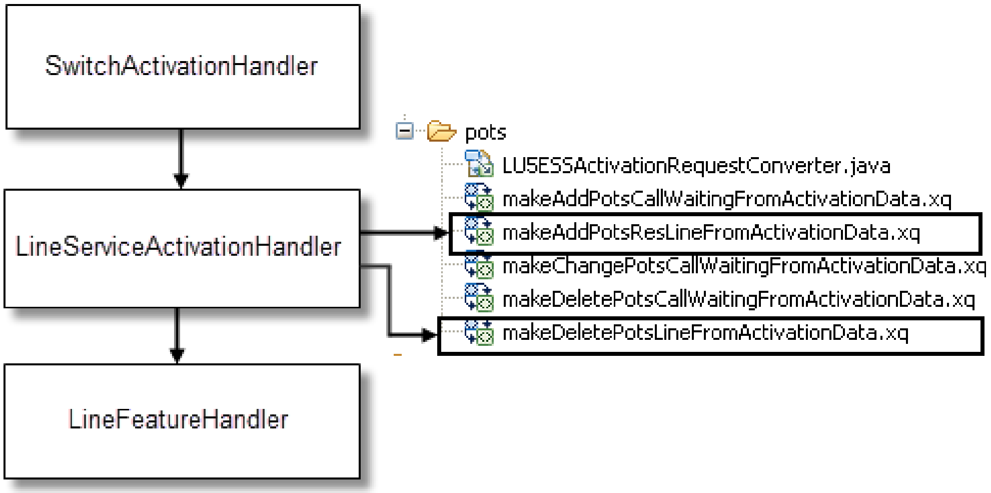

SwitchActivationHandler
LineServiceActivationHandler
LineFeatureHandler
pots
LU5ESSActivationRequestConverter.java
makeAddPotsCallWaitingFromActivationData.xq
makeAddPotsResLineFromActivationData.xq
makeChangePotsCallWaitingFromActivationData.xq
makeDeletePotsCallWaitingFromActivationData.xq
makeDeletePotsLineFromActivationData.xq

Figure 6–9 shows the XQuery workflows that map CSDL commands for adding, changing, or deleting call waiting, which is a line feature (or line option).

Figure 6–9 Mapping CSDL Commands for a POTS Line Feature (Call Waiting)

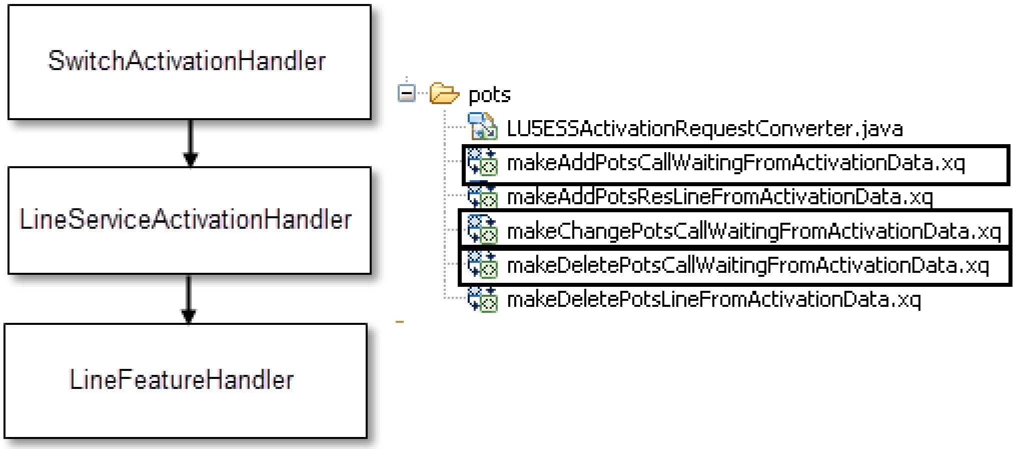

SwitchActivationHandler
LineServiceActivationHandler
LineFeatureHandler
pots
LU5ESSActivationRequestConverter.java
makeAddPotsCallWaitingFromActivationData.xq
makeAddPotsResLineFromActivationData.xq
makeChangePotsCallWaitingFromActivationData.xq
makeDeletePotsCallWaitingFromActivationData.xq
makeDeletePotsLineFromActivationData.xq

Figure 6–10 shows the XQuery workflows that map CSDL commands for adding DSL service.

