CHAPTER 18

Working with ebXML Traffic

This chapter describes how to work with services that exchange ebXML data. It covers these topics:

About ebXML

Electronic Business XML, or ebXML for short, is a set of standards designed to facilitate the exchange of business-process information across systems and organizations. In practical terms, ebXML allows organizations to exchange information—such as purchase orders, invoices, or any other data that needs to be shared—in a conventional, message-oriented fashion.

ebXML message are sent using a protocol named EbXML Message Service (ebMS). ebMS is based on SOAP and SOAP with attachments, and relies on either HTTP or SMTP for transport.

The Cisco ACE XML Gateway supports ebXML services. This means that you can create handlers and service descriptors for external ebXML services.

The ACE XML Gateway lets you validate ebXML traffic in a variety of ways. It can also apply other policy features to ebXML traffic, such as access control, routing rules, and so on.

The ACE XML Gateway can validate these ebXML-specific features:

- **ebXML Headers**. The ACE XML Gateway verifies that an ebXML message has a MessageHeader SOAP Header. This element must be formed according to the ebMS 2.0 specification. Any deviation from the specification causes a Validity exception.
- **ebXML Manifest**. An ebXML message may have a Manifest SOAP Body. If it does, the ACE XML Gateway can make sure that it is formed according to the ebMS 2.0 specification.
- **XML Signature**. If signatures are in the header and your policy requires signature checking, the signatures are checked for validity. The signatures may include references to one or more SOAP attachments with the Content-Id (cid:xxx) URL scheme.

- **Attachment Manifest**. The ACE XML Gateway can ensure that a ebXML manifest, if present, contains a reference for every attachment. If the manifest contains a reference for which no attachment exists, a validity exception is raised. If the payload contains an attachment which is not mentioned in the Manifest, a warning log level event is generated, but the attachment is not discarded (the behavior specified by the ebMS 2.0 specification).
- **Attachment Schema**. The Manifest may provide Schema IDs for each attachment. If these IDs are present and the policy requests it, each attachment will be tested for schema validity.

The ACE XML Gateway supports both synchronous and asynchronous ebXML message exchange. When a consumer submits a SOAP message over HTTP, the message is tested for ebXML validity and submitted over HTTP to an ebXML server. For synchronous messages, the response from the server is tested for ebXML validity and returned to the client. For asynchronous messages, if the message is delivered to the server successfully, a 204 response is returned to the client.

Handling ebXML Service Traffic

This section describes how to configure a ACE XML Gateway policy for ebXML service traffic. The procedure is in two parts:

- Create the service descriptor, which contains settings for the ACE XML Gateway interface to the backend service provider (or destination of a consumer-initiated message).
- Create the service handler, which contains settings for the interface with the service consumer.

Create the ebXML Service Descriptor

A service descriptor controls the interface between the ACE XML Gateway and the message recipient, typically a service container.

To create a service descriptor for an ebXML service:

Step 1 As an `Administrator` user or `Privileged` user with the `Routing` role, click the **Virtual Services** link in the navigation menu.

Step 2 In the routing page, choose **Service Descriptor** from the **Create a New** menu, and click the **Create** button**.**

The **Step 1 of 5: Service Protocol** page appears.

Step 3 From the protocol menu, choose the type of ebXML message traffic the service should accept. Options are:

- **HTTP Post ebXML**—For synchronous ebXML traffic. In this case, the consumer expects a response.
- **Asynchronous HTTP Post ebXML**—For ebXML traffic in which a response is not expected.
- **Asynchronous URL Lookup-Based ebXML**—To have the ACE XML Gateway make a routing decision based on a destination URL lookup table. This mechanism supports both outgoing SMTP and HTTP messages.

  To use this option, in addition to the steps described in this section, you will need to set up a lookup table, as described in "Using ebXML Lookup Tables" section on page 18-189.

Step 4 Click **Continue**.

Step 5 In the **Step 2 of 5: General Information** page, type a unique name for the service descriptor. This name is used only within the policy.

