covisint
a subsidiary of Compuware Corporation

HTTP(S) Connector

August 2004
Product Revision Level 2.0.0

Table of Contents

INTRODUCING HTTP(S) CONNECTOR

This user guide contains information you will need to use HTTP(S) Connector's ***Post-Post*** mode to connect to Covisint Connect data messaging. With the ***Post-Post*** type of connectivity described in this guide, you can use a Web server and programmatic HTTP(S) client to exchange information with your trading partners(TP). To use the HTTP(S) Connector, you should have prior knowledge of the HTTP(S) protocol, Web servers and TCP/IP networking basics. The connectivity test implies basic knowledge of EDI document standards. The intended audience for this document includes TP integration developers, network engineers and customer connection support staff and third-party organizations. Contact connectsupport@covisint.com with questions or concerns.

Overview - HTTP(S) Connector

The HTTP(S) Connector is a component of the Covisint Connect messaging hub. It supports two modes of connectivity - "Post-Post" mode and "Mailbox" mode. This guide is dedicated to the "Post-Post" mode. There is a different guide available for "Mailbox" mode.

The HTTP(S) Connector in the "Post-Post" mode allows a TP with a standard HTTP(S) server and/or client software to perform the following tasks:

- Connect to Covisint using a unidirectional or bi-directional TCP/IP link over the public Internet, VPN tunnel or ANX/ENX networks.
- Submit business documents to be processed by Covisint using a regular programmatic client on the inbound (to Covisint) HTTP connection.
- Get business documents from Covisint using a regular Web server on the outbound (from Covisint) HTTP connection.
- Perform secure communications over SSL (HTTPS), if required..

All TPs have the choice of inbound-only, outbound-only or bi-directional connectivity with the HTTP(S) Connector "Post-Post" mode.

Functional Description – HTTP(S) Connector

Inbound Flow

On the inbound flow to Covisint (*Figure 1*), the HTTP(S) Connector accepts client posts, extracts message data and optional routing information from the HTTP POST request and sends it to the Covisint Connect Document Recognition Service. The HTTP(S) Connector acknowledges all successful posts by sending a Tracking ID back to the client.

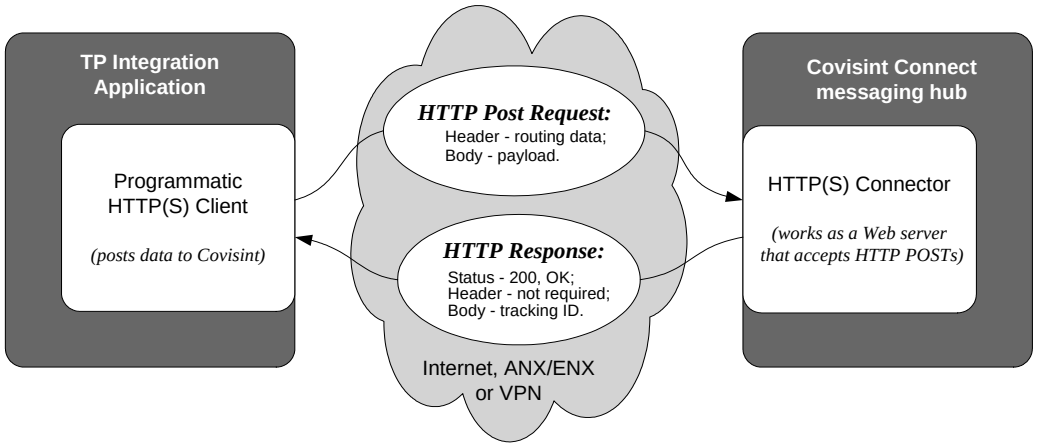

Figure 1: Inbound Flow to Covisint

Outbound Flow

On the outbound flow from Covisint (*Figure 2*), the HTTP(S) Connector posts the message to the TP's Web server. The message payload is contained in the body of the HTTP POST request. The Tracking ID is supplied in the HTTP POST request header.

