webMethods

webMethods Ariba Supplier OnRamp

User's Guide

VERSION 2.0

webMethods, Inc.
3930 Pender Drive
Fairfax, VA 22030
USA
703.460.2500
http://www.webmethods.com

Document ID: ADAPTER-ARIBASUPPLIER-UG-20-20041203

Contents

About This Guide

This guide supplements the *webMethods Ariba Supplier OnRamp User's Guide Version 1.5.1* by providing information about the webMethods Ariba Supplier OnRamp 2.0. This release implements the latest cXML 1.2 specification announced by Ariba, Inc. The new cXML standard introduces two new document types, MasterAgreementRequest and InvoiceDetailRequest. The cXML 1.2 standard also supports new transaction authentication by using digital signatures.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Italic	Identifies variable information that you must supply or change based on your specific situation or environment. Identifies terms the first time they are defined in text. Also identifies service input and output variables.
Narrow font	Identifies storage locations for services on the webMethods Integration Server using the convention *folder.subfolder:service*.
`Typewriter font`	Identifies characters and values that you must type exactly or messages that the system displays on the console.
UPPERCASE	Identifies keyboard keys. Keys that you must press simultaneously are joined with the "+" symbol.
\	Directory paths use the "\" directory delimiter unless the subject is UNIX-specific.
[ ]	Optional keywords or values are enclosed in [ ]. Do not type the [ ] symbols in your own code.

Additional Information

The webMethods Advantage Web site at http://advantage.webmethods.com provides you with important sources of information about webMethods components:

- **Troubleshooting Information.** webMethods provides troubleshooting information for many webMethods components in the webMethods Knowledge Base.
- **Documentation Feedback.** To provide documentation feedback to webMethods, go to the Documentation Feedback Form on the webMethods Bookshelf.
- **Additional Documentation.** All webMethods documentation is available on the webMethods Bookshelf.

webMethods.

CHAPTER 1

Using cXML Document Types

Overview

This section describes the new cXML 1.2 document types, MasterAgreementRequest and InvoiceDetailRequest. The configuration and processing of the MasterAgreementRequest document is similar to the OrderRequest document. For details, see "Creating MasterAgreementRequest Handlers" and "Creating MasterAgreementRequest Handler Services" on page 10. The InvoiceDetailRequest document type does not require configuration of the handler but you must configure the URL that receives the document. For instructions, see "Configuring the URL for InvoiceDetailRequest" on page 9.

MasterAgreementRequest Documents

Master Agreements enable buyers to establish a commitment for goods and services with suppliers. They represent a common mechanism for managing supplier and budget commitments, and they enable buyers to negotiate better discounts by basing the discounts on future purchases, while enabling suppliers to more accurately forecast demand.

The MasterAgreementRequest document defines the Master Agreement created by the buying organization. It specifies beginning and end dates, and the committed maximum and minimum values of the agreement. It also lists maximum and minimum values and quantities for individual items. The MasterAgreementRequest document consists of MasterAgreementRequestHeader and AgreementItemOut. The MasterAgreement-RequestHeader contains the Master Agreement common to all contained items. The AgreementItemOut specifies the requirements of a particular line item that is part of the Master Agreement contract. For details about the attributes and elements, see the *cXML User's Guide Version 1.2*.

The Ariba Supplier OnRamp 2.0 provides sample services to create a MasterAgreement-Request test message and submit it to the configured URL. To test this transaction, see "Using the Test Module" on page 18.

InvoiceDetailRequest Documents

The cXML InvoiceDetail transaction enables suppliers to send invoices to buying organizations or marketplaces. This transaction supports invoice details for a wide variety of business scenarios, including standard invoices, credit memos, debit memos, and receipts. Suppliers use cXML invoices to bill buying organizations or marketplaces for provided products or services.

The supplier sends a cXML document of type, InvoiceDetailRequest, to let the buyer know about the list of products that are being delivered against an order placed by the buyer. The document is routed to the buyer through the Ariba Network. The OnRamp allows the supplier to create InvoiceDetailRequest and posts it to the specified URL in the Ariba Network to be routed to the buyer. The supplier can make use of Trading Networks to post the document to the Ariba Network. Trading Networks can be used in the case of InvoiceDetail to make use of its logging and event mechanism. Ariba Network, after

receiving the InvoiceDetailRequest, responds with a generic response indicating whether the request is genuine/authenticated.

