

Commerce One MarketSite™

XML Portal Connector 4.1 Migration Guide



Hacienda Business Park, Bldg # 4
4440 Rosewood Drive
Pleasanton, CA 94588

www.commerceone.com

Commerce One MarketSite™

XML 4.1 Migration Guide

Version 1.0

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.

© 2001 Commerce One, Inc.

All rights reserved.

Commerce One, Many Markets. One Source., Global Trading Web, BuySite, MarketSite, Global Trading Platform, XML Common Business Library, XCBL, XML Development Kit, XDK, XML Commerce Connector, XCC, XML Portal Connector, XPC, MarketSite Builder, SupplyOrder, Content Engine, Content Refinery, Auto Catalog Publisher, XML Content Pipeline, XCP, Commerce One Auction Services and iExtract are either trademarks or registered trademarks of Commerce One, Inc. Enterprise Buyer and MarketSet are trademarks of Commerce One, Inc. and SAPMarkets, Inc. All other company, product, and brand names are trademarks of their respective owners.

This documentation and the software described constitute proprietary and confidential information protected by copyright laws, trade secret and other laws. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of Commerce One.

May 2001

Part Number 13-4.1-11-1

Contents

Preparing to Migrate	1
Migration Tasks	2
Setting Up Security	3
Certificates	3
Export Keystore Data	3
Import Keystore Data	4
Configuring MarketSite Communication	4
Migrating Filestore Directories	5
Migrating the Invoker	5
Processing xCBL 2.0 Documents	6
Migrating Generic EDI	7
Migrating DocSender Sample	8
Migrating XPC Framework-based Services	10
Migrating HotFS	11
Migrating AbstractDocumentService	12

XML Portal Connector (XPC) 4.1 Migration Guide

This document provides the steps to migrate XPC 3.2 and XPC 4.0 to XPC 4.1. It also provides guidelines for migrating AbstractDocumentService from XCC 3.2 to 4.1 and HotFS to 4.0.

Throughout this document it is assumed that XPC 4.1 is installed. However, XPC 4.0 might be sufficient to perform some tasks. Please refer to the *Migration Tasks* on page 2.

Note Consider installing XPC 4.0 or 4.1 on a different machine than where XPC 3.2 is installed. Installing XPC 4.1 on the same machine as XPC 3.2 makes XPC 3.2 unusable. It is not possible to revert back to XPC 3.2.

Preparing to Migrate

1. Shut down XPC3.2.
2. Backup MessageStore archive files located under the following directories:
 - <<install:root>>/messagestore
 - <<install:root>>/errormessagestore
 - <<install:root>>/runtime/servers/defaultserver/lostfound/lostdocs.prop
3. Export the Keystore data. See *Export Keystore Data* on page 3
4. Save the property files (default.prop) located in the directory <<install:root>>/runtime/servers/defaultserver/config/service
5. Save the server property files located in the directory <<install:root>>/runtime/servers/defaultserver/config/startup
6. Save any customized SOX schema files and SOX bean classes.

Migration Tasks

Perform the following migration steps.

Task	Document Reference	Required?	Tasks That Can Be Performed in XPC 4.0
1. Install XPC 4.0, if necessary	<i>XML Portal Connector Installation and Administration Guide, Version 4.0</i>	Yes	
2. Install XPC 4.1	<i>XML Portal Connector Installation and Administration Guide, Version 4.1</i>	Yes	
3. Import the Keystore data	<i>Setting Up Security</i> on page 3	No	Yes
4. Configure the MarketSite Communication	<i>Configuring MarketSite Communication</i> on page 4	Yes	No
5. Migrate the Filestore Directories	<i>Migrating Filestore Directories</i> on page 5	No	No
6. Migrate the Invoker	<i>Migrating the Invoker</i> on page 5	No	No
7. Migrate the Generic EDI	<i>Migrating Generic EDI</i> on page 7	No	No
8. Place the saved SOX schema and SOX bean classes, if any, into the new installation.		No	No
9. Migrate XPC Framework based services	<i>Migrating XPC Framework-based Services</i> on page 10	No	No
10. Migrate stand alone client applications	<i>Migrating DocSender Sample</i> on page 8	No	Yes