Figure 6–10 Mapping CSDL Commands for Creating the Physical and Virtual Connections for DSL

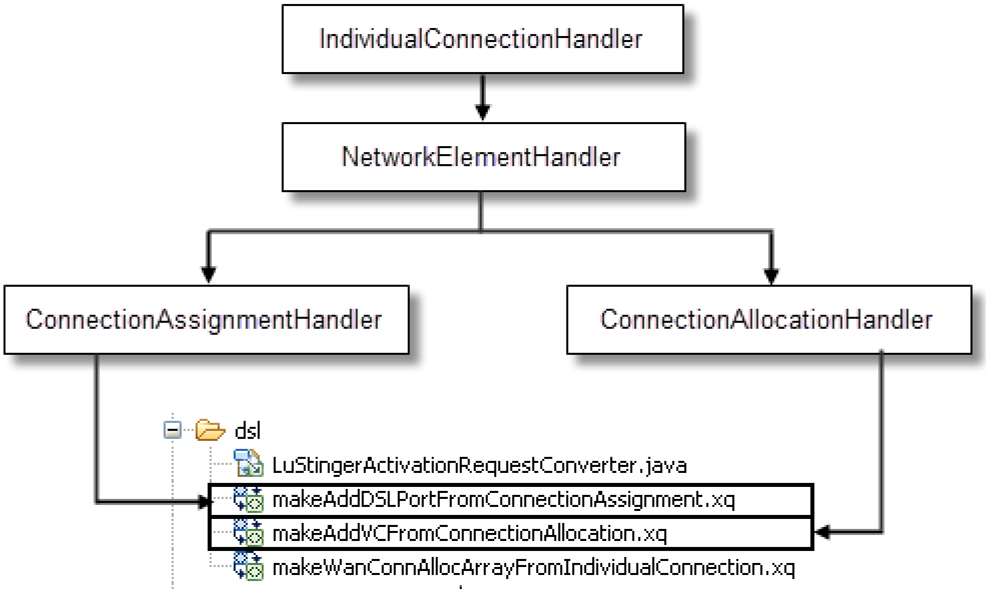
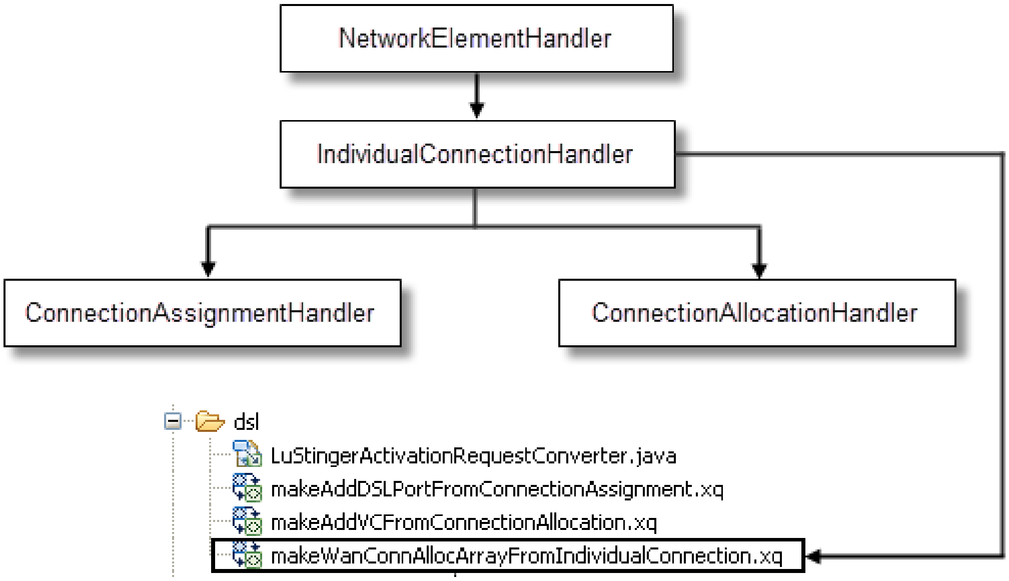

IndividualConnectionHandler
NetworkElementHandler
ConnectionAssignmentHandler
ConnectionAllocationHandler
dsl
LuStingerActivationRequestConverter.java
makeAddDSLPortFromConnectionAssignment.xq
makeAddVCFromConnectionAllocation.xq
makeWanConnAllocArrayFromIndividualConnection.xq

Figure 6–11 shows the XQuery workflows that map CSDL commands for creating an array of connections that represent the DSL service.