The Ariba Supplier OnRamp 2.0 provides sample services to send a request of type InvoiceDetailRequest to the configured URL. The sample services are:

- wm.ariba.supplier.test:sendCXMLInvoiceDetailRequest - Sends the InvoiceDetailRequest document to the configured URL and writes logging information. This service should be used for transactions without Trading Networks.
- wm.ariba.supplier.test:sendCXMLInvoiceTN - Invoked from the processing rule created by the Ariba Supplier OnRamp. The Ariba Supplier OnRamp sends the InvoiceDetailRequest document to Trading Networks where the processing rule is executed. When the processing rule is executed, the InvoiceDetailRequest document is sent to the configured URL and the corresponding information is logged. This service should be used for transactions with Trading Networks.

To test this transaction, see "Using the Test Module" on page 18.

Configuring the URL for InvoiceDetailRequest

Before using the Ariba Supplier OnRamp to send the InvoiceDetailRequest document, you must configure the URL that receives the document.

To configure the URL for InvoiceDetailRequest

1. In the Integration Server Administrator, click **Ariba OnRamp** under **Adapters**.
2. On the **Config** page, go to the **Invoice Detail Request Config** section.
3. Enter the following parameters for the URL of the buyer or host that expects the InvoiceDetailRequest document.

For this parameter...	Specify...
protocol	http or https.
Hostname	Hostname of the buyer or other host.
Port	Port number for the specified hostname.
Path	Path to the service expecting the InvoiceDetailRequest document. For example, service/transaction/cxml.asp.

Using cXML Document Types without Trading Networks

This section describes how to configure the Ariba Supplier OnRamp to support the MasterAgreementRequest document types in a supplier's site that does not use Trading Networks. For complete instructions on the configuration of document handlers, see the *webMethods Ariba Supplier OnRamp User's Guide Version 1.5.1.*

To use the InvoiceDetailRequest document type without Trading Networks, see "InvoiceDetailRequest Documents" on page 8.

Creating MasterAgreementRequest Handlers

A MasterAgreementRequest handler is a service that you create and implement to receive and process new MasterAgreementRequests, updates, and deletions (cancellations). The MasterAgreementRequest handler is responsible for mapping the cXML Master Agreement to MasterAgreementRequests/cancellations for the underlying system. The Ariba Supplier OnRamp's cXML processing mechanism, built into the wm.b2b.cxml:receiveCXML service, invokes the handler when buyers send MasterAgreementRequest messages.

To create and implement an MasterAgreementRequest handler, you perform the following tasks:

1. Create an appropriate service for the handler, using the *Config* module.
2. Implement the handler's functionality using the Ariba Supplier OnRamp API services.

Creating MasterAgreementRequest Handler Services

A MasterAgreeementRequest handler is invoked with a Master Agreement record argument that represents the actual received message. The MasterAgreeementRequest handler is also invoked with an operation argument that represents the Master Agreement operation that is being requested (new, update, delete).

The Master Agreement must return a status in its *serviceError* output parameter. If an error was encountered while processing the MasterAgreeementRequest, the *serviceError* must be set to indicate the error. If no error was set, the Ariba Supplier OnRamp returns a response with the status code of 200 that indicates that the MasterAgreementRequest was successfully received, and that the request is being processed.

Using cXML Document Types with Trading Networks

This section describes how to configure the Ariba Supplier OnRamp to support the MasterAgreementRequest and InvoiceDetailRequest documents in a supplier's site that uses Trading Networks. For complete instructions on the configuration of processing rules and initialization of documents, see the *webMethods Ariba Supplier OnRamp User's Guide Version 1.5.1.*

Trading Networks must contain document attributes that are used to describe parts of the cXML Master Agreement and Invoice Detail requests. Trading Networks must also contain document type definitions describing the cXML requests. You use the Ariba Supplier OnRamp's *Config* module to add these resources to Trading Networks.

The Ariba Supplier OnRamp will also install cXML document attributes in Trading Networks (fields in the cXML requests that are determined to be important).

Note: You can modify documents by adding or removing attributes. You also can modify the attributes and related queries to match your document. For instructions, see the *webMethods Trading Networks User's Guide*.

MasterAgreementRequest Document Types and Document Attributes

Document Types and Attributes that will be installed by the Ariba Supplier OnRamp include the following:

Document Type	Description
cXML_MasterAgreementRequest_new	cXML MasterAgreementRequest document with operation new.
cXML_MasterAgreementRequest_update	cXML MasterAgreementRequest document with operation update.
cXML_MasterAgreementRequest_delete	cXML MasterAgreementRequest document with operation delete.

