XML PORTAL CONNECTOR (XPC)™

INSTALLATION AND ADMINISTRATION GUIDE

Version 4.0

Commerce One
Corporate Headquarters
4440 Rosewood Drive
Pleasanton, CA 94588-3050

XPC Installation and Administration Guide, Version 4.0

February 2001

COMMERCE ONE, Inc. Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted.

Contents

Preface

Purpose of This Guide

XML Portal Connector (XPC) is a platform for applications and trading partners to exchange documents over MarketSite. XPC manages a secure MarketSite connection for runtime document exchange. XPC can run:

- Within a marketplace to connect hosted applications and business services.
- At a trading partner site to connect enterprise applications to MarketSite.

For integration development, XPC provides a component-based architecture with an extensible, plug-and-play environment that trading partners can tailor to their specific needs.

The *XPC 4.0 Installation and Administration Guide* provides information necessary to install, configure, and maintain XPC at a Trading Partner site. It does not address installing and maintaining XPC with a marketplace. For information using XPC within a marketplace, refer to *MarketSite 4.0 Installation Guide for NT.*

This guide includes:

- Instructions for installing XPC
- Instructions for connecting to MarketSite
- Information for maintaining and troubleshooting the installation.

Audience

This book is for administrators responsible for installing and managing XPC 4.0. It assumes administrators are familiar with MarketSite, xCBL documents, and the environment in which they are installing XPC.

Related Information

The following manuals are available on your XPC installation CD or from the Internet.

Document	Description
XPC Developer Guide and API Reference	General information about XPC and information about customizing XPC installations.
XML Interconnectivity Guide	Information about the structure of Commerce One’s xCBL business documents.
XDK Pro Developer Guide	Documents how to use Commerce One XML Parser (CXP), how to use the SOX to Java compiler, and how to interface with CXP through SAX
xCBL Online Reference Guide is available at *www.xcbl.org*	Documents xCBL 3.0
HotFS Installation and Configuration Guide	Describes how to install and configure HotFS including XCC configuration, sample scenarios, and client and service setup and execution.

How to Use This Guide

The information in this book is organized for the installation and configuration of XPC for trading partners. The following table describes each chapter in this guide:

Chapter	Description
1	Requirements and instructions for installing locally.
2	Registrations and configurations for connecting to MarketSite.
3	Security, firewalls, and authentication.
4	Advanced configuration for special options.
5	Administration, maintenance, and pending documents.
6	Reading event logs and troubleshooting.

Documentation Conventions

This guide follows these typographical conventions:

Convention	Meaning / Example(s)
<install:root>	All pathnames are expressed relative to the root directory of the XPC installation.
italics	Indicates emphasis, a book name, or the name of a variable. Emphasis example: ■ Do *not* perform this procedure until you have backed up your data. ■ Back up your data *before* performing this operation. Book name example: ■ *Enterprise Buyer Administration Guide* Variable name in text: ■ Enter the *servername* in the Server field.
`monospace text`	Indicates file and path names, printed or displayed computer output, code samples, or text you type at the keyboard. Text typed at the keyboard is not case sensitive unless so noted, such as when typing commands at a terminal window on Solaris or UNIX operating systems.
`[ ]`	Text enclosed in brackets denotes a variable name in code. These are often represented as separate words with underscores to denote one continuous string: ■ `http://www.[site_name].com/`

<table>
<tr><th>Convention</th><th>Meaning / Example(s)</th></tr>
<tr><td>{ }
and
|</td><td>Curly braces ({and}) and vertical bars (|) are used together to separate options:
■ {true | false}
■ {on | off}
■ {low | medium | high}
In addition, vertical bars are used in text to denote menu paths:
■ File | Import... | File...</td></tr>
<tr><td>/ \</td><td>Slashes (/) are used to separate directory names on Solaris and other UNIX operating systems. Backslashes (\) are used to separate directory names on Windows NT and Windows 2000.</td></tr>
</table>

If You Need Help

If you have purchased XPC software directly from Commerce One, Commerce One Technical Support is available to you by contacting one of the following:

- csa@commerceone.com
- By fax at 925-520-6060
- Hotline at 925-520-5959 or 800-949-8939
- http://www.commercecone.com/solution/tech.htm

If you purchased XPC software from a different source, such as your GMP or Systems Integration Partner, contact that source for technical support.

1 Installing XPC on Windows NT

This chapter provides the steps required to install and test connections for XPC 4.0 on Windows NT. Information and instructions are given in the following sections:

- Summary of Setup Steps on page 1-1
- System Requirements on page 1-2
- Removing a Previous XPC Installation on page 1-4
- Adding JDK to Path Variable on page 1-4
- Installing XPC on page 1-4
- Testing the Installation Locally on page 1-6

Note .... If your XPC installation is to be fully integrated with MarketSite 3.x or MarketSite 4, use the MarketSite documentation to install XPC as part of your MarketSite installation.

Summary of Setup Steps

The steps for setting up a test installation include installing locally and connecting with MarketSite. For a production installation, configure for security and configure advance options as needed.

All Setup Steps are listed in this table; refer to the last column of the table for locations of complete instructions.

Step	Action	Reference
1	Verify that your XPC server is ready for installation.	System Requirements on page 1-2
2	Remove previous XPC installation, if it exists.	Removing a Previous XPC Installation on page 1-4
3	Install the files from your XPC CD.	Adding JDK to Path Variable on page 1-4
4	Use the XPC Invoker to verify the local installation.	Testing the Installation Locally on page 1-6
5	Register in MarketSite 3.x and configure for https. or Register in MarketSite 4.0 and configure for SonicMQ.	Connecting to MarketSite on page 2-1
6	Configure and restart XPC.	Configuring XPC on page 2-7
7	Send a round-trip ping to MarketSite.	Testing Your Installation on page 2-9
8	**This completes testing of your basic installation and MarketSite communication.**	
9	Add security for your Production installation.	Security for Production Systems on page 3-1
10	Plan and configure advanced options in your Production installation.	Advanced Configuration on page 4-1
11	Test your Production installation.	Testing Your Production Installation on page 3-13
12	Review Administration and Troubleshooting methods.	Administration on page 5-1 and Troubleshooting on page 6-1.

System Requirements

The hardware and software requirements are listed in this section. The example environment described in this section is used to test XPC at Commerce One.

Verify that you have the system requirements before following installation instructions.

Hardware Requirements

The current release of XPC with SonicMQ (when both are run as NT Services) will occupy a minimum memory footprint of about 120 MB and a disk footprint of about 500kb. The memory footprint includes the entire Java runtime environment and the XPC server. System requirements include:

- Memory: 512 MB minimum
- Diskspace: 100 MB minimum

Note .... The default heap (-mx) requirement for JVM is 512M. If large messages are sent frequently, increase the JVM -mx to 1024M.

Software Prerequisites

Commerce One certifies XPC 4.0 for the following environment:

- Windows NT 4.x Server with Service Pack 6a
- If you plan to use https transport, it must be installed. SonicMQ 3.0 reliable messaging is bundled with XPC and is used internally in the XPC server, whether you use https and/or SonicMQ to connect to MarketSite.
- 128 bit or 56 bit encryption with Secure Socket Layer (SSL), X.509 certificate
- Java 2. XPC 4.0 requires and supports JDK 1.2.2_006 or higher without HotSpot. It does not support JDK 1.3. Sun makes the JDK 1.2.2_006 version available at http://java.sun.com/products/archive/j2se/1.2.2_006/index.html.

  You must add the JDK bin directory to your system environment Path variable before installing XPC (refer to Adding JDK to Path Variable on page 1-4). If it is not present, a message such as *Unable to open Password file* or *Unable to open broker.key file* displays and SonicMQ, an integral feature of XPC 4.0, will not install.

Note .... You can have multiple JVMs on your machine, and also have HotSpot VM installed. The installer will ask you for the location of a 1.2.2_006 (or higher within the 1.2.2 series of the Sun JVM) version of java.exe. It will then place this absolute PATH into the XPC invocations of java. XPCs invocations of java also use the **-classic** option to force the use of the Classic VM, even if the HotSpot VM is present.

Removing a Previous XPC Installation

Before beginning the installation procedure, you must first **Stop** and **Unregister** the XCC server and then remove any existing XPC installations from your computer. Follow these steps to remove any XCC and XPC installation files.

1. To stop XPC, select **Start | Programs | commerceone | XML Portal Connector 3.2 | Stop**.
2. To unregister the XCC server, select **Start | Programs | commerceone | XML Portal Connector 3.2 | Unregister**.
3. Select **Start | Settings | Control Panel**. Double-click on the **Add/Remove Programs** icon on the Control Panel. The **Add/Remove Programs Properties** dialog is displayed. Select the **Install/Uninstall** tab, find and select **XML Portal Connector 3.2**. Then click on the **Add/Remove** button from the dialog box. (Commerce One does not support the Repair and Update options.)
4. .Manually delete the transmitter\ccs\certs directory and all its subdirectories and files in the commerceone directory.

Note .... If you try to install XPC without first deleting the transmitter\ccs\certs directory, the initsecureclient script will fail.

Adding JDK to Path Variable

Because XPC requires Java, you must add the JDK location to the Path Variable.

1. From the **Start** menu, select **Setting | Control Panel | System**. The **System Properties** window is displayed.
2. Select the **Advanced** tab and then **Environment Variables**.
3. In the **System variables** section of the **Environment Variables** dialog, select **Path** and click on **Edit...**
4. Add the JDK bin location, usually c:\jdk1.2.2\bin, to the semi-colon delimited Path string. Click **OK** buttons on the Edit, Environment Variables, and System Properties windows to save your new Path variable.

Installing XPC

Before installing, determine:

- The name of your XPC server.

- XPC location. You can install in the default location or specify another directory for XPC files. The default location is recommended.
- Location for SonicMQ source directory.

Using your installation CD, follow these steps to install XPC on a Windows NT system.

1. Find the **setup.exe** file and double-click on it.

   The **InstallShield Wizard** Welcome Window is displayed.

2. To proceed with the installation, click **Next.**

   The **License Agreement** Window is displayed.

3. Review the terms of the agreement. If you agree with the terms of the License Agreement, click **Yes.**

   The **Find Java Virtual Machine** dialog is displayed.

4. Select or browse to a different directory to select a JVM executable. The installer checks the version. If it is not the correct version, a message is displayed.

   When the correct version of JVM has been found, the **XPC Server Name** Window is displayed.

5. Enter the name of your XPC server and click **Next**.

   The **Destination Location** Window is displayed. The Destination Folder is the location where XPC is to be stored.

6. If you click **Next,** the wizard installs XPC in **c:\commerceone\Xpc**. If you prefer to change the XPC location, **Browse** to a different directory to select another location and click **Next**.

Note .... If you choose to enter a directory, be aware that blank spaces are not allowed. For example, the path **c:\Program Files\XPC** is invalid.

The **Choose Source Directory** window prompts you to enter the location for SonicMQ installer files.

7. Enter the directory for SonicMQ installer files.

   The **Start Copying Files** window is displayed.

8. The **Start Copying Files** window requests that you review your selections before continuing. When you are satisfied with your settings, click **Next.**

9. The wizard installs XPC and displays the progress on the **Setup Status** window. During this time, your xCBL schema and event catalog structures are established, several transmission and security processes are enabled, and property files are

updated with the <install:root> directory you entered as your Destination Location and XPC Location.

Setup message dialogs display messages about the SonicMQ installation process.

After a few minutes, the **InstallShield Wizard Complete** Window is displayed.

10. The **InstallSheild Wizard Complete** Window asks you to choose whether or not to restart your computer now. Before using XPC, you must restart your computer. Choose to restart your computer now or later, and click **Finish**.

    You have completed the XPC installation. After your computer is restarted, proceed to the next steps to verify that your settings are correct.

11. To verify your installation, after rebooting, select **Start |Settings | Control Panel**. Click on the **Services** icon. On the **Services** window, verify that **CCSNTService** (XPC) and **SonicMQService** are started.

Testing the Installation Locally

Before connecting to MarketSite, follow these instructions to verify a correct local installation.

1. To verify your communication, select **Start** | **Programs | XMLPortal Connector | Invoker.** The **XPC Invoker** window is displayed.
2. On the **XPC Invoker Window**, click on the **Ping** button in the lower left corner.
3. The installation is successful if the response message, similar to the response shown here, ends with **Pong**.

```
<?soxtype urn:x-
commerceone:document:com:commerceone:ccs:doclet:ping:Ping.sox$1.0?><?import
urn:x-commerceone:document:com:commerceone:ccs:doclet:ping:Ping.sox$1.0?>
<Pong><PingInfo>nr=0</PingInfo>
<ServerTime>Tue Feb 06 21:49:59 PST 2001</ServerTime>
<ServerConfig>VeoServer</ServerConfig>
<ServerUser><UserName>SYSTEM</UserName>
<UserTimezone>America/Los_Angeles</UserTimezone>
<UserRegion>US</UserRegion>
<UserLanguage>en</UserLanguage>
<UserHomeDir>C:\commerceone\Xpc\runtime\servers\defaultserver</UserHomeDir>
<UserCurrentDir>C:\WINNT\system32</UserCurrentDir>
</ServerUser>
```

```
<ServerHost><HostName>cingersoll-lt</HostName>
<HostAddress>10.10.9.188</HostAddress>
</ServerHost>
<ServerOS><OSName>Windows NT</OSName>
<OSVersion>4.0</OSVersion>
<OSArch>x86</OSArch>
</ServerOS>
<ServerConnection><ConnectionPort>Unknown Port</ConnectionPort>
<ConnectionPath>Unknown Path</ConnectionPath>
<ConnectionProtocol>Unknown Protocol</ConnectionProtocol>
<ConnectionMethod>Unknown Method</ConnectionMethod>
<ConnectionQuery>Unknown Query</ConnectionQuery>
<ConnectionAuthType>Unknown Auth Type</ConnectionAuthType>
<ConnectionClientAgent>Unknown Agent</ConnectionClientAgent>
</ServerConnection>
<ServerJava><JavaVersion>1.2.2</JavaVersion>
<JavaVendor>Sun Microsystems Inc.</JavaVendor>
<JavaCompiler>symcjit</JavaCompiler>
<JavaJRE>false</JavaJRE>
</ServerJava>
<ServerSecurity><SecurityServerRealm>No Realm</SecurityServerRealm>
<SecurityServerCertificate>No Server Certificate</
SecurityServerCertificate>
<SecurityUserKeyStore>No Keystore</SecurityUserKeyStore>
</ServerSecurity>
<ServerQoS><PerformanceMetric>0.0</PerformanceMetric>
<NetworkMetric>0.0</NetworkMetric>
<MemoryAvailable>7522272</MemoryAvailable>
<DiskAvailable>0</DiskAvailable>
</ServerQoS>
</Pong>
```

4. If the response is unsuccessful, the response is similar to the following. In this case, access the systemStartup or debug files in the`<Install:root>/runtime/`

servers/<SERVERNAME>/logs directory and read the error messages.

```
<?soxtype urn:x-
commerceone:document:com:commerceone:ccs:doclet:error:Error.sox$1.0?><?impo
rt urn:x-
commerceone:document:com:commerceone:ccs:doclet:error:Error.sox$1.0?>
<Error><Code>INVOKER_ERRORGENERIC</Code>
<DefaultMessage><Message><MessageString>ERROR:
com.commerceone.ccs.excp.comm.communicator.TimeoutException: !!!
sendAndReceive: Timeout &apos;10000&apos; expired</MessageString>
<LanguageCode>en</LanguageCode>
</Message>
</DefaultMessage>
<Severity>Error</Severity>
</Error>
```

2 Connecting to MarketSite

This chapter provides the steps required to connect to MarketSite and test the communication between your XPC installation and MarketSite. Information and instructions are given as listed in the table of steps.

The following SonicMQ-related terms are used in this chapter.

Inter- Broker port	Port on which the local sonic broker listens for connections from the portal sonic broker.
Sonic to Broker port	Port on which the local sonic broker listens for connections from sonic clients running in either the local XPC server or the Invoker.
Sonic Node Name	The node name for the local sonic installation. This name is chosen by the trading partner and submitted to MarketSite Builder (MSB). Must be unique within a particular portal.
Portal Node Name	SonicMQ node name for the portal's sonic installation.
Portal Address	Address for the portal's SonicMQ installation.
Portal Port	Port on which the portal's sonic broker listens for connections from other sonic brokers.
Production Mode	A mode of sonic configuration that uses certificates for broker to broker authentication.

.

Step	Action	Reference
1	Determine which version of MarketSite with which to connect.	Which Version of MarketSite? on page 2-2
2	Register yourself as a Trading Partner on MarketSite. And for MarketSite 4.0, register with SonicMQ Broker and register the TP destination address.	Registering in MarketSite 3.x on page 2-2 or Registering in MarketSite 4.0 on page 2-3
3	For MarketSite 4.0, receive SonicMQ registration confirmation email.	
4	Configure XPC for communication with the Marketsite you registered with at Step 2.	Configuring XPC on page 2-7
5	Run PingMarketSite.	Testing Your Installation on page 2-9

Which Version of MarketSite?

Will you connect with MarketSite 3.x or MarketSite 4.0?

MarketSite 3.x	MarketSite 4.0
MarketSite operator registration	Trading Partner self-registration
https transport	SonicMQ transport and/or https

Registering in MarketSite 3.x

You will need the value for the HTTPS Port of your XPC server to register with MarketSite 3.x. The default value is 4433, which is created at installation. If you wish to change this port before registering, refer to Configuring XPC on page 2-7. This is optional and not normally necessary.

Request a copy of *MarketSite Builder Installation and Administration Guide* for MarketSite 3.x from your MarketSite operator. This manual gives the details you need to use MarketSite Builder for:

- Trading Partner Registration
- Manage Document Destination

Registering in MarketSite 4.0

Here are the defaults for the data you need to register with MarketSite 4.0. These defaults are created when XPC is installed. If you wish to change any of this information before registering, refer to Configuring XPC on page 2-7. This is optional and not normally necessary.

- **HTTPS Port** is 4433
- **Inter-Broker Port** is 2506
- **Sonic Node Name** is the <fully-qualified hostname> of the machine on which XPC has been installed. For example, if XPC has been installed on venus.commerceone.com, the sonic node name will default to venus.commerceone.com.
- **Sonic Queue Name** is XPC_<fully-qualified hostname>_ConnectorInbound with any dots in hostname replaced by underscore'_'. For example, if XPC has been installed on venus.commerceone.com, the queue name will default to XPC_venus_commerceone_com_ConnectorInbound.

Request a copy of *MarketSite Builder User Guide: Trading Partners for MarketSite4.0* from your MarketSite operator. This manual gives the details you need to use MarketSite Builder for:

- Trading Partner Registration
- Manage Document Destination
- Manage Sonic MQ Broker

Receiving Confirmation Emails

After registering with MarketSite 4.0, you will receive two email confirmations, a Trading Partner Registration email and a SonicMQ Broker Registration Confirmation email.

Trading Partner Registration Email

Here is an example trading partner email. The information from this email that is needed in Configuring XPC on page 2-7 is highlighted in **bold**.

```
Subject: Trading Partner Registration

TP Administrator:
```

```
Login information:

    User Id: gzhao
    Organization Id: TomasZhao
    Password: 8e8484ZP

Your company registration has been completed with the following
information:

    Company name: Supplies For All
    Contact name: Zhao Tomas
    Contact Email address: tomas.zhao@commerceone.com

    TPID: e4bf0456-7810-1000-8c3f-0a0a01090001
    Authorizing Entity ID: e4bf0456-7810-1000-8c3f-0a0a01090001
    DUNS:
    EANID:
    LegacyId0:
    LegacyId1:

Basic business profile information:

    Visibility Code: NMM
    Commodity Code (Buy):
    Commodity Code (Supply):
    URL:
    Logo URL:
    Description:
    Terms and Conditions:

    Phone number:
    Fax number:
```

```
    Street:
    City: Cupertino
    State:
    Postal Code:
    Country: US
    Language: en
    Timezone: PST
    Currency: USD

Administrator contact information:

    Name: Zhao Tomas
    Title:
    Email address: tomas.zhao@commerceone.com
    Phone number:
    Fax number:
    Street:
    City: San Francisco
    State:
    Postal Code:
    Country: US
    Language: en
    Timezone: PST
    Currency: USD

Business contact information:

    Name:
    Title:
    Email address:
    Phone number:
    Fax number:
    Street:
```

City:
State:
Postal Code:
Country:
Language:
Timezone:
Currency:

System configuration information:

Portal MPID: b2b7a208-780e-1000-9b62-0a0a09340001
Profile service MPID:
System account ID: GeorgeZhao
System account password: LQ78vGn6
Portal URL: jliu-lt.commerceone.com

SonicMQ Broker Registration Confirmation

You will receive an email confirming your SonicMQ registration that looks something like this. The information from this email that you use in Configuring XPC on page 2-7 is highlighted in **bold**.

Subject: Sonic MQ Broker Registration Confirmation

Your sonic MQ broker registration has been completed with the following information:

Node name: TomasZhao.com
Address type: CertAddress
IP address: 10.10.9.35
Port number: 2506
Queue name: XPC_ConnectInbound
Queue type: inbound

The following is the information for the portal node:

```
Portal node name:  commerceone.com
Portal IP address: snickers.commerceone.com
Portal port number: 2506
Portal IP address (backup):
Portal port number (backup):
```

Configuring XPC

With XPC installed and Trading Partner registration and administration with MSB completed, follow these steps to configure XPC for communication with the MarketSite with which you registered.