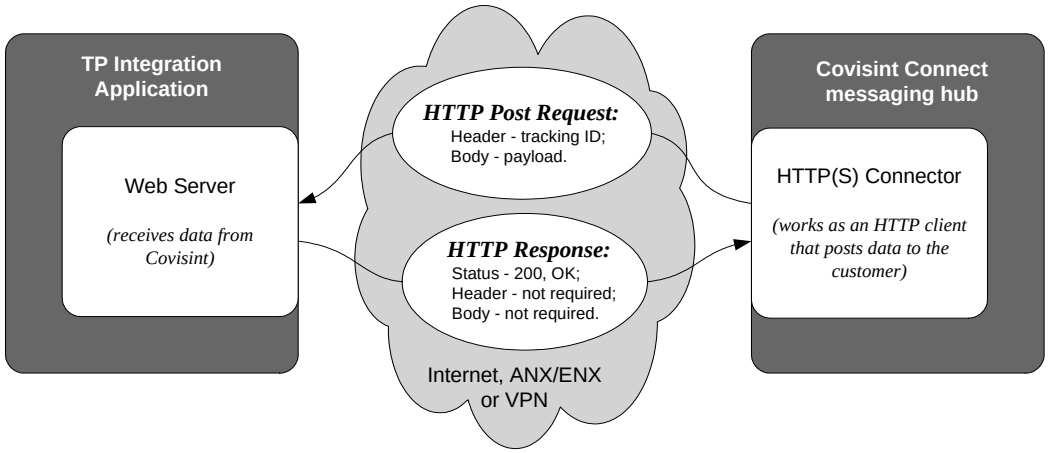

Figure 2: Outbound Flow from Covisint

How to perform Initial Setup – HTTP(S) Connector

Obtain User ID, Password and URL from Covisint

Please contact connectsupport@covisint.com to get your unique username, password and URL. You will establish your Trading Partner Profile and initial Trading Partner relationships when you register. Also, Covisint will provide you with an account and password to access the administration Web application that allows you to manage your TP profile and check the transaction log. Credentials for HTTP Connector data access and the administration Web application are different for security reasons.

Determine Type and Attributes Of Network Connection

Before you can connect, some necessary communication requirements must be satisfied. The following table lists the connectivity information for the HTTP(S) Connector:

Network	IP Address	Port
Public Internet	64.37.249.63 (https://messaging.covisint.com)	443
ANX	206.18.241.63	443
ENX	To be determined, contact connectsupport@covisint.com	TBD
VPN	To be determined, contact connectsupport@covisint.com	TBD
Messaging Admin on public Internet	https://connect.covisint.com	443

When using the public Internet, secure HTTP over SSL (HTTPS) is required. For a highly secure VPN connection, plain HTTP should be used since double encryption reduces communication channel performance. When using secure ANX or ENX networks, you may request HTTPS for additional connection-level security, though it's not required.

Provide Covisint with Type and Attributes of Network Connection to Your Web Server

Send your connectivity information (the network type, IP address, URL, username, password) and connection test scenario to connectsupport@covisint.com. Use the TP Web Server Configuration Form in Appendix 1.

Testing Credentials and Connection

The easiest way to test your Covisint inbound connection is to launch a Web browser and make an attempt to access the "ConnectionTest" URL:

1. Open the browser and enter the following URL:

   http(s)://host:port/invoke/HTTPConnector.Inbound.Handlers/ConnectionTest

2. Supply your valid credentials when prompted.
3. Your HTTP request dump should appear on the browser screen, as shown below:

```
**** HTTP Request Dump Started.

**** HTTP Request Header:

HTTP/1.1 200 OK
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
Host: 10.68.1.63:44300
Connection: Keep-Alive
Cookie: ssnid=1981v7MXEBRgUh0rBOtNoiQ1AAn4ZkY=555563
Authorization: Basic 2UpWthZRQW5vtay5bc3RYXRYZcj=

**** HTTP Request Body:

<null>

**** HTTP Request Dump Finished.
```

The connection is not established if you get a timeout with "**The page cannot be displayed**" error on the browser screen. The credentials are not valid if you are repeatedly prompted for a username and password.

How to conduct the One-Way Test – HTTP(S) Connector

The one-way test is a process of sending a test document from the TP to Covisint (inbound connection) or from Covisint to the TP (outbound connection).

Test Inbound-only (TP to Covisint) Connection

1. Prepare a valid EDI or EDIFACT business document with your ID as both sender and receiver (find a sample document in Appendix 2).
2. Submit the prepared message to the URL supplied by connectsupport@covisint.com. Use your programmatic client to do this (see paragraph **Post Message from Programmatic Client** for details).
3. Allow several seconds for message processing.
4. Find the internal Covisint tracking ID in the HTTP response and use it to track your message inside the messaging hub with help from the administration Web application, https://connect.covisint.com.

Test Outbound-only (Covisint to TP) Connection

1. The outbound test can be initiated from Covisint only. Please contact the Covisint support team at connectsupport@covisint.com to schedule the test.
2. Have your Web server ready to accept messages from Covisint.
3. Ask the Covisint support team to send a test transaction.
4. Check your Web server for the inbound request; it should contain the message sent by Covisint.

How to conduct the Loopback Test – HTTP(S) Connector

Note:
The loopback test is possible only when you have two-way HTTP(S) connectivity established with Covisint.

The loopback test is a process of sending a test document to and from yourself. There are two ways to execute this task.