Document Attribute	Description
operation	cXML Request operation type. For *OrderRequest*: new, update, or delete. For *PunchOutSetupRequest*: create, inspect, or edit.
language	The language used for the cXML request. This field may be set, depending on whether the cXML request has its *xml:lang* attribute set.
cXML_SenderIDDomain	cXML Request Header *From* credential domain.
cXML_ReceiverIDDomain	cXML Request Header *To* credential domain.
cXML_RoutingAgentIDDomain	cXML Request Header *Sender* credential domain.
cXML_RoutingAgentID	cXML Request Header *Sender* identity.
cXML_MarketplaceSenderDomain	XML Request Header *From* credential domain, for the credential with type *marketplace*.
cXML_MarketplaceSenderID	cXML Request Header *From* credential identity, for the credential with type *marketplace*.
cXML_AgreementID	The procurement system agreement ID for this request.
cXML_AgreementType	Whether the agreement refers to a value or quantity.
cXML_AgreementDate	The date and time that the agreement request was created. This is different from the effective and expiration date of the agreement.
cXML_EffectiveDate	The date that the agreement is available for ordering or releases.
cXML_ExpirationDate	The date the agreement is no longer available.

In addition, the Ariba Supplier OnRamp uses the following attributes that are predefined in Trading Networks to represent cXML field values:

Document Attribute	Description
senderID	Represents the cXML Header *From* credential identity.
receiverID	Represents the cXML Header *To* credential identity.
documentID	The cXML *payloadID*.

InvoiceDetailRequest Document Types and Document Attributes

Document Types and Attributes that will be installed by the Ariba Supplier OnRamp include the following:

Document Type	Description
cXML_InvoiceDetailRequest_new	cXML InvoiceDetailRequest document with operation new.
cXML_InvoiceDetailRequest_delete	cXML InvoiceDetailRequest document with operation delete.
cXML_InvoiceID	A supplier-generated identifier for the Invoice. Identical to the Invoice Number that appears at the top of a physical Invoice.
cXML_InvoicePurpose	Purpose of the invoice. ■ *standard* - (Default) A standard billing statement from the supplier to the buying organization. ■ *creditMemo* - A credit memo for issuing credit to the buying organization. isHeaderInvoice must be yes. Also, the elementInvoiceDetailSummary/ DueAmount must be a negative amount. ■ *debitMemo* - A debit memo for billing a balance owed by the buying organization. isHeaderInvoice must be yes. Also, the element InvoiceDetailSummary/ DueAmount must be a positive amount.

Document Attribute	Description
cXML_InvoiceDate	Date and time that the Invoice was created (should be earlier than the cXML timestamp).
cXML_IsInformationOnly	Indicates whether the buying organization needs to take action. ■ *yes* - Invoice is for the buying organization's information only (no action needs to be taken by the buying organization). ■ *Not specified* - (Default) Invoice is functional. The buying organization needs to take action upon receiving this document (submit payment or accept credit).

Initializing the cXML Document Attributes and Document Types

To initialize the cXML document attributes and types in Trading Networks

1. Start the Server Administrator. See the *webMethods Integration Server Administrator's Guide* for instructions.
2. In the Ariba Supplier menu in the navigation area, select **Config**. The Ariba Supplier OnRamp Configuration screen is displayed.
3. Click the **Use Trading Networks Support** check box and click the **Update** button next to it. The page displays the fields that support Trading Networks.
4. Click the **Initialize** button next to the **Document Attributes and Type Definitions** label in the Trading Networks Configuration section of the page. This adds all the cXML document attributes and type definitions into Trading Networks. The OnRamp warns you that if you proceed, all cXML attributes and document type definitions that exist in Trading Networks will be overwritten with these default values.
5. Click **OK** in response to the warning pop-up.

Creating a Processing Rule

You can define a processing rule for each operation of a MasterAgreementRequest and InvoiceDetailRequest document type, or you create one rule to apply to all operations. The purpose of a processing rule is to invoke custom code when MasterAgreement-Request and InvoiceDetailRequest operations are executed. To create a processing rule, you:

- Use the *Config* module to specify the operation and the names of the rule and its associated service and package.
- Use Trading Networks to specify conditional processing criteria for the rule and to move the rule above Default Rule.
- Use the Developer to define the Integration Server service that the rule invokes.