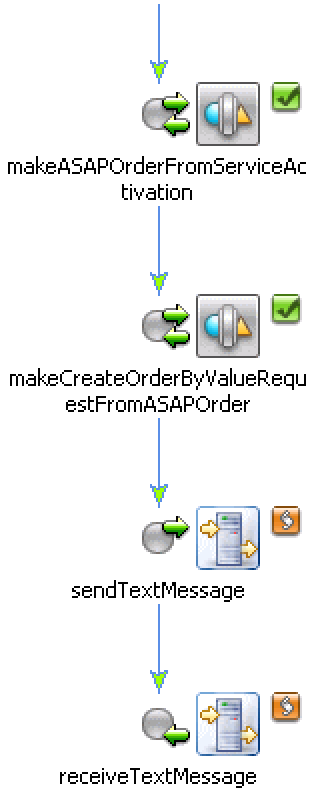
Figure 6–11 Mapping CSDL commands for creating an array of connections that represent the DSL service


NetworkElementHandler
IndividualConnectionHandler
ConnectionAssignmentHandler
ConnectionAllocationHandler
dsl
LuStingerActivationRequestConverter.java
makeAddDSLPortFromConnectionAssignment.xq
makeAddVCFromConnectionAllocation.xq
makeWanConnAllocArrayFromIndividualConnection.xq

Step 4: Building and Sending the ASAP Order

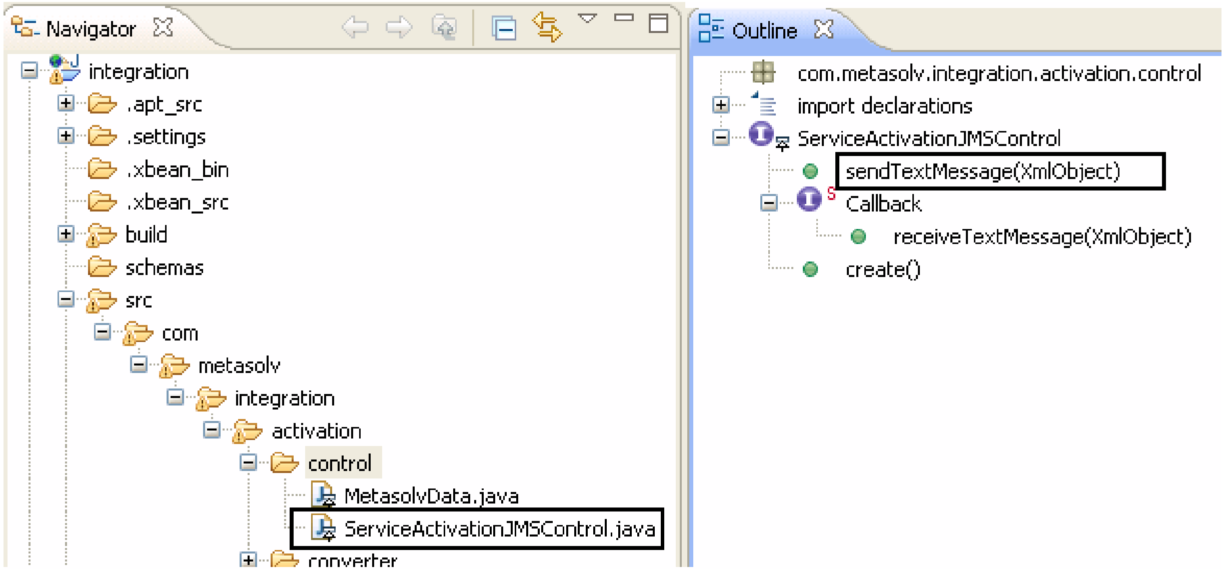
ActivationOrderManager builds the ASAP order from the CSDL commands mapped to MetaSolv Solution data and sends the order to ASAP as shown in Figure 6–12.

Figure 6–12 ActivationOrderManager Group that Creates and Sends the ASAP Order


makeASAPOrderFromServiceActivation
makeCreateOrderByValueRequestFromASAPOrder
sendTextMessage
receiveTextMessage