Test with the Roundtrip Document

1. Prepare a valid EDI business document with your ID as both sender and receiver (find a sample document in Appendix 2).
2. Submit the prepared message to the URL supplied by connectsupport@covisint.com. Use your programmatic client to do this (see paragraph **Post Message from Programmatic Client** for details).
3. Allow several seconds for message processing.
4. Check your Web server for the inbound request. It should contain your original message, returned back.
5. Run a binary comparison utility against the initial and loopback messages.

Test Using Explicit Roundtrip Routing Data

1. Prepare any valid EDI or EDIFACT business document as a test message (find a sample document in Appendix 2)..
2. Prepare the explicit routing data - sender TP ID, receiver TP ID and message type. Use your TP ID as the value for both sender and receiver. TP IDs should be formed as a string concatenation of "**EDI Interchange ID Qualifier**" "**~**" (tilde) and "**EDI Interchange ID**" itself, e.g. "**ZZ~0123456789**." Form the message type as a string concatenation of the message format prefix "**x12**", "**~**" (tilde) and 3-digit numeric value that identifies "**EDI Transaction Set Control Number**" (transaction type), e.g. "**x12~830**."
3. Submit the prepared message to the URL supplied by connectsupport@covisint.com using your programmatic client. Pass extra routing data either in the HTTP request header or in the URL string, as described in the paragraph **Post Message from Programmatic Client.**
4. Please follow steps 3 -5 from the ***Test with the Roundtrip Document*** above.

The loopback test is successful when you get the initial document back unchanged.

How to Post Message from Programmatic Client – HTTP(S) Connector

Post Message from Programmatic Client

The HTTP(S) Connector accepts an HTTP(S) POST request from any programmatic client that supports the HTTP 1.1 protocol specification. The message payload should be contained in the body of the POST. The content type setting on the HTTP header should have a value of "**application/octet-stream.**" The internal Covisint tracking ID is returned as a small XML snippet in the body of the HTTP response, formed in reply to the customer's POST request.

For all client posts, use the URL string supplied by Covisint during your initial setup.

Request header requirement:

Content-type: application/octet-stream

Response header to expect:

Content-type: text/xml

Response body to expect:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
    <TrackingId>a117a21b-ef33-12b7-1432</TrackingId>
</Response>
```

Post Message with Explicit Routing Data

As an additional feature, the HTTP(S) Connector allows you to perform channel-driven routing, which controls the message routing process at the Covisint messaging hub. Typically the routing data is extracted from the message body during the document recognition process and doesn't require any customer intervention. However, if you want to override the routing data contained in the message body itself, or to specify routing data explicitly, e.g. for a message that doesn't contain any routing data at all, you can do so by adding three extra parameters either to the HTTP request header or to the URL request string (whichever is easier to implement - your choice)

Pass routing data in the HTTP request header

Request header requirement:

Content-type: application/octet-stream
From-party-id: Covisint-defined Partner ID
To-party-id: Covisint-defined Partner ID
Message-type: Covisint-defined Message Type

Pass routing data in the URL request string

Add the following to the Covisint-supplied URL string:

?from_party_id=xxxx&to_party_id=yyyy&message_type=zzzz

Add to the request header:

Content-type: application/octet-stream

In both cases of channel-driven routing, the data that is contained in the HTTP header or URL string overrides the result of the document recognition process of the messaging hub.

Reliable Delivery

HTTP is not a reliable delivery protocol. To ensure a message is delivered properly, the programmatic HTTP client must perform a series of retries in case of error. The HTTP Connector retry policy doesn't differentiate the transient and fatal errors. Every transmission failure causes the retry. The recommended retry session should contain no less than three attempts with the repeat period no less than 30 seconds. Retry sessions may be repeated several times, unless the error is eliminated. If the error persists, your integration software should generate an application alert.

How to Receive Message with Web Server – HTTP(S) Connector

The HTTP(S) Connector transmits outbound data with the help of regular HTTP "POST" requests from the programmatic client to the TP's Web server. The programmatic client at Covisint end- supports regular HTTP, cookies, SSL, basic and client certificate authentication..

Your Web server that accepts HTTP requests from Covisint must satisfy several minimum requirements:

- Basic HTTP server functionality according to RFC 2616
- Support for HTTP over SSL
- Basic authentication support
- Binary-safe HTTP POST body processing

You can use any software that meets the above-mentioned requirements, including Apache HTTP Server, BEA WebLogic, IBM WebSphere, webMethods Integration Server, Microsoft Internet Information Server, etc.

HTTP(S) Connector POST Request Structure

The HTTP(S) Connector forms the standard HTTP request according to RFC 2616. The HTTP request header is extended with two parameters: **Tracking-id** (the unique internal Covisint messaging hub ID) and **Content-type** (defaulted to "**application/octet-stream**"). Additional header parameters can be specified in the TP profile, if needed. The HTTP request body carries the message payload.

POST Request sample:

```
HTTP/1.0 200 OK
User-Agent: Mozilla/4.0 [en] (WinNT; I)
Accept: image/gif, */*
Host: 10.68.1.63:44300
Authorization: Basic RyYptYZRtW5pb2UaWj53vchQWRZ=
Content-type: application/octet-stream
Tracking-id: B38CA888-1E9F-CCD2-92DA-3F82C4EE1DB2
Cookie: ssnid=567QzFho|nKbG7Yn7qaMzqWzy1|nw=555563
Content-Length: 256

<HTTP Request Body goes here>
```

Reliable Delivery

HTTP is not a reliable delivery protocol. To ensure the message is delivered properly, the programmatic HTTP client at Covisint performs a series of retries in case of error. The HTTP client doesn't differentiate the transient and fatal errors. Every error situation is considered a reason for the retry. Retry settings - number of attempts and retry period - are configurable in the TP profile.

Asynchronous Message Processing

It is important to understand that message processing at the TP destination application must be performed asynchronously. This means the Web server must acknowledge the successful HTTP data transfer immediately, without waiting for the TP business application to process the message completely. This approach, known as **loose integration**, guarantees the transport-level connection doesn't consume server and/or network resources for extended periods.

APPENDIX

TP Web Server Configuration Form

Complete the form below and submit to connectsupport@covisint.com.

Covisint TP Web Server Configuration Form	
Covisint-assigned Trading Partner ID	
Trading Partner company name	
Company postal address	
Corporate Web Site	
Administrative contact (e-mail, phone, pager)	
Technical contact (e-mail, phone, pager	
Network type (Internet, ANX, ENX, VPN)	
Web server DNS name	
Web server IP address	
Inbound URL	
Username	
Password	Send in separate e-mail
Connection test scenario	

Sample EDIFACT DELFOR Document

Please use this sample EDIFACT DELFOR document for one-way and loopback tests. Insert your Covisint-assigned TP ID by executing a find and replace for the substring YOUR_ID_HERE.

sample_delfor.edifact

Common Errors – HTTP(S) Connector

To indicate an error response, HTTP(S) Connector always uses standard HTTP response codes with detailed error descriptions passed in the HTTP format as the response body. See the most common error situations below.

Error Description	HTTP Status Code	HTTP Status Message	Response Body
No error	200	OK	~
Access denied	403	Forbidden	"Access denied" error as the webMethods IS HTML error report
Unknown service or resource in the URL string	403	Forbidden	"Unknown Service Exception" as the webMethods IS HTML error report
Common HTTP(S) failure	403	Forbidden	403 Forbidden - Service Error - HTTP Connector failure <details>
Common Web server error	500	Server Error	~

Errors on Outbound Message Flow from Covisint

The HTTP(S) Connector programmatic client expects standard HTTP status codes to be returned from the TP's Web Server. The successful HTTP transfer is acknowledged by any HTTP status code in the range 200-299. The HTTP transfer is a failure if the HTTP status code is less than 200 or greater than 300.

Glossary

The following terms are used in the guide:

ANX - Automotive Network Exchange.

Basic Authentication - A secure method of authentication that uses HTTP and HTTP over SSL (HTTPS), in which the server authenticates via the user name and password obtained from the HTTP client

Client Certificate Authentication - A secure method of authentication that uses HTTP over SSL (HTTPS), in which the server and, optionally, the client authenticate each other with Public Key Certificates.

EDI - The Electronic Data Interchange group of standards.

ENX - European Network Exchange.

HTTP – Hypertext Transfer Protocol, an application-level protocol standard (defined in RFC2616) used on IP networks, such as the Internet, to transfer Web content between computers.

HTTP(S) Connector – A messaging hub component that provides access to messaging hub services using the HTTP(S) protocol.

ISA, GS, ST - EDI standard definitions for the specific parts of a business document.

"Post-Post" Mode - A communication method, supported by the HTTP(S) Connector, that implies the usage of HTTP POST requests for business data transfers in both directions - from a TP to Covisint and from Covisint to a TP.

Programmatic HTTP(S) Client – A software component in custom applications that implements the HTTP and SSL protocol RFCs and uses them to communicate with the Web (HTTP) servers in automatic mode.

SSL - Secure Socket Layer protocol that provides data encryption, server authentication, message integrity and optional client authentication for TCP/IP connections.

TCP – A transport-level protocol widely used on IP networks, including the Internet.

TP – Trading Partner

VPN - Virtual Private Network.