1. From the **Start** menu, select **Programs |XML Portal Connector 4.0 | Configure**. The **Configure XPC** window is displayed with the following sections:
   - XPC Server
   - Local Communication
   - Portal Communication
   - Transport Preferences
   - Sonic / Https Communication
   - Proxy Configuration
2. The **XPC Server** box contains defaults created during your installation session of the machine on which you have installed XPC. These are described in Configuring Additional Options on page 4-5.
3. The **Local Communications** section contains the defaults created during your installation. The defaults and examples are given in Registering in MarketSite 3.x on page 2-2 and Registering in MarketSite 4.0 on page 2-3.

Warning! If you change **Local Communications** information after having registered with MarketSite, you will need to amend your MarketSite registration.

4. For **Portal Communication**, the first box is **Trading Partner Information**.·
   - For **TPID**, enter the value of your TPID (for MarketSite4.0) or the value of your MarketParticipantID (MPID for MarketSite3.x). The TPID value is given in your Trading Partner registration email.
   - For **System Account Id**, enter the value of your System account ID (for

MarkteSite4.0) or the value of your System ID (for MarketSite3.x). The System Account ID is given in your Trading Partner registration email.

- For **System Account Password**, enter the value of your System account password (for MarketSite4.0), or the value of your System Password (for MarketSite3.x). The password is given in your Trading Partner registration email.

Ignore the **Certificate Manager** button at this time. You will use it to configure security for production mode; refer to Security for Production Systems on page 3-1 for details.

5. To configure your **Transport Preferences**, select Preference1 or Preference2 for sonic and/or https. You must select here the same choice(s) as you entered in MarketSite Builder (Manage Document Destination).

6. If you are using SonicMQ, select the **sonic** tab. What you enter here determines a routing connection from your XPC installation to the MarketSite where you are registered.

   - For the **Production Mode** checkbox, certificates are used to authenticate between Portal and Trading Partner brokers if it is checked; otherwise userid/password is used for authentication to Portal from Trading Partner. If you have registered with a test MarketSite, this checkbox will be unchecked.
   - For **Portal Address**, enter the Portal IP address from your Sonic MQ Broker registration email.
   - For **Portal Port**, enter the Portal port number from your Sonic MQ Broker registration email.
   - The **Portal Node Name**, enter the Portal Node Name from your Sonic MQ Broker registration email.

7. If you are using https, select the **https** tab. Your XPC server properties are updated with the information you enter here.

   - For **Portal Address**, if the Portal IP Address is not specified in the Trading Partner registration email, ask MarketSite operator.
   - For **Portal Port**, if the Portal Port is not specified in the Trading Partner registration email, contact your MarketSite operator for this information.

8. If you wish to configure a Proxy Server at this time, refer to Configuring Additional Options on page 4-5.

9. Review your entries. If you find an error, press either the Clear button or the Reset button at the bottom of the window and re-enter your data.

   - The **Clear** button clears all data.

- The **Reset** button reverts to the previously saved settings.

When you have verified that the information is correct, press the **Save** button to update the property files.

A message dialogs reminds you to restart the XPC Server.

10. To exit the Configure XPC window, press the **Exit** button.

11. For your settings to take effect, stop and restart the XPC server (from the **Start** menu, **Programs | XML Portal Connector 4.0 | Stop** and then **Programs | XML Portal Connector 4.0 | Start.**

Testing Your Installation

To test the communication between your XPC server and MarketSite, run PingMarketSite, as in these steps.

1. Run <install:xpc>\bin\pingMarketSite.bat:

```
cd <install:xpc>\bin
pingMarketSite
```

A self-addressed ping document is sent to MarketSite.

2. MarketSite routes the ping document to the Trading Partner destination address that you registered, your XPC address.

3. A ping service run on your XPC server receives the ping document and generates a pong document and sends the pong document to MarketSite.

4. MarketSite routes the pong document to your address.

5. Upon receipt of the pong document, you have successfully completed the test of your MarketSite configuration.

You can also send the ping document to MarketSite with the ping service running, and receive the pong document from MarketSite. To do this, you can add one argument of -recipient <MarketSite MPID for MarketSite3.x/Portal MPID for MarketSite4.0 from Trading Partner Registration email> in <install:xpc>\bin\pingMarketSite.bat script.

Because this script uses the com.commerceone.sample.xpc.docsender.DocSender class, you can find the detail information how to invoke this class in the README.txt under <install:xpc>\sample\com\commerceone\sample\docsender

3 Security for Production Systems

This chapter describes how to set up XPC for production on Windows NT. It includes the following sections:

- Firewall and Network Requirements on page 3-1
- Certificate Manager on page 3-3
- Authentication for Production Using https on page 3-3
- Authentication for Production Using SonicMQ on page 3-9
- Testing Your Production Installation on page 3-13.

Firewall and Network Requirements

This section details security considerations forn deploying the XPC Server and applications into a Trading Partner (TP) site with existing security, firewall, and proxy policies, and LAN topologies. The XPC adminstrator must use the firewall and network information given here while considering local Information Technology (IT) practices.

XPC 4.0 Server and SonicMQ Broker cannot currently be deployed in the DMZ due to a XPC 4.0 installation limitation.

XPC 4.0 supports two transfer or transport protocols, HTTP/S and JMS-based SonicMQ over Secure Sockets Layer (SSL):

- Inbound/Outbound https traffic from the internet is always to/from a MarketSite Portal Router. The maximum document size for https is 4 MB.
- Inbound/Outbound JMS/Sonic MQ over SSL traffic is always to/from a MarketSite Sonic MQ Broker. Maximum document size for SonicMQ is 10 MB. SonicMQ is

automatically installed during XPC installation.

Planning for XPC Deployment

In planning for XPC Deployment, note that:

- The XPC server and its SonicMQ Broker must be installed on the same NT machine in an internal protected network, that is, behind the firewall.
- In addition to having the XPC Server in a protected network, Commerce One strongly recommends that passwords stored in application property files be protected by encryption. All passwords used by the XPC server and Sonic MQ Broker are encrtypted for you by XPC.
- Correct Microsoft NT Lock Down procedures are strongly recommended. Refer to Microsoft Secure NT documentation in http://www.microsoft.com/ntserver/security/exec/overview/Secure_NTInstall.asp.

Internal Firewall Requirements

The following table summarizes the requirements for the XPC and transport connections.

	https	**SonicMQ**
Target server	XPC https server	SonicMQ Broker SSL server
Protocol	https	ssl
Firewall TCP Port	4433	4433

Proxy Firewall Requirements

If you are using the Proxy option (refer to Using Forward Proxy on page 4-2), the firewall requirements are as follows.

XPC applications can optionally connect to MarketSite through a local forward (SSL) proxy over both https and SonicMQ/SSL Transport. Commerce One supports both standard https and SSL Proxies. It is recommended that the Proxy Server address can be resolved by the local DNS available to the XPC application. If local DNS cannot resolve the proxy address, then configure the explicit IP address of the proxy Server in the XPC application in the Proxy Host field of the Configure XPC window.

Certificate Manager

Commerce One products use a web protocol to communicate with each other. As a web server, a Commerce One product uses the Secure Sockets Layer (SSL) to encrypt transactional data that travels outside of its own Local Area Network (LAN).

The Certificate Manager is a Commerce One tool for you to use to manage your KeyStore file, request and import a Certificate Authority root as Trusted Root, import a Server Certificate, and import a MarketSite Server Certificate as Trusted Client.

The authentication procedure differs between https and SonicMQ.

Authentication for Production Using https

Acquiring authentication requires pre-planning. It is recommended that you review the preparation and steps before you begin the steps.

Preparation for Authentication

Start now to arrange for your server certificate so that you are not delayed by Step 5. Contact your Certificate Authority and determine how much time their process requires.

Following the Steps

The steps detailed in this section for setting up security for https are summarized in this table. Note that to perform Steps 1 and 12, the XPC Server must be stopped. To stop the XPC Server, select **Start | Program | XML Portal Connector 4.0 | Stop**. .

Step	Description
1.	With the XPC server stopped, make a backup of keystore.
2.	Create a new server entry.
3.	Create a certificate request.
4.	Submit that certificate request
5.	Receive server certificate from your Certificate Authority (CA).
6.	Import the certificate.
7.	Set the entry as active.

Step	Description
8.	Optionally, configure trusted CA root certificates.
9.	Trust your MarketSite.
10.	Enable Client Authentication.
11.	Change the keystore password.
12.	With XPC Server stopped, make a new backup of keystore.

1. To make a backup copy of keystore, find the serverstore file in the **<install:root>/runtime/servers/<your server name>/certs directory.** Make a copy of that file and save it with a different name, such as **xpc_serverstore_default.**

a) From the **Configure XPC** window, click the **Certificate Manager**button in the center of the window. The **Certificate Manager** window is displayed.

b) In the **Certificate Manager** window, enter the following information:

- **Server** - Choose the server where the keystore is located. The Certificate Manager will auto populate this field with your options; choose the correct one from the drop-down menu.
- **Password** - The password for the KeyStore you will open. The password for default test Keystore is *admin*, you change this password in task 11.

b) Click **Open a Keystore**. This opens the main window for the Certificate Manager.

2. To create a new server entry, select the **Configure Server Entries** tab. Then click **Create**. The **Create Certificate Entry** dialog is displayed.

a) Complete these fields in the Create Certificate Entry dialog.

Field Name	Description
Certificate Entry Name	The name you choose for your new entry.

Field Name	Description
Algorithm Type	For algorithm type, RSA is the default. Choose **RSA** for an RSA keypair. Choose **DSA** for a DSA keypair.
Key Size	This determines the size of the keys. **56-bit** version has a default of 1024 bit, but can also be 512 bit. **128-bit** version has a default of 1024 bit, but can also be 512 bit or 2048 bit.

b) Click the **Create Certificate Entry** button.

c) A window shows confirmation that your server entry was created successfully. Click the **OK** button.

d) The **Configure Server Entries** window displays your new entry. The Certificate Details section lists *Dummy Certificate!*

3. To create a certificate request, select the **Create Certificate Request** tab. The **Create Certificate Request** window is displayed.

a) Complete all fields marked with a *****. Remaining fields are optional.

Caution .... The Certificate Manager requires a minimum of fields. Some Certificate Authorities require more fields than others and the format of each field may vary depending on the Certificate Authorities. Please refer to your Certificate Authorities documentation to determine which fields you need.

Field Name	Description
Certificate Entry Name	From the drop down menu, select the entry for which you are creating a certificate.
***Common Name**	This is the common name of the entry or server name, such as MyServer.CommerceOne.com. Be aware that SonicMQ requires a different format for common name.
Email Address	Email address of the server administrator.
***Organization**	Your company name.
Organizational Unit	Your organization within your company.

Field Name	**Description**
Locality	Your city.
State	Your state. This must be the full name, not the abbreviation. For example, use *California* not *CA*.
***Country**	Two-letter country code for your country. For example, US for United States.
Algorithm Type	The algorithm type you chose when you created the entry. This field will auto populate.
Key Size	The key size you chose when you created the entry. This field will auto-populate.

b) With the fields completed, click **Create Certificate Request**.

c) You are prompted to enter a name and location for the certificate request file you are creating. The default is **certreq.p10**. The extension p10 indicates these files are in PKCS10 format. Make any changes as needed, and click **Save** to continue.

d) The Save certificate request prompt confirms that the request file has been generated. Select **Copy to System Clipboard** to copy the request to the clipboard where it is easily accessible when you submit the request, or **Done** to finish and save the file.

4. In this step, you submit the certificate request you created in the previous step to a Certificate Authority (CA) to have the security certificate generated. Each CA has its own method for fulfilling certificate requests. The instructions here outline the online system used by Verisign OnSite. For more information about Verisign services, visit their web site at *http://www.verisign.com.*

a) When a company registers with Verisign, Verisign assigns a special Secure Server Enrollment web page for that company. The company’s administrator submits certificate requests at this page.

b) For each certificate request, the Verisign certificate request wizard asks for the exact server and directory name where the keystore was created.

c) The wizard requests that you cut and paste the entire contents of your certificate request file into the **Enter CSR Information** text box. To do this, open your certificate request file in an ASCII text editor such as Notepad (do not use a word processor such as Word that inserts formatting or control characters). Copy the entire contents and paste the entire contents including the
----BEGIN NEW CERTIFICATE REQUEST---- and

----END NEW CERTIFICATE REQUEST---- markers.

d) When you and the Verisign wizard are finished, the Verisign system generates the certificate. The certificate is sent to the administrator’s email address.

5. Receive server certificate from your Certificate Authority, either by email as in the Verisign example, or by some other method.

6. To import the certificate, select the **Configure Server Entries** tab. Then click to select the entry where you wish the certificate imported. With the entry selected, click **Import**. The available import formats are:

- Base 64
- DER
- PEM
- PKCS7
- PKCS12

Note .... The certificate must match the keypair stored in that entry and generated based on a certificate request for that entry.

Note .... The Certificate Manager will dynamically determine the format of the file you are importing, therefore the extension of the file is not important.

a) From the resulting browser window, choose the certificate file you want to import and click **Open**.

b) If the trusted root for this certificate authority is not yet within your keystore, a dialog prompts you to trust the Root certificate. Click **Yes** to continue.

c) When the certificate has been imported, a confirmation window is displayed. Click **OK** to finish.

Caution .... If the trusted root certificate is not available within the file being imported, you may need to import the root certificate separately first. To do so, follow the steps listed in Task 8, Configure Trusted CA Root Certificates.

7. Marking an entry as active indicates to your XPC server to use that entry from the keystore during startup. To set an entry as active, click on its name in the Entries dialog to select it, then click **Set as Active**. A confirmation screen is displayed when the entry is set as active. Click **OK**. A * precedes the name of an active entry in the Entries dialog.

Note .... You must restart the server after you have set an entry as active in order for the change to take effect.

8. Configuring a trusted CA root certificate is not necessary if the trusted root certificate is included in the certificate you import from your Certificate Authority.

To configure a trusted root, select the **Configure Trusted Root** tab and click **Import**. The **Select certificate file to import** dialog is displayed. Select the Trusted Root CA file to import and click **Open**. When the operation is complete, a confirmation dialog is displayed. Click **OK**.

9. To trust a MarketSite, get a copy of the MarketSite certificate from your MarketSite operator, typically by email.

Note .... The certificate must be in PKCS7 format.

a) Go to the **Configure Trusted Client** tab. Click **Import**. The **Select certificate file to import** dialog is displayed.

b) Select the Trusted client file and click **Open**. When the operation is complete, confirmation dialog is displayed. Click **OK**.

10. With XPC in Test mode, Client Authentication should be disabled. With XPC in Production mode, it should be enabled. To enable or disable Client Authentication, select the **Configure Trusted Client** tab. Click on disable or enable to change the setting.

Note .... You must restart the server after you have enabled or disabled Client Authentication in order for the change to take effect.

11. To change a keystore password, select the **Change Password** tab. Then:

a) Enter your old password.

b) Enter the new password.

c) Re-type the new password to confirm it.

c) Click **Change KeyStore Password**.

12. Ensure that the XPC Server is stopped. To make a new backup copy of keystore, find the serverstore file in the **<install:root>/runtime/servers/<your server name>/certs directory.** Make a copy of that file and save it with a different name, such as xpc_serverstore_https_production.

a) In the **Certificate Manager** window, enter the following information:

- **Server** - Choose the server where the keystore is located. The Certificate Manager will auto populate this field with your options; choose the correct one from the drop down menu.
- **Password** - The password for the KeyStore you will open.

b) Click **Open a KeyStore**.

Authentication for Production Using SonicMQ

Acquiring authentication requires pre-planning. It is recommended that you review the preparation and steps before you follow the steps.

Preparation for Authentication

Start now to arrange for your server certificate so that you are not delayed by Step 5. Contact your Certificate Authority and determine how much time their process requires.

Following the Steps

The steps detailed in this section for setting up security for SonicMQ are summarized here..

Step	Description
1.	Create a new server entry.
2.	Create a certificate request.
3.	Submit that certificate request.
4.	Receive server certificate from your Certificate Authority (CA).
5.	Import the certificate.
6.	Configure SonicMQ Broker.
7.	Import Trusted CA Root Certificate from MarketSite (optional).
8.	Make a backup file of the keystore file.
9	Stop and restart SonicMQ Broker.

Follow these steps to set up authentication for production using the Certificate Manager.

1. From the **Configure XPC** window, press the **Manage Certificate** button at the bottom of the window. The **Certificate Manager** window is displayed. If this is the first time you access this window, enter **admin** for the password and then set your own password. If you changed this password while performing the https steps, use the new password now.

2. To create a new server entry, select the **Configure Server Entries** tab. Then click **Create**. The **Create Certificate Entry** dialog is displayed.

   a) Complete these fields in the Create Certificate Entry dialog.

Field Name	Description
Certificate Entry Name	The name you choose for your new entry.
Algorithm Type	For algorithm type, RSA is the default. Choose **RSA** for an RSA keypair. Choose **DSA** for a DSA keypair.
Key Size	This determines the size of the keys. **56-bit** version has a default of 1024 bit, but can also be 512 bit. **128-bit** version has a default of 1024 bit, but can also be 512 bit or 2048 bit.

   b) Click the **Create Certificate Entry** button.

   c) A window shows confirmation that your server entry was created successfully. Click the **OK** button.

   d) The **Configure Server Entries** window displays your new entry. The Certificate Details section lists *Dummy Certificate!*

3. To create a certificate request, select the **Create Certificate Request** tab. The **Create Certificate Request** window is displayed.

   a) Complete all fields marked with a *****. Remaining fields are optional.

Caution**.....** The Certificate Manager requires a minimum of fields. Some Certificate Authorities require more fields. Please refer to your Certificate Authorities documentation to determine which fields you need.

Caution **.** The Common Name of the certificate must be the same as the XPC Broker node name for authentication to proceed correctly.

Field Name	Description
Certificate Entry Name	From the drop down menu, select the entry for which you are creating a certificate.
***Common Name**	This is the common name of the certificate and must be the same as XPC broker node name.
Email Address	Email address of the server administrator.
***Organization**	Your company name.
Organizational Unit	Your organization within your company.
Locality	Your city.
State	Your state. This must be the full name, not the abbreviation. For example, use *California* not *CA*.
***Country**	Two-letter country code for your country. For example, US for United States.
Algorithm Type	The algorithm type you chose when you created the entry. This field will auto populate.
Key Size	The key size you chose when you created the entry. This field will auto-populate.

b) With the fields completed, click **Create Certificate Request**.

c) You are prompted to enter a name and location for the certificate request file you are creating. The default is **certreq.p10**. The extension *p10* indicates these files are in PKCS10 format. Make any changes as needed, and click **Save** to continue.

d) The **Save certificate request** prompt confirms that the request file has been generated. Select **Copy to System Clipboard** to copy the request to the clipboard where it is easily accessible when you submit the request, or **Done** to finish and save the file.

4. In this step, you submit the certificate request you created in the previous step to a Certificate Authority (CA) to have the security certificate generated. Each CA has its own method for fulfilling certificate requests. The instructions here outline the online system used by Verisign OnSite. For more information about Verisign services, visit their web site at *http://www.verisign.com*.

a) When a company registers with Verisign, Verisign assigns a special Secure Server Enrollment web page for that company. The company's administrator submits certificate requests at this page.

b) For each certificate request, the Verisign certificate request wizard asks for the exact server and directory name where the keystore was created.

c) The wizard requests that you cut and paste the entire contents of your certificate request file into the **Enter CSR Information** text box. To do this, open your certificate request file in an ASCII text editor such as Notepad (do not use a word processor such as Word that inserts formatting or control characters). Copy the entire contents and paste the entire contents including the
----BEGIN NEW CERTIFICATE REQUEST---- and
----END NEW CERTIFICATE REQUEST---- markers.

d) When you and the Verisign wizard are finished, the Verisign system generates the certificate. The certificate is sent to the administrator's email address.

5. Receive server certificate from your Certificate Authority, either by email as in the Verisign example, or by some other method.

To import the certificate, select the **Configure Server Entries** tab. Then click to select the entry where you wish the certificate imported. With the entry selected, click **Import**. The available import formats are:

- Base 64
- DER
- PEM
- PKCS7
- PKCS12

Note .... The certificate must match the keypair stored in that entry and generated based on a certificate request for that entry.

Note .... The Certificate Manager dynamically determines the format of the file you are importing; therefore the extension of the file is not important.

a) From the resulting browser window, choose the certificate file you want to import and click **Open**.

b) If the trusted root for this certificate authority is not yet within your keystore, a dialog prompts you to trust the Root certificate. Click **Yes** to continue.

c) When the certificate has been imported, a confirmation window is displayed. Click **OK** to finish.

Caution..... If the trusted root certificate is not available within the file being imported, you may need to import the root certificate separately first. To do so, follow the steps listed in Step 8 of HTTPS section.

6. Select the **Configure SonicMQ Broker** tab. Enter the password in the **Broker Keystore Password** field and re-enter the password in the **Confirm Password** field. Click **Configure**.

Note .... You can enter any value you would like for the password. It is used for encryption of the certificate file.

7. Select the **Configure SonicMQ Broker** tab. Click **Import Trusted Root**. Select the trusted root certificate file from File Browser, and click **OK**.

8. To make a new backup copy of keystore, find the serverstore file in the <install:root>\runtime\servers\<your servername>\certs directory. Make a copy of that file and save it with a different name, such as xpc_serverstore_https_sonic_production. (If you chose to go through HTTPS steps first and then the SonicMQ steps, at this point, the keystore contains certificate information for both HTTPS and SonicMQ).

9. Stop and restart SonicMQ Broker Service. Using **Start | Settings | Control Panel | Services**, stop then restart SonicMQService.

Note .... Note... SonicMQService needs to be restarted in order for the newly installed certificate to take effect.

Testing Your Production Installation

To test the communication between your XPC server and MarketSite, contact your MarketSite operator and request a Ping test. When the MarketSite operator pings your XPC installation and receives a response, your installation is verified as communicating correctly.