Step 6 If the ebXML service type is based on HTTP (and not SMTP), an additional setting appears in the **General Information** page, the **Server** menu. Choose the destination server by choosing it from the list. If the server has not been specified as a HTTP server resource in the policy, click the **Add a New Server** button and configure the new server settings.

Step 7 Click **Continue**. The **Step 3 of 5: Service Interface** page appears.

Step 8 The service interface configuration varies depending on the type of ebXML message service, as follows:

For HTTP Post ebXML and Asynchronous Post ebXML:

a. In the **Path on Server** field, enter the calling URL for the actual service on the backend server.

b. The SOAP version can be left at its default value, SOAP 1.1, since ebXML is only specified to support SOAP 1.1. The SOAPAction should be "ebXML".

c. Configure settings for authenticating the ACE XML Gateway to the backend server, if required, such as HTTP Basic Authentication, WSS UsernameToken, or SAML Token values. The values configured here are passed with the message.

d. If desired, specify a service time threshold in milliseconds. If the ACE XML Gateway does not receive a response from the backend service within the threshold time, an event indicating this fact is logged. This is useful for quality of service considerations.

For ebXML Messages with URL Lookup:

a. If you have already uploaded the lookup table file into the ACE XML Manager, as described in "Using ebXML Lookup Tables" section on page 18-189, choose it from the menu. Otherwise, click **Upload** and follow the on-screen instructions to load the table.

b. If there are HTTPS destinations in the URI mappings of the lookup table, specify certificate verification and public/private key settings for those destinations, as appropriate.

c. If desired, specify a service time threshold.

Step 9 Click **Continue**.

The **Step 4 of 5: Request Message Specification** page appears.

Step 10 Messages to the backend service are automatically validated to ensure ebXML validity (by checking the validity of ebXML message headers).

In the **Request Message Specification** page, configure additional validation options desired. These cover manifest validation, attachments, and XML signature checking.

The following points apply to these configuration settings:

- A manifest describes the attachments of an ebXML message. When you verify that the attachments match the manifest, the ACE XML Gateway checks whether the message contains the correct number of attachments, whether all ContentIDs are accounted for, and so on.
- The Manifest specifies the location of an XML Schema by URL. If the XML schema with the given URL has previously been uploaded to the ACE XML Manager, from an exactly matching URL, and schema validation has been enabled, the ACE XML Gateway will use this schema. The ACE XML Gateway will not dynamically retrieve schemas from the network.
- If the manifest specifies an attachment that is not part of the message but is instead referenced with a URI, the ACE XML Gateway can reject the message. For security reasons, the ACE XML Gateway never attempts to use non-local resources. Although you can configure your handler or service descriptor to allow URI-based resources, it is recommended that you reject messages having manifest entries that reference objects not in the message.

- Notice that you can specify signature requirements for the ebXML message or the message attachments. The ACE XML Gateway will decrypt PKCS7 (S/MIME) encrypted attachments before validating signatures on them. If the message attachments are encrypted, you need to specify the key for decrypting the message prior to validation.
- If you want to require attachments to be signed, all attachments must be signed. However, a descriptor or handler that does not require attachments to be signed can accept signed attachments without error.

Step 11 For HTTP Post messages (which are synchronous), the Response Message Specification page appears. If appropriate, configure validation requirements for the message returned by the server. As for the outgoing request, you can configure manifest validation, attachment validation, and XML signature checking for the response body.

Step 12 Click **Continue** to finish the service descriptor.

You can now create a handler for the service, as described next.

Create the ebXML Handler

To create a handler for an ebXML service:

Step 1 As an Administrator or Privileged user with the operations role in the web console, open the **Virtual Services** browser.

Step 2 Click the **Create** button with **Handler** selected in the **Create a new** menu.

Step 3 From the protocol list, choose the type of ebXML message traffic appropriate for the service. Options are:

- **HTTP Post ebXML**. For synchronous ebXML traffic. In this case, a response with a body is expected from the backend ebXML service.
- **Asynchronous HTTP Post ebXML**. For ebXML traffic in which a response is not expected from the backend service.
- **SMTP ebXML**. For asynchronous ebXML traffic passed as SMTP-based email. To use this option, you need to enable the SMTP MTA listener on the ACE XML Gateway, as described in "Enabling the Gateway to Receive SMTP Traffic" section on page 18-191.