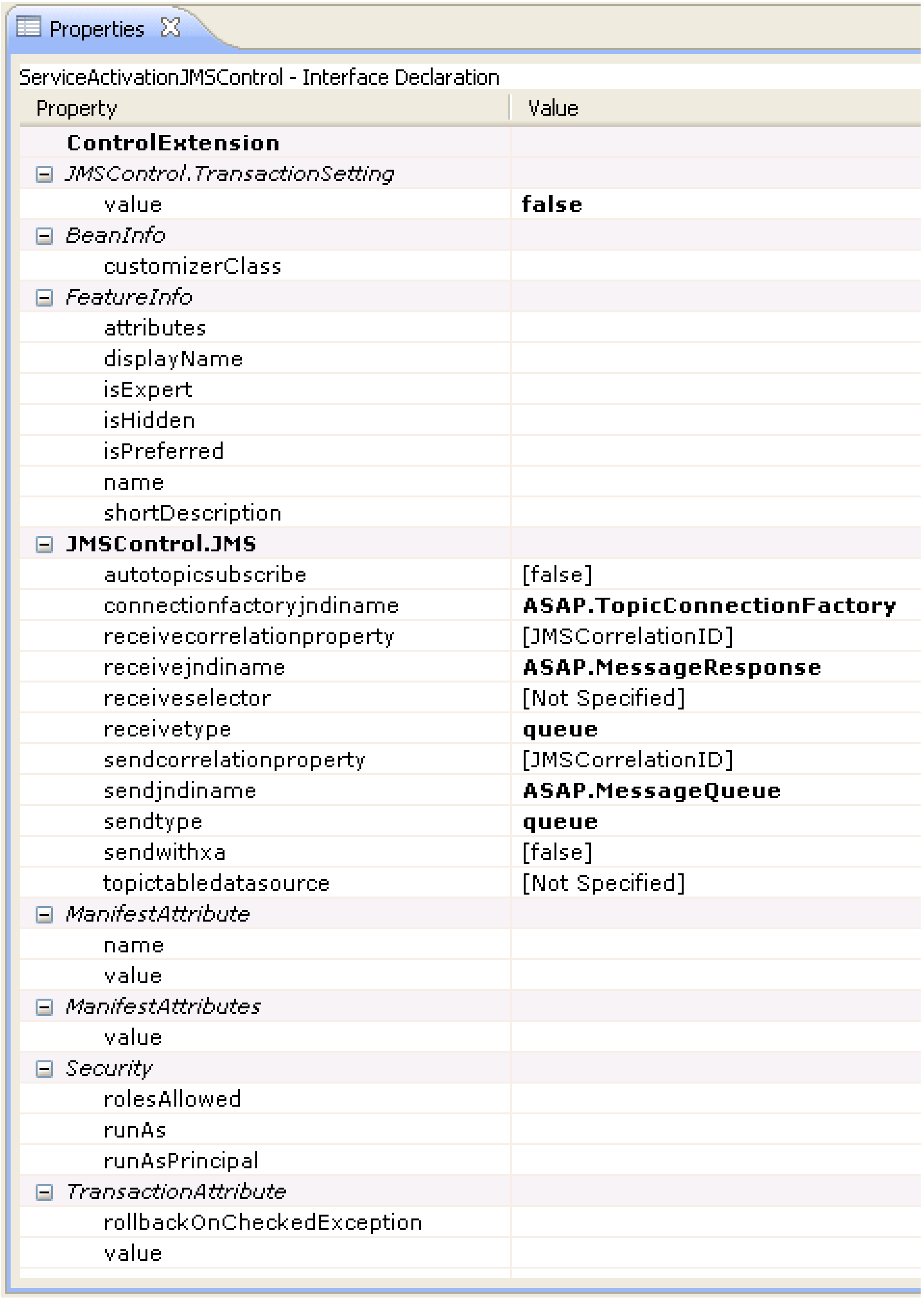
ActivateOrderManager sends the text message that contains the request by calling the sendTextMessage method in the ServiceActivationJMSControl shown in Figure 6–13. The result is a message posted on ASAP.MessageQueue.

Figure 6–13 ServiceActivationJMSControl



ServiceActivationJMSControl is a basic WebLogic Integration control that you can configure. The properties defined for the ServiceActivationJMSControl are the same as the foreign JMS server as shown in Figure 6–14.

Figure 6–14 Properties for ServiceActivationJMSControl



ServiceActivationJMSControl - Interface Declaration

Property	Value
ControlExtension	
⊟ *JMSControl.TransactionSetting*	
value	**false**
⊟ *BeanInfo*	
customizerClass	
⊟ *FeatureInfo*	
attributes	
displayName	
isExpert	
isHidden	
isPreferred	
name	
shortDescription	
⊟ **JMSControl.JMS**	
autotopicsubscribe	[false]
connectionfactoryjndiname	**ASAP.TopicConnectionFactory**
receivecorrelationproperty	[JMSCorrelationID]
receivejndiname	**ASAP.MessageResponse**
receiveselector	[Not Specified]
receivetype	**queue**
sendcorrelationproperty	[JMSCorrelationID]
sendjndiname	**ASAP.MessageQueue**
sendtype	**queue**
sendwithxa	[false]
topictabledatasource	[Not Specified]
⊟ *ManifestAttribute*	
name	
value	
⊟ *ManifestAttributes*	
value	
⊟ *Security*	
rolesAllowed	
runAs	
runAsPrincipal	
⊟ *TransactionAttribute*	
rollbackOnCheckedException	
value	

To see the properties defined for the foreign JMS server for the flow-through package, see "Step 2: Setting Up the Foreign JMS Server".