Task	Document Reference	Required?	Tasks That Can Be Performed in XPC 4.0
11. Migrate HotFS	<i>Migrating HotFS</i> on page 11	No	Yes
12. Migrate AbstractDocumentService	<i>Migrating AbstractDocumentService</i> on page 12	No	Yes

Setting Up Security

Certificates

The SecurityMigrateTool32 script exports XPC 3.2 keystore data and imports that data to XPC 4.1. It retains the certificates and encryption used with XPC 3.2.

If XPC 4.1 connects to a different MarketSite or uses a different encryption level, import the Certificates using CertManager described in the section “Certificate Manger” in the chapter “Security for Production Systems” in the *XML Portal Connector Installation and Administration Guide, Version 4.1*.

Export Keystore Data

To export keystore data from the XPC 3.2 server to the XPC 4.1 server, run the SecurityMigrateTool32 script:

1. Insert the XPC 4.1 CD on the XPC 3.2 server.

Note: Your CD will contain either 128-bit or 56-bit security information. The examples in this section use 128-bit.

2. Open a DOS command window and create the following directory:

C:\CommerceOne\Migration

3. Copy the files from

[Drive name]:\XPC_41\128-bit\Migration

to

C:\CommerceOne\Migration

4. Type: `cd C:\CommerceOne\Migration`

5. Type: `./SecurityMigrateTool32.bat`

The following directory now contains the keystore data to import to the XPC 4.1 server:

C:\temp\xpc-security-migration*

Import Keystore Data

On the XPC 4.1 server, follow these steps to run the SecurityMigrateTool40 script. This step will backup all existing security data before over-writing it.

1. If XPC 4.1 is on a different machine, copy the exported data into C:\temp\xpc-security-migration*

2. Insert the XPC 4.1 CD on the XPC 4.1 server.

3. Open a DOS command window and type:

`cd [Drive Name]: XPC_41\128-bit\Migration`

Note: Your CD will contain either 128-bit or 56-bit security information depending on the location of your site.

4. Type: `./SecurityMigrateTool40.bat`

5. Enter the XPC 4.1 server root path at the prompt.

For example:

`d:\CommerceOne\xpc41\runtime\servers\defaultserver`

The migration tool updates the files under the `certs` directory

6. Restart the server(s).

Configuring MarketSite Communication

MarketSite offers two methods of transport: SonicMQ and HTTPS.

■ SonicMQ transport configuration

- If you are a new Trading Partner to MarketSite 4.0 follow the steps in the section "Registering in MarketSite 4.0" in the chapter "Connecting to MarketSite" of the *XML Portal Connector Installation and Administration Guide, Version 4.0*.
- If you have previously registered with MarketSite 4.0 but want to use SonicMQ which is now available in XPC 4.1, follow the steps under the bullet "Manage SonicMQ Broker" in the *XML Portal Connector Installation and Administration Guide, Version 4.0*.
- HTTPS transport configuration

If you are using the HTTPS as a transport method, do the following:

1. Run **Start | Programs | XML Portal Connector 4.0 | Configure**
2. In the Transport Preferences box , select HTTPS for Preference 1 and select "None" for Preference 2 .

Select the HTTPS tab and enter the portal address and portal port.

4. In the Communication box, change the HTTPS Port if needed but keep default settings for the rest of the fields.
5. Save and exit.

After you have set up the transport method, follow the instructions in the section "Configuring XPC" in the chapter "Connecting to MarketSite" of the *XML Portal Connector Installation and Administration Guide, Version 4.0*.

Migrating Filestore Directories

The directories used by the FileStore Component for preconfigured services in XPC3.2 are not created in XPC 4.1. You must recreate the directories to use the new directory structure under `<<install:root>>/filestore` in XPC 4.1 or use XPC Manager to reconfigure the services.

