eMarketPlace Catalog™

Installation and Administration Guide

Version 2.0

December 2001

Publication: December 18, 2001 5:34

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

www.commerceone.com

eMarketPlace Catalog Installation and Administration Guide, Version 2.0

December 2001

Preface ix
Related Documentation ix
If You Need Help x

1. Overview 1-1
Introduction 1-1
Solution Architecture 1-1
Integration with Content Production Applications 1-1
Integration with Procurement Applications 1-3
Security 1-5
SiteMinder-based Security 1-5
OCI-based Security 1-8
Application Architecture 1-10
Application Interface 1-12
Search Capabilities 1-13
Administrative Features 1-14
Agents 1-14
Actions 1-15
Chores 1-15
Administrator Interfaces 1-15
Installation Overview 1-16
Installation Sequence 1-17

2. eMPC Installation 2-1
SQL Server Installation 2-1
SQL Server Host Requirements 2-1
Installation 2-2
eMPC Application Installation 2-2
eMPC Application Host Requirements 2-2
Installation 2-3
Starting the eMPC Application 2-3
Starting the eMPC Application as a Service 2-3
Starting the eMPC Application as a Program 2-3
Accessing the Administrator Interfaces 2-5
Accessing the User Interface 2-5

3. Initial Configuration 3-1
Prepare the SQL Server Database 3-1
Set the Virtual Memory 3-2
Edit the eMPCDiskServer and eMPCSQLConnector Agents 3-3
Configuring the TempDirectory Attribute (Optional) 3-5
Automatic Database Configuration 3-5

4. JRun eMPC Proxy Web Server Installation and Integration 4-1
Overview 4-1
Prerequisites 4-2
eMPC Proxy Installation and Setup 4-2
Install the eMPC Proxy 4-2
Set the eMPC Proxy Application Properties 4-6
Optional IIS Configurations 4-11
Creating a Separate eMPC Web Site for Multiple Application Hosting in IIS 4-11
Configuring IIS for HTTPS 4-12

5. Configuring MarketSiteBuilder (MSB) for eMPC 5-1
Overview 5-1
Implementation 5-2
Install and Configure SiteMinder Web Agent 5-2
Create the eMPC Service for End-user Catalog Access 5-3
Create the eMPC Catalog Admin Service 5-4
Create the eMPC Upload Admin Service 5-4
Create a Role to Contain eMPC Administrator Services 5-5
Expose the eMPC (End-user) Service to Trading Partner Users 5-5
Enable SiteMinder WebAgent 5-5

6. Procurement Application Integration 6-1
Integrating with Enterprise Buyer Desktop 6-2
Creating and Defining the eMPC Supplier in EBD 6-2
On Demand Supplier Registration 6-4
Integrating with Enterprise Buyer Professional 6-5
EBP Configuration 6-5
SAPSupplierLookupTable 6-6
OCI Mapping Table 6-7

7. Actions and Chores 7-1
Actions 7-1
MonitorEvents 7-1
UpdateCurrencyConversionMatrix 7-2
SaveApplication 7-2
xCBLConnectUnconnectedCatalogs 7-3
xCBLTransferCatalogs 7-3
Chores 7-4
Scheduling Chores 7-4
Types of Chores 7-5

8. Additional Administrative Tasks 8-1

Update Currency Exchange Rates 8-1

9. Localization 9-1

Localizing the User Interface 9-1
Spelling Parameters 9-1
Language Files 9-2
Language File Syntax 9-2
Language Files Types 9-4
Creating Language Files 9-6
Preconfigured Languages 9-6
Localizing to a Preconfigured Language 9-7
Localizing to a Non-Preconfigured Language 9-7
Localizing the Online Help 9-8

A. Setting up eMPC in a High Availability Environment A-1

Overview A-1
Prerequisites A-1
Software Requirements A-1
Hardware Requirements A-2
Additional Requirements A-2
Catalog Server Installation A-2
Create a New Cluster Group for the eMPC Catalog Server A-2
Create a Physical Disk Resource for the eMPC Catalog Server A-2
Install the eMPC Catalog Server A-3
Create a Generic Service Resource for the eMPC Catalog Server A-3
Create an IP Address Resource for the eMPC Catalog Server A-4
Create a Network Name Resource for the eMPC Catalog Server A-5
Upload Server Installation A-5
Create a New Cluster Group for the eMPC Upload Server A-5
Create a Physical Disk Resource for the eMPC Catalog Server A-6
Install the eMPC Upload Server A-6
Create a Generic Service Resource for the eMPC Upload Server A-7
Create an IP Address Resource for the eMPC Upload Server A-8
Create a Network Name Resource for the eMPC Upload Server A-9
Edit the Registry A-9
Editing the Registry on the Second Node A-10
Editing the Registry on the First Node A-10
Test the Installation A-10
Create the CartState File Share A-11
Create a New Cluster Group for the CartState File Share A-11
Create a Physical Disk Resource for the File Share A-11
Create a File Share Resource A-12
Create an IP Address Resource for the File Share A-12

Create a Network Name Resource for the File Share A-13
Test the File Share Creation A-13
Create the TransferDirectory File Share A-14
Create a New Cluster Group for the TransferDirectory File Share A-14
Create a Physical Disk Resource for the File Share A-14
Create a File Share Resource A-15
Create an IP Address Resource for the File Share A-15
Create a Network Name Resource for the File Share A-16
Test the File Share Creation A-16
Complete Your Setup A-17

B. Configuring eMPC in a MarketSite 3.2 Environment B-1
Installation and configuration B-1

C. Troubleshooting C-1
eMPC Application-related Log Files C-4
eMPC Proxy-related Log Files C-5
IIS Log Files C-5
JRun Admin Log Files C-5
JRun eMPC Proxy Log Files C-5
eMPC Proxy Log Files C-5

Preface

This book, *eMarketPlace Catalog Installation and Administration Guide*, describes the concepts and procedures you need to know to install and administer Commerce One's eMarketPlace Catalog (eMPC) application. It is intended for system administrators responsible for installing the eMPC application and using it to generate catalogs for a MarketSite/MarketSet portal. These system administrators must have expert-level knowledge of the following:

- Commerce One's MarketSite or MarketSet platforms
- Microsoft SQL Server
- Microsoft Windows 2000
- Network terminology and concepts
- XML Common Business Library (xCBL) standards

Related Documentation

Read the following documents for more information on the eMPC application and eMPC-related topics:

- *Content Engine Administration Guide*
- *Catalog Staging Area Installation Guide for SQL Server*
- *Catalog Staging Area Administration Guide*
- *xCBL 3.0 for Catalog Content*
- *xCBL Web Site (http://www.xcbl.org)*
- *MarketSite Builder Administration Guide*

If You Need Help

In addition to the information in this manual, the eMPC application has a built-in on-line help system. To access the online-help for a specific page, go to that page and click on the help icon, which appears as follows:

Also, each Commerce One installation with a support contract has one or more persons designated to contact Technical Support. If you cannot resolve a problem by using the Commerce One manuals or online help, ask the designated person to contact Technical Support via email **(csc@commerceone.com)** or phone **(925-941-5959)**.

1 Overview

Introduction

Commerce One's eMarketPlace Catalog (eMPC) is an application that system administrators in e-marketplaces can use to produce a multi-supplier catalog. System administrators host and maintain the catalog at either a private or public e-marketplace, and end-users such as buyers access the catalog with an Internet browser.

Solution Architecture

In MarketSite/MarketSet environments, the eMPC application integrates with content production and management applications as well as e-procurement applications. This integration streamlines the processing and flow of information from supplier organizations to prospective buyers. Buyers can access the eMPC catalog from their browsers and use the eMPC application's intuitive yet powerful search capabilities to find the products they wish to purchase.

Integration with Content Production Applications

The eMPC application generates on-line catalogs for use with MarketSite and MarketSet portals. To generate these on-line catalogs, the eMPC application requires catalog data from supplier organizations, such as product lists, product attributes, supplier information, pricing information, and so forth, which it receives in the form of xCBL documents.

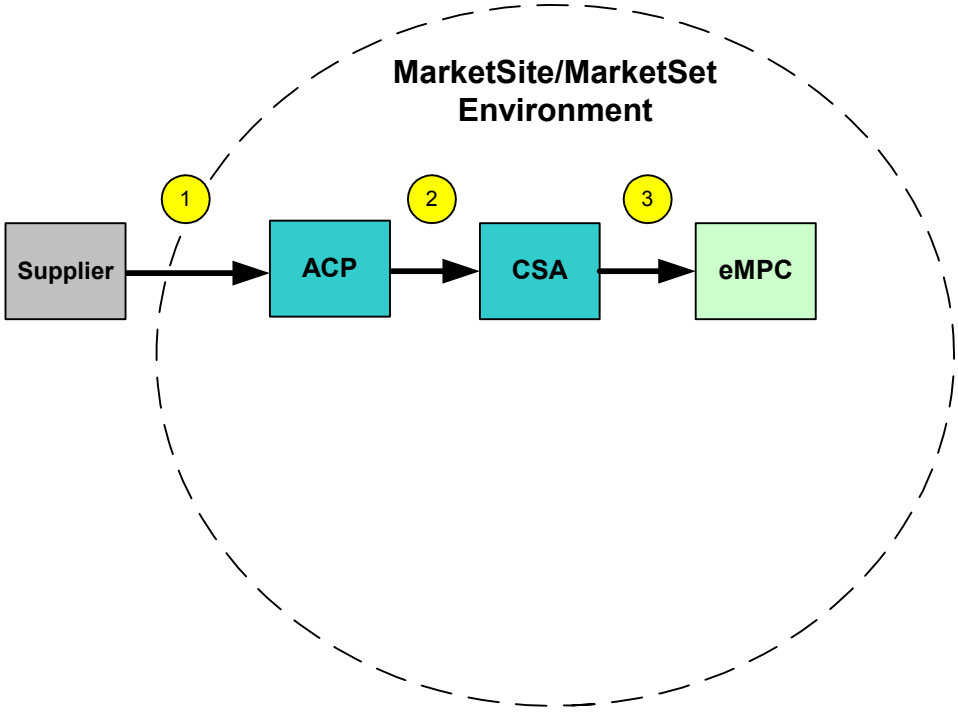
Supplier organizations load catalog data directly into MarketSite or MarketSet environments using the Auto Catalog Publisher (ACP) application. The ACP application is a Marketplace gateway for suppliers that automatically processes high quality content into a format that is acceptable to eProcurement applications. Supplier content of lower quality requiring additional processing (such as catagorization) may be processed by a licensed Commerce One Content Refinery before being submitted to the ACP application.

After processing the catalog data, the ACP application transmits the xCBL document to the eMPC application via the Catalog Staging Area (CSA), which is a set of tools that enable administrators to control and manage content in an on-line catalog.

Note: The eMPC application exclusively accepts xCBL documents that undergo processing by the ACP application. Some of these documents may originate from the XCP refinement of catalog data from suppliers.

Figure 1-1 illustrates the flow of data between the ACP, XCP, CSA, and eMPC applications.