Step 5: Getting the Service Activation Response

When the request is sent to ASAP, the MSSOutboundEventHandler listens on the specified channel for an event from ASAP. The following ASAP events are possible:

- createOrderResponse

This event indicates that the initial data was accepted and an order was created. The following event types can result from the order being created in ASAP.

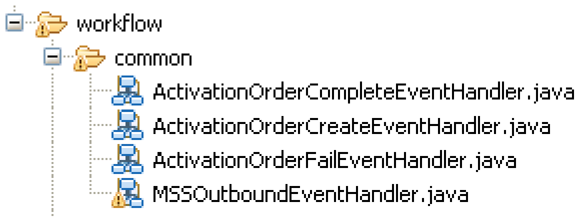
- orderCreateEvent: the order is successfully created.
- orderCompleteEvent: the order is successfully completed and activation occurs.
- orderFailEvent: the order is successfully created but fails on deployment.

- createOrderFail

  This event indicates the initial data was not accepted, and the order was failed without being created in ASAP.

Figure 6–15 shows the Event Handler workflows.

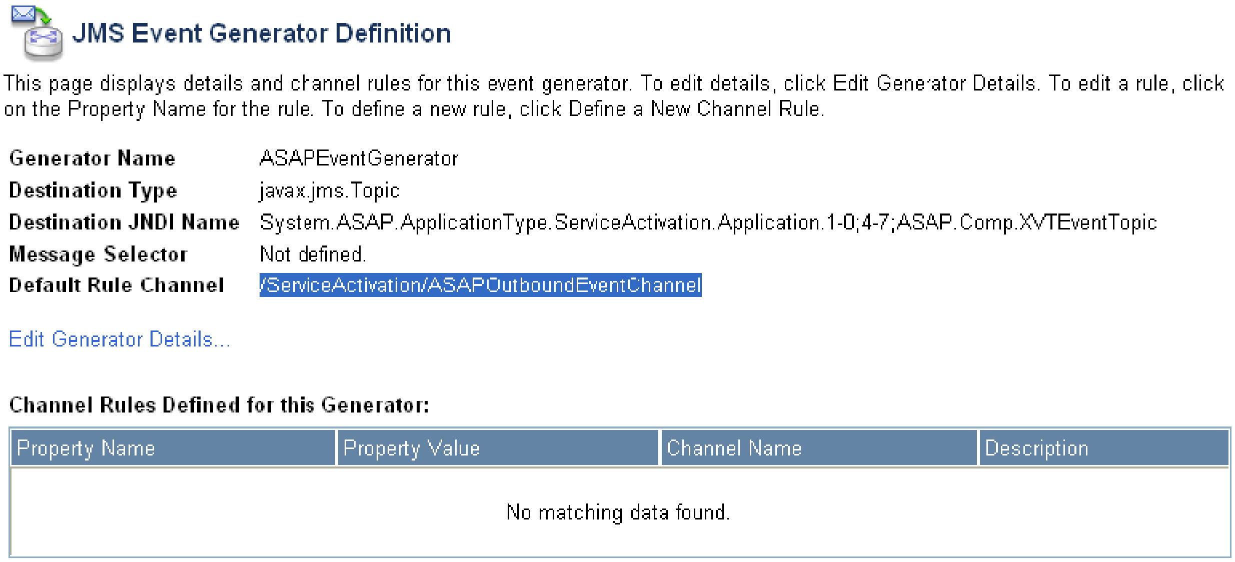
Figure 6–15 Event Handler Workflows



```
workflow
    common
        ActivationOrderCompleteEventHandler.java
        ActivationOrderCreateEventHandler.java
        ActivationOrderFailEventHandler.java
        MSSOutboundEventHandler.java
```

Figure 6–16 shows the ASAPOutboundEventChannel. All event workflows subscribe to this channel. See "Step 4: Creating the ASAP Event Generator" for information on how the channel was set up for the flow-through package.

Figure 6–16 ASAPOutboundEventChannel



JMS Event Generator Definition

This page displays details and channel rules for this event generator. To edit details, click Edit Generator Details. To edit a rule, click on the Property Name for the rule. To define a new rule, click Define a New Channel Rule.