4 Advanced Configuration

This chapter provides the steps for adding special options to a production installation of XPC on Windows NT. The steps include:

1. Configure advanced options and restart XPC. Refer to Configuring Additional Options on page 4-5.
2. Request your MarketSite operator complete a Ping test. Refer to Testing Your Production Installation on page 3-13.

Planning for Special Options

Items in this section address unique or special situations that require advance planning. These items are optional.

Adjusting Java Heap Size

XPC 4.0 requires and supports JDK 1.2.2_006 or higher without HotSpot. It does not support JDK 1.3. Sun makes the JDK 1.2.2_006 version available at http://java.sun.com/products/archive/j2se/1.2.2_006/index.html.

The recommended java heap size is 512 MB. If you choose to modify the java heap size, use the **-Xms** option for setting the minimum heap size for use by the Java VM. Set the parameter in these files:

- <install:root>/bin/ccsntservice.prop
- <install:root>/bin/launch.bat
- <install:root>/runtime/servers/defaultserver/config/startup/exec.prop

Then restart the XPC server.

Using Forward Proxy

Outbound (forward) http and https traffic headed to MarketSite can be routed through a proxy server located in the demilitarized zone (DMZ). Outbound communications through a proxy server provides support for any http/https based proxy server that follows the standard HTTP1.1 proxy connection protocol.

You enter the Proxy Host and Proxy Port settings on the **Configure XPC** window (**Start | Program | XML Portal Connector 4.0 | Configure**). The settings can be changed anytime after initial installation.

Using the Versioning Library

The Versioning Library is a set of Java classes that transforms a given xCBL document from one version to other. It is integrated and shipped with XPC 4.0. The Versioning Library Service is installed as an XPC Auxiliary Service as part of the XPC installation.

The Commerce One application listed in the first column uses the xCBL version listed in the second column.

Application	xCBL Version
BuySite 6.x	2.0
SupplyOrder 2.x	2.0
XPC 3.x	2.0
Galerie 3.x	2.0
Enterprise Buyer Desktop (EBD) 2.0 (formerly BuySite 7.0)	2.2
SupplyOrder 3.0	2.2
MarketSet Connector	3.0
XPC 4.0	3.0

You and your trading partner can exchange any of the documents listed here. The Versioning Library transforms the documents in either direction.

xCBL Document		xCBL Document
2.0 Purchase Order	↔	2.2 PurchaseOrder
2.0 PurchaseOrder Response	↔	2.2 OrderResponse
2.0 StatusResult	↔	2.2 OrderStatusResult
2.2 PurchaseOrder	↔	3.0 Order
2.2 OrderResponse	↔	3.0 OrderResponse
2.2 ChangeOrder	↔	3.0 ChangeOrder
2.2 ChangeOrderResponse	↔	3.0 OrderResponse
2.0/2.2 PriceCheck	↔	3.0 PriceCheck
2.0/2.2 PriceCheckResult	↔	3.0 PriceCheckResult
2.0/2.2 AvailabilityCheck	↔	3.0 AvailabilityCheck
2.0/2.2 AvailabilityCheckResult	↔	3.0 AvailabilityCheckResult
2.0/2.2 OrderStatus	↔	3.0 OrderStatus
2.0 OrderStatusResult	↔	3.0 OrderStatusResult
2.2 Invoice	↔	3.0 Invoice
2.2 ASN	↔	3.0 ASN

For information on disabling the Versioning Library as an Auxiliary Service, refer to Administration on page 5-1.

The Versioning Library is installed with default properties, as listed here. If you need to change these settings, refer to Troubleshooting on page 6-1 for more information.

Property	**Your Data**	**Description**	**Values and Default**
transformation.registry		Full path name of the TransformRegistry.xml file, which has meta data about transformations.	transformation.registry=c:/commerceone/xpc/schema/TransformRegistry.xml By default, the file is installed under <install:root>/schema.
transformation.internalversion		The xCBL version supported by the Application. Incoming documents are converted to this before application recieves it. For example, Buysite 7.0, EBD 2.0, and SupplyOrder 3.x support xCBL version 22 and use transformation.internalversion =22	Valid values are xCBL versions, 20|22|30. The default is 30. transformation.internalversion=30
transformation.externalversion		Lowest xCBL version supported by any Trading Partner sending you transactions.	Valid values are xCBL versions, 20|22|30. The default is 20. transformation.externalversion=30

Enabling and Disabling the Versioning Library

To enable the Versioning Library transformations, edit the file **<install:root>/runtime/servers/defaultserver/config/startup/service-start.prop**. Find the line that begins **startup.aux.services=** and add **XCBLTransformService** to that line.

To disable the Versioning Library transformations, set the configuration **transformation.internalversion** and **transformation.externalversion** to the same version as your applications.

For example, to disable transformation for SupplyOrder 3.0, set transformation.internalversion=22 and transformation.externalversion=22. Because SupplyOrder 3.0 supports xCBL version 2.2, outgoing transmissions will not be transformed. If an incoming document is not xCBL 2.2, it is transformed to xCBL 2.2.

Note that there is no transformation available from xCBL 2.2 Invoice Document to xCBL 2.0 because there is no Invoice document in xCBL 2.0. If there is no transformation available, then the document is unchanged.

Similarly, if you are testing and using XPC 4.0, set transformation.internalversion=30 and transformation.externalversion=30. This disables transformation for testing.

Configuring Additional Options

Using your worksheet, enter your advanced options using the Configure XPC user interface.

1. Select **Start | Programs | XML Portal Connector 4.0 | Configure.** The **Configure XPC** window is displayed. The name of your XPC server is shown in the middle of the title bar.
2. Review the **XPC Server** information in the first box and edit as needed. To adjust your message level, consider these debug levels:
   - 0 Debug. All messages are saved, including those intended for developers.
   - 1 Information
   - 2 Warning
   - 3 Error
   - 4 Critical
   - 5 Fatal. Only information about Fatal errors are saved.
3. Ignore the **Preconfigure Trading Partner** button. This is used by the developers.
4. The **Local Communication** box is described in Configuring XPC on page 2-7.
5. The **Portal Communication** box is described in Configuring XPC on page 2-7.
6. If you are using Forward Proxy, click the **Enable Proxy** button in the **Proxy** box. Then enter the **Host** and **Port** of your Proxy Server.
7. Review your entries. If you need to recover the original entries, click on **Reset**. To clear the entries, click on **Clear**. When you are finished, click on **Save** and then **Exit**.

5 Administration

This chapter describes the XPC administration tasks. The sections in this chapter include the following:

- Starting and Stopping the XPC Server on page 5-1
- Backing Up SonicMQ Keystore on page 5-2
- Using the XPC Invoker on page 5-4
- Working with Pending Documents on page 5-8
- Maintaining the XPC Server on page 5-9.

Starting and Stopping the XPC Server

When making any changes to the XPC installation or when maintaining the XPC server, you will need to stop and restart the server. The recommended way to start and stop the server is:

- **To stop the XPC server,** select **Start | Programs | XML Portal Connector 4.0 | Stop**. This stops both XPC and SonicMQ.
- **To start the XPC server**, select **Start | Programs | XML Portal Connector 4.0 | Start**. This starts both XPC and SonicMQ.

Running XPC as an NT Service

CCSNTService and SonicBroker service must be started to start XPC, and they can be started in any sequence. The XPC defaults with these services are set to start automatically upon system startup.

However, if the default system account does not have sufficient permissions, privileges, or resources to run the XPC, configure these services as follows.

1. To access the services control panel, select **Start | Settings | Control Panel | Services**. **SonicMQService** and **CCSNTService** are displayed on the Services panel.
2. Click on **SonicMQService** and then click **Startup** and configure according to your site requirements:
   - To prevent NT from starting XPC automatically, set **Startup Type** to **Manual** instead of Automatic.
   - To configure an account for running XPC, configure **Log On As** section. Provide a valid NT user name and password with required rights.
3. Click on **CCSNTService** and then click **Startup** and configure according to your site requirements:
   - To prevent NT from starting XPC automatically, set **Startup Type** to **Manual** instead of Automatic.
   - To configure an account for running XPC, configure **Log On As** section. Provide a valid NT user name and password with required rights.

Here is a summary of methods for starting and stopping the XPC Server.

Method	**To Start:**	**To Stop:**
Start menu	**Start | Programs | XML Portal Connector 4.0 | Start**	**Start | Programs | XML Portal Connector 4.0 | Stop**
Command line	a) **cd <install:root>\SonicMQ\bin; startbr** b) **cd <install:root>\bin; ccsserver_<SERVERNAME>**	Ctrl-C in the DOS window
From .bat file	**cd <install:root>\bin; StartService**	**cd <install:root>\bin; StopService**

Backing Up SonicMQ Keystore

Make copies of the SonicMQ keystore to avoid recovery difficulties. If your keystore should corrupt and you do not have a backup, follow the recovery instructions in Certificate Manager Keystore on page 6-11.

XPC Envelope Archive

XPC archives envelopes exchanged with MarketSite for asynchronous transactions. Both incoming and outgoing messages are stored under the <install:root>\messagestore directory:

- Inbound envelopes are archived immediately upon receipt in the server. Any transformation of the envelope, such as changing to the xCBL internal version, occurs *after* archiving. The Inbound archive thus contains the external envelope.
- Outbound envelopes issued through a timed service are archived *before* a version transformation. The Outbound archive thus contains the internal envelope.

Outbound envelopes transmitted through a stand-alone transmitter are not stored in the XPC archive.

Archived outbound and inbound envelopes are associated with one another as part of the same document exchange. Such a transaction consists of a request plus its response envelope. Two envelope header properties link the response to the request:

- Both have the same **Correlation_ID** property value.
- The response identifies the **Message_ID** property of the request in its **Reference_ID** property.

From the archive, the status of a transaction can be inferred based on the following cases:

- Pending: A request envelope with no response envelope
- Completed: A request envelope with a business response
- Error: A request envelope with an error response

XPC Invoker and the Envelope Archive

Use the XPC Invoker to view the Envelope Archive and perform the following administrative tasks as needed:

- List historical transactions for a specific document type
- Display the request and response envelopes of a transaction including header properties, primary document, and attachments.
- Display the status of the transaction, which can be Pending, Completed, or Error
- Reissue a transaction.

Using the XPC Invoker

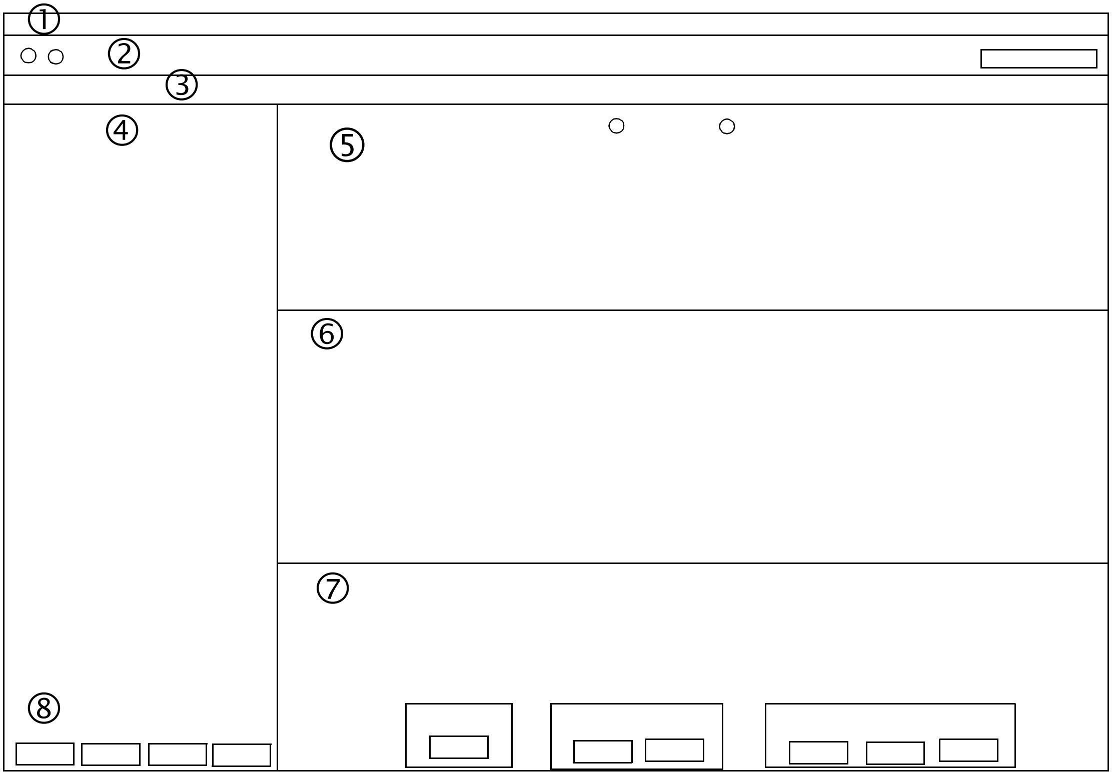
The Invoker has two modes: Test and Production. Developers use the Test mode to verify that the their changes are working correctly.

As an XPC administrator, use the Production mode to monitor and maintain a *live* XPC installation. Maintenance includes resubmitting *pending* purchase orders. Purchase orders are considered pending when no response has been received.

XPC Invoker User Interface

The XPC Invoker User Interface is composed of these sections:

1. Title bar with Commerce One logo
2. Mode selection, Archive directory, and Refresh button
3. Transaction Filter
4. Transaction Table
5. Envelope
6. Document
7. Attachments
8. Action buttons



To switch between Test and Production mode in Section 2, click on the Mode radio buttons at the top left corner of the **XPC Invoker** window:

- In **Test** mode, all functions are available.
- In **Production** mode, the Invoker cannot create new transactions, delete transactions, browse to a new archive directory, save, or edit envelopes. Only Asynchronous transactions are available for display. Here, the Invoker defaults to the XPC file system envelope archive specified in the **Archive Directory** field of the **Configure XPC** window (**Start | Programs | XML Portal Connector 4.0 | Configure**).

Transaction Filter

For ease of viewing, the Invoker filters the views of the archive by Transaction Type (Price Check, Purchase Order, Availability Check, and Order Status). You can change the view by selecting a different transaction in the drop down box containing all known transaction types in Section 3. If envelopes are found in the archive that are not part of a specified transaction type, they are placed in the type *Unknown*.

Archive Display

The Archive Display shows the currently viewed archive path and allows you to refresh the view of the archive or select a new archive. See Section 2.

In Production Mode, you cannot browse to a new directory or directly edit the directory path; only the refresh button is available. The Invoker defaults to the XPC file system envelope archive specified in the **Archive Directory** field of the **Configure XPC** window (**Start | Programs | XML Portal Connector 4.0 | Configure**).

Transaction Table

The Transaction Table, Section 4, lists all transactions located in the selected directory. A transaction consists of a request envelope and its response envelope.

This table contains two columns: Date Requested and Status:

- **Date Requested**—is the date that the envelope was sent, thus, initiating the transaction.
- **Status**—is the status of the transaction. These statuses are: **New**, **Pending**, and **Completed**.
  - **New**—a transaction has been created but has not been initiated. The request has not been sent out.
  - **Pending**—the request envelope has been queued for transmission.
  - **Completed**—a transaction has been created, the request has been sent out, and the response has been received.

The transaction buttons include:

- **New—**allows you to create a new transaction by selecting a document template. Document templates are out-of-the-box xCBL documents.

  Selecting this button brings up a file selection dialog box. When you select a file, the Invoker creates a new transaction of the appropriate type, selects the archive of the newly created transaction, and displays the newly created envelope. This envelope is not yet saved. If you want to save it, click the **Save** button.

- **Send**—sends the request envelope of a selected transaction. The request envelope is always saved on transmission.
  For synchronous transactions, the Invoker waits for a response and automatically displays the response if the transaction type is still selected.
- **Delete**—deletes the transaction selected in the archive display. The Invoker deletes both the request and response of the transaction.

Envelope

Use the Request/Response buttons in Section 5 to view either the request envelope or the response envelope for a transaction.

The envelope header properties include:

Header Property	Description
Sender ID	Identifies the sender of the envelope.
Receiver ID	Identifies the recipient of the envelope. This field can be edited in Test mode.
Message ID	The unique identifier of the envelope.
Document Type	Type of document, such as PriceCheckRequest or PriceCheckResult, carried by the envelope.
Request Mode	Mode of transmission for the envelope. Valid request modes are sync, peer-peer, and oneway.
Date Sent	Date the envelope was sent.
Correlation ID	The unique identifier used to associate a request envelope with the corresponding response envelope.

To edit the Document, select the **Edit** button. To select a new document, click the **Change** button. A file selection dialog box is displayed; select a new xCBL document from the list. If the XML file you select is not valid, an error message is displayed.

The Envelope buttons in Section 7 allow you to save, change, add, and remove attachments from the envelope.

- **Save—**allows you to save the currently displayed envelope. If the envelope has been edited, the edited envelope is saved as a new transaction and a new row appears in the transaction display. If you do not save the old envelope, it will be lost. This function is only available in Test Mode.

- **Change**—selects an xCBL document from the file system. The selected document is displayed, but a new transaction is not created until the envelope is saved or sent out. This function is only available in Test Mode. If the selected xCBL document is different from the original envelope's document type, then the new transaction and envelope are displayed in the appropriate archive upon saving or sending the edited envelope.
- **Add—**selects a file from the file system and attaches it to the envelope. This function is only available in Test Mode.
- **Remove—**removes a selected attachment from the envelope. This function is only available in Test Mode.
- **Name**—specifies a URI for an attached file. You must select a file from the attachment display and then select the **Name** button. This action launches a dialog box which takes your input for the attachment name. All attachment names must be properly formed URIs. If the attachment is already named, this name is displayed in the dialog box. This function is only available in Test Mode.

Working with Pending Documents

The Invoker correlates and displays envelopes within the XPC archive and displays the dates and status. Transactions that contain a request and response pair are considered complete and display the status of **Completed**. If a transaction contains only the sent request, the status is **Pending**. If the request has not been sent out, the status is **New**.

If a transaction has been pending for an extended period of time, the transaction is considered *stale* and the administrator can resend the request.

Because the Invoker does not contain a server, it cannot receive asynchronous responses. To accommodate this needed functionality, a special mechanism has been put in place. When XPC receives an envelope whose sender is TEST, XPC *transmits* the response to the file system. This location is specified in the **default.prop** file of the TransmitterService.

The Invoker polls the file system for asynchronous responses. The location the Invoker polls in Test mode is specified in **<install:root>\etc\config\client.prop** with the **xpc.invoker.asyncReceptionDirectory** variable. The default value for this variable is **<install:root>/asyncDir**. All the envelopes placed in this directory are considered by the Invoker as an asynchronous response of some kind and are matched with its request using the envelope’s correlation ID. For asynchronous reception of documents, the Invoker must look in the same directory as the TransmitterService transmitted to.

In Production mode, the Invoker polls the archive of the XPC-asynchronous requests and responses to the file system. The **xpc.invoker.production.archiveDirectory** of the Invoker must be set to the same value in the **default.prop** file of the MessageStoreService. The default value for both is **<install:root>\messagestore**. For all asynchronous services, archiving must be turned on.

Maintaining the XPC Server

The tasks for maintaining the XPC Server include clearing the:

- XPC log files
- SonicMQ files
- Archive directory.

Clearing the XPC Log Files

The logs are created in the directory:

```
<Install:root>/runtime/servers/<SERVERNAME>/logs
```

where SERVERNAME is the name of the XPC server you assigned during installation, such as defaultserver.

When you stop the XPC Server for maintenance, delete the log files or archive for future use. How often you need to the clean logs directory depends on high low you set the Debug level. The lower the Debug level, the larger the files become. To reset the Debug level, select **Start | Programs | XML Portal Connector 4.0 | Configure**. On the **Configure XPC** window, change the first field, the **Debug Level** field.

If you delete unused files while the server is running, you cannot delete the file currently being used. If you attempt to do so, you will get an error message stating that the file is locked and cannot be deleted.

Clearing the Archive Directory

Clear the archive directory periodically, as the traffic requires. Before moving files from the archives, make sure they are not a pending transaction. To do so, access the Invoker in Production mode and check the status.

Checking Security Certificate for Expiration

All certificates have a valid period. Be sure to view the **Configure Server Entries** tab of the Certificate Manager tool to determine the expiration date of your original certificate entry. Note this date in a reminder file.

***Caution*.....** **Before a certificate expires, you must replace it with a valid certificate.** Replace the expiring certificate with a valid certificate 14 days before the expiration date of the certificate. If you do not, XPC will stop working.

To replace an expiring certificate with a valid certificate, follow the steps as you did initially; refer to Authentication for Production Using https on page 3-3 or Authentication for Production Using SonicMQ on page 3-9. That is:

- For https, create a new server entry, create and submit a certificate request,and import the certificate.
- For SonicMQ, create a new server entry, create and submit a certificate request, and import the certificate.

Note .... After you import the new certificate, be sure to stop and start the SonicMQ Broker using **Start | Settings | Control Panel | Services**.

6 Troubleshooting

This chapter describes the most common problems with XPC and how to correct them.

If you have purchased XPC software directly from Commerce One, Commerce One Technical Support is available to you by contacting one of the following:

- csa@commerceone.com
- By fax at 925-520-6060
- Hotline at 925-520-5959 or 800-949-8939
- http://www.commercecone.com/solution/tech.htm

If you purchased XPC software from a different source, such as your GMP or Systems Integration Partner, contact that source for technical support.