Figure 1-1 Document Flow Between the ACP, CSA, and eMPC Applications



1. A supplier creates a document that contains catalog data and sends it to the ACP application, which verifies the format of the document.
2. The ACP application posts the processed xCBL document to the CSA application. Administrators use the CSA application to approve or reject catalogs, and determine which commodities Buyers browsing the catalog can view.
3. An eMPC Administrator uses the CSA application to approve or reject the

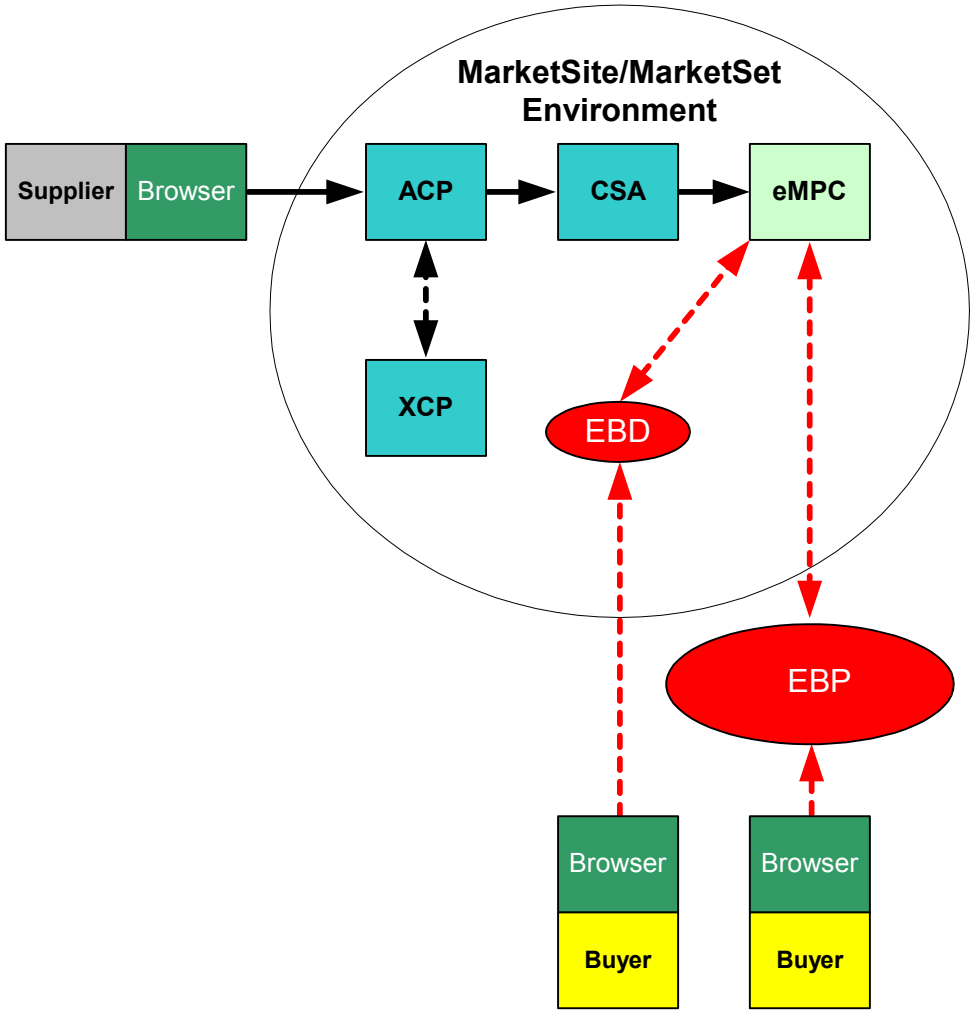
xCBL document. The CSA application then posts approved catalog data and Buyer view settings to the eMPC application.

Integration with Procurement Applications

The eMPC application uses SAP's Open Catalog Interface (OCI) protocol. This allows for out-of-the-box integration with various hosted (existing at the same portal as the eMPC application) and external (existing outside of the portal) procurement applications, such as Commerce One's Enterprise Buyer Desktop (EBD) application and SAP's Enterprise Buyer Professional (EBP) application. Integrating the eMPC application with one or more procurement applications enables buyers to access an eMPC catalog from their procurement application, search for products, add products to a shopping cart, and have their selections automatically returned to the requisition form within the procurement application for approval and ordering.

Figure 1-2 illustrates the flow of data in a MarketSite/MarketSet environment. The illustration shows an eMPC application that is accessible by both hosted and external procurement applications. Both types of procurement applications are depicted as red ellipses. Perforated arrows indicate data exchanges between the eMPC application, procurement applications, and the user's browsers; solid arrows indicate catalog data being sent from suppliers to the eMPC application through the XCP, ACP, and CSA applications.

Figure 1-2 The eMPC Application in a MarketSite/MarketSet Environment



Security

The eMPC application is compatible with two types of security setups:

- SiteMinder-based security — For use with an eMPC application that is accessible by one or more procurement applications that are hosted at the same portal. This type of security works with MarketSite 4.x and MarketSet 2.x.
- OCI-based security — For use with an eMPC application that is accessible by one or more procurement applications that may or may not be hosted at the same portal. This type of security works with MarketSite/MarketSet environments that use the MarketSite 3.1 (or higher) platform.

Most portals are likely to choose SiteMinder-based security for single-sign-on integration with other MarketSite Portal services. The decision to choose OCI-based security will be influenced by the use of MarketSite 3.x (for which SiteMinder is not available) or by the existence of non-hosted procurement applications. It is not necessary to choose OCI-based security to support eMPC OCI integration with procurement applications, as the eMPC application integrates via OCI regardless of the security scheme.

SiteMinder-based Security

You can use the Netegrity SiteMinder product to create a secure operating environment for an eMPC application that is accessible by one or more procurement applications that are hosted at the same portal. This type of security works with MarketSite/MarketSet environments that use the MarketSite 4.x platform. You can also use the SiteMinder product to create an environment in which users have seamless access to services for which they are authorized, but are prevented from accessing services for which they are not authorized. This is known as Single Sign-On (SSO).

Introduction to SiteMinder

Versions 4.1 and higher of the MarketSite platform come with the Netegrity SiteMinder product. The SiteMinder product provides you with the tools necessary to create a secure MarketSite/MarketSet environment that uses SSO.

The SiteMinder product has the following components:

- SiteMinder Database — The Netegrity SiteMinder product has its own LDAP database, which contains user, role, and service information created in the MSB interface.
- SiteMinder Web Agents — SiteMinder Web Agents regulate access to the applications and services in a MarketSite/MarketSet environment. Most setups require the use of multiple SiteMinder Web Agents: one for every application or service hosted in that MarketSite/MarketSet environment. With SiteMinder-based security, end-users do not know the URL for the application that they are trying to access. Instead, they are given the URL for that application's SiteMinder Web Agent. If the SiteMinder Web Agent

authenticates the user, the user can access the application.