Generator Name	ASAPEventGenerator
Destination Type	javax.jms.Topic
Destination JNDI Name	System.ASAP.ApplicationType.ServiceActivation.Application.1-0;4-7;ASAP.Comp.XVTEventTopic
Message Selector	Not defined.
Default Rule Channel	/ServiceActivation/ASAPOutboundEventChannel

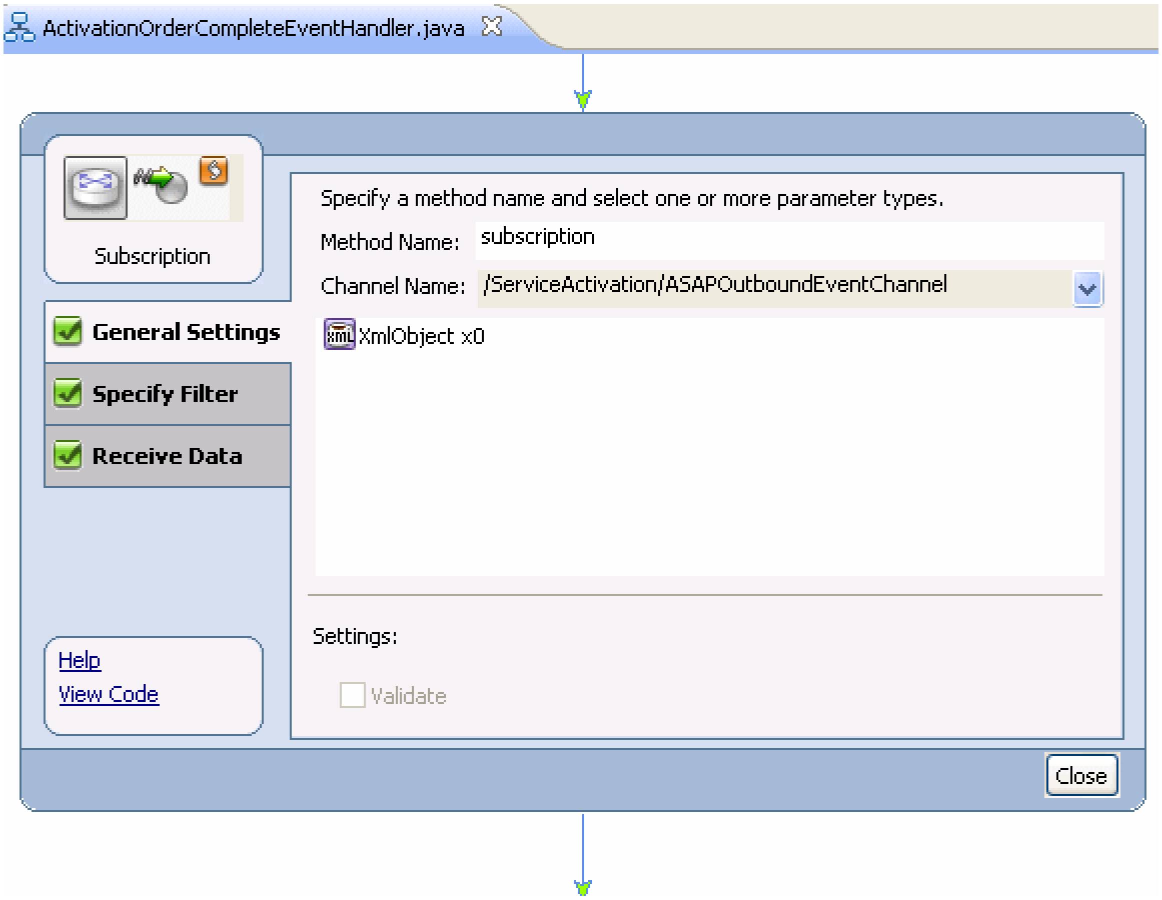
Channel Rules Defined for this Generator:

Property Name	Property Value	Channel Name	Description
No matching data found.			

When ASAP posts an event to a topic, the event generator moves the event to the appropriate channel. Workflows listen on the channel, waiting for an event that is specific to them.

Figure 6–17 shows the properties for a method named Subscription.

Figure 6–17 Subscribing to a Channel



Specify a method name and select one or more parameter types.