This chapter includes the following sections:

- XPC Logging on page 6-2
- Reading Debug Messages on page 6-2
- XPC Invoker Connection Refused on page 6-6
- Certificate Manager Keystore on page 6-11
- Troubleshooting the Versioning Library Service on page 6-7

XPC Logging

XPC has three key logs: System Startup, Event, and Invoker. If you experience problems with starting XPC, sending events, or using the XPC Invoker, check the XPC log files.

The following table lists the logs with their file names and descriptions.

Log	**File Name**	**Description**
System Startup	runtime\servers\defaultserver\logs\systemStartupLog_*timestamp*	Contains information describing the startup of XPC. If a particular service is not starting up, inspect this file.
Event	runtime\servers\defaultserver\logs\eventlog_*timestamp*	Contains event logs generated by XPC or specific components. If a component throws an exception, the XPC default exception handler will add a new entry in this file.
Invoker	bin\clienteventlog	Contains information about the Invoker. If the Invoker has errors, they are stored in this file.
Debug	runtime\servers\defaultserver\logs\debut-default	This file has debug trace messages.

Reading Debug Messages

Developers insert debug messages into the code for both their own use and yours. These messages can be filtered by severity to give useful information about the server.

Debug message syntax is simply a string that can contain substitution variables provided at runtime. Debug messages are in English only and can have misspellings or grammatical errors. Each message is placed in the file on a separate line. Here is an example debug message from the debug log:

```
April 18, 2000 5:17:00 PM GMT-07:00: Debug.DEBUG: Configuration is:
localhost=jmiller.commerceone.com archive=archive user=xccarchiver
```

Breaking up the message, it reads as follows:

April 18, 2000 5:17:00 PM GMT-07:00:	Time when message is generated in system local time.
Debug.DEBUG:	Label that lists the debugging level of the message. The lower the number, the more messages are displayed: 0 Debug.DEBUG 1 Debug.INFO 2 Debug.WARNING 3 Debug.ERROR 4 Debug.CRITICAL 5 Debug.FATAL If the debugging level is set to 0, all messages are seen. If set to 2, only warnings messages and higher-level messages are seen.
Configuration is: localhost=dbmachine.commerceone.com archive=archive user=xccarchiver	Text of the debug message, with parameters filled in. For example, the previous message includes the configuration information of the machine name where the message store archive is, the name of the archive, and the user name.

Tip: To quickly review a log, search for the words WARNING or ERROR. Expect to find CRITICAL and FATAL errors at the end of the file, when the server probably stopped or crashed.

Reading Event Output

Events are formatted into XML syntax, which is defined in event catalogs. Events can be localized into different languages. Each Event message is logged onto a separate line of the file. Here is an example event item from an Event Log.

```
<EVENT><TIME>November 9, 1999 3:49:33 PM GMT-08:00</TIME>
<MILLIS>942191373381</MILLIS>
<KEY>CCS_COMM_TRANSMITTER_3041_TransmitterCreationSuccessful</KEY>
<TEXT>Transmitter creation successful.Transmitter Type = http.</TEXT>
<CAT>CCS_COMM</CAT><SUBCAT>TRANSMITTER</SUBCAT><NUMID>3041</NUMID>
<SEV>0</SEV><TYPE>STATUS</TYPE><LANG>en</LANG><PARM>http</PARM></EVENT>
```

In addition to the syntax for the event from the catalog, the example message includes:

- <TIME>, a date/time stamp in local time
- <MILLIS>, the date/time in milliseconds
- <LANG>, the language the text is generated in
- <PARM>, a single parameter

Here is the example message, explained item by item.

Item	**Explanation**
<EVENT>	Beginning of the event.
<TIME>November 9, 1999 3:49:33 PM GMT-08:00</TIME>	Date/time stamp in local time when event was generated
<MILLIS>942191373381</MILLIS>	Date/time in milliseconds when event was generated
<KEY>CCS_COMM_TRANSMITTER_3041_TransmitterCreationSuccessful</KEY>	The *event key* is essentially the name of the event. Note that an event number is embedded in the name of the event.
<TEXT>Transmitter creation successful. Transmitter Type = http.</TEXT>	The event message with parameter values. In this example, the 'Type = +1' has become 'Type = http' due to parameter substitution. A second parameter would be noted by a +2, a third by +3, and so on.

Item	Explanation
<CAT>CCS_COMM</CAT>	Category of the event. In this example this is a communications event. The event name begins with this value.
<SUBCAT>TRANSMITTER</SUBCAT>	Subcategory of the event. This event is from the transmitter subcategory, so was probably generated from the Transmitter service. This is also part of the event name.
<NUMID>3041</NUMID>	Event number that was embedded in the event name.
<SEV>0</SEV>	Severity of the event. A severity of 0 means this is a normal event, showing normal operation.
<TYPE>STATUS</TYPE>	Type of event. This is a status event, used to provide information on the operation of the server. Other types are AUDIT or DEBUG..
<LANG>en</LANG>	Language used to generate this event. Event catalogs are intended for localization and this value should match the language of <TEXT>.
<PARM>http</PARM>	Parameter passed to <TEXT>. This can be repeated for additional parameters.
</EVENT>	End of the event.

Avoiding Known Problems

Review this table to ensure that you have completed all known installation and administration tasks required for keeping your XPC installation running smoothly.

	Description	Solution or Explanation
XPC Startup fails	If you have not configured otherwise, NT uses default system account, which might not have proper permission and resources.	Refer to Running XPC as an NT Service on page 5-1 for instructions to set the Log On As properties.
classpath directory	The **default** file in the **etc\classpath** directory requires a carriage return or a blank space at the end of the last line.	If you do not use the carriage return or a blank space, the line is not read.

XPC fails after months of running successfully	The Authentication Certificate has expired.	Refer to Checking Security Certificate for Expiration on page 5-10.
Envelope Size		Envelopes have a 10 MB size limit including attachments.
Installation	You cannot install to a directory that contains a blank space in the name (for example, File Name).	It is recommended that you install XPC to the **<drive>commerceone\xpc** directory on NT.
Invoker Error Message	If you select invalid xCBL documents, an error message is displayed.	
Invoker Response Attachments	Response envelopes have an attachment identified by message ID. This attachment is the security credential information provided in the reply.	
Java Heap Size	An out of memory error occurs when the heap size is not large enough. The recommended java heap size is 512MB.	Refer to Adjusting Java Heap Size on page 4-1.
SonicMQ/XPC Installation stops	An error message such as *Unable to open broker.key file* or *Unable to open Password file* is displayed.	Ensure that the JDK bin directory is in the PATH environment variable before you begin installation.

XPC Invoker Connection Refused

An error message like the one below indicates that the Invoker is not authorized to communicate with XPC. Verify that the Invoker's security properties have been properly configured.

```
<?soxtype urn:x-
commerceone:document:com:commerceone:ccs:doclet:error:Error.sox$1.0?>
<?import urn:x-
commerceone:document:com:commerceone:ccs:doclet:error:Error.sox$1.0?>
<Error><Code>INVOKER_ERRORGENERIC</Code>
<DefaultMessage><Message><MessageString>ERROR:
com.commerceone.xdk.excp.metadox.send.TransferException: Error while
silently connecting: org.w3c.www.protocol.http.HttpException:
java.net.ConnectException: Connection refused</MessageString>
```

```
<LanguageCode>en</LanguageCode>
</Message>
</DefaultMessage>
<Severity>Error</Severity>
</Error>
```

Troubleshooting the Versioning Library Service

Versioning Library parameters are stored in:

```
<install:root>\runtime\servers\<servername>\config\service\AuxiliaryService
.XCBLTransformService.XCBLTransformService\default.prop
```

Verifying Versioning Library

To verify that VL is running on XPC server, follow these steps.

1. Access the directory:
`<install:root>/runtime/servers/<servername>/config/startup`

where <install:root> is the directory where XPC is installed and <servername> is the name of the XPC server.

2. Locate the file named **service-start.prop** and open it in Notepad or any text editor.

3. Verify that the entries shown in **bold** are present.

```
startup.aux.services=EventManager,LogService,XCBLTransformService
```

4. Verify that the following entry is present. It starts the xCBLTransformService.

```
service.XCBLTransformService.code=com.commerceone.versiongateway.service.XC
BLTransformService
```

Versioning Library Transformations

The Versioning Library transforms to the closest possible version. For example, if requested to transform an xCBL 3.0 Invoice to xCBL 2.0, it will transform it to xCBL 2.2, because Invoice is not defined in xCBL 2.0.

Note that the Versioning Library will not transform the envelope/document if:

- There is no valid transformation available, such as from INVOICE document from xCBL 22 to xCBL 20 since there is no INVOICE document defined in xCBL 20.

- The document being passed to Versioning Library is not xCBL document.
- The document is not in one of the following namespaces:

```
urn:x-commerceone:document:com:commerceone:CBL01:CBL01.sox$1.0
urn:x-commerceone:document:com:commerceone:CBL:CBL.sox$1.0
urn:x-commerceone:document:com:commerceone:XCBL30:XCBL30.sox$1.0
```

To verify that the transformations are being made as intended, test the transformation using the Transformation Test tools shipped with XPC, as described in Testing Versioning Library Transformations on page 6-8. If there is no transformation available, transform.bat tool displays the exception NoTransformationFound.

Testing Versioning Library Transformations

Use the Transformation Test tools shipped with XPC to confirm that your transformations work as you expect. The document transformation tool, transform.bat, transforms xCBL documents. The envelope transformation tool, transformEnv.bat, transforms xCBL documents inside envelopes.

The two tools run as command-line applications from the XPC server. You can use these tools to test the Versioning Library without running the XPC Server and using the XPC Invoker.

Document Transformation Tool

This tool, transform.bat, transforms an xCBL document from one version to another. To execute:

1. Change to the XPC bin directory, for example:

```
cd <install:root>\bin
```

2. Execute the tool using the following format:

```
transform.bat <destination CBL version> <CBL document1> [<CBL document2> …]
```

where the destination xCBL version is the target xCBL version; supported values are 20, 22, and 30. To transform more than one CBL document to the destination version, specify up to seven file names, separating the names with spaces. All transformed files are placed in the <install:root>\out directory with a prefix of o_ on the original name. For example:

```
transform.bat 30 c:\temp\PriceCheckResult.xml
```

transforms the document PriceCheckResult.xml to xCBL version 3.0. If PriceCheckResult.xml is already in version 3.0, no transformation is done. The transformed file will be in <install:root>\out\o_PriceCheckResult.xml.

To edit transform.bat to add or delete file names or change the default output directory, as described in Customizing Transform.bat on page 6-9.

Customizing Transform.bat

Transform.bat uses XPC launch.bat to set the classpath, then launches the jvm to start the transformation-testing tool class. To customize the Document Transformation Tool, edit transform.bat, which is in this format:

```
call launch -Dout.directory=%XPCROOT%\out -Dschema=%XPCROOT%\schema
com.commerceone.versiongateway.tools.transformation.Transform %1 %2 %3 %4
%5 %6 %7 %8
```

where:

out.directory specifies the output directory of the transformed documents.

schema specifies the directory where the sox schema for xCBL 2.0, xCBL 2.2, xCBL 3.0, and the Versioning Library is located. The schema directory also stores the TransformRegistry.xml file.

%1 %2 … specifies the standard input to the class file. %1 is the destination version. %2 is the first xCBL document to transform. %1 and %2 are mandatory. %3 and higher are additional, optional xCBL documents.

Envelope Transformation Tool

The Envelope Transformation Tool, transformEnv.bat, transforms xCBL documents that are inside envelopes. This tool calls the transformIn API when receiving documents and the transformOut API when sending out documents. You can use transformEnv.bat to:

- Create an envelope
- Transform an envelope in the inbound direction
- Transform an envelope in the outbound direction.

To run the Envelope Transformation Tool, follow these steps.

1. To execute, you must be in the XPC bin directory, for example:

```
cd <install:root>\bin
```

2. Enter the name of the file and options in the format shown here. The options are defined in the following table.

```
transformEnv.bat [-options]
```

Option	Description
sender	Sender for envelope creation. Default is Sender.
receiver	Receiver for envelope creation. Default is Receiver.
destination	Destination of output.
p	Schema path.
create <CBL document>	Create an envelope for the CBL document specified.
i <envelope>	Transform the envelope in the inbound direction in the version specified by -j.
o <envelope>	Transform the envelope in the outbound direction in the version specified by -k.
j <internal version>	Specify the internal version. Supported values are 20, 22, 30. If not specified, defaults to 22.
k <external version>	Specify the external version. Supported values are 20, 22, 30. If not specified, defaults to 20.
?	Command syntax

In the following examples, all use the defaults for sender and receiver, which are Sender and Receiver.

In this example, the specified PurchaseOrder is in the directory d:/commerceone/xpc/out with a file name of o_PurchaseOrder.xml.txt:

```
TransformEnv -p d:/commerceone/xpc/schema -c d:/temp/
PurchaseOrder.xml -d d:/commerceone/xpc/out
```

The next example uses the Versioning API transformIn to transform the envelope d:/temp/PurchaseOrder.txt into the internal xCBL version 2.2.

```
TransformEnv -p d:/commerceone/xpc/schema -i d:/temp/
PurchaseOrder.txt -d d:/commerceone/xpc/out -j 22
```

The API will try to extract the xCBL document version 22 from the main document or from the list of attachments. If it does not exist, the API will then try to transform the document to 2.2. The output will be a new envelope with an xCBL 2.2 document as the main document. The new file is named o_PurchaseOrder.txt.

The last example uses the transformOut API to transform the envelope d:/temp/PurchaseOrderResponse.txt to xCBL version 2.0. All intermediate transformations are stored as attachments. The output will be an envelope with the new transformed xCBL 2.0 document stored as the main document and the original xCBL 2.2 document stored as an attachment. The new file is named o_ PurchaseOrderResponse.txt.

```
TransformEnv -p d:/commerceone/xpc/schema -o d:/temp/
PurchaseOrderResponse.txt -d d:/commerceone/xpc/out -k 20
```

Certificate Manager Keystore

If your keystore file becomes corrupt and you have a backup file for your keystore file, simply replace the corrupt file with your backup file.

If you do not have a keystore backup file, follow these instructions to create a new keystore.

1. Access the XPC bin directory, for example, <install:root>\bin.
2. From the bin directory, start the script **CertMgrCreateNewKeyStore.bat.** A **Certificate Manager** dialog is displayed, showing a **Create a new KeyStore** button.
3. Enter the password for the new keystore and click on the **Create a new KeyStore** button. The **Warning** dialog displays the message: New Keystore was successfully created. But there is no server entries, please create a server entry first.
4. Click on the **OK** button. The **Certificate Manager** window is displayed with the **Configure Server Entries** tab selected.
5. From this point, complete the authentication for https or SonicMQ as you did initially; refer to Authentication for Production Using https on page 3-3 or Authentication for Production Using SonicMQ on page 3-9. That is:
   - For https, create a new server entry, create and submit a certificate request,

import the certificate, and set the entry as active.

- For SonicMQ, create a new server entry, create and submit a certificate request, import the certificate, and configure the SonicMQ Broker.

A MarketSite 4.0 Core Events

This appendix lists events logged by the MarketSite platform. For events related to the SonicMQ Broker, refer to SonicMQ Broker Events on page -1.

The **Description** and **Action** columns suggest an explanation or resolution for the event. Numerical symbols, such as +1, indicate parameters that will be inserted into the message text.

The Severity (**Sev**) column refers to the level of severity:

0	Success
10	Informational
20	Warning
30	Diagnostic
40	Error

#	Event	Description	Sev	Action
1	CCS_ADM_1000_LogMgrErr	Could not instantiate LogManager object	40	Examine the error message included with the event. Usually this is a result of being unable to open the log file for logging event messages.

2	CCS_ADM_1005_BadDocErr	+1 : got bad document: +2	40	Generic error message for a bad document. Examine the error for the action to take.
3	CCS_ADM_1010_ProcRunning	A process named [+1] is already running: +2	40	The daemon server will not be able to start a new server, because a process is already running. Correct by killing the process and restarting the daemon server.
4	CCS_ADM_1015_ProcTblFull	The process table is full. Can't run process [+1]: +2	40	Stop the daemon server and kill all of the associated java processes. This should not happen under normal circumstances.
5	CCS_ADM_1016_ProcNotManaged	The server [+1] is not managed by this management server	40	Stop the daemon server and kill all of the associated java processes. This should not happen under normal circumstances.
6	CCS_ADM_1017_ProcStartTimeout	Timeout while trying to start server [+1]	40	Timeout occurred during the start of the server. Stop the daemon, remove all java processes, and try again.
7	CCS_ADM_1020_HttpdFatal	*** [httpd]: fatal error, exiting !	40	Unused event
8	CCS_ADM_1025_HttpdInitErr	Couldn't initialize HTTP server: +1	40	Could not initialize the server. Fatal error, stop the daemon, remove all java processes, and try again.
9	CCS_ADM_1030_ProcIntr	Interrupted in Process.waitFor (): +1	40	Interrupt was sent to the server while being managed by the daemon. No corrective action possible.
10	CCS_ADM_1035_ProcDbgIntr	Process debugger thread interrupted: +1	40	Debugging event should not be seen in the field.
11	CCS_ADM_1040_ProcDbgIsNull	The debugged Process instance is null. Can't debug	40	Debugging event should not be seen in the field.

12	CCS_ADM_1045_ProcDbgIOErr	Caught IOException while debugging subprocess [+1]: +2	40	IOException occurred during debugging. Not an event that should be seen in the field.
13	CCS_ADM_1050_ProcStartIOErr	Caught IOException while trying to start subprocess [+1]: +2	40	IOException occurred. Examine event message for root cause and correct.
14	CCS_ADM_1055_ProcStartSecurityErr	Caught SecurityException while trying to start subprocess [+1]: +2	40	Security Error occurred when trying to start the server from the daemon. Daemon process may not have sufficient permissions to run sub processes. May need to modify security on system so that daemon can run with sufficient permissions.
15	CCS_ADM_1060_ProcUnkillableErr	Couldn't kill process [+1]: +2	40	Error occurred while trying to kill a server through the daemon. May need to kill the server directly through the task manager or process manager.
16	CCS_ADM_1065_McastInstErr	Couldn't instantiate IP multicast service: +1	40	Error occurred during a multicast daemon discovery in the Admin Console. No longer used
17	CCS_ADM_1070_McastAddrErr	Couldn't create IP Multicast socket: +1	40	Error occurred during a multicast daemon discovery in the Admin Console. No longer used
18	CCS_ADM_1075_McastIOErr	Couldn't send document to IP Multicast address [+1]: +2	40	Error occurred during a multicast daemon discovery in the Admin Console. No longer used.
19	CCS_ADM_1080_XMLConvErr	Couldn't convert XML string [+ 1] to a document bean: +2	40	Unused event
20	CCS_ADM_1085_XMLStreamErr	Couldn't write XML string to output stream: +1	40	Unused event

21	CCS_ADM_1090_URNCatLoadErr	Syntax error in urncatalog: + 1	40	Unused event
22	CCS_ADM_1095_CouldntLoadProps	Unable to load properties from file [+ 1]. Exception text: +2	40	Error message in event should suggest the needed remedy. For example, if the property file was not readable, then it is possible that the permissions on the file or directory may need to be changed.
23	CCS_ADM_1100_RemoteCmdErr	Caught Exception while retrieving result from remote command: + 1	40	Unused event.
24	CCS_ADM_1105_MalformedURLErr	Malformed URL [+1]: +2	40	Unused event.
25	CCS_ADM_1110_SerializableErr	Couldn't de-serialize +1 object: +2	40	Unused event.
26 (*)	CCS_ADM_1115_NoServerSelected	No server selected	40	
27 (*)	CCS_ADM_1120_NoServiceSelected	No service selected	40	
28	CCS_ADM_1125_StartSvcErr	WARNING! Couldn't START/ STOP service [service name]	40	
29 (*)	CCS_ADM_1130_StartSrvErr	WARNING! Couldn't START/ STOP server [+1]	40	
30	CCS_ADM_1135_NoServerInfoErr	WARNING! No configuration information available for servers. Can't start any servers!	40	Check the server config directory path
31 (*)	CCS_ARCHIVE_STORE_100_Test	A sample archive store - event error message +1 +2.	40	Not used
32	CCS_ARCHIVE_STORE_1000_ArchiveInitialized	Document archiving initialized to: <Machinename>, storage file location: <StorageLoc>	40	Event indicates that the archive has been initialized normally. Parameters are the database name and the machine name where the database exists.