To create a processing rule

1. Start the Server Administrator. See the *webMethods Integration Server Administrator's Guide* for instructions.
2. In the Ariba Supplier menu in the navigation area, select **Config**. The Ariba Supplier OnRamp Configuration screen is displayed.
3. Click the **Use Trading Networks Support** check box and click the **Update** button next to it. The page displays the fields that support Trading Networks.
4. Click the **Create** button next to the **MasterAgreementRequest** or **InvoiceDetailRequest Processing Rule** label. The **Create Trading Networks Processing Rule** for the selected document type is displayed.
5. Complete the following fields and click the **Submit** button:

Field	Value
Operation	Select **new**, **update**, or **delete**. Select **any** if you want the rule to apply to all operations.
Rule Name	The name of the rule.
Rule Description	Optional. A text description of the rule.
Service Name	The name of the service that will be associated with the rule. Type the name in the following format: folderName:serviceName
Package Name	The name of the package in which the service will be located.

6. View the rule from the Trading Networks Console to confirm that the processing rule was created and to verify that the rule invokes the service you specified.

7 Move the processing rule above **Default Rule** (in Trading Networks 6.1, new processing rules are added to the bottom of the list) and modify the processing rule to create more restrictive conditions when the rule is executed. For more information, see the *webMethods Trading Networks User's Guide*.

8 Using the Developer, display the service stub you specified and implement your MasterAgreementRequest or InvoiceDetailRequest processing inside the sequence labeled **ADD YOUR CODE HERE**.

webMethods.

CHAPTER 2

Using the Log and Test Modules

Overview

This section describes the new message categories for logging. It also describes the testing functions for the new MasterAgreementRequest and InvoiceDetailRequest document types.

Using the Logging Facility

The Ariba Supplier OnRamp 2.0 has new log entries that can be configured to log messages related to the MasterAgreementRequest and InvoiceDetailRequest processes.

Message Categories and Log Entry Types

The following new categories of log messages are added:

- MasterAgreementRequests - Messages related to the MasterAgreementRequest/ Response transaction.
- InvoiceDetailRequests - Messages related to the InvoiceDetailRequest/Response transaction.

For details about how to enable logging and about other log entry types, see *webMethods Ariba Supplier OnRamp User's Guide Version 1.5.1.*

Using the Test Module

The Ariba Supplier provides a Test module that enables you to test your Master Agreement implementation and InvoiceDetailRequest. You access the Test module from the **Test** menu selection in the Server Administrator.

You can send the test message, MasterAgreementRequest, to the receiveCXML service as if it was sent from Ariba Network. You can view the status of the transaction using the Logs module. You also can send the InvoiceDetailRequest test message to an URL specified on *Config* module as if it was buyer's URL.

The test messages can reside in files, or you can create messages using the Test module.

Testing a MasterAgreementRequest Message

You can test the MasterAgreementRequestHandler by creating and sending simple MasterAgreementRequest messages.

To create and send MasterAgreementRequest test message

1. Start the Server Administrator. See the *webMethods Integration Server Administrator's Guide* for instructions.
2. In the Ariba Supplier menu in the navigation area, select **Test**. The Test module is displayed.
3. Select the test type **MasterAgreementRequest**. The Test cXML Request Processing screen is displayed.
4. Set the cXML **From**, **To**, and **Sender** credentials by clicking the **Set Credentials** button and completing the following fields to specify the credentials of the sender and receiver of the document. (To send empty values, click the check boxes next to the fields.)

Field	Value
Identity	An ID string representing the entity.
Domain	Select one of the following values: ■ **DUNS**. If the ID is a DUNS number. ■ **NetworkId**. If the ID is an Ariba Network ID. ■ **AribaNetworkUserId**. If the ID is an Ariba Network user ID.
Shared Secret	For the Sender's Credential, this must be the supplier's Shared Secret, which is configured on both the Ariba CSN Network and on the Ariba Supplier OnRamp's *cXML*.
Digital Signature	Optional. For Sender's Credential only.
Signature Type	Optional. For Sender's Credential only. The Digital Signature Type.
Signature Encoding	Optional. For Sender's Credential only. The Digital Signature Encoding.
User Agent	For Sender's Credential only. The sending system's User Agent Identity.