Migrating the Invoker

Invoker in XPC 4.1 contains the following enhancements:

- Ping button, which submits a sample Ping document to XPC.
- Invoker supports the following xCBL 3.0 documents:

AdvanceShipNotice, Auction, AvailabilityCheckRequest, AvailabilityToPromise, Invoice, Order, OrderStatusRequest, PaymentRequest, PlanningSchedule, PriceCheckRequest, ProductCatalog, Quote, RemittanceAdvice, ShippingSchedule, TimeSeries, and TradingPartner.

Processing xCBL 2.0 Documents

The sample XPC services included with XPC 4.1 are backwards-compatible. The most recent set of sample services are capable of processing both xCBL 2.0 and xCBL 3.0 documents.

XPC 4.1 does not include the xCBL 2.0 sample documents. However to obtain the XPC 3.2 sample documents (AvailabilityCheck, OrderStatus, PriceCheck, and PurchaseOrder) perform the following steps:

1. Copy the XPC 3.2 sample documents to the corresponding directory in your new XPC 4.0 installation.

For example, copy:

```
$XPC32ROOT\sample\xpc\instances\*.*
```

to

```
$XPC40ROOT\sample\xpc\instances\
```

2. Modify the

```
$XPC40ROOT\runtime\servers\defaultserver\config\startup\service-start.prop
```

to include the XPC 3.2 Document services, as follows:

- To the startup.application.services parameter, add the following:

```
startup.application.services=XPCPurchaseOrder
```

- In the # Service Specs section, add the following:

```
service.XPCPurchaseOrder.code=com.commerceone.xpc.service.XPCDocumentServiceservice.XPCPurchaseOrder.args=initialThreads=0,maximumThreads=1,queueCode=com.commerceone.ccs.kernel.queue.ThreadedEnvelopeQueue
```

3. Modify the \$XPCROOT\etc\config\tranMap.prop file as follows:

- Append the following to transactionTypes parameter.

```
transactionTypes=PriceCheck,PurchaseOrder,AvailabilityCheck,OrderStatus
```

■ Add the following:

```
#TransactionType = DocumentTypes in a given Transaction
```

```
PriceCheck.elements=PriceCheckRequest,PriceCheckResult,Error
```

```
PriceCheck.displayName=Price Check
```

```
PriceCheck.requestMode=sync
```

```
PriceCheck.sync.timeout=10000
```

```
PurchaseOrder.elements=PurchaseOrder,PurchaseOrderResponse,Error
```

```
PurchaseOrder.displayName=Purchase Order
```

```
PurchaseOrder.requestMode=peer-peer
```

```
PurchaseOrder.sync.timeout=10000
```

```
AvailabilityCheck.elements=AvailabilityCheckRequest,AvailabilityCheckResult,Error
```

```
AvailabilityCheck.displayName=Availability Check
```

```
AvailabilityCheck.requestMode=sync
```

```
AvailabilityCheck.sync.timeout=10000
```

```
OrderStatus.elements=OrderStatusRequest,OrderStatusResult,Error
```

```
OrderStatus.displayName=Order Status
```

```
OrderStatus.requestMode=sync
```

```
OrderStatus.sync.timeout=10000
```

■ Append the following list of documents to requests:

```
requests=PurchaseOrder
```

Migrating Generic EDI

The following changes have been made to Generic EDI (GEDI).

- The location of the `map.txt` file which XPCGEDIOutbound service uses was moved from `<<install:root>>/gedi/tpid_map` to `<<install:root>>/tpid_map`.
- Descriptor and CreateGEDIEnvelope components have been changed. If you have modified the configuration of these components, consult the component description in the “Generic EDI Components” Appendix in the *XML Portal Connector Developer Guide and API Reference* to make the proper change.
- GEDI services are enabled/disabled from XPCManager, not from SetupUI.

Migrating DocSender Sample

DocSender Sample demonstrates a preferred way of sending documents to MarketSite in a standalone client mode.