33	CCS_ARCHIVE_STORE_1010_InitializationFailed	Could not open data storage: <Machinename>, <DBName>, <DBUser>, <DBPassword>, <ErrorMsg>	40	Parameters include the host name, database name, user name, an empty string instead of the password, and the message from the exception that was thrown when initialization failed. Verify that the connection can be made with these values using a database tool like SQL Query analyzer
34	CCS_ARCHIVE_STORE_1020_ErrorInDocumentRetrieval	Error in document <DocId> retrieval: <ErrorMsg>	40	Verify from a SQL client if the document with the given id can be fetched
35	CCS_ARCHIVE_STORE_1030_ErrorInDocumentStorage	Error in document <DocId> storage: <ErrorMsg>	40	Verify from a SQL client if the document with the given id can be fetched
36 (*)	CCS_ARCHIVE_STORE_1040_CanNotCreateDirectory	Could not create directory <Directory>	40	**DEPRECATED –** Signals that the directory needed to store the documents could not be created. This is no longer needed in post 3.1 MarketSite
37 (*)	CCS_UTL_QUEUE_8010_QueueExistsInfo	Queue: +1 : already exists.	10	
38 (*)	CCS_UTL_QUEUE_8015_IllegalQueueNameError	Illegal Queue Name: +1 .	40	
39 (*)	CCS_UTL_QUEUE_8020_QueueCreationError	Queue: +1 : creation error, +2	40	
40 (*)	CCS_UTL_QUEUE_8025_QueueDeletionError	Queue: +1 : deletion error, +2	40	
41 (*)	CCS_UTL_QUEUE_8026_QueueReceiveError	Queue: +1 : receive error, +2	40	
42 (*)	CCS_UTL_QUEUE_8027_QueueSendError	Queue: +1 : send error, +2	40	

43 (*)	CCS_UTL_QUEUE_8028_QueueReplyError	Cannot send reply message as no reply queue info. was supplied	40	
44 (*)	CCS_UTL_QUEUE_8029_QueueCloseError	Queue: +1 : close error, +2	40	
45	CCS_BLOX_SVC_2100_ServiceStarted	Service <fully qualified service name> started.	0	
46	CCS_BLOX_SVC_2101_ServiceStopped	Service <fully qualified service name> stopped.	0	
47	CCS_BLOX_SVC_2102_ServiceInitialized	Service <fully qualified service name> initialized.	0	
48	CCS_BLOX_SVC_2103_ServiceDestroyed	Service <fully qualified service name> destroyed.	0	
49	CCS_BLOX_SVC_2104_InvalidQueueConfiguration	Invalid service configuration: <service name: cannot specify a queue name for a memory queue> <service name: queue owner has to be service> <service name: queue type must be either sonic or memory> <service name: memory queue cannot have SonicThreadedEnvelopeQueue as queue code> <service name: sonic queue has to have sonic threaded envelope queue as queue code>	40	Check the service queue configuration based on the event message generated
50	CCS_BLOX_SVC_2105_UseDefaultQueueConfiguration	Use Default Configuration: <use default queue code|type|owner>	10	
51	CCS_BLOX_DOC_2110_DocumentReplyPublished	Document Reply successfully published.	0	No action
52	CCS_BLOX_DOC_2111_DocumentSuccessfullyReceived	Document successfully received (Abstract Document Service).	0	No action

53	CCS_BLOX_DOC_2112_DocumentReceiveFailed	Failed to handle document: <CCSEnvelopeEventContext>	40	Check the corresponding ID in the document and the LDAP entry
54	CCS_BLOX_PROT_2113_StartingDocumentSecurityCheck	Starting document security check.	10	No action
55	CCS_BLOX_PROT_2114_BadProtectionRef	Service had bad protection reference.	40	Verify that the service is in the right category.
56	CCS_BLOX_PROT_2115_ServiceAuthorizationFailed	Service authorization failed.	30	Check the userId and password for the service in the corresponding property file
57	CCS_BLOX_COMM_2116_ClassNotFound	Failed to find <CommAgentImpl> class	40	Check the corresponding jar files is in the folder and the jar file name is in the classpath in the script.
58	CCS_BLOX_MGR_2117_FactoryCreationFailure	Failed to instantiate factory <Comm Agent factory>, <class name>, <ClassNotFoundException >	40	Verify the service name is correct in the service-start.prop
59	CCS_BLOX_MGR_2118_RouterCreationFailure	Failed to instantiate router <class name>, <ClassNotFoundException >	40	Verify the jar files are in the right folder and classpath.
60	CCS_BLOX_MGR_2119_ServiceInitFailure	Service initialization failed: <FileNotFoundException>	40	Service init failed, check the service name is correct in the service-start.prop and the corresponding jar file is in right folder and the classpath is correct.
61	CCS_BLOX_MGR_2120_NoServiceClass	No class specified for service <service_name>	40	Verify the service name is correct in the service-start.prop and the corresponding jar file is in the right folder and the classpath set correctly

62	CCS_BLOX_MGR_2121_ServiceClassNotFound	Failed to find class <class_name> for service <service name>	40	Check the service name is correct and the corresponding jar files is in the folder and in the classpath.
63	CCS_BLOX_MGR_2122_ServiceInitFailed	Initialization of service <service_name> failed	40	Service init failed, check the corresponding requirement for the service, such as the required queue name or file existing or not, etc. and other related exception.
64	CCS_BLOX_MGR_2123_ServiceCreationFailed	Creation of service < service_name> failed, <class name>, <exception>	40	Service creation failed. Check for correct service name, the corresponding requirement for the service, such as the required queue name or file existing or not, other related exceptions and so on.
65	CCS_BLOX_MGR_2124_FileNotFound	<service startup properties> file < service startup file> not found	40	Check that the service-start.prop files are in the right folder, or the root path is correct.
66	CCS_BLOX_MGR_2125_FileNotLoaded	Failed to load <service startup properties> file < service startup file>: <IOException>	40	Files could not be found or open. Check if the service-start.prop files exist or not.
67	CCS_BLOX_DOC_2126_ClassNotFound	Failed to find < Envelope> class <com.commerceone.xdk.swi.metadox.meta.Envelope>	40	Check if required jar files are in the right folder.
68	CCS_BLOX_SVC_2127_TimeoutFailed	Failed to handle timeout: <Exception>	40	Timout happened, check the network connection and other exception.
69	CCS_BLOX_DOC_2128_NoHandlerMethod	No service method to handle document	40	Check the doc corresponding app service is in the service-start.prop
70	CCS_BLOX_DOC_2129_ReplyToOnewayRequestIgnored	Reply to one-way request ignored.	10	Cannot deliver a response to ONEWAY document; check the document sending mode.

71	CCS_BLOX_MGR_2130_SvcMgrInitFailed	Service Manager initialization failed, exiting....	40	Shut down the server, check configuration and restart the server
72	CCS_BLOX_PROT_2200_ServiceAuthorizationSuccess	Service authorization success.	0	No action
73	CCS_BLOX_SVC_2210_CanNotGenerateError	Can not generate Error	40	Check the senderID of the original document.
74	CCS_BLOX_SERVER_10010_ServerNameNotSet	Server Name not set	40	Check and Set the Server Name in the server-start.prop
75	CCS_BLOX_SERVER_10020_ServerIDNotSet	Server ID not set	40	Check and Set the Server ID in the server-start.prop
76	CCS_BLOX_SERVER_10030_ServerTypeNotSet	Server Type not set	40	
77	CCS_BLOX_SERVER_10040_ServerGroupNotSet	Server Group not set	40	
78	CCS_BLOX_SERVER_10035_ServerModeInvalid	Server Mode is Invalid	40	Make sure that ccs.server.mode in the server-start.prop is either service_loopback or real.
79	CCS_BLOX_SERVER_10000_ServerInitFailed	Server Initialization Failed!!	40	Not used!
80	CCS_BLOX_SVC_10040_ServerNameIDTypeSet	Server Name, ID, Type set. Server Name : <ServerName> Server ID : <ServerId> Server Type : <ServerType>	40	No action
81	CCS_BLOX_SERVER_10050_ServerSonicSyncResponseQueueNotSet	Server Sonic MQ Sync Response Queue Name is not set.	10	No action
82	CCS_BLOX_SERVER_10050_ServerSonicEnabledAndSonicSyncResponseQueueNotSet	Server is Sonic Enabled and the Server Sonic MQ Sync Response Queue Name is not set.	40	Make sure that the property ccs.server.sonicmq.syncresponsequeue.name in the server_start.prop has the sync response queue name.
83	CCS_BLOX_SERVER_10060_ServerSonicEnabledKeyNotSet	Server Sonic MQ Enabled Key is not set. Setting it to false.	10	No action

84	CCS_BLOX_SERVER_10061_ServerSonicEnabledFalse	Server Sonic MQ Enabled Key is set to FALSE.	0	No action
85	CCS_BLOX_SERVER_10062_ServerSonicEnabledTrue	Server Sonic MQ Enabled Key is set to TRUE.	0	No action
86	CCS_BLOX_SERVER_10063_Caught_DocumentExchangeException	A DocumentExchangeException is caught by ADS. <Error Message>. The message is rolled back."	40	
87	CCS_BLOX_SERVER_10064_Caught_Exception	An exception is caught by the ADS. <Error Message>. The message is rolled back.	40	
88	CCS_COMM_AGENT_3001_AgentInitialized	Comm Agent initialized.	0	No action
89	CCS_COMM_AGENT_3002_AgentStarted	Comm Agent started.	0	No action
90	CCS_COMM_AGENT_3003_AgentStopped	Comm Agent stopped.	0	No action
91	CCS_COMM_AGENT_3004_AgentDestroyed	Comm Agent destroyed.	0	No action
92	CCS_COMM_AGENT_3005_ReplyFailure	Failed to send reply: +1	30	Not used
93	CCS_COMM_AGENT_3006_SecondaryReply	Secondary reply received; ignoring.	30	Not used
94	CCS_COMM_AGENT_3007_NotAReply	Not a reply document; ignoring.	30	Something failed while creating the reply for a synchronous request. The request needs to be sent again.
95	CCS_COMM_AGENT_3008_NO_MESSAGE_STORE_FOUND	No message store found for archive: service type <type>, name <name>	30	If archiving is desired, then a MessageStoreService must be configured for this server.
96	CCS_COMM_AGENT_3009_MESSAGE_STORE_FOUND	Message store found for archive	0	No action

97	CCS_COMM_SERVLET_3011_HttpdInitFail	Failed to initialize http daemon.	40	This is a critical error. If the daemon doesn’t start, the server will not start. Try starting it again, manually. If it fails, call customer support.
98	CCS_COMM_SERVLET_3012_ReplyTimedOut	No reply document generated before timeout, envelope id = <Envelope Id>	40	The reply was not generated before timeout for this sync request specified by the envelope id. The request needs to be sent again.
99	CCS_COMM_JMS_3013_ReplyTimedOut	No reply document generated before timeout, envelope id = <envelopeId>.	40	
100	CCS_COMM_JMS_3014_DocumentExchangeError	Document Exchange Error in the JMS CommService. <exception>.	40	
101	CCS_COMM_SERVLET_3015_DocumentExchangeError	Document Exchange Error in the CommHttpServlet. <exception>	40	
102	CCS_COMM_SERVLET_3016_GeneralError	CommHttpServlet Error. <exception>.	40	
103	CCS_COMM_SERVLET_3017_FailedToStoreMessage	Failed to archive message as requested	40	
104	CCS_COMM_SERVLET_3018_RequestTimedout	Request timed out	20	
105	CCS_COMM_SERVLET_3019_IOError	IO Related Error	40	
106	CCS_COMM_SERVLET_3020_MessageTooBig	Message too big. Size = <size>, Limit = <limit>.	40	
107	CCS_COMM_TRANSMITTER_3040_TransmitterCreationFailed	Transmitter creation failed.Transmitter Type = <protocol, exception>	40	
108	CCS_COMM_TRANSMITTER_3041_TransmitterCreationSuccessful	Transmitter creation successful.Transmitter Type = <protocol>	0	

109	CCS_COMM_TRANSMITTER_ 3042_ResourceCreationFailed	Resource creation failed in <serviceName> method = <method>	40	
110	CCS_COMM_TRANSMITTER_ 3043_TransmitterPropertyError	Transmitter Property Error. Resource creation failed in <serviceName> method = <method>	40	
111	CCS_COMM_TRANSMITTER_ 3044_NoTradingPartnerFound	No Trading Partner found. Service Name : <serviceName>. Method : <method>. Trading Partner Not found : <exception>	40	
112	CCS_COMM_TRANSMITTER_ 3050_DocumentListenerCreation Failed	Transmitter creation failed. Transmitter Type = +1. (TYPE IS NOT SPECIFIED IN CODE)	40	
113	CCS_COMM_TRANSMITTER_ 3051_DocumentListenerCreation Successful	Transmitter creation successful. Transmitter Type = +1. **(NOT USED ENYWHERE)**	0	
114	CCS_COMM_TRANSMITTER_ 3060_DocumentResponderCreatio nFailed	Transmitter creation failed. Transmitter Type = +1. (TYPE IS NOT SPECIFIED IN CODE)	40	
115	CCS_COMM_TRANSMITTER_ 3061_DocumentResponderCreatio nSuccessful	Transmitter creation successful. Transmitter Type = +1. **(NOT USED ENYWHERE)**	0	
116	CCS_COMM_TRANSMITTER_ 3070_DocumentServantCreationF ailed	Transmitter creation failed. Transmitter Type = +1. (NOT USED ENYWHERE)	40	
117	CCS_COMM_TRANSMITTER_ 3071_DocumentServantCreationS uccessful	Transmitter creation successful. Transmitter Type = +1. **(NOT USED ENYWHERE)**	0	
118	CCS_COMM_TRANSMITTER_ 3080_TransmitterCachingState	Transmitter Caching State : <cache value>	10	

119	CCS_COMM_TRANSMITTER_3081_TransmitterCachingRefreshRate	Transmitter Caching RefreshRate : <refreshRate>.	10	
120	CCS_COMM_TRANSMITTER_3090_TransmitterInfoInitializationError	TransmitterInfo initialization error	40	
121	CCS_COMM_TRANSMITTER_3091_TransmitterCacheSuccessful	Using cached Transmitter.	0	
122	CCS_COMM_JMS_ENVELOPEPUBLISHED_3092	Envelope published from JMS queue into router.	10	
123	CCS_COMM_JMS_JMSPUBLISHED_3093	Envelope published from router into JMS queue.	10	
124	CCS_COMM_AGENT_ENVELOPEPUBLISHED_3094	Envelope published from Communication input into router.	10	
125	CCS_COMM_AGENT_COMMPUBLISHED_3095	Envelope published from router into communication synch response.	10	
126	CCS_COMM_MCAST_3100_FailedToStartService	Couldn't start Multicast listener on [+1:+ 4: +5] + 6	40	Look in event log and/or debug-default as to why the MulticastListener could not be created, taking note of he address and port #.
127	CCS_COMM_MCAST_3110_CommAgentProcessingTimeout	Timeout in CommAgent while processing document: +1 + 4	40	Look in event log and/or debug-default as to why it timed out, taking note of the stackTrace.
128	CCS_COMM_JMS_3120_FailedToCreateConnection	Failed to create JMS connection	40	Look in event log and/or debug-default as to why it was not able to connect to the Sonic Broker, taking note of the broker URL and the username/password used.

129	CCS_COMM_JMS_3121_FailedToCreateQueue	Failed to create JMS queue	40	Look in event log and/or debug-default as to why it was not able to create the queue identity, taking note of the queueName passed in.
130	CCS_COMM_SONIC_COMM_SERVICE_3150_NO_MESSAGE_STORE_FOUND	No message store found for archive: service type <type>, name <name>	40	The message store was not among the list of known services. Check the service-start.prop to ensure it has been turned on.
131	CCS_COMM_SONIC_COMM_SERVICE_3151_MESSAGE_STORE_FOUND	Message store found for archive	0	The message store was found. If the system was configured to start with a message store then this is good. However if it was configured to start without one then check the service-start.prop to ensure that it was not turned on.
132	CCS_COMM_SONIC_COMM_SERVICE_3152_INVALID_SONIC_HEADER	Invalid Sonic header	40	
133	CCS_COMM_SONIC_COMM_SERVICE_3153_ENVELOPE_CONSTRUCTION_FAILED	Envelope construction failed, error: +1	40	Look in event log and/or debug-default as to why it was not able to create the envelope out of the byte message, taking note of the stackTrace.
134	CCS_COMM_SONIC_COMM_SERVICE_3154_AUTHENTICATION_FAILED	Authentication failed, error: +1	40	Look in event log and/or debug-default as to why the client could not be authenticated, taking note of the username/password used.
135	CCS_COMM_SONIC_COMM_SERVICE_3155_HANDLE_TO_ROUTER_QUEUE_FAILED	Handle to router failed, error: +1	40	Look in event log and/or debug-default as to why we got this event, taking note of the stackTrace.
136	CCS_COMM_SONIC_COMM_SERVICE_3156_MESSAGE_TOO_BIG	Message is too big size = +1, max = +2	40	Message is too big.

137	CCS_COMM_SONIC_COMM_SERVICE_3157_ENVELOPE_ARCHIVING_FAILED	Envelope archiving failed, error: +1	40	Failed to archive envelope. Check if DB is up, check on disk space.
138	CCS_COMM_SONIC_COMM_SERVICE_3158_COMMIT_FAILED	Commit failed, error: +1	40	Look in event log and/or debug-default as to why not able to commit the transaction, taking note of the stackTrace.
139	CCS_COMM_SONIC_COMM_SERVICE_3159_SEND_ERROR_REPLY_FAILED	Send error-reply failed, error: +1	40	Look in event log and/or debug-default as to why not able to send the Error reply, taking note of the stackTrace.
140	CCS_COMM_SONIC_COMM_SERVICE_3160_INITIALISATION_FAILED	SonicCommService initialisation failed error <error>	40	Look in event log and/or debug-default as to why the SonicCommService was not able to initialize, taking note of the stackTrace.
141	CCS_COMM_SONIC_COMM_SERVICE_3161_READING_JMS_HEADER_FAILED	Reading property from jms header failed, error: +1	40	Look in event log and/or debug-default as to why we had a hard time reading the JMS header, taking note of the stackTrace.
142	CCS_COMM_SONIC_COMM_SERVICE_3162_PROPERTY_NOT_SET_IN_JMS_HEADER	Property +1 not set in jms header	40	Look in event log as to which property was not set in the JMS header.
143	CCS_COMM_SONIC_COMM_SERVICE_3163_ROLLBACK_FAILED	Rollback failed, error: +1	40	Look in event log and/or debug-default as to why we had a hard time rolling back the transaction, taking note of the stackTrace.
144	CCS_COMM_SONIC_COMM_SERVICE_3164_ERROR_READING_BYTES_FROM_JMS_MESSAGE	Error reading bytes from jms message, error: +1	40	Look in event log and/or debug-default as to why we had a hard time reading the bytes from the JMS message, taking note of the stackTrace.

145	CCS_COMM_SONIC_COMM_SERVICE_3165_JMS_MESSAGE_IS_NOT_BYTES_MESSAGE_TYPE	JMS Message is not BytesMessage type	40	Look in event log and/or debug-default as to why the JMS message was not an instance of ByteMessage, taking note of the stackTrace. And then take a look in the DMQ for the message.
146	CCS_COMM_SONIC_COMM_SERVICE_3166_SEND_TO_DEAD_MESSAGE_QUEUE_FAILED	Send to SonicMQ.deadMessage failed, error: +1	40	Look in event log and/or debug-default as to why we had a hard time sending the message to the DMQ, taking note of the stackTrace.
147	CCS_COMM_SONIC_COMM_SERVICE_3167_XCBL_VERSION_TRANSFORMATION_IN_FAILED	XCBLVersion transformation 'in' failed, error: +1	40	Look in event log and/or debug-default as to why the xCBL version transformer burped on a transform in, taking note of the stackTrace.
148	CCS_COMM_SONIC_COMM_SERVICE_3168_XCBL_VERSION_TRANSFORMATION_IN_SUCCEEDED	XCBLVersion transformation 'in' succeeded	0	Transformation on an inbound message was successful.
149	CCS_COMM_SONIC_TRANSMITTER_3200_PROPERTY_NOT_SET_IN_JMS_HEADER	Property +1 not set in jms header	40	Look in event log as to which property was not set in the JMS header.
150	CCS_COMM_SONIC_TRANSMITTER_3201_READING_JMS_HEADER_FAILED	Reading property from jms header failed, error: <ErrorMsg>	40	
151	CCS_COMM_SONIC_TRANSMITTER_3202_AUTHENTICATION_FAILED	Authentication failed, error: <ErrorMsg>	40	Look in LDAP and make sure the security settings are correct. (Note: from the event, the operator will not be able to find out the sender ID. If possible we may want to augment this event with senderid information).
152	CCS_COMM_SONIC_TRANSMITTER_3203_XCBL_VERSION_TRANSFORMATION_OUT_FAILED	XCBLVersion transformation 'out' failed, error: +1	40	Not Used

153	CCS_COMM_SONIC_TRANSMITTER_3204_XCBL_VERSION_TRANSFORMATION_IN_FAILED	XCBLVersion transformation 'in' failed, error: <ErrorMsg>	40	The CBL schema may be missing or you may have an incorrect version.
154	CCS_COMM_SONIC_TRANSMITTER_3205_XCBL_VERSION_TRANSFORMATION_OUT_SUCCEEDED	XCBLVersion transformation 'out' succeeded	0	Not Used (No action required)
155	CCS_COMM_SONIC_TRANSMITTER_3206_XCBL_VERSION_TRANSFORMATION_IN_SUCCEEDED	XCBLVersion transformation 'in' succeeded	0	Not Used (No action required)
156	CCS_COMM_HTTP_TRANSMITTER_3250_XCBL_VERSION_TRANSFORMATION_OUT_FAILED	XCBLVersion transformation 'out' failed, error: +1	40	Not Used
157	CCS_COMM_HTTP_TRANSMITTER_3251_XCBL_VERSION_TRANSFORMATION_IN_FAILED	XCBLVersion transformation 'in' failed, error: +1	40	Not Used
158	CCS_COMM_HTTP_TRANSMITTER_3252_XCBL_VERSION_TRANSFORMATION_OUT_SUCCEEDED	XCBLVersion transformation 'out' succeeded	0	Not Used (No action required)
159	CCS_COMM_HTTP_TRANSMITTER_3253_XCBL_VERSION_TRANSFORMATION_IN_SUCCEEDED	XCBLVersion transformation 'in' succeeded	0	Not Used (No action required)
160	CCS_COMM_AGENT_TRANSMITTER_3300_XCBL_VERSION_TRANSFORMATION_OUT_FAILED	XCBLVersion transformation 'out' failed, error: <ErrorMsg>	40	The CBL schema may be missing or you may have an incorrect version.
161	CCS_COMM_AGENT_TRANSMITTER_3301_XCBL_VERSION_TRANSFORMATION_IN_FAILED	XCBLVersion transformation 'in' failed, error: <ErrorMsg>	40	The CBL schema may be missing or you may have an incorrect version.