Step 4 Click **Continue**. The handler configuration page appears.

Step 5 Enter a name for the handler and choose a handler group. The name is only for internal use. You can set the other handler options on this page as desired. When developing the policy, it is recommended that you enable message body logging.

Step 6 Click **Continue**. The **Consumer Interface** page appears.

Step 7 Configure the consumer interface for the service as follows:

- For ebXML over HTTP handlers, choose the port and exposed local path. This is the path that, along with the hostname, makes up the URI the consumer uses to invoke the service.
- For ebXML over SMTP handlers, specify the email address to which consumer-initiated messages for this service are to be addressed. The email address should be in standard form: `<name>@<domain>.<ext>`. The ACE XML Gateway MTA will accept email message addressed to the domain specified in this address.

You can leave the other settings on the handler page at their default values. Briefly described, these settings are:

- **SOAP version** should be SOAP 1.1, since ebXML only specifies support this SOAP version.
- **SOAPAction** is ignored as specified by ebXML. However, the value "ebXML" (with quotes) is encouraged.
- For ebXML services, the **SOAP Method** and **SOAP Method Namespace** settings in the handler are unused by the Gateway.

Step 8 Click **Continue**.

Step 9 By default, incoming messages for the handler will be validated as ebXML, and dropped if they are not. In the **Request Message Specification** page, you can configure additional validation options, including manifest validation, attachment validation, and XML signature checking. For more information, see Step 5 in "Create the ebXML Service Descriptor" section on page 18-186.

Step 10 Click **Continue**.

Step 11 Configure the **Response Message Specification** page as appropriate. The settings on the page varies as follows:

- For asynchronous message handlers, no configurable properties exist, since a response message is not expected.
- For synchronous message types (HTTP POST ebXML), the **Response Message Specification** page appears. You can enable a *mixed* mode of synchronous-or-asynchronous traffic over HTTP by checking the option **Allow asynchronous delivery; this response can be an empty (HTTP 204) message** on a synchronous handler-service pair.
- The message can be an empty message, such as a 204 response, or a response with a body. The ACE XML Gateway can verify the response body just as it does the incoming ebXML message. That is, you can configure manifest validation, attachment validation, and XML signature checking for the response body.

Step 12 Click **Continue**.

Step 13 Click **Add a Route Now** to specify a service descriptor to which this handler will route messages. (If the service descriptor does not yet exist, you can finish without adding a route and add it later.)

Step 14 Select the name of the service descriptor you created from the menu. Like for other types of handlers, you can have multiple routes with ebXML handlers, which are chosen by matching criteria you specify in the route configuration page.

Step 15 Click **Save Changes** and deploy the policy to have the settings enforced by the ACE XML Gateway.

Using ebXML Lookup Tables

A service descriptor is the ACE XML Gateway policy object that represents the backend settings for a service virtualized by the ACE XML Gateway. For most types of services, you can specify a destination service by simply configuring the server IP address or hostname and the target service URL.

For ebXML services, you can similarly define a message recipient. However, ebXML service descriptors provide an additional routing mechanism, called URL lookup.

With URL lookup, the ACE XML Gateway determines the destination of a message dynamically, using a table made up of key-destination values. The ACE XML Gateway arrives at a key by combining elements of the ebXML message header. It then checks the lookup table for a matching key. If the key is found, the message is routed to the URL associated with the key.

Note URL lookup works with asynchronous ebXML service message types only.

To set up the URL lookup routing, enter the values of the key and the corresponding destination URL in a text file. There are no requirements for the name of the file. It will need to be accessible by file system or URL to the computer from which you access the ACE XML Manager web console.

The destination URL can be an HTTP, HTTPS, or email address. For HTTPS destinations, an SSL client keypair and server certificate can be specified, as described in section "Create the ebXML Service Descriptor" section on page 18-186.