The new DocSender Sample contains the following features:

- Ability to send to the XPC server via SonicMQ.
- Allows users to specify all parameters from either the command line or the `client.prop` (bin directory) file.
- Supports all document types and transmission modes (synchronous, peer-peer and one-way).
- Ability to generate and transmit randomized purchase order document with specified size.
- We recommend using the new DocSender Sample directly from the installation and preserving the existing configuration. To do this, perform the following steps:
 1. Back up the old `bin/client.prop` for later reference.
 2. Install XPC 4.1.
 3. Start the configureUI.
 4. Enter the HTTP transport preferences (portal address and portal port) used in the old `client.prop` file.
 5. configureUI will modify the `bin/client.prop` file.
 6. If you have more additional properties in your existing `bin/`

`client.prop` file and need to hand-modify the property file, use the new `bin/client.prop` as the base, and add the properties from the old `client.prop` file. Do not directly overwrite the new `client.prop` using the old one.

7. For more information on DocSender Sample, refer to the section “Setting Up a Client Environment” in the appendix “Using a Transmitter API” in the *XML Portal Connector Developer Guide and API Reference*.

Migrating XPC Framework-based Services

You can migrate the XPC Framework-based services including XPC3.2 out-of-box services as follows:

1. Place the `default.prop` file for the service to be migrated in the XPC 4.1 installation directory.

For example, if the `default.prop` file for the original service named "XPCPurchaseOrder" is located at:

```
c:\commerceone\xpc3.2\runtime\servers\defaultserver\config\service\DocumentService.XPCDocumentService.XPCPurchaseOrder.1_0\default.prop
```

and XPC 4.1 is installed at `C:\commerceone\xpc4.1` with `defaultserver` as the server name, copy the file to

```
c:\commerceone\xpc4.1\runtime\servers\defaultserver\config\service\DocumentService.XPCDocumentService.XPCPurchaseOrder.1_0\default.prop
```

2. Edit the `default.prop` to remove any line continuation characters and space characters on the left and right side of the character '=' which separates the key and value pair.
3. Start XPCManager. Refer to the "XPC Manager" chapter in the *XML Portal Connector Developer Guide and API Reference*.
4. Click the "Add New Service" button from the XPC Manager Services page. Enter the service name and select the service type.
 - If you select "Document", enter only the `initialThreads` and `maximumThreads` using the "Add New Keys" box.
 - If you select "Timed", enter the necessary keys and corresponding values.

For available keys, refer to the "XPC Manager" chapter in the *XML Portal Connector Developer Guide and API Reference*, section "Configuring a Service"/ "Service Level Configuration" / item number 4.

For a specific value to be used, refer to your original setting which can be found in the file `service-start.prop` in the XPC 3.2 installation.

5. Click the "Save" button to save the service level configuration. This completes the XPC Framework based service migration.

Migrating HotFS

To migrate from HotFS 3.2 to HotFS 4.0, preserving your current configuration, follow these steps:

1. Back-up the following files:

```
{XPCROOT}\bin\config.xml  
{XPCROOT}\runtime\hotFS\hotfsServiceconfig.xml
```

2. Install the new HotFS 4.0.

3. Replace the {XPCROOT}\bin\config.xml and {XPCROOT}\runtime\hotFS\hotfsServiceconfig.xml with the backed-up ones.

4. Edit the file {XPCROOT}\bin\config.xml to add the new transformation elements:

```
<TransmissionConfigurations>  
<HotFSConfigurations>  
...  
  <TransformationSettings>  
    <Registry>C:/commerceone/Xpc/schema/  
TransformRegistry.xml</Registry>  
    <InternalVersion>20</InternalVersion>  
    <ExternalVersion>20</ExternalVersion>  
  </TransformationSettings>  
</TransmissionConfigurations>  
</HotFSConfigurations>
```

where the values for entries *<Registry>*, *<InternalVersion>*, and *<ExternalVersion>* can be obtained from your {XPCROOT}\bin\client.prop file. Refer to the *XML Portal Connector Installation and Administration Guide, Version 4.0* for information on customizing these values.