Method Name: subscription

Channel Name: /ServiceActivation/ASAPOutboundEventChannel

XmlObject x0

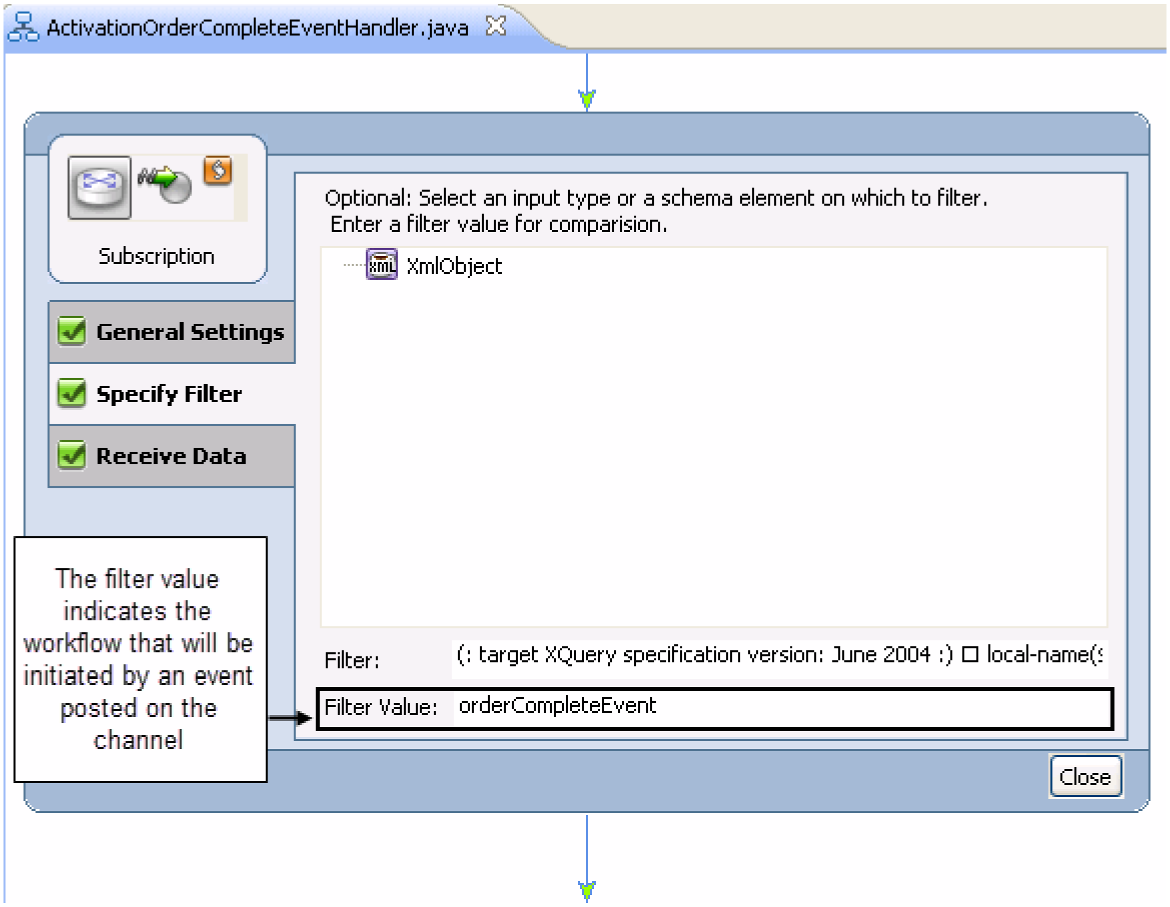
Settings:

□ Validate

A filter determines when an event is intended for a workflow.

Figure 6–18 shows the filter set for the Subscription method shown in Figure 6–17.

Figure 6–18 Filter Value



Optional: Select an input type or a schema element on which to filter.
Enter a filter value for comparision.

Filter: (: target XQuery specification version: June 2004 :) □ local-name(

Filter Value: orderCompleteEvent

The filter value indicates the workflow that will be initiated by an event posted on the channel

7 Troubleshooting

This chapter describes common POTS and DSL problems and the causes of these issues for troubleshooting purposes.

POTS

For more information on the POTS issues presented here, see "Best Practices for Setting Up MetaSolv Solution Data".

Issue: networkElementIdentifier (LineServiceActivationType) is not populated for a line service.

Cause:

- The switch equipment is not associated with a network element that has an element type associated with a technology/role of voice/switch.
- NEID field is not populated on the properties window for the network element.
- The port was changed in design without disconnecting the line and adding a new line. The Activation API does not currently return port information for both the old and the new port on a line service. The workaround when a port must be changed is to disconnect the line and add a new line with the same telephone number.

Issue: lineEquipmentNumber (LineServiceActivationType) is not populated for a line service.

Cause: Node Address is not populated on the equipment containing the assigned port. Node Address is captured on the equipment specification.

Issue: originatingEquipmentType (LineServiceActivationType) is not populated for a line service.

Cause: The value was not entered in the **Mounting Type** field on the equipment specification for the shelf equipment.