- SiteMinder Policy Server — Each installation of the Netegrity SiteMinder product contains one instance of the SiteMinder Policy Server. The SiteMinder Policy Server resides on its own machine and serves data stored in the SiteMinder database to the SiteMinder Web Agents.

Conceptual Overview

When a user logs into the MarketSite/MarketSet environment through the MarketSite Builder (MSB) application, the MSB application checks its LDAP database to authenticate the user and determine which services they can access in the MarketSite/MarketSet environment. The MSB application then creates a credential cookie that contains the user's credential information, and places the credential cookie on the user's machine.

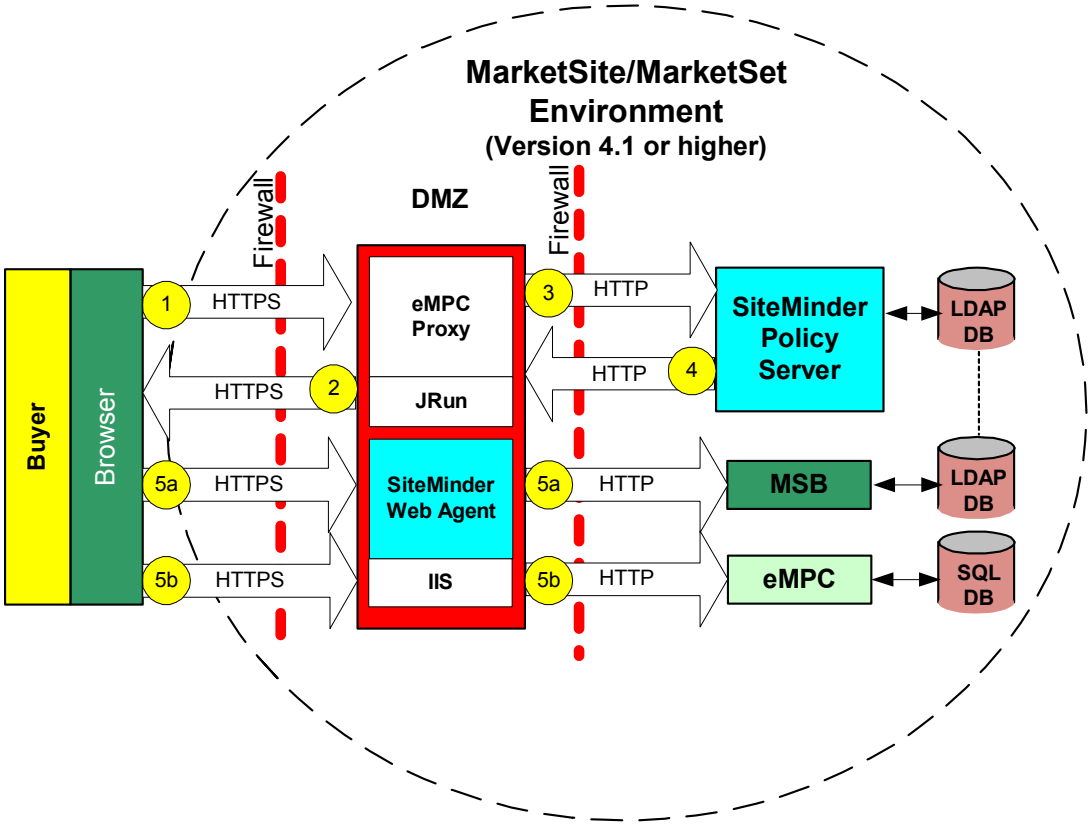
When a user wants to access a particular application after logging into the MarketSite/MarketSet environment, they enter the URL for that application's SiteMinder Web Agent. The user's browser then sends the credential cookie, set by a successful MSB login, to the SiteMinder Web Agent.

Note: If the user deletes the credential cookie from their machine or the credential cookie expires, the user must once again log into the MarketSite/MarketSet environment through the MSB application.

If the SiteMinder Web Agent does not receive the credential cookie, access to the application is denied and the user is sent to the MSB login page; if the SiteMinder Web Agent finds the credential cookie, it connects to the SiteMinder Policy Server to determine if the user is authorized to access the application. The SiteMinder Policy Server queries the SiteMinder database and returns the authorization information (either positive or negative) to the SiteMinder Web Agent. If the results of the SiteMinder Policy Server query are positive, the user can access the application; if the results are negative, the user is sent to the MSB login page.

Figure 1-3 illustrates an eMPC application operating in a SiteMinder-based secure environment.

Figure 1-3 SiteMinder-based Security



1. A user enters the URL for the eMPC application's SiteMinder Web Agent in a browser.
2. The SiteMinder Web Agent checks the user's machine for the MSB-placed credential cookie.
3. If the MSB-placed credential cookie is not found, the end-user is sent to the MSB application login screen; if the SiteMinder Web Agent finds the MSB-placed credential cookie on the user's machine, the SiteMinder Web Agent checks with the SiteMinder Policy Server to find out if the end-user is authentic and authorized to access the eMPC application.
4. The SiteMinder Policy Server queries its database (which is synchronized with the MSB database) to find the user's authentication and authorization information, and informs the SiteMinder Web Agent as to whether or not the user is both authentic and authorized to access the application.

5. The SiteMinder Web Agent does one of the following:
   a. If the end-user is either unauthentic or unauthorized to use the eMPC application, the Web Agent sends the end-user to the MSB application login screen.
   b. If the end-user is both authentic and authorized to use the application, the Web Agent allows the request to pass to the eMPC Proxy, which redirects the request to the internal eMPC server.

OCI-based Security

If you do not use SiteMinder security, you can still create a secure operating environment for the eMPC application by using SAP's OCI protocol. OCI-based security is ideal for an eMPC application that is accessible by procurement applications that may or may not be hosted at the same portal. It is for use with MarketSite/MarketSet environments that use the MarketSite 3.1 (or higher) platform.