Lookup Key

The fields from the ebXML message header that compose the URI matching key are:

- `Service`
- `Action`
- `CPAId`
- `To-PartyId`

To construct the key, provide field values next to a corresponding URI in the lookup table file. The field values should be entered in the order shown and comma separated (no space). The URI should be on the same line, separated from the key by a single tab character.

For example, consider the structure of the message shown in Example 18-1.

Example 18-1 Sample message

```
<SOAP-ENV:Header>
 <eb:MessageHeader>
  <eb:From>
    <eb:PartyId context="uriReference">
      urn:company.com
    </eb:PartyId>
  </eb:From>
  <eb:To>
    <eb:PartyId context="uriReference">urn:acme.com</eb:PartyId>
  </eb:To>
  <eb:CPAId>cpa1</eb:CPAId>
  <eb:ConversationId>1234567.1953.react.nene</eb:ConversationId>
  <eb:Service>OrderProc</eb:Service>
  <eb:Action>ProcNewOrder</eb:Action>
  <eb:MessageData>
  ...
</eb:MessageData>
...
 </eb:MessageHeader>
...
</SOAP-ENV:Header>
```

Notice the values of the key fields: Service, Action, CPAId and To-PartyId. To have any message with the value of the fields shown in the sample directed to a particular address, you would construct the mapping as follows:

```
OrderProc,ProcNewOrder,cpa1,acme.com   mailto:jw@example.com
```

As shown, the field values are comma-separated. A single tab must be used to separate the last field from the destination address.

In the example, the URL is a mail-to address. http or https destinations are also supported. To use an http or https address as the destination, simply include the protocol specification in the URL, as in Example 18-2.

Example 18-2 Sample URL Look-up items

```
OrdProc,ProcOrder,cpa1,acme.com   mailto:user@sample.com
OrdProc,ProcOrder,cpa2,acme.com   http://sample.com/myservice
```

There are several additional points to note about lookup mappings:

- A mapping in the lookup table must specify all fields: `Service`, `Action`, `CPAId`, and `To-PartyId`.
- For a message to match the key, each key field must match the corresponding value of the message header. If not, the message is dropped and an exception is raised.
- A message can be matched to only one destination address. If two entries exist with the same key, the last entry is used.
- Comments can be added to the file by typing a hash symbol (#) at the start of the line.

Uploading a Lookup Table

After creating the file, you can use its mappings in an ACE XML Gateway policy by uploading the table in the web console as follows:

Step 1 Click the **ebXML Lookup Tables** link from the navigation menu.

Step 2 Provide a name for the table. This name is used internally to the web console only.

Step 3 Depending on the location of the file, either click **Browse** and identify the file in the file chooser dialog or specify the file URL.

Step 4 Click **Upload**.

When finished, the table is available as a menu option in the configuration settings that appear when creating ebXML service descriptors for SMTP-, lookup-based services.

Enabling the Gateway to Receive SMTP Traffic

ebXML messages can be transported over HTTP or SMTP. If your services use SMTP, you must open the Mail Transport Agent (MTA) listening port of each Gateway in your deployment. Opening the port enables the ACE XML Gateway to receive incoming ebXML traffic as email messages.

Note These instructions describe how to set up the MTA included with the ACE XML Gateway. You can also configure an external MTA, as describe in section "Configuring an External MTA" section on page 18-193

By default, the incoming email listening port is closed on ACE XML Gateways. (Note that a `sendmail` process is active by default, however, to enable the transmission of email notifications of log events to system administrators.)

To open the MTA listening port, you will need to access the shell interface on the ACE XML Gateway appliance with an administrator account.

To enable the MTA listening port:

Step 1 From either a secure SSH client or a console connection to the appliance, log into the shell interface of the ACE XML Gateway appliance as `root` user. The Main Menu appears.

Note For more information on accessing the shell, see "Logging Into the Shell Interface" in the *Cisco ACE XML Gateway Administration Guide*.

Step 2 Choose the **Advanced Options** item from the **Main Menu**. The shell interface displays the Advanced Options screen.

Step 3 Choose the `MTA Configuration` item from the **Advanced Options** page. The shell interface displays the `MTA Menu`.

Step 4 Choose the `Open MTA Port` menu item. Choosing this item toggles the status of the listening port. The name of the menu item changes to `Close MTA Port`.