5 Set MasterAgreementRequest Header fields. Specify the following fields:

Field	Value
Operation	Specifies the type of the agreement request. Select one of the following: ■ **new** (Default) ■ **update** ■ **delete** - Used to cancel an existing agreement. The delete request should be an exact replica of the original request.
Agreement Date	The date and time the agreement request was created. This is different from the effective and expiration date of the agreement.
Expiration Date	Specifies the date the agreement is no longer available.
Agreement ID	The procurement system agreementID for this request.
Agreement Type	Specifies whether the agreement refers to a value or quantity.
Effective Date	Specifies the date the agreement is available for ordering or releases.
Parent Agreement Payload ID	Optional. PayloadID for the corresponding parent document from which this agreement is derived.

6 To add detail records, click the **Add Agreement Item** and perform the following steps:

a Wait for the browser page to fully load. The page is a form containing fields representing the cXML *AgreementItemOut* Record, as defined in the cXML User's Guide. At a minimum, you should enter the ItemOut field.

b *AgreementItemOut* can contain the following child elements:

Field	Value
MaxAmount	Optional. Contains the maximum amount for this particular line item.
MinAmount	Optional. Contains the minimum amount for this particular line item.

Field	Value
MaxReleaseAmount	Optional. Indicates the item level maximum amount per release.
ItemOut	Required. A line item that is part of the Master Agreement. The lineNumber attribute in the **ItemOut** specifies the corresponding lineNumber on the Master Agreement in the Procurement Application. The quantity attribute in the **ItemOut** should be set to "one" and ignored at the Master Agreement implementation processing stage.

c Click the **Submit** button. You should see the line item in the **Agreement Items** table.

7 Click the **Create MasterAgreementRequest Message** button.

8 Click the **Send cXML Request** button. The response content is displayed.

Testing an InvoiceDetailRequest Message

You can test InvoiceDetailRequest transaction by sending a test message to the URL specified on the *Config* module.

To create and send InvoiceDetailRequest test message

1 Start the Server Administrator. See the *webMethods Integration Server Administrator's Guide* for instructions.

2 In the Ariba Supplier menu in the navigation area, select **Test**. The Test page is displayed.

3 Select the test type **InvoiceDetailRequest**. The Test cXML Request Processing screen is displayed.

4 Set the cXML **From**, **To**, and **Sender** credentials by clicking the **Set Credentials** button and completing the following fields to specify the credentials of the sender and receiver of the document. (To send empty values, click the check boxes next to the fields.)

Field	Value
Identity	An ID string representing the entity.
Domain	Select one of the following values: ■ **DUNS**. If the ID is a DUNS number. ■ **NetworkId**. If the ID is an Ariba Network ID. ■ **AribaNetworkUserId**. If the ID is an Ariba Network user ID.

Field	Value
Shared Secret	For the Sender's credential, this must be the supplier's Shared Secret, which is configured on both the Ariba CSN Network and on the Ariba Supplier OnRamp's *cXML*.
Digital Signature	Optional. For Sender's Credential only.
Signature Type	Optional. For Sender's Credential only. The Digital Signature Type.
Signature Encoding	Optional. For Sender's Credential only. The Digital Signature Encoding.
User Agent	For Sender's Credential only. The sending system's User Agent Identity.

5 Set Invoice Detail Request Header fields. Specify the following fields:

Field	Value
Invoice ID	A supplier-generated identifier for the Invoice. Identical to the Invoice Number that appears at the top of a physical Invoice.
Is Information Only?	Indicates whether the buying organization needs to take action. Select one of the following: ■ **yes** - Invoice is for the buying organization's information only (no action needs to be taken by the buying organization). ■ **Not specified** - (Default) Invoice is functional. The buying organization needs to take action upon receiving this document (submit payment or accept credit).
Purpose	Purpose of the invoice. Select one of the following: ■ **standard** - (Default) A standard billing statement from the supplier to the buying organization. ■ **creditMemo** - A credit memo for issuing credit to the buying organization. isHeaderInvoice must be yes. Also, the elementInvoiceDetailSummary/DueAmount must be a negative amount. ■ **debitMemo** - A debit memo for billing a balance owed by the buying organization. isHeaderInvoice must be yes. Also, the element InvoiceDetailSummary/DueAmount must be a positive amount.

Field	Value
Operation	How this document is acting on the invoice. Select one of the following: ■ **new** - (Default) Creates a new invoice. ■ **delete** - Cancels an existing invoice. The PayloadID of the existing invoice must be specified in a Document Reference.
Invoice Date	Date and time that the Invoice was created (should be earlier than the cXML timestamp).