Conceptual Overview

OCI security involves the use of a dedicated security servlet running on the Web Server, which exists between first and second firewalls. When a buyer uses a procurement application to attempt to access the eMPC application, the procurement application passes the buyer's user credentials (username and password) and BUYER_MPID to the security servlet. The security servlet checks the user credentials and BUYER_MPID against the data in the MarketSite platform database. If the user is authentic and authorized for access, the proxy server directs the user to the eMPC application.

OCI-based security requires you to install the following components on a single machine existing between the initial and secondary firewalls:

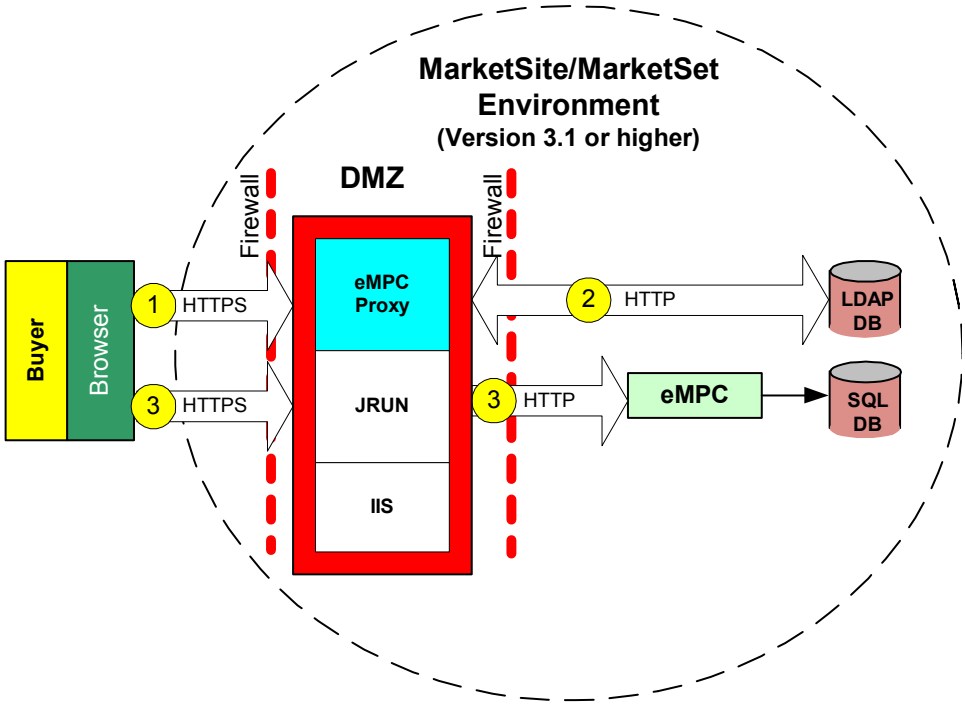
- A web server
- JRun application server
- eMPC Proxy Servlet

As with SiteMinder-based security, users are not given the URL for the eMPC application. Instead, they are given the URL for the eMPC servlet. When a user attempts to access the eMPC application using the eMPC servlet URL, the user's procurement application sends the OCI Username and Password data to eMPC Servlet. The eMPC Servlet checks to see if an entry for the specified username, password, and buyer_mpid exists in the LDAP database. If it does not exist, the eMPC Servlet denies the user access; if it does exist, the eMPC Servlet sends the user to the eMPC application.

While OCI-based security option does not support SSO capability between the eMPC application and other MarketSite/MarketSet applications, it does support transparent login between procurement applications and the eMPC application. This means that users can use their procurement application to seamlessly access the eMPC application without having to log in multiple times.

Figure 1-4 illustrates an eMPC application operating in a OCI-based secure environment.

Figure 1-4 OCI-based Security.



1. The user attempts to access the eMPC application by entering the URL for the eMPC Proxy in a browser.

Note: The eMPC Proxy requires that users provide a valid MPID, Username, and Password combination before they connect to the eMPC application; otherwise, the eMPC application displays an "Authorization Denied" screen. If a user has successfully connected and then changes the MPID or user identify to spoof someone else, the servlet disconnects that user from the eMPC application and displays the "Authorization Denied" screen.

2. The eMPC Proxy Servlet checks the LDAP database to see if the user is authentic and authorized to access the eMPC application.
3. If the user is either unauthentic or unauthorized to use the application, the eMPC Servlet denies the user access; if the user is both authentic and authorized to use the application, the eMPC Servlet sends the user to the application.