162	CCS_COMM_AGENT_TRANSMITTER_3302_XCBL_VERSION_TRANSFORMATION_OUT_SUCCEEDED	XCBLVersion transformation 'out' succeeded	0	No action required.
163	CCS_COMM_AGENT_TRANSMITTER_3303_XCBL_VERSION_TRANSFORMATION_IN_SUCCEEDED	XCBLVersion transformation 'in' succeeded	0	No action required
164	CCS_DCLT_1_Test	A sample utility CCS_DCLT - event error message +1 +2.	40	Not Used
165	CCS_DCLT_4010_XmlConversionError	XML conversion error.	40	Not Used
166	CCS_DCLT_4015_UnsupportedEncodingError	UTF8 encoding not supported.	40	Not Used
167	CCS_KRNL_5010_QueueCreationError	Failed to create a Queue using class: <cls_name>	40	
168	CCS_KRNL_5015_ServiceQueueingError	Failed to queue envelope to service	40	
169	CCS_KRNL_5025_GoodwillServiceWarning	GoodwillService: Got <message id>.	20	
170	CCS_KRNL_5026_SyncDocumentNoSubscriberWarning	Sync Document has no subscruber: Envelope Id: <message id>.	40	
171	CCS_KRNL_5028_AsyncDocumentNoSubscriberWarning	Async Document has no subscriber: Envelope Id: <message id>.	40	
172	CCS_KRNL_5027_SyncDocumentErrorDropped	Sync Document dropped: Envelope Id: <message id>.	40	
173	CCS_KRNL_5029_ServerCannotSendErrorDocumentOutForSyncRequest	Server cannot send Error document out as response to sync request : Envelope Id: <message id>.	40	
174	CCS_KRNL_5130_PublishError	Publish Error!!	40	

175	CCS_KRNL_5131_CAUGHT_DOCUMENT_EXCHANGE_EXCEPTION	Router catches a DocumentExchangeException. <exception message>. The message is rolled back	40	
176	CCS_KRNL_5132_CAUGHT_EXCEPTION	The router catches an exception. " + <exception message> + ". The message is rolled back."	40	
177	CCS_KRNL_5210_		40	
178	CCS_KRNL_5250_QueueCreationSuccess	MSMQ : <name> created successfully. Journal = <on/off> , Delivery = <express/recoverable>	0	
179	CCS_KRNL_5255_QueueExistsInfo	MSMQ : Queue: <name> already exists. Journal = <on/off> , Delivery = <express/recoverable>	10	
180	CCS_KRNL_5260_IllegalQueueNameError	MSMQ: Illegal Queue Name: <name> .	40	
181	CCS_KRNL_5265_QueueCreationError	MSMQ: Queue: <name> creation error!	40	
182	CCS_KRNL_5270_QueueReceiveError	MSMQ: MsmqDequeue: Message receives error.	40	
183	CCS_KRNL_5271_QueueReceiveSuccessful	MSMQ:MsmqDequeue: Message receives successful.	0	
184	CCS_KRNL_5275_QueueSendError	MSMQ: MsmqEnqueue: Message sends error.	40	
185	CCS_KRNL_5276_QueueSendSuccessful	MSMQ: MsmqEnqueue: Message send successful.	0	
186	CCS_KRNL_THREAD_5310_ServiceThreadException	Unhandled exception in service thread: <service thread information> <Java StringWriter information>	40	
187	CCS_KRNL_5410_		40	

188	CCS_KRNL_SONICMQ_6000_CONNECTION_FAILED		40	
189	CCS_KRNL_SONICMQ_6001_InitDefaultQueueError	In SonicThreadedEnvelopeQueue:initDefaultQueue can create default queue resource, <JMSException>	40	
190	CCS_KRNL_SONICMQ_6002_QueueBrowserError	In SonicThreadedEnvelopeQueue: isEmpty Queue browser error. < JMSException >	40	
191	CCS_KRNL_SONICMQ_6003_QueueSendError	In SonicThreadedEnvelopeQueue: enQueue Message Send Error. < JMSException >	40	
192	CCS_KRNL_SONICMQ_6004_Receive_NULL_JMS_Message	In SonicThreadedEnvelopeQueue:deQueue A Null JMS message is received. Possible reason is connection dropped during the receive.	40	
193	CCS_KRNL_SONICMQ_6005_QueueReceiveError	In SonicThreadedEnvelopeQueue:deQueue cannot receive message. < JMSException >	40	
194	CCS_KRNL_SONICMQ_6006_GET_MSG_LEN_PROP_FAILED	In SonicThreadedEnvelopeQueue:deQueue cannot receive message. < JMSException >	40	
195	CCS_KRNL_SONICMQ_6007_CREATE_QUEUE_RESOURCE_FAILED	In SonicThreadedEnvelopeQueue:createQueue, queue resource creation failed. <Exception>	40	
196	CCS_KRNL_SONICMQ_6008_COMMIT_FAILED	In SonicThreadedEnvelopeQueue:commitTransaction commit failed. <JMSException>	40	

197	CCS_KRNL_SONICMQ_6009_ROLLBACK_FAILED	In SonicThreadedEnvelopeQueue :rollbackTransaction rollback failed. <JMSException>	40	
198	CCS_KRNL_SONICMQ_6010_ROLLBACK_FAILED	40" In SonicThreadedEnvelopeQueue :rollbackTransaction rollback failed	40	
199	CCS_KRNL_SONICMQ_6011_INVALID_ROUTER_SERVICE_QUEUE_CONF	Invalid queue configuration. If router queue is memory, service queue must be memory queue	40	
200	CCS_KRNL_SONICMQ_6012_UNKNOWN_QUEUE_OWNER	The queue configuration specifies an unknown queue owner	40	
201	CCS_KRNL_SONICMQ_6013_CONNECTION_INIT_FAILED	SonicMQ connection initialization failed	40	
202	CCS_KRNL_SONICMQ_6014_RECEIVE_EXCEPTION_OnException	In OnException, receive exception in the JMS exception listener. +1	40	
203	CCS_KRNL_SONICMQ_6015_CONNECTION_DROPPED	SonicMQ broker connection dropped unexpectedly. +1	40	
204	CCS_KRNL_SONICMQ_6016_CONNECTION_DROPPED	SonicMQ broker connection dropped unexpectedly	40	
205	CCS_KRNL_SONICMQ_6017_CREATE_QUEUE_FAILED	In JMSQueuePasser:setNextQueue, set next queue failed. +1	40	
206	CCS_KRNL_SONICMQ_6018_SEND_QUEUE_FAILED	In JMSQueuePasser:enqueue, send queue failed. +1	40	
207	CCS_KRNL_SONICMQ_6019_NON_SESSIONTHREAD_PASSED	In ThreadedEnvelopeQueue :initQueue, a non sessionThread is passed to initQueue.	40	

208	CCS_KRNL_SONICMQ_6020_NON_SESSIONTHREAD_PASSED	In ThreadedEnvelopeQueue :getEnvelopeListener, a non sessoinThread is passed to initQueue.	40	
209	CCS_KRNL_SONICMQ_6021_NON_SESSIONTHREAD_PASSED	In BasicEnvelopeQueue :getEnvelopeListener, a non sessionThread is passed to initQueue.	40	
210	CCS_KRNL_SONICMQ_6022_NON_SESSIONTHREAD_PASSED	In BasicEnvelopeQueue :getEnvelopeListener, a non sessionThread is passed to initQueue.	40	
211	CCS_KRNL_SONICMQ_6023_NON_INTEGER_TIMETOLIVE	Sonic Interface: envelope time to live parameter is not integer. +1	40	
212	CCS_KRNL_SONICMQ_6024_CONNECT_TO_BROKER	Connect to broker: +1 as +2	40	
213	CCS_KRNL_SONICMQ_6025_SESSIONTHREAD_INIT_FAILED	JMS QueueSessionThread init failed: +1	40	
214	CCS_KRNL_SONICMQ_6026_INVALID_QUEUE_CONFIGURATION	Invalid Queue Configuration: +1.	40	
215	CCS_KRNL_SONICMQ_6027_NULL_QUEUE_PASSER	NULL Value: +1.	40	
216	CCS_KRNL_SONICMQ_6028_CAN_NOT_CREATE_ENVELOPE		40	
217	CCS_KRNL_SONICMQ_6029_MESSAGE_TO_DEADLETTER_QUEUE		20	
218	CCS_KRNL_SONICMQ_6030_MESSAGE_TO_DEADLETTER_QUEUE_FAILED		40	
219	CCS_SEC_1_Test	A sample utility CCS_SEC_CONN - event error message +1 +2.	40	

220	CCS_SEC_CONN_6010_ClientConnectionSucceeded	This event indicates that the HTTPS/JMS Client connection has succeeded and application level authentication of has completed. Event Parameters (only for HTTPS client): **- *ConnectionName*** Remote Host Name of HTTPS client connecting to HTTPS Server. **- *Fixed Message*** (for both HTTPS and JMS connections): “Client connection succeeded.”	0	No action. Informational Event.

221	CCS_SEC_CONN_6020_AuthenticationFailed	This event indicates that the application level authentication of Trading Partner (TP) Client has Failed. Event Parameters: HTTPS Connections: ***ConnectionName***HTTP/S Client RemoteUserName TargetName Target Resource URL being Accessed on the HTTPS Server State General HTTP/S based TP Authentication Error message ErrorDetails Specific run-time error message for TP Authentication Failure over HTTPS connections. For JMS/SonicMQ Broker based TP connections, following events attributes are available: ***State*** General JMS-based TP Authentication Error message ***ErrorDetails*** Specific run-time error message for TP Authentication Failure over JMS/SonicMQ connections.	40	Check if the TP application is connecting multiple times with same authentication failure result, possibly due to incorrect {MPID, UserId, Password}. Notify TP organization that they may need to reset their password if they have forgotten the right password or use correct password. Possible Denial-of-service attack. Need to use proper safety measures and network monitoring to track possible illegal client connections to MarketSite, particularly over HTTPS.

226	CCS_SEC_CONN_6050_ServerLDAPConnectionFailed	Connection to LDAP Server Failed (during TP Password Authentication). Event Parameters: ***LDAPSERVER*** LDAP Host Name. ***2) USERNAME*** LDAP Connection UserId (used by the MarketSite Platform Security Subsystem)	40	Check if the Netscape LDAP Server HostName, UserId, Password are correctly configured in the *security.prop* file. Check that the LDAP Server is alive.
227	CCS_SEC_REG_ENVELOPECONTEXT_3000	REG Envelope Context.	10	227
228	CCS_SEC_REG_INITSTART_3010	REG Init Start.	10	228
229	CCS_SEC_REG_INITCOMPLETE_3015	REG Init Complete.	10	229
230	CCS_SEC_REG_HANDLEBASICTPSTART_3020	REG Handle Basic TP Profile Start.	10	230
231	CCS_SEC_REG_HANDLEBASICTPCOMPLETE_3025	REG Handle Basic TP Profile Complete.	10	231
232	CCS_SEC_REG_HANDLEBASICTPLDAPSTART_3030	REG Handle Basic TP Profile LDAP Start.	10	232
233	CCS_SEC_REG_HANDLEBASICTPLDAPCOMPLETE_3035	REG Handle Basic TP Profile LDAP Complete.	10	233
234	CCS_SEC_REG_HANDLEBASICTPMEMBERSTART_3040	REG Handle Basic TP Member Profile Start.	10	234
235	CCS_SEC_REG_HANDLEBASICTPMEMBERCOMPLETE_3045	REG Handle Basic TP Member Profile Complete.	10	235
236	CCS_SEC_REG_HANDLEBASICTPMEMBERLDAPSTART_3050	REG Handle Basic TP Member Profile LDAP Start.	10	236
237	CCS_SEC_REG_HANDLEBASICTPMEMBERCOMPLETE_3055	REG Handle Basic TP Member Profile LDAP Complete.	10	237

238	CCS_SEC_REG_HANDLEUPDATETPSTART_3060	REG Handle Update TP Start.	10	238
239	CCS_SEC_REG_HANDLEUPDATETPCOMPLETE_3065	REG Handle Update TP Complete.	10	239
240	CCS_SEC_REG_HANDLEUPDATETPMEMBERSTART_3070	REG Handle Update Member Start.	10	240
241	CCS_SEC_REG_HANDLEUPDATETPMEMBERCOMPLETE_3075	REG Handle Update Member Complete.	10	241
242	CCS_SEC_REG_NODEPROFILEINITSTART_3080	REG Node profile Init Start.	10	242
243	CCS_SEC_REG_NODEPROFILEINITCOMPLETE_3085	REG Node Profile Init Complete.	10	243
244	CCS_SEC_REG_HANDLELOOKUPTPSTART_3090	REG Handle Basic TP Lookup Start.	10	244
245	CCS_SEC_REG_HANDLELOOKUPTPCOMPLETE_3095	REG Handle Basic TP Lookup Complete.	10	245
247	CCS_SEC_CONN_CONNCONSTRUCTSTART_3010	This event indicates a new TP HTTPS/JMS Connection creation event. Event Parameters: - No variable parameters; - Fixed message: "Conn Constructor Start."	10	Informational event. No action.
248	CCS_SEC_CONN_CONNCONSTRUCTCOMPLETE_3015	This event indicates a new TP HTTPS/JMS Connection creation was completed. Event Parameters: - No variable parameters; - Fixed message: “Conn Constructor Complete.”	10	Informational event. No action.

249	CCS_SEC_CONN_CONNVALIDATESTART_3020	This event indicates start of a new TP/JMS Authentication event. Event Parameters: - No variable parameters; - Fixed message: "Conn Validate Start."	10	Informational event. No action.
250	CCS_SEC_CONN_CONNVALIDATECOMPLETE_3025	This event indicates start of a new TP/JMS Authentication event. Event Parameters: - No variable parameters; - Fixed message: "Conn validate end."	10	Informational event. No action.
251	CCS_SEC_CONN_CONNKEYSTORE_3090_OPENFAILURE	This event indicates problems in opening the HTTPS Server keystore file. Event Parameters: - ***KeystoreFilePath*** Fully-qualified path of the server keystore file. - ErrorDetails Run-time error message during keystore opening failure.	40	Check if the HTTPS Server keystore file exists in the <KeystoreFilePath>. If not, check if the *https-server.prop* file has the correct keystore file name in the property *iaik.jigsaw.ssl.keystore* Check if the password of the keystore is properly configured in the https-Server.prop file in the property *iaik.jigsaw.ssl.keystore.password* (and also encrypt the password before going to Production).

252	CCS_SEC_CONN_CONNKEYSTORE_3095_PWDDECRYPTIONERROR	This event indicates problems in opening the HTTPS Server keystore file due to the keystore password not successfully decrypted. Event Parameters: - ***KeystoreFilePath*** Fully-qualified path of the server keystore file. - ErrorDetails Specific run-time error message during keystore password decryption failure.	40	Check to make sure that keystore file is not corrupt (e.g., by opening it with MarketSite Certificate Management Tool). Check to see if the keystore file was not migrated from an old Weak Encryption MarketSite installation since we do not support re-using of keystores (and HTTPS server certificates) from 40-bit installations into 128-bit installations. Re-create keystore and re-install HTTPS server certificate.
253	CCS_SEC_CONN_6060_SetJMSClientSSLPropSucceeded	This event indicates that the properties for SonicMQ JMS client SSL connection in the client.prop have been successfully set.	0	No action. Informational Event.
254	CCS_SEC_CONN_6065_SetJMSClientSSLPropFailed	This event indicates problems in setting the properties for SonicMQ JMS client SSL connection in the client.prop. Event Parameters: - ***EXCEPTIONMESSAGE*** Specific exception message.	40	Check the following properties in security.prop: - jms.client.ssl.enable - jms.client.ssl.provider.class -jms.client.ssl.cipher.suites - jms.client.ssl.client.requireTrustedRoot - jms.client.ssl.client.trustedRoot.dir Check if any properties above are missing or have an invalid value.

255	CCS_SEC_HTTPSSERVER_6060_UnTrustedCARootClientConnectionEstablished	This event indicates that the HTTPS Client connecting to the HTTPS Server has untrusted CA Root certificate, but by default configuration we are accepting the HTTPS connection (since the client certificate is unexpired and signature is valid.) Event Parameters: - SubjectDN Subject Name of HTTPS Client Certificate - IssuerDN Issuer Name (i.e., CA Name) of HTTPS Client Certificate - Fixed Message: "HTTPS Client's Cert has untrusted CA Root, client connection succeeded:”	0	Informational event. No action.

256	CCS_SEC_HTTPSSERVER_6065_UnTrustedCARootClientConnectionFailed	This event indicates that the HTTPS Client connecting to the HTTPS Server has untrusted CA Root certificate, and due to HTTPS Server configuration requiring client certificates to be issued from Trusted CAs, the connection is rejected at SSL handshake time. Event Parameters: - SubjectDN Subject Name of HTTPS Client Certificate - IssuerDN Issuer Name (i.e., CA Name) of HTTPS Client Certificate - Fixed Message: "HTTPS Client's Cert has untrusted CA Root, client connection failed:”	40	Using the MarketSite Certificate Manager, check if IssuerDN matches one of the trusted CAs in the Server Keystore. If the IssuerDN does not match one of the trusted CAs, decide if that CA should be trusted and acquire the Trusted CA Root from the client site (typically another MarketSite HTTPS Client connecting to present MarketSite over HTTPS using certificates). Re-configure the HTTPS Server to not require Trusted CA root from clients by changing the https-server.prop file property: *iaik.jigsaw.ssl.client.requireTrustedRoot to false* **Note:** Option 2 is not recommended during normal production, only testing deployment.

257	CCS_SEC_HTTPSCLIENT_6065_UnTrustedServerCARootAndClientConnectionFailed	This event indicates that the HTTPS Client application while connecting to a HTTPS Server has detected that the HTTPS Server certificate has been issued by UnTrusted CA and hence the HTTPS Client has aborted the connection due its CA Trust Policy not allowing such Untrusted Server connection. Event Parameters: - SubjectDN Suibject Name of HTTPS Server Certificate - IssuerDN Issuer Name (i.e., CA Name) of HTTPS Server Certificate	40	Check if the target HTTPS Server's issuing CA is trusted by the HTTPS Client keystore. Normally if the HTTPS Client is running as part of HTTPS Server servlet application, the effective Trusted CA will be in the Server keystore. Hence, using the MarketSite Certificate Manager, check if the CA is trusted. If not, acquire the CA Root of the target HTTPS Server and import it as a Trusted CA Root. By default enforcing Trusted CA Policies for target HTTPS Servers is disabled. Decide if default behavior is appropriate. E.g., a HTTPS Client running inside a HTTPS Server based Async or SyncTransmission Service can disable Trusted CA requirement from its target HTTPS Server by changing the *security.prop* file property: *Security.httpServerTrustedRoot.required* to *false.*

258	CCS_SEC_HTTPSSERVER_6065_NoServerCertificateFoundInKeyStore	This event indicates that the HTTPS Server failed to bootstrap due to server certificate not available in the Server Keystore. Event Parameters: ***-ServerCertificateEntryName*** Entry Name of HTTPS Server certificate entry in the ***https-server.prop*** file property: ***iaik.jigsaw.ssl.rsa.keyAndCertificate*** - KeyStoreFilePath Fully-qualified file path to HTTPS Server keystore file.	40	Check if the https-server.prop file has the correct certificate entry name in the *https-server.prop* file (that is, entry name corresponds to the real HTTPS Server certificate that is imported into the keystore using the Certificate Manager). 2) Check if the keystore file name is correct in *https-server.prop* file.
259	CCS_SEC_DOCUMENTSERVICE_6065_DataDecryptionError	This event indicates that a data decryption error occurred. Event Parameters: - Fixed Message: “Data Decryption Error:” - ErrorDetails Specific run-time failure reasons.	40	Report error details to Commerce One Tech Support.
260	CCS_SEC_REG_SONICBROKERCREATIONFAILED_6070	Creation Of Broker +< TPBROKERURL>, For Site Node < SITENODENAME>, Failed	40	260
261	CCS_SEC_REG_SONICBROKERWARNING_6075	Broker < TPBROKERURL > For Site Node < SITENODENAME> is of Local Address Type. Not Creating Entry in Portal’s Config Server	20	261
262	CCS_SEC_REG_SONICBROKERDELETIONFAILED_6080	Deletion Of Broker +1 , For Site Node + 2, Failed	40	
263	CCS_SVC_LOSTANDFOUND_7000_GotDocument	Lost and Found service got document : +1	30	N/A

264	CCS_SVC_LOSTANDFOUND_7001_SYNC	Lost and found got sync request, sending back error	30	Service Router – No service is subscribing to this document. Hence, this synchronous request is ending up in the L&F. The client should see an exception. Check if all the services are up and running. Portal Router – The synch transmitter service cannot send the request out to the XPC. Check the transmission settings for the SyncTransmitterService
265	CCS_SVC_LOSTANDFOUND_7002_ASYNC	Lost and found got async document.		The AsyncTransmitterService cannot send the asynchronous document out. Check the transmission settings for the AsyncTransmitterService.
266	CCS_SVC_LOSTANDFOUND_7003_NO_MESSAGE_STORE_FOUND	No message store found for lost and found: service type <type>, name <name>	30	N/A
267	CCS_SVC_LOSTANDFOUND_7004_MESSAGE_STORE_FOUND	Message store found for lost and found service	0	N/A
268	CCS_SVC_REMOTESUBSCRIPTION_7010_SubscriptionSuccess	Remote Subscription from Node : +1 is successful with id = +2	10	N/A
269	CCS_SVC_REMOTESUBSCRIPTION_7011_UnSubscriptionSuccess	Node : +1 , with id = +2 successfully unsubscribed	10	N/A
270	CCS_SVC_VIRTUALNODE_7012_QueueCreateFail	Queue : +1 could not be created using class : +2 for the Node : +3	40	N/A
271	CCS_SVC_SUBSCRIPTION_7013_SendSubscription	Sending Remote Subscription to Main CCS Server	20	N/A