DSL

For more information on the DSL issues presented here, see "Best Practices for Setting Up MetaSolv Solution Data".

Issue: networkElementIdentifier (NetworkElementType) is not populated for a network element.

Cause: NEID field is not populated on the properties window for the network element.

Issue: activityCd (ConnectionAssignmentType) is not populated on a connection assignment.

Cause: The connection is not on the order. The activityCd is populated only if the item is the order. A connection is included in the export when it is not on the order if it is between two elements that are included in the path of an ordered virtual connection.

Glossary

The glossary contains a list of terms and their definitions as they relate to Oracle Communications MetaSolv Solution (MSS) documentation:

Activation Framework

The activation framework consists of the MetaSolv Solution XML APIs and the WebLogic Workshop. The environment lets you create workflows for the APIs that accomplish the tasks you want to integrate between MetaSolv Solution and ASAP.

Activation Task

The activation task is a task on a provisioning plan (in MetaSolv Solution) assigned to an order that triggers a gateway event to set the activation process in motion. Although a gateway event can be associated with any task in the provisioning plan, Oracle recommends creating a task called Activate for the purpose of triggering the gateway event. The Activate task should be placed as close to the Due Date task on the provisioning plan as possible.

Atomic Service Description Layer (ASDL)

An ASDL is a lower-level ASAP command that describes atomic operations to be applied to a network element by ASAP. When the Service Activation Request Manager (SARM) translates (CSDL) commands into ASDL commands, the parameters required for each ASDL command are obtained from the originating CSDL command.

Common Service Description Layer (CSDL)

A CSDL is an ASAP command associated with a particular work order. The CSDL command is associated with one or more operations on one or more network elements. Each CSDL command has a configuration record that you can set up. This record contains the following attributes:

- name: the CSDL name should begin with the constant "C" to represent CSDL, followed by a series of tokens separated by the underscore ("_") character. Specifically, `C_<technology>_<software load>_<action>_<service>`. All of the characters contained in a CSDL command must appear in uppercase, for example:

  ```
  C_HLRPS_MSP8_ADD_CFB
  ```

- rollbackOnFailure: a true or false flag that indicates whether rollback is required for a CSDL.
- priority: the priority of a CSDL within the work order.
- failEvent: the event that is triggered if a CSDL fails. The event can either be system or custom.

- completeEvent: the event that is triggered if a CSDL completes successfully. The event can be system or custom.

Custom Attribute

Any parameter required by an ASAP cartridge but not defined in MetaSolv Solution must be defined as a custom attribute. Custom attributes can be defined for network elements, connections, and the relationship between the physical and virtual connection.

For example, the ASAP integration for the Nortel DSLAM does not require equipment information, but it does require a parameter called Profile Name. MetaSolv Solution does not carry this parameter on the equipment specification, but the parameter can be defined as a custom attribute for the DSLAM network element.

DSL (Digital Subscriber Line)

DSL offers broadband service over ordinary copper telephone lines. Most DSL technologies allow the simultaneous transmission of both analog and data signals on the same telephone line.

Gateway Event

A gateway event is triggered when a special task associated with an order is worked in the MetaSolv Solution Work Management module. For example, when the activation task is worked in the Work Management queue, an activation gateway event for the type of service to be activated is triggered and sent to the integration layer’s event processor, which uses the Activation API to request the appropriate data from MetaSolv Solution to send to ASAP.

POTS (Plain Old Telephone Service)

A POTS order consists of an electrical path (line) between a service provider switch and a subscriber. The line is associated with an individual telephone number that can be used for incoming and outgoing telephone calls. The line can also have associated features such as call forwarding or call waiting.

WebLogic Integration Message Broker Channel

A Message Broker channel has similar properties to a Java Message Service (JMS) topic, but is optimized for use with WebLogic Integration processes, controls, and event generators. Within a WebLogic Integration application:

- Message Broker Publish controls are used by process or Web service instances to publish messages to a Message Broker channel.
- Event generators that receive outside events route them as messages to a Message Broker channel.
- Subscription start nodes start processes upon receipt of a message from a Message Broker channel. This constitutes a static subscription to the channel.
- Message Broker Subscription controls are used by process or Web service instances to receive messages from a Message Broker channel. This constitutes a dynamic subscription to the channel.
- Publishers to a Message Broker channel can pass message metadata with the message. This metadata can be received by the subscriber as a parameter.

Channel files define the channels available in a deployed application. To restrict the messages routed to static or dynamic subscribers, XQuery filters can be applied against message metadata (if the metadata is typed XML) or message body (if the body is string or typed XML). All subscribers registered to receive a message on a channel receive the message, subject to any filters they have set up.