Application Architecture

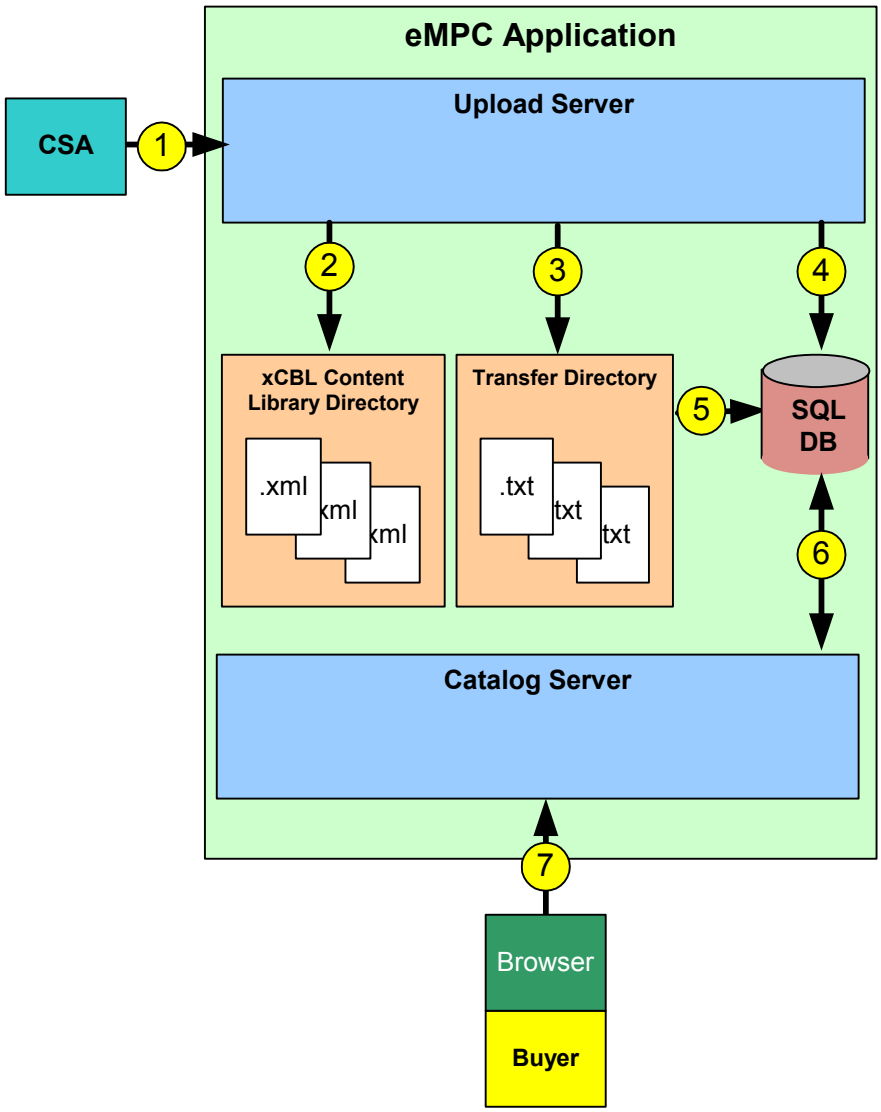
The core functionality of the eMPC application is based on Commerce One's Content Engine technology. The eMPC application utilizes two or more instances of a modified Content Engine application running on a single machine and sharing a single SQL database running on a separate machine. One instance of the Content Engine functions as the Upload Server while the other one functions as a Catalog Server.

Note: For information on advanced eMPC installations, see Appendix A, Setting up eMPC in a High Availability Environment.

The Upload Server is the point of entry for the eMPC application. It receives an xCBL document from the CSA via HTTP, and stores a copy of the xCBL document on disk in the xCBL Content Library Directory, which contains a current copy of every xCBL catalog in the file system of the computer hosting the eMPC application. The Upload Server also generates SQL Bulk Upload files that contain the catalog data from the xCBL document. These SQL Bulk Upload files appear in the Transfer Directory, which is a storage area that is accessible by both the SQL database and the Upload Server. After generating the SQL Bulk Upload files, the Upload Server notifies the SQL database, which retrieves the files from the Transfer Directory and stores the catalog data. The Catalog Server uses the catalog data in the SQL database to render on-line catalogs that end-users can access with their browsers.

Figure 1-5 illustrates the eMPC architecture, as well as the flow of an xCBL document through the eMPC application.

Figure 1-5 eMPC Application Architecture and Document Flow



1. The CSA application loads an xCBL document into the Upload Server.
2. The Upload Server checks the xCBL Content Library Directory and does one of the following:
   - If the xCBL file contains data for a new catalog, the Upload Server copies the xCBL file to the xCBL Content Library Directory.
   - If the xCBL file contains new data for an existing catalog, the Upload Server renames the previous version of the xCBL file by adding a ~#~ extension to the file, in which # corresponds to the sequence in which the file was received. The Upload Server then creates a new version of the existing xCBL catalog that contains the new catalog data, and stores this version in the

xCBL Content Library Directory.

3. The Upload Server generates a SQL Bulk Upload file (in text format) that contains the catalog data, and places it into the Transfer Directory.
4. The Upload Server notifies the SQL Server database about the existence of the SQL Bulk Upload file.
5. The SQL Server database retrieves the SQL Bulk Upload file from the Transfer Directory.
6. The SQL Server database transfers the catalog metadata to the Catalog Server, which queries the SQL Server database for catalog data.
7. An end-user accesses the Catalog Server via a browser.

Application Interface

Both end-users and administrators access the eMPC application and its catalogs through an Internet browser, and navigate the interface using familiar Web techniques, such as clicking on hyperlinks. The eMPC application, however, renders different interfaces for different users. End users (such as buyers) access a user interface on the Catalog Server that restricts them to only performing functions such as browsing catalogs and selecting products. Administrators access administrator interfaces on both the Catalog Server and Upload Server components. These administrator interfaces have controls that allow administrators to perform functions such as scheduling Chores and customizing the appearance of the eMPC user interface. Administrators can also configure the eMPC application to render catalogs in a specific language if they are creating catalogs for international users.

Caution: The eMPC administrator interfaces are very powerful, giving administrators access to numerous capabilities that can significantly alter the behavior of the eMPC application. Most of these capabilities are not essential to normal eMPC administration and are therefore not covered in this book. Commerce One recommends that you only use the capabilities described in this book; however, if you want more information on these undocumented features, see *Content Engine Administration Guide*.

If a user or administrator knows the URL for an interface and has the proper authorization, the user can access that interface from any computer with a browser without having to install any other applications on their computer.

Search Capabilities

The eMPC application has different types of search capabilities that enable system administrators to accomplish administrative tasks quickly, and end-users to find products with ease. These search capabilities are:

- Taxonomic Search
- Parametric Search
- Keyword Search

Taxonomic Search

A taxonomy defines a structure for a classification of objects. Companies and industry organizations have various taxonomies to define products within certain vertical markets such as electronic components and industrial components.

A taxonomic search is a search through a classification system or taxonomy that enables users to find all instances of a particular category or sub-category. The eMPC application's taxonomic search capability allows end-users and administrators to traverse such a structure by following the appropriate links.