272	CCS_SVC_SUBSCRIPTION_7014_SubscriptionSuccess	Subscription Successful with Main CCS Server , Id = +1	20	N/A
273	CCS_SVC_SUBSCRIPTION_7015_SubscriptionFail	Remote Subscription Un-Successful with Main CCS Server, Returned : +1	40	N/A
274	CCS_SVC_SUBSCRIPTION_7016_Unsubscribe	UnSubcribing from Main CCS Server	10	N/A
275	CCS_SVC_SUBSCRIPTION_7017_UnsubscribeSuccess	UnSubscribe successful from Main CCS Server	10	N/A
276	CCS_SVC_SUBSCRIPTION_7018_UnsubscribeFail	UnSubscribe unsuccessful with Main CCS Server, Returned : +1	40	N/A
277	CCS_SVC_FACTORY_7019_CreationFailure	Failed to create +1 using +2: +3	40	This message could be seen at server startup when the server is loading the different factory objects that are internal to the server. The message should indicate the corresponding factory object that it is failing to load. This event should not arise and if it does, indicates a serious internal problem.
278	CCS_SVC_TRANSMITTER_7020_TransmitFailure	Failed to transmit envelope id = +1: +2	40	There is an error in transmitting a document from the (A)syncTransmitterService. The error could be due a misconfiguration of the corresponding service, or an error in the underlying transport mechanism (Sonic or HTTPS).
279	CCS_SVC_FORWARD_7021_DocumentForwarded	Document <documentname> from <sender> forwarded to <receiver>	10	N/A

280	CCS_SVC_FORWARD_7022_ForwardingFailed	Document < documentname> from < sender> failed to forward to < receiver>: <reason>	40	N/A
281	CCS_SVC_PING_7023_PingSent	Ping document sent	10	N/A
282	CCS_SVC_PING_7024_PongReceived	Pong document received	10	N/A
283	CCS_SVC_VIRTUALNODE_7025_InitFailed	Initialization failed for node +1: +2	40	N/A
284	CCS_COMM_SONIC_COMM_SERVICE_7026_XCBL_VERSION_TRANSFORMATION_OUT_FAILED	XCBLVersion transformation 'out' failed, error: +1	40	There was an error in transforming the envelope from one version of CBL to the other. Check if the transformation service is up and running and if it is then it is configured correctly.
285	CCS_SVC_SyncTransmitterService_7030_NotFound	Sync Transmitter Service not found in ServiceRegistry. Server unusable.	40	The SyncTransmitterService is not specified in the service-start.prop file for the server. Make sure that there is a specification for the SyncTransmitterService in the file.
286	CCS_SVC_SyncTransmitterService_7031_EnvelopeTransmitted	Envelope successfully transmitted from SyncTransmitterService	10	N/A
287	CCS_SVC_AsyncTransmitterService_7040_NotFound	Async Transmitter Service not found in ServiceRegistry. Server unusable.	40	N/A
288	CCS_SVC_AsyncTransmitterService_7041_EnvelopeTransmitted	Envelope successfully transmitted from AsyncTransmitterService	10	N/A

289	CCS_SVC_TRANSMITTER_9000_ErrorForAnError	Got an Error for an Error document. EnvelopeId: +1. ReferenceId: +2. Exception: +3"	40	This event is generated when there is an error while transmitting an error document from the AsyncTransmitterService. Check the transmission settings and the underlying transport (HTTPS or Sonic) settings for the AsyncTransmitterService.
290	CCS_SVC_TRANSMITTER_9050_ErrorForAnErrorIgnored	Cannot deliver and error for an error. Error document ignored. EnvelopeId: +1. ReferenceId: +2. Exception: +3	40	This event is usually preceded by the previous one and just indicates that in case when there is an error in transmitting an error document for the AsyncTransmitterService, the platform drops the error document.
291	CCS_SVC_TRANSMITTER_9051_EnvelopeDroppedBecauseMatchedExcludeAddress	+1: Envelope dropped because excludeAddress = +2	10	This event is generated when the (A)syncTransmitterService receives an envelope that was sent from an address that this server is configured to reject.
289	CCS_UTL_URI_1_Test	A sample utility URI - event error message +1 +2.	40	N/A

B SonicMQ Broker Events

SonicMQ Events are located in:

<SonicMQ installation root>/events/sonic.log

There several types of SonicMQ events, but two types are the most seen:

- Connection Exception Code Messages
- Broker Events

Connection Exception Codes are seen in the event log when a problem occurs during an attempt to establish a connection to another broker. The following exceptions may be seen:

- ***ENetworkFailure*** - the remote broker is not started or is an invalid broker
- ***java.net.ConnectionException*** - remote broker is not started or is an invalid broker
- ***EInauthenticClient*** - an invalid user name or password was used

Note .... If the event log displays the exception code: ***EUserAlreadyConnected,*** restart the Redirector Service, a utility service that receives all events generated by the SonicMQ broker and sends them to the event log. In the Services window, find *redirectService* and restart it.

Broker Events are logged during the normal operation of the broker. Most of the messages which are logged for these events are self-explanatory. View the event log to obtain complete information about the message.

These types of events are logged in the event log. They are generated when:

connect	A connection is opened
disconnect	A connection is closed
dmqstatus	Amessages is added to the Dead Message Queue if the Dead Message Queue has exceeded a user-specified fraction of its maximum size.
drop	A connection is lost without the client being disconnected. For example, if the client dies.
redirect	A client connect attempt is redirected to another server in a SonicMQ cluster because of load balancing
reject	A connection is rejected
undelivered message	See **Undelivered Message Event** immediately after this table.

An **Undelivered Message Event** occurs when a message is moved to the Dead Message Queue. If messages can not be routed the XPC Server they are moved from a broker's *Routing Queue (*SonicMQ.routingQueue*)* to the *Dead Letter Queue (SonicMQ.deadMesage)*.

The event log will hold events related to these un-routed messages. This appendix contains a summary of the types of undelivered messages along with the codes for the events which are generated when a message is moved to the Dead Message Queue.

The event log entry for an Undelivered Message Event contains a numerical *Reason Code*. Use the reason code number to locate information about the event in the Reason Code listing here.

For more information, select **Start | Programs | Progress SonicMQ | Documentation** to consult the SonicMQ documentation: *SonicMQ Programming Guide* (Chapter 7) and *SonicMQ Deployment* (Chapter 3).

- **Reason code 1:** **UNDELIVERED_TTL_EXPIRED**

  Description: *The TTL has expired*

  This applies only to messages sent in synchronous mode, for example, from a service such as *PriceCheck*)

When a message is sent, an optional parameter, TTL, can be placed in the message header. TTL (Time to Live) is set according to GMT. If a message arrives at a SonicMQ broker and the TTL is expired, the message is moved to the Dead Message Queue.

- **Reason Code 3:** **UNDELIVERED_ROUTING_INVALID_NODE**

  Description: *The node name is invalid*

  This error occurrs when a node attempts to route a message to another node and there is no connection information for this node in the routing connection table. If a node name was incorrectly specified during the installation, the broker will need to be re-installed.

  Reason code 4: **UNDELIVERED_ROUTING_INVALID_DESTINATION**

  Description: ***Destination queue is invalid***

  This error occurs when a SonicMQ broker receives a message intended for a global queue which does not exist on that broker. It will then be placed in the Dead Message Queue for that broker. If a queue name was incorrectly specified during the installation, the broker will need to be re-installed.

- **Reasom code 5:** **UNDELIVERED_ROUTING_TIMEOUT**

  Description: *Connection Cannot Be Established Before Routing Timeout*

  This error occurs when a message is sent to the Routing Queue for a server and a connection to the receiving SonicMQ broker has been lost. After the delay period expires, asynchronous messages will be moved to the Dead Message Queue. The delay period is set in the broker configuration file (broker .ini) and the default is 7200 seconds. This is the only type of undelivered message which is not moved immediately to the Dead Message Queue.

- **Reason code 6:** **UNDELIVERED_ROUTING_INDOUBT**

  Description: *An unacknowledged messages has exceeded its INDOUBT_TIMEOUT*

  When a broker sends a persistent message which is not acknowleded (due to a network or server failure), the message goes into an *indoubt* state. If a connection is re-stablished later and the message is then acknowledged, it is removed from that state (the message is never re-sent). If the failure period exceeds the

INDOUBT_TIMEOUT, then the message is sent to the Dead Message Queuee.

- **Reason code 7:** **UNDELIVERED_ROUTING_CONNECTION_AUTHENTICATION_FAILURE**

  Description: *The Connection Authentication has failed*

  This error occurs when a connection to another broker fails due to an authentication problem. For example, the remote broker does not accept the userid/password which was used to establish the connection. The message goes to the Dead Message queue of the sender.

- **Reason code 8:** **UNDELIVERED_ROUTING_CONNECTION_AUTHORIZATION_FAILURE**

  Description: *The Connection Authorization has failed*

  A connection has been successfully established between brokers, but the name of the sending broker is not recognized as a valid sender by the remote broker. The message goes to the Dead Message queue of the sender.

- **Reason code 9:** **UNDELIVERED_MESSAGE_TOO_LARGE**

  Description: *The message is too large*

  A sending broker attempts to queue a message on another broker but the size of the messages is larger than that which the queue will accept. The message goes to the Dead Message queue of the sender.

C Advanced Security

This appendix covers advanced security topics. It discusses optional Client-side Trusted CA Validation, presents a table of all pre-installed Trusted CA roots, and contains information on Trading Partner passwords.

Client-side Trusted CA Validation

CA Trust Policies on the HTTPS Client

When HTTPS client applications (such as XPC 4.0, the Async Tranmission Service, Sync Tranmission Service) make connections to target HTTPS Servers, you can use an optional security policy to verify that the target HTTPS Server is identifying itself with a SSL Server certificate that was issued by a trusted third party CA.

By default, the HTTPS client will accept any server certificate CA to be trusted. Further details are discussed below.

Enforcing CA trust policies on the HTTPS client-side can further improve the security of the HTTPS client applications. This is because it only allows connections to servers that use an SSL certificate issued by a third party CA that is trusted by the HTTPS client application.

Configuring CA Trust Policies for HTTPS Client

For XPC HTTPS client applications, the setting of this CA trust policy option is defined in <install:root>/runtime/servers/<servername>/config/startup/security.prop as follows:

```
security.httpsServerTrustedRoot.required=false
```

this is default value.

To enable this security policy option, we can configure this property as:

```
security.httpsServerTrustedRoot.required=true
```

all target server CAs must be trusted

If you choose this option, you must determine if your target HTTPS Server(s) will be configured with a SSL Server certificate issued by a trusted CA. For example, for XPC Servers running at a BuySite or at an NMM site, they will only connect back to MarketSite and hence only MarketSite's Server CA needs to be trusted. However, at a MarketSite, all CAs for all target HTTPS Servers must be configured to be trusted.

Default HTTPS Client-side Behavior

The default is set to false because of configuration complexity of "registering" all target HTTPS Servers CAs to be trusted. However, the HTTPS client applications will always validate that the target server certificate chain signatures are valid and certificates are not expired.

Managing CA Trust Policies of Target HTTPS Server

If this option is enabled, your XPC operator must verify that the issuing CA of the target HTTPS Server certificate is trusted. If the CA of the target server is not trusted by the XPC installation and this CA Trust Policy is enabled, then you must complete the following steps:

1. Acquire CA Root of Target Server

The operator must acquire that CA root from the target site. The CA certificate of the target server may be acquired via an e-mail from target sites.

2. Trust the new CA Root

Use the Certificate Manager tool to import the CA root certificate that needs to be trusted by that HTTPS client domain.

Detecting Un-Trusted CA Validation

When the CA root certificate is not trusted, you will get the following run-time error message:

HTTPS Client application has detected a server certificate by an untrusted CA Root, therefore client connection failed.

Additionally, an event is logged that the HTTPS client has refused to connect to the target HTTPS Server because the CA of the target server is not trusted. An event of this form will be logged in the event log file:

```
<EVENT><TIME>November 15, 2000 9:38:33 PM GMT-08:00</
TIME><MILLIS>974353113473</
MILLIS><KEY>CCS_SEC_HTTPSCLIENT_6065_UnTrustedServerCARootAndCl
ientConnectionFailed</KEY><TEXT>: HTTPS Client application has
detected a server certificate by an untrusted CA Root, therefore
client connection failed: subject:
C=US,ST=California,OU=MarketSiteOps,O=MarketSitePartnerCompany,
L=Mountain View,EMail=admin@yourmarketsite.com,CN=rsa-user ,
issuer: CN=Commerce One (Local) Self-signed CA,OU=Market Site -
Server Security,O=Commerce One Certificate Server -
CA,L=Mountain View,C=USA</TEXT><CAT>CCS_SEC</
CAT><SUBCAT>SECURITY</SUBCAT><NUMID>6065</NUMID><SEV>40</
SEV><TYPE>STATUS</TYPE><LANG>en</
LANG><PARM>C=US,ST=California,OU=MarketSiteOps,O=MarketSitePart
nerCompany,L=Mountain
View,EMail=admin@yourmarketsite.com,CN=rsa-user</
PARM><PARM>CN=Commerce One (Local) Self-signed CA,OU=Market
Site - Server Security,O=Commerce One Certificate Server -
CA,L=Mountain View,C=USA</PARM></EVENT>
```

Pre-installed Trusted CA Root Certificates

MarketSite HTTPS and SonicMQ Broker have a SSL Server supporting Trusted CA Roots for the following Certificate Authorities: Baltimore, Entrust, Thawte and Verisign. Details of these Trusted CAs are provided in the following table.

Third Party CA Name	Friendly Name	Trusted CA Root DN

Baltimore/ GTE Cybertrust	BTCTRoot_12-20-00	CN=Baltimore CyberTrust Root, OU=CyberTrust,
		O=Baltimore, C=IE

Entrust	Entrust2048CA	CN=Entrust.net Certification Authority (2048), OU=(c) 1999 Entrust.net Limited, OU=www.entrust.net/CPS_2048 incorp. by ref. (limits liab.), O=Entrust.net

Entrust	EntrustGss164	CN=Entrust.net Secure Server Certification Authority,
		OU=(c) 2000 Entrust.net Limited, OU=www.entrust.net/SSL_CPS incorp. by ref. (limits liab.),
		O=Entrust.net

Entrust	EntrustServerCA	CN=Entrust.net Secure Server Certification Authority, OU=(c) 1999 Entrust.net Limited, OU=www.entrust.net/CPS incorp. by ref. (limits liab.), O=Entrust.net, C=US

Baltimore/ GTE Cybertrust	GTECTGlobalRoot_12-20-00	CN=GTE CyberTrust Global Root, OU=GTE CyberTrust Solutions, Inc.,
		O=GTE Corporation, C=US

Baltimore/ GTE Cybertrust	GTECTRoot_12-20-00	CN=GTE CyberTrust Root, O=GTE Corporation, C=US

Thawte	ThawtePremiumServerCA	EMail=premium-server@thawte.com,CN=Thawte Premium Server CA,
		OU=Certification Services Division, O=Thawte Consulting cc, L=Cape Town,ST=Western Cape, C=ZA

Third Party CA Name	Friendly Name	Trusted CA Root DN
Thawte	ThawteServerCA	EMail=server-certs@thawte.com,CN=Thawte Server CA,OU=Certification Services Division,O=Thawte Consulting cc,L=Cape Town,ST=Western Cape,C=ZA
Verisign	VerisignC1G2	OU=VeriSign Trust Network,OU=(c) 1998 VeriSign, Inc. - For authorized use only,OU=Class 1 Public Primary Certification Authority - G2,O=VeriSign, Inc.,C=US
Verisign	VerisignC1G3	CN=VeriSign Class 1 Public Primary Certification Authority - G3,OU=(c) 1999 VeriSign, Inc. - For authorized use only,OU=VeriSign Trust Network,O=VeriSign, Inc.,C=US
Verisign	VerisignC2G2	OU=VeriSign Trust Network,OU=(c) 1998 VeriSign, Inc. - For authorized use only,OU=Class 2 Public Primary Certification Authority - G2,O=VeriSign, Inc.,C=US
Verisign	VerisignC2G3	CN=VeriSign Class 2 Public Primary Certification Authority - G3,OU=(c) 1999 VeriSign, Inc. - For authorized use only,OU=VeriSign Trust Network,O=VeriSign, Inc.,C=US
Verisign	VerisignC3G2	OU=VeriSign Trust Network,OU=(c) 1998 VeriSign, Inc. - For authorized use only,OU=Class 3 Public Primary Certification Authority - G2,O=VeriSign, Inc.,C=US
Verisign	VerisignC3G3	CN=VeriSign Class 3 Public Primary Certification Authority - G3,OU=(c) 1999 VeriSign, Inc. - For authorized use only,OU=VeriSign Trust Network,O=VeriSign, Inc.,C=US
Verisign	VeriSignC4G2	OU=VeriSign Trust Network,OU=(c) 1998 VeriSign, Inc. - For authorized use only,OU=Class 4 Public Primary Certification Authority - G2,O=VeriSign, Inc.,C=US

Third Party CA Name	Friendly Name	Trusted CA Root DN
Verisign	VerisignC4G3	CN=VeriSign Class 4 Public Primary Certification Authority - G3,OU=(c) 1999 VeriSign, Inc. - For authorized use only,OU=VeriSign Trust Network,O=VeriSign, Inc.,C=US
Verisign	VeriSignIntlRoot	OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign,OU=VeriSign International Server CA - Class 3,OU=VeriSign, Inc.,O=VeriSign Trust Network
Verisign	VeriSignPubC1	OU=Class 1 Public Primary Certification Authority,O=VeriSign, Inc.,C=US
Verisign	VeriSignPubC2	OU=Class 2 Public Primary Certification Authority,O=VeriSign, Inc.,C=US
Verisign	VeriSignPubC3	OU=Class 3 Public Primary Certification Authority,O=VeriSign, Inc.,C=US
Verisign	VerisignRSACA	OU=Secure Server Certification Authority,O=RSA Data Security, Inc.,C=US

Trading Partner Password Authentication Protocol

The Trading Partner client, such as an EBD client or other XPC client using JMS/SonicMQ or HTTPS protocol, sends the authentication data as part of SSL encrypted connection. The authentication data contains three fields that are included in the HTTP/S header or JMS/SonicMQ message header.

The specific authentication fields are described below:

■

Authentication Data	Description	HTTP/S Header Field	JMS/SSL Header Field
User ID	The registered userid of the sender who is a member of a registered trading partner at MarketSite; this is typically the friendly name or shortname of the Trading Partner.	RFC 2617Standard HTTP Basic Authentication {UserId/ Password}Header Data.	Application-level JMS/SonicMQ header key: "BasicUser"
Password	The password of the member; this is typically the Password of Trading Partner organization since we support only one password per Trading Partner. *Note:* The minimum character length of a password is 8 characters.	RFC 2617 Standard HTTP Basic Authentication {UserId/ Password}Header data.	Application-level JMS/SonicMQ header key: "BasicPassword
MarketParticipant ID (MPID)	A Globally unique ID of the trading partner organization that is issued at registration time to the Buyer or Supplier trading organization registered at MarketSite. MPIDs are immutable across time and space and are issued to identify a Trading Partner at authentication time. *Note*: MPID is IETF standard compliant UUIDs that are generated as 16-bytes UUIDs, but stored/transmitted in their 36-character string representation format.	Application-level HTTP header key: "marketparticipantid"	Application-level JMS/SonicMQ header key: "marketparticipantid"

D Property Files

This appendix describes how property files are used in XPC 4.0. Only properties related to XPC are listed. It is highly recommended that the properties should not be changed directly. Instead, tools such as **Configure XPC**, **XPC Manager**, or **Certificate Manager** should be used to make any change to the properties.

Note .... **Configure XPC** and **Certificate Manager** are documented in this manual and **XPC Manager** is documented in the *XPC Developer Guide and API Reference*.

Key property files are listed in this table. Locations for these property files vary and are listed throughout this appendix. You can edit properties that are highlighted by changing them on the Configure XPC window, as described in Configuring XPC on page 2-7 and Configuring Additional Options on page 4-5.

Property File	**Description**
http-runner.prop	Holds some basic information including the http port and the debug level information; also indicates whether the server is run securely
https-server.prop	Holds the https port and security information
security.prop	Security information needed by the server
server-start.prop	Information used by the server at startup time
service-start.prop	Information on each of the services to be run on this server

Property File	Description
default.prop	Each service has its own property file, usually named default.prop
client.prop	Client property file for the client software, usually named client.prop

Server Configuration Files

The initial XPC server configuration is done when you install XPC. After you install, use tools such as **Configure XPC, XPC Manager,** and **Certificate Manager**. The properties in the server files set up the port numbers and file locations, specify services to run, and give code locations.

Property files are located in the runtime directory installation for each server and the daemon server as shown:

<install:root>\runtime\servers\<ServerName>\config\startup

```
<install:root>\runtime\daemon\config\startup
```

http-runner.prop

This is the smallest property file. It holds initial properties such as the unsecure http port and the security switch for the server.