6 Set Invoice Detail Header Indicator fields. By default all indicators are set to **no**.

Field	Value
Is Header Invoice?	Select one of the following: ■ **yes** - Header invoice. Invoice uses InvoiceDetailHeaderOrder, which contains header level invoice information without item details ■ **Not specified** - Detail invoice. Invoice uses InvoiceDetailOrder, which contains item details.
Is VatRecoverable?	Select one of the following: ■ **yes** - The entire invoice is VAT (Value Added Tax)-recoverable. ■ **no** - The entire invoice is not VAT - recoverable.

7 Set Invoice Detail Line Indicator. This record indicates that invoicing details exist at invoice line level.

Field	Value
Is Tax Inline?	Select one of the following: ■ **yes** - Tax (Tax) is provided at invoice line level. ■ **no** - Tax is not provided at invoice line level.
Is Shipping Inline?	Select one of the following: ■ **yes** - Shipping (InvoiceDetailLineShipping) is provided at invoice line level. ■ **no** - Shipping is not provided at invoice line level.

Field	Value
Is Accounting Inline?	Select one of the following: ■ **yes** - Accounting distribution (Distribution) is provided at invoice line level. If isHeaderInvoice is true, this indicator must not be specified, because Distribution is available only at item level. ■ **no** - Accounting distribution is not provided at invoice line level
Is Special Handling Inline?	Select one of the following: ■ **yes** - Special handling (InvoiceDetailLineSpecialHandling) is provided at invoice line level. ■ **no** - Special handling is not provided at invoice line level.
Is Discount Inline?	Select one of the following: ■ **yes** - Discount (InvoiceDetailDiscount) is provided at invoice line level. ■ **no** - Discount is not provided at invoice line level.

8 To add detail records, click on the following buttons:

   a **Invoice Partner** - a party involved in invoicing, including the issuer of the invoice and the person sold to.

   b **Invoice Detail Orders** - the invoice information of an order with item details, used only when isHeaderInvoice is false (not specified).

   c **Invoice Detail Header Orders** - the header invoice information of a purchase order, without item details, used only when isHeaderInvoice = “yes”.

   d **Invoice Detail Summary** - the summary information of an invoice. See cXML User's Guide for details about these detail records.

9 Click the **Create InvoiceDetailRequest Message** button.

10 Click the **Send cXML Request** button. The request will be sent to the URL specified on *Config* module.

webMethods.

CHAPTER 3

Using Transaction Authentication

Overview

The Header element of the cXML document contains addressing and authentication information. The Header is the same regardless of the specific Request or Response cXML message. The main elements of the Header are From, To, and Sender. Each of the elements contains the Credential element that allows you to specify identification and authentication values.

Credential has the following attributes:

- *domain*. Specifies the type of credential. This attribute allows documents to contain multiple types of credentials for multiple authentication domains. For example, the domain of the messages sent on the Ariba Supplier Network can be the AribaNetworkUserId to indicate an email address, DUNS for a D-U-N-S number, or NetworkId for a pre-assigned ID.
- *type*. (Optional) Requests to or from a marketplace identify both the marketplace and the member company in From or To Credential elements. In this case, the credential for the marketplace uses the type attribute, which is set to the value "marketplace".

Credential contains an Identity element and optionally a Shared Secret, Digital Signature, Signature Type and Signature Encoding. For the details about these elements, see the cXML User's Guide.

Digital Signatures

The Ariba Supplier OnRamp allows suppliers to specify a digital signature and stores the signature in the repository. For each cXML document, the wm.b2b.cxml:receiveCXML service checks for the authenticity of the document by verifying the digital signature specified in the *Sender* Credentials of the document. The Test module of the Ariba Supplier OnRamp provides fields where the digital signature can be specified for a specific cXML message to be tested.

Complete the following steps to configure digital signatures.

To configure digital signatures

1. In the Integration Server Administrator, click **Ariba OnRamp** under **Adapters**.
2. Click **cXML**. The **cXML Message Fields** page appears.
3. Go to the **Digital Signature** section and provide the following fields.

Field	Value
Digital Signature	Supplier's digital signature. The cXML.org recommends using a self-contained PK7 format and the current timestamp. For example, Ariba ORMS 5.1P4.
Signature Type	The type of digital signature. For example, PK7 self-contained.
Signature Encoding	The signature encoding in the XML stream. For example, Base64.

4 Click **Update**.

Index