For example, consider a taxonomy that contains the super-category "Writing Instruments" and sub-categories under that super-category, such as "Pens," "Pencils," and "Markers." A user looking for the "Pens" sub-category must first click the link of the appropriate super-category ("Writing Instruments"). When the user does this, the eMPC application displays all instances of the Writing Instrument category in the catalog—for instance, all pens, pencils, and markers. Users can quickly narrow down the list by clicking a sub-category.

Note: The default taxonomy used by the Content products is UNSPSC, but the products also support custom taxonomies.

Parametric Search

Categories in a catalog have certain attributes, such as color, size, manufacturer, and so on. The eMPC application's parametric search capability enables users to search for categories and items based on their attributes. End-users can use this capability to locate either suppliers or items for which they are looking.

The parametric search capability works well in conjunction with the taxonomic search capability. When a user uses the taxonomic search to specify a category, the eMPC application returns a list of attributes for that category. Users can then enter values for any of those attributes, and the eMPC application returns instances that match those values.

For example, to search for blue pens, a user selects the category "Pen." This brings up a list of attributes for the category "Pen," such as Color and Type. If a user is looking for a blue pen, they enter the value "blue" in the Color attribute field and submit the search.

Parametric searches take advantage of the fact that data is structured; the results are usually closer to the user's needs than a conventional string search.

Keyword Search

The keyword search capability allows you to search the following fields of each item for a particular word or phrase:

- Short description
- Long description
- Supplier part number
- Manufacturer part number

The eMPC application returns set of tabs identifying the categories of the results to facilitate the user's search experience.

Note: The eMPC application does not utilize the keyword search capabilities of MS SQL Server.

Administrative Features

The eMPC administrator interfaces contain the following features:

- Agents
- Actions
- Chores

You access these features using the Administrator Interfaces of the Upload Server and Catalog Server components.

Agents

The eMPC applications consists of various Agents. Agents are modules that perform various functions, such as controlling the appearance of the User Interface, transferring data between the eMPC application and external applications, generating files, etc.

Basic eMPC administration involves configuring Agents by setting the values of their parameters. Administrators access these parameters by using links on the administrator interfaces.

Note: For a deeper understanding of Agents and their uses, see *Content Engine Administration Guide*.

Actions

The eMPC application can execute certain tasks called Actions. These Actions accomplish tasks such as importing and exporting data, and establishing connections to external data sources.

Basic eMPC system administration involves performing simple Actions. Administrators use links on the administrator interfaces to access parameters that initiate Actions.

Chores

Chores are configurations that cause the eMPC application to perform certain Actions either at regular intervals or a particular time in the future. The eMPC application comes with several predefined Chores.

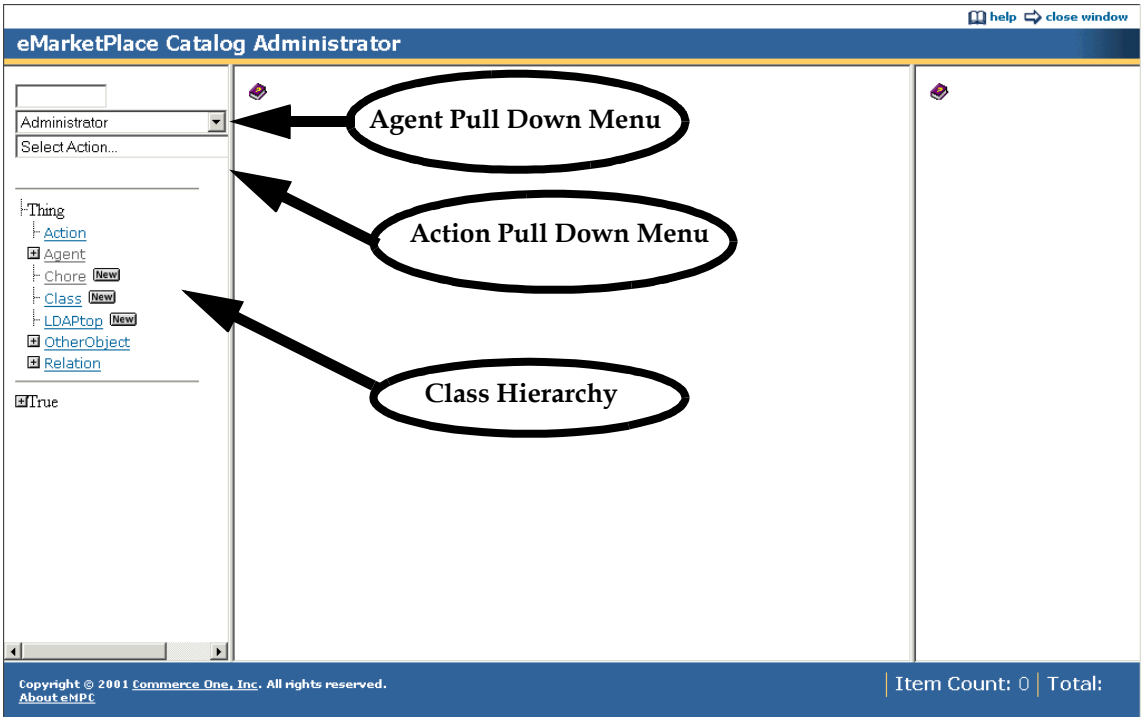
Administrator Interfaces

When you open an eMPC administrator interface, it has the following additional features that are not found on the end-user interface:

- Agent Pull Down Menu — Contains a list of Agents. Select an Agent from this list to view its parameters.
- Action Pull Down Menu — Contains a list of Actions. Select an Action from this list to execute it.
- Class Hierarchy — Contains links for different classes of eMPC objects, such as Chores, Agents, and Actions. When you click a link, the right-side pane lists all the instances of the class you selected. Also, the parametric search mechanism displays in the middle pane. You can use the parametric search to find specific instances of the selected class.

Figure 1-6 shows an eMPC administrator interface. To access the administrator interfaces, see Starting the Upload Server Component on page 2-3 and Starting the Catalog Server Component on page 2-4.