Properties	Value	Description
org.w3c.jigsaw.port	Any available port Default is 8008	Port opened for the http connection, even if the server is run securely. So the port should haven't been used.
ccs.server.security	true/false Default is true	If set to true, the server will also read https-server.prop and security.prop, start an https connection, for the port specified in the org.w3c.jigsaw.port of https-server.prop file and the server is designated a secure server.
ccs.server.debug.level	0-6 where: 0=debug, 1=info, 2=warning, 3=error, 4=critical, 5=fatal	This property sets the debugging level for the server. In full production systems, set to a high level to avoid a lot of messages. For debugging a faulty server, particularly during an initial installation, set to a low number.

ccs.server.xdk.dir	Default is<install:root>\schema	This property indicates a local file system for schema resolution and is used if local file based schema resolution is used, which is specified in ccs.server.entitymgr.fs of server-start.prop
ccs.server.debug.filename	Default is <install:root>\runtime\servers \<ServerName>\logs\debug-default	This property indicates a local file for debug messages. If not used, messages go to standard out and are displayed on the screen if the server is run in isolation, which is useful for initial debugging of a server.
ccs.server.debug.file.overwrite	true/false	If set to true, the debug log is overwritten with every start of the system. Defaults to false.

https-server.prop

This file contains properties for the https connections.

Properties	**Value**	**Description**
org.w3c.jigsaw.port	Any available port Default is 4433.	Port opened for the https connection. This is set during the installation.
org.w3c.jigsaw.client.bufsize	8192	Output buffer size. **Do not change.**
org.w3c.jigsaw.request.timeout	3000000	https server request timeout property. **Do not change.**
iaik.jigsaw.ssl.enabled	True	Server running with SSL. **Do not change.**
iaik.jigsaw.ssl.security.provider	iaik.security.provider.IAIK	IAIK is the security provider. **Do not change.**
iaik.jigsaw.ssl.tempRSAKey	Generate	Part of SSL bootstrapping. **Do not change.**
iaik.jigsaw.ssl.dhPararmeters	Pre-generated	Part of SSL bootstrapping. **Do not change.**
iaik.jigsaw.ssl.keystore	Default is serverstore.	Default keystore to use to start the server. Use the **Certificate Manager** to change the value; refer to Certificate Manager on page 3-3.

iaik.jigsaw.ssl.keystore.password		Default password for default keystore; should be encrypted after install. The default password is: *admin* Clear text should be used only for testing. Can be encrypted 128bit or 56bit. Use the **Certificate Manager** to change the value; refer to Certificate Manager on page 3-3.
iaik.jigsaw.ssl.rsa.keyAndCertificate	Default is Test-ServerCert	Default entry name for keystore needed to connect to the keystore. In the daemon this is rsa-user-daemon.
Cipher suites is not being displayed due to length		**Do not change.**

Below are the authentication flags in the same file, https-server.prop. The values differ between Test and Production modes.

Properties	**Value**	**Description**
iaik.jigsaw.ssl.client.authentication	true/false Default is false	True, the default for Production mode, means client authenticate using certificate or uid/pswd. False, the default for Test mode, means client authenticate using only uid/pswd. You can change the value with **Configure XPC** by enabling client authentication (Production mode) or disabling client authentication (Test mode).
iaik.jigsaw.ssl.client.requireCertificate	true/false Default is false	True, the default for Production mode, requires certificate from client. False, the default for Test mode, does not require certificate from client. You can change the value with **Configure XPC** by enabling client authentication (Production mode) or disabling client authentication (Test mode).
iaik.jigsaw.ssl.client.requireTrustedRoot	true/false Default is false	True checks to see if client certificate’s root is trusted. False, the default, does not check to see if client certificate’s root is trusted. You can change the value to True in Production mode (optional).

security.prop

This file is for additional security properties, other than the certificates and other jigsaw information in the https-server.prop file.

Properties	Value	Description
jms.client.ssl.enable	True	SonicMQ JMS client SSL connection. **Do not change.**
jms.client.ssl.provider.class	Progress.message.net.ssl.iaik.iaikSSLImpl	IAIK is the security provider. **Do not change.**
jms.client.ssl.cipher.suites	Not listed	Define all cipher suites used by sonic ssl. **Do not change.**
jms.client.ssl.client.requireTrustedRoot	False	Use trust root certificates to verify server certificates. **Do not change.**
jms.client.ssl.client.trustedRoot.dir	<CADir>	Directory to store trusted root certificates.

The values differ between Test and Production modes as enforced by the Certificate Manager.

Properties	Value	Description
security.httpsServerTrustedRoot.required	true/false Default is false	True checks whether server root certificate is trusted. False, the default, does not check whether server root certificates is trusted. You can change the value to True in Production mode (optional).

security.anonymousclients.allowed	true/false Default is true	True, the default for Test mode, allows anonymous clients. False, the default for Production mode, does not allow anonymous clients. You can change the value with **Configure XPC** by enabling client authentication (Production mode) or disabling client authentication (Test mode).
security.clientauth.disable	true/false	True, the default value for Test mode, disables client authentication. False, the default value for Production mode, enables client authentication. You can change the value in **Configure XPC** by enabling client authentication (Production mode) or disabling client authentication (Test mode).

service-start.prop

This property file contains startup information for the server itself.

Properties	**Value**	**Description**
serverstartup.logfile.dir	Default is \logs, the directory of <install:root>\runtime\servers\<ServerName>\ogs	Directory from the servers runtime root directory for startup log information. The root directory is <install:root>/runtime/servers/<ServerName>.
serverstartup.logfile.allevents	Default is systemStartupLog, which is <install:root>\runtime\servers\<ServerName>\logs\systemStartupLog	Event subscription to all events go in this file under the serverstartup.logfile.dir.
serverstartup.default.lang	en Default is en	The default language for the server. The default is English.
serverstartup.default.eventcatalog.dir	Default is \eventcatalog, the directory of <install:root>\runtime\servers\<ServerName>\eventcatalog	Directory from the server runtime root directory for the event catalog. The root directory is <install:root>\runtime\servers\<ServerName>.

ccs.server.entitymgr.fs	true/false Default is *true* for file system resolution.	A boolean flag for using the Filesystem for Schema resolution or not. If this is true, then the one below should be false and vice-versa.
ccs.server.entitymgr.ldap	true/false Default is false.	A boolean flag for using the ldap for Schema resolution or not. If this is true, then the one above should be false and vice-versa.
ccs.server.router.impl.code	Default is com.commerceone.ccs.kernel.router.ServiceRouter	Routing component. **Do not change.**
ccs.server.router.impl.args	Default is initialThreads=1,maximumThreads=9,queueCode=com.commerceone.ccs.kernel.queue.ThreadedEnvelopeQueue,queueType=memory,queueOwner=router	These are the arguments of the Document Router code. Valid arguments include: queueOwner= service OR router queueName = <some name>, required only if queueType=sonic queueType= memory OR sonic queueCode= com.commerceone.ccs.kernel.sonicmq.SonicThreadedEnvelopeQueue for sonic queueType queueCode= com.commerceone.ccs.kernel.queue.ThreadedEnvelopeQueue for memory queueType Use the XPC Manager to change this property.
ccs.server.sonicmq.enabled	true/false Default is true.	True indicates server is using Sonic queues as the primary transport mechanism.
ccs.server.sonicmq.syncresponsequeue.name	Default is XPC_<servername>_SyncResponse	Name of the response queue needed for synchronous messages. This needs to be set if ccs.server.sonicmq.enabled is true.

service-start.prop

There are two sections to this file:

- Top section lists services to be run on the server.
- Bottom section contains a listing for each service of the class name to be run to create this service, and the parameters to be passed into the service.

Service startup list section

The following are the properties of a service start section:

Properties	Value	Description
startup.base.services	Default is ConfigurationService,AdminService,AsyncTransmitterService,SyncTransmitterService	Base services
startup.comm.services	Default is CommServletService,SonicCommService	Communication services. Use XPC Manager to edit.
startup.aux.services	Default is EventManager,LogService,MessageStoreService,XCBLTransformService	Auxiliary services
startup.application.services	Default is PingService	Application services. Use XPC Manager to define document and timed services.

Services are categorized and are started in this order of the categories:

1. Base services
2. Auxiliary (aux) services
3. Application services
4. Communication (comm) services.

This prevents documents from being received at a server before all services are up and available to receive them. Most business services run as application services. A service will not run on a server, unless it is named in the lists of services to be started.

Service definition section

The **Service definition** section includes entries for all of the services named in the list section. For each service the code is specified to tell the server what class implements the service. The name used is the same name used in the service startup list. A line specifying the arguments to be passed to the service can be added. For most services that use queues to receive documents, this requires a specification of the queue code to be used.

It is possible to specify services for services that are not used. If the service is not listed in the startup section, the entries are ignored. For XPC document and timed services, the Args are defined by the XPC Manager.

The first entries here specify the code for the CommServletService and the code for the SonicCommService. In addition, the argument queueName, and number of threads are listed for the SonicCommService, which tells the SonicCommService to use a Sonic queue for its input queue.

Properties	**Value**	**Description**
service.CommServletService.code	Default is com.commerceone.ccs.comm.receiver.servlet.CommServletService	Class used to create the CommServletService.
service.SonicCommService.code	Default is com.commerceone.ccs.comm.receiver.sonic.SonicCommService	Class used to create the SonicCommService.
service.SonicCommService.args	Default is initialThreads=4,maximumThreads=9,queueName=XPC_<sonicnodename>_ConnectorInbound	Arguments for SonicCommService.
service.LostAndFoundService.code	Default is com.commerceone.ccs.service.LostAndFoundService	Code for LostAndFoundService
service.LostAndFoundService.arg	Default is LostAndFound message could also go to dead message queue with the args of store=file, initial Threads=4, maximumThreads=9,queueCode=com.commerceone.ccs.kernel.sonicmq.SonicThreadedEnvelopeQueue.queueType=sonic,queueOwner=service.queueName=SonicMQ,deadMessage, readonly=false	
service.ConfigurationService.code	Default is com.commerceone.ccs.service.config.ConfigurationManager	Code for Configuration Service

service.ConfigurationService.args	Default is initialThreads=0,maximumThreads=1,queueCode=com.commerceone.ccs.kernel.queue.ThreadedEnvelopeQueue,queueType=memory,queueOwner=service	Valid arguments for Configuration Service include:
		queueOwner= service OR router
		queueName = <some name>, required only if queueType=sonic
		queueType= memory OR sonic

service.AdminService.code	Default is com.commerceone.ccs.service.admin.AdminService	Code for AdminService

service.AdminService.args	Default is initialThreads=0,maximumThreads=1,queueCode=com.commerceone.ccs.kernel.queue.ThreadedEnvelopeQueue,queueType=memory,queueOwner=service	Valid arguments for AdminService include:
		queueOwner= service OR router
		queueName = <some name>, required only if queueType=sonic
		queueType= memory OR sonic

service.EventManager.code	Default is com.commerceone.ccs.service.event.EventManager	Code for EventManager
service.LogService.code	Default is com.commerceone.ccs.service.log.LogService	Code for LogService
service.XCBLTransformService.code	Default is com.commerceone.versiongateway.service.XCBLTransformService	Code for XCBLTransformService

Service Property Files

Service default.prop configuration files are located in a directory for each service to run on the server. The directory name is created from the type of service, the name of the service, and the name of the class as in this example:

<install:root>\runtime\servers\<portalrouter\xpc>\config\services\AuxiliaryService.

MessageStoreService.MessageStoreService.1_0 \default.prop

Log Service default.prop File

The following properties are set in the default.prop file of the LogService. This service subscribes to all events, writing them out to a server.

Properties	Value	Description
event.logfile.dir	Default is <install:root>/runtime/servers/<ServerName>/logs	Directory where the event logs are stored. Change this property if you wish to store events at a different location.
event.logfile.allevents	Default is eventlog which is <install:root>/runtime/servers/<ServerName>/logs/eventlog	Send all events to the event.logfile.dir directory with the given name, including ccs events. Each log file name is appended with the timestamp for when the log started.
event.logfile.ccsevents	Default is ccslog which is <install:root>/runtime/servers/<ServerName>/logs/ccslog	Send only the ccs events to the given filename with timestamp appended.

AsyncTransmitter and SyncTransmitter Service default.prop file

The default.prop file contains information on how the server identifies itself when making a connection, and the location of destinations it is trying to reach. The Transmitter service uses this authentication information while making connections to other locations.

Properties	Value	Description
marketparticipantid	Example: 55d8d1d6-77b1-1000-87d6-0a0000200001	MPID is obtained through MSB registration or MPID for test. Use Configure XPC to change this property.
authpref	cert/uidpswd Default is cert	Authentication preference for http/https. This can be cert (with ssl transport level certificate authentication) or uidpswd (with user level userid/password authentication). When Trading Partner connects to Portal Router, uidpswd is used. When Portal Router connects to Trading Partner, only cert is valid.
userid	Default is admin	ID used by the service when making the connection. Use Configure XPC to edit.

password		Encrypted (for production) or clear text (for test only) password used for authentication. To change the password, use Configure XPC.
sonicmq.authpref	none/uidpswd	Authentication preference for the sonic connection,.This can be uidpswd (with user level authentication of userid/password) or none. When Trading Partner connects to Portal Router, uidpswd is used. When Portal Router connects to Trading Partner, none should be used.

The following properties are the source of transmission information.

Properties	**Value**	**Description**
ccs.comm.tx.fs	true/false	A boolean flag for using the file system for transmission information or not. If this is true, then the one below should be false and vice-versa
ccs.comm.tx.ldap	true/false	A boolean flag for using the ldap for transmission information or not. If this is true, then the one above should be false and vice-versa

client.prop Property File

This property file is in the directory where the client application is started, or where it is specified when TransmitterFactory object is instantiated. There are two client.prop files that are configured by changes to Configure XPC.

<install:root>/bin contains an Outbound client.prop. This client.prop picks up the MarketSite connectivity configuration in Configure XPC. PingMarketSite and DocSender use this client.prop.

<install:root>\etc\config contains a client.prop that is pointed Inbound. It picks up the local configurations in Configure XPC. Invoker uses this client.prop.

Most properties in this file are set during the installation of the server and many of them are duplicates of properties that are listed in the server property files.

Properties	Value	Description
ccs.comm.em.fs	true/false	Boolean whether to use the filesystem for schema resolution or not. If set to true, the ccs.comm.em.ldap property should be false.
ccs.comm.em.fs.path	<install:root>\schema	If the above is set to true, this is the path where the schema documents are located.
ccs.comm.em.ldap	true/false	Boolean whether to use the ldap for schema resolution or not. If set to true, the ccs.comm.em.fs property should be false.

Client code can use events. The following parameters direct the event handling.

Properties	Value	Description
ccs.comm.event.language	en Default is en	Language to be used. Default is English.
ccs.comm.event.catalogloaddir	Default is <install:root>\runtime\servers\<ServerName>\eventcatalog\	Location of the event catalogs (a local directory).
ccs.comm.event.outputfile	Default is <install:root>\runtime\servers\<ServerName>\logs\clienteventlog	Location of the file where events from the client are placed.

The following specify the source of transmitter properties.

Properties	Value	Description
ccs.comm.tx.fs	true/false	Boolean whether or not to use the filesystem for transmission information. If set to true, the ccs.comm.tx.ldap property should be false.
ccs.comm.tx.ldap	true/false	Boolean whether or not to use the ldap for transmission information. If set to true, the ccs.comm.tx.fs property should be false

The following specify the cache information for transmitter:

Properties	**Value**	**Description**
ccs.comm.transmitter.cache	true/false	Use cache or not for transmitter including http(s), sonic.
ccs.comm.transmitter.cache.refresh	Example: 100	Time in second to cache the transmitter information.

To send a document, the following transmission properties must be available.

Properties	**Value**	**Description**
ccs.comm.transmitter.destination.name	Example: 55d8d1d6-77b1-1000-87d6-0a0000200001,+	List of destination MPIDs separated by comma(,)
<destinationMPID>.doctype	Example: Ping	List of document type separated by comma(,)
+.+.docformat	xml	Format used by the document, usually xml
+.+.protocols	http and/or ,https and/or ,sonic	Protocols for sending the document
+.+.protocol.http.args	Example: preference=3,destinationaddress=http://localhost:8008/xcc,qos=guaranteed,code=com.commerceone.ccs.comm.sender.http.HttpTransmitter	There should be one of these entries for each protocol type, including a preference count to help direct the transmitter to using the preferred protocol.
<destinationMPID>.<doctype>.protocol.https.args	Example: preference=1,destinationaddress=https://localhost:443/xcc,qos=guaranteed,code=com.commerceone.ccs.comm.sender.https.HttpsTransmitter	There should be one of these entries for each protocol type, including a preference count to help direct the transmitter to using the preferred protocol.
<destinationMPID>.<doctype>.protocol.sonic.args	Example: preference=2,destinationaddress=<nodename>::<queuename>	This is the syntax of the sonic transport.

Versioning Properties in client.prop

Key	Value	Description
transformation.registry	Full Path of TransformRegistry.xml file Default is <install:root>/schema/TransformRegistry.xml	TransformRegistry.xml file has meta data about transformations.
transformation.internalversion	20|22|30	The xCBL version supported by the client program. All incoming documents are converted to this xCBL version (if transformation exists) before client handles it.
transformation.externalversion	20|22|30	The lowest xCBL version supported by a Trading Partner your application transacts with. If the value of this key is the same as the value of above, the transformation is disabled.

Sonic Properties in client.prop

Sonic properties to set up by the Sonic Transport.

Properties	Value	Description
+.+.protocols	sonic	Designates SonicMQ as the available protocol.
+.+.protocol.sonic.args	preference=1,destinationaddress=<nodename>::<queuename>	The destinationaddress is the destination queue name.

Sonic broker connection parameters required.

Properties	Value	Description
sonicmq.broker.url	<tcp/ssl>://<BrokerMachine>:<BrokerPort4Client>	<tcp/ssl>://<BrokerMachine>:<BrokerPort4Client> Use Configure XPC to change.
sonicmq.broker.username		User id needed to connect to the broker as a JMS client.

sonicmq.broker.password		This is the encrypted password for the above user name.
sonicmq.syncresponsequeue.name		This is the response queue for the client to use to receive responses for synchronous requests sent.

These close modes are supported for SonicMQ.

Properties	**Value**	**Description**
sonicmq.connection.close	idle:<NumOfSeconds>/ always/never Default is idle:30	The idle mode closes connection idle for specified number of seconds. The idle timeout is in seconds, and should not be set lower than 30. The always mode always closes a connection after document send and reopens it before the next send. The never mode never closes the connection.

The following specify the sonic authentication preference.

Note .... This also can be set through Properties object instead of in client.prop.

Properties	**Value**	**Description**
sonicmq.authpref	uidpswd/none Default is uidpswd.	Authentication preference for the sonic connection. This can be uidpswd (with user level authentication of userid/password) or none. When Trading Partner connects to Portal Router, uidpswd should be used. When Portal Router connects to Trading Partner, none should be used.

If sonicmq.broker.url (described above) is defined as SSL, then these additional SSL parameters are required.

Properties	**Value**	**Description**
jms.client.ssl.enable	True	Enables SonicMQ JMS client SSL connection. **Do not change.**
jms.client.ssl.provider.class	Progress.message.net.ssl.iaik.iaikSSLImpl	Implementation class for the SSL provider. **Do not change.**

jms.client.ssl.cipher.suites	SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_RSA_WITH_RC4_MD5,SSL_RSA_WITH_RC4_SHA,SSL_RSA_WITH_IDEA_CBC_SHA,SSL_RSA_WITH_DES_CBC_SHA,SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA,SSL_DH_DSS_WITH_DES_CBC_SHA,SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA,SSL_DH_RSA_WITH_DES_CBC_SHA,SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA,SSL_DHE_DSS_WITH_DES_CBC_SHA,SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA,SSL_DHE_RSA_WITH_DES_CBC_SHA,SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA,SSL_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA,SSL_RSA_EXPORT1024_WITH_RC4_56_SHA,SSL_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA,SSL_RSA_EXPORT_WITH_RC4_40_MD5,SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5,SSL_RSA_EXPORT_WITH_DES40_CBC_SHA,SSL_DH_DSS_EXPORT_WITH_DES40_CBC_SHA,SSL_DH_RSA_EXPORT_WITH_DES40_CBC_SHA,SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA,SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	Cipher suites **Do not change.**.
jms.client.ssl.client.requireTrustedRoot	true/false	Use trust root certs to verify server certs. If false, the client is not required to have the trusted root of the server's certificate. Otherwise, the client needs to have the root of the server certificate.
jms.client.ssl.client.trustedRoot.dir	Example: <install:root>/certs/ca	Directory to store trusted root certificates, required only if jms.client.ssl.client.requireTrustedRoot is true

The following properties are set at the bottom of the **Configure XPC** user interface. When XPC makes outbound connection to MarketSite, the document could be routed through a proxy server located in the DMZ (demilitarized zone).

Outbound communications through a proxy server provide support for any http/https based proxy server that follows the standard HTTP 1.1. Use Configure XPC to edit.

Properties	Value	Description
proxySet	true/false Default is false	If true, outbound connection are routed to proxy server. If false, outbound connection is not routed to proxy server. If false, the following four properties keys are ignored.
proxyHost	<ProxyMachine>	Host machine where proxy server is installed. This value must be the fully qualified network address of the proxy server host machine. Example: ,proxyHost=sslproxy.mycompany.com This value should be the same as https.proxyHost below.
proxyPort	<ProxyPort>	Http port where proxy server is configured to allow http connection. This value should be the same as https.proxyPort below. Note: This port needs to be configured correctly on proxy server to allow successful outbound connection on that port. See your chosen proxy server's documentation on how to configure ports.
https.proxyHost	<ProxyMachine>	Host machine where proxy server is installed. This value must be the fully qualified network address of the proxy server host machine. Example: ,https.proxyHost=sslproxy.mycompany.co m This value should be the same as proxyHost above.
https.proxyPort		Https port where proxy server is configured to allow https connection. This value should be the same as proxyPort above. Note: This port needs to be configured correctly on proxy server to allow successful outbound connection on that port. See your chosen proxy server's documentation on how to configure ports.