The appliance is now configured to listen on port 25 as a standard SMTP server.

Note If the first item in the MTA Menu screen is Close MTA Port rather than Open MTA Port, the MTA port is already open.

Step 5 After the listening port is opened, it's possible that the ACE XML Gateway MTA will now receive email in its postmaster mailbox. The postmaster address is a standard administrative address for MTA's (as required by the SMTP protocol). It does not affect incoming or outgoing Gateway traffic.

If desired, you can modify the address so that mail to the postmaster is sent to another location, or keep the default, in which case the postmaster mailbox is the root user's mailbox on the ACE XML Gateway. To do so, from the MTA Menu, choose the second item, Configure postmaster address, and follow the on-screen instructions to configure the postmaster mailbox.

Step 6 When finished, return to the **Advanced Options** menu by choosing the third option from the MTA Menu and close the console session.

The ACE XML Gateway is now configured to receive service traffic in the form of email. Repeat this procedure for each ACE XML Gateway in your deployment.

Configuring an External MTA

As described in "Enabling the Gateway to Receive SMTP Traffic" section on page 18-191, each Gateway includes an internal Mail Transport Agent (MTA) for handling incoming and outgoing email traffic to the ACE XML Gateway.

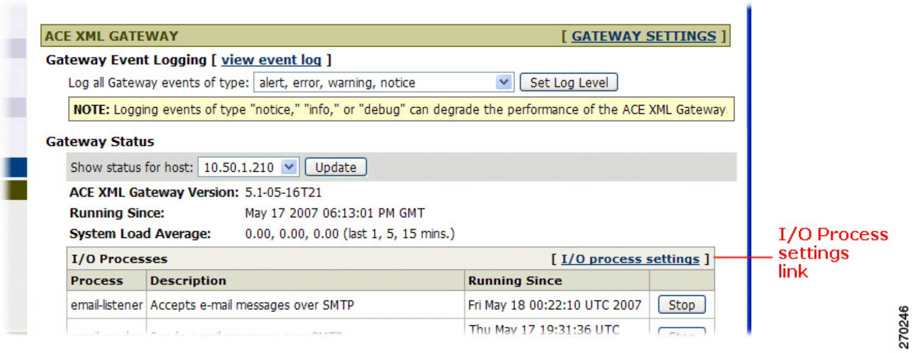
Alternatively, you can set up an external MTA for the ACE XML Gateway to use, such as a Microsoft® Exchange server.

To set up an external MTA:

Step 1 As an `Administrator` user or as a `Privileged` user with the `Operations` role in the console, click **System Management** in the **Administration** group of the navigation menu.

Step 2 Under the ACE XML Gateway settings, click the **I/O process advanced settings** link, which appears next to the I/O Processes.

Figure 18-1 Accessing external MTA set up



ACE XML GATEWAY [GATEWAY SETTINGS]

Gateway Event Logging [view event log]

Log all Gateway events of type: alert, error, warning, notice | Set Log Level

NOTE: Logging events of type "notice," "info," or "debug" can degrade the performance of the ACE XML Gateway.

Gateway Status

Show status for host: 10.50.1.210 | Update

ACE XML Gateway Version:	5.1-05-16T21
Running Since:	May 17 2007 06:13:01 PM GMT
System Load Average:	0.00, 0.00, 0.00 (last 1, 5, 15 mins.)

I/O Processes [I/O process settings]

Process	Description	Running Since	
email-listener	Accepts e-mail messages over SMTP	Fri May 18 00:22:10 UTC 2007	Stop
		Thu May 17 19:31:36 UTC	

Step 3 In the **I/O Process Settings** page, enter the hostname of the MTA in your environment in the SMTP server field, as well as its SMTP listening port (conventionally port 25).

Step 4 The VM (virtual memory) size limit applies to the e-mail sender component of the ACE XML Gateway. This setting protects the sender from becoming consumed by an email attack. Depending on your application, you may want to increase, decrease, or accept the default. The default size 512000 should be sufficient in most cases.

Step 5 Click **Save Changes**.