5. If you want to support a proxy server, add the following additional entries to the {XPCROOT}\bin\config.xml file:

```
<TransmissionConfigurations>  
<HotFSConfigurations>  
...  

```

```
<ProxyServerSettings>
  <ProxySet>true</ProxySet>
  <ProxyHost>abc</ProxyHost>
  <ProxyPort>80</ProxyPort>
  <HttpsProxyHost>abc</HttpsProxyHost>
  <HttpsProxyPort>80</HttpsProxyPort>
</ProxyServerSettings>

<TransformationSettings>
  <Registry>C:/commerceone/Xpc/schema/
TransformRegistry.xml</Registry>
  <InternalVersion>20</InternalVersion>
  <ExternalVersion>20</ExternalVersion>
</TransformationSettings>
</TransmissionConfigurations>
</HotFSConfigurations>
```

Migrating AbstractDocumentService

To migrate XCC 3.2 services that you developed, follow these steps:

1. Recompile the services with the library shipped with XPC 4.1. Use JDK 1.2.2 on XPC 4.0 or JDK 1.3_02 on XPC 4.1. Refer to the *XML Portal Connector Installation and Administration Guide, Version 4.0* and *XML Portal Connector Installation and Administration Guide, Version 4.1* for details.
2. Add the services.

The ConfigService tool adds services to the properties file (service-start.prop) and adds the queue to the Sonic broker if SonicMQ is the queue type in the service argument. The format of server-start.prop has not changed since XPC 3.2. The ConfigService tool is particularly beneficial for SonicMQ configuration.

Here is a description of how to use it:

1. Invoke ConfigService by running <<install:root>>/bin/ConfigService.bat from the DOS command prompt.

2. In the "Root Directory" field, enter the root directory of XPC installation.

For example, `d:\commerceone\xpc41`

In the "Server Directory" field, enter the path name (after the root directory) of the XPC server to which this service is to be added.

For example, `runtime\servers\xpc1`.

3. Click the "OK" button.

4. Click the "Add Service" button.

- In the "Service Name" field, enter the name of the service to be added. For example, "PingService".
- In the "Service Type" field, enter the type of service. It could be Base Service, Application Service, Comm Service, Auxiliary Service, and Timed Service. Typically "Application Service" is selected.

- In the "Package Name of Service Byte Code" field, enter the package name of the service byte code. For example, "com.commerceone.ccs.service.ping.PingService".

- In the "Service Start-up Arguments" field, enter the startup arguments required by the service. Typically, this is a list of key-value pairs separated by a comma.

- If you select "Memory Queue Type", the arguments are:

```
queueType=memory,queueCode=com.commerceone.ccs.kernel.queue.ThreadedEnvelopeQueue,maximumThreads=1,initialThreads=0,queueOwner=service
```

Note: Be sure to uncheck the "Create Queues" box near the bottom of the screen.

- When sonic queue type is selected, the arguments are:

```
queueType=sonic,queueName=XPC_<ServerGroupName>_<ServerName>_<ServiceName>,queueCode=com.commerceone.ccs.kernel.sonicmq.SonicThreadedEnvelopeQueue,maximumThreads=1,initialThreads=0,queueOwner=service
```

where *<ServerGroupName>* should be the same as *<ServerName>* (e.g. XPC1) and *<ServiceName>* is the same as entered in the "Service Name" field.

Note: It is important to have the queue name follow this convention.

- In the "Service Classpath" field, enter the classpath specification required by the service. Typically this is a list of path names separated by a semicolon.

For example,

D:\commerceone\xpc1\lib\ccs_server.jar.

- In the "Service Directory" field, copy the XPC 3.2 or XCC 3.x service directory located at
<XPC3.2Install>\runtime\servers\<ServerName3.2>\config\service
to XPC 4.1
<XPC4.1Install>\runtime\servers\<ServerName4.1>\config\service.

Then use the "Browse" button to browse to this directory.

Note: Be sure to check the "Create Queues" box near the bottom of the screen.

- Click on "Add Service" button.