Figure 1-6 eMPC Administrator Interface

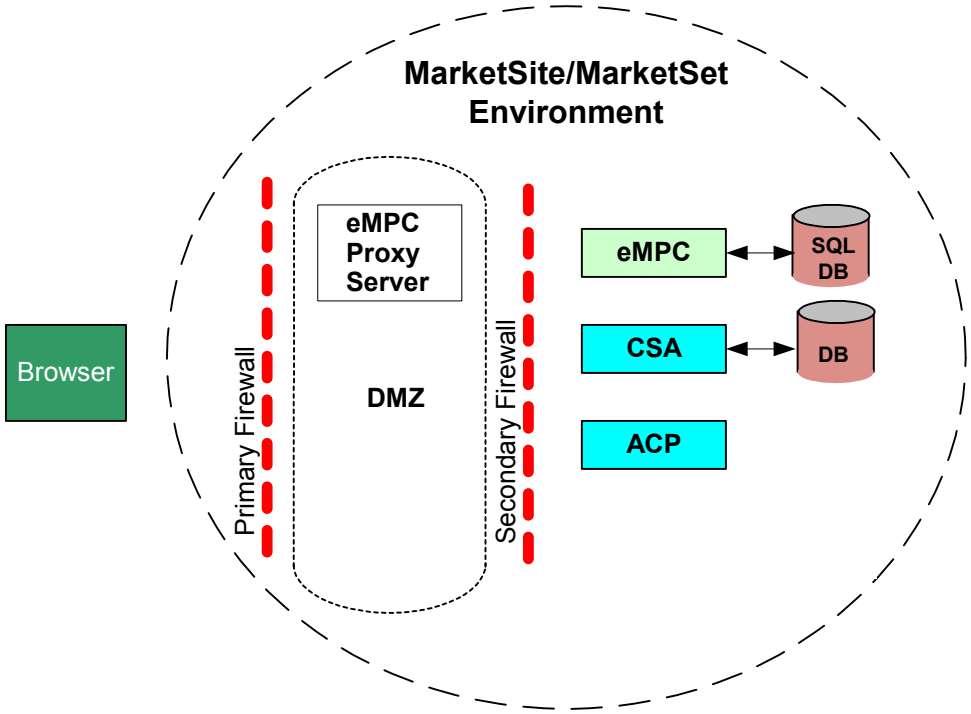


Installation Overview

The eMPC application works in conjunction with a SQL Server database and the eMPC proxy. These three components reside on three separate machines in a MarketSite/MarketSet environment. The eMPC application and SQL Server database reside on machines behind the secondary firewall, while the web proxy server resides on a machine in the “demilitarized” zone (DMZ) between the primary and secondary firewalls.

Figure 1-7 illustrates the topology of a MarketSite/MarketSet environment featuring the eMPC application. Other applications, such as the ACP and CSA applications, are shown as well.

Figure 1-7 Topology of a MarketSite/MarketSet Environment Featuring the eMPC Application



Installation Sequence

Perform the installation in the following order:

1. SQL Server database installation
2. eMPC application installation and configuration
3. eMPC Proxy Server installation and configuration
4. MSB Integration configuration
5. Procurement application integration

2 eMPC Installation

This chapter describes the requirements and procedures for installing a basic version of the eMPC application. If you plan on using the eMPC application in an advanced or high-availability setup, your system may need to meet additional requirements, and you might have to follow additional procedures that are not included in this chapter.

SQL Server Installation

SQL Server Host Requirements

The eMPC application requires SQL Server 2000 SP 1 database application. The machine on which you install the SQL Server database application must be behind the secondary firewall in your MarketSite/MarketSet environment. It must also meet or exceed the hardware and software requirements listed in this section.

Note: Changes occur automatically to the SQL Server database configuration the first time you process a catalog in the eMPC application.

Hardware Requirements

- 1 GB RAM minimum for small catalogs, 2 GB RAM strongly recommended.
- 10 GB hard disk minimum for small catalogs with only a few thousand items, 40 GB hard disk recommended for larger catalogs
- Dual Pentium 3 processors

Software Requirements

See your SQL Server database documentation for the latest software requirements.

Installation

See your SQL Server database documentation for instructions on installing the SQL Server database application.

Note: You can optionally run the eMPC's SQL Server database on the same machine on which you are hosting the Catalog Staging Area's SQL Server database. If you do this, be sure to increase the amount of RAM and disk space accordingly.

eMPC Application Installation

eMPC Application Host Requirements

The machine on which you install the eMPC application must be behind the secondary firewall in your MarketSite/MarketSet environment. It must also meet or exceed the hardware and software requirements listed in this section.

Hardware Requirements

- Dual 700 MHz dual processors
- 1 GB RAM for small catalogs, 2 GB RAM strongly recommended, 4 GB RAM if the eMPC and SQL Server applications are hosted on the same machine
- 10 GB for small catalogs, 40 GB strongly recommended, 50 GB if the eMPC and SQL Server applications are hosted on the same machine

Note: RAM and CPU resources greatly affect the performance of the eMPC application. Larger eMPC catalogs may require more memory and processing power than specified in this section.

Software Requirements

- Microsoft Windows 2000 Server Edition with Service Pack 2
- One of the following browsers:
  - Internet Explorer version 5.5 Service Pack 1 or higher
  - Netscape Navigator version 6.0
- Microsoft Data Access Components (MDAC) 2.6 for Windows 2000

Installation

After making sure that your system meets both the hardware and software requirements, install the eMPC application by running the Installation Wizard.

To run the Installation Wizard:

1. Insert the eMPC application CD-ROM into the CD-ROM drive of the machine on which you plan to perform the installation.
2. Click **Start** and select **Run...**. The **Run** window appears.
3. In the **Run** window, type **D:\nt\setup.exe** in the **Open** field.

Note: This book assumes that the letter of your CD-ROM drive is "D." If your CD-ROM is labeled with a letter other than "D," replace "D" with that letter.

4. Click **OK**.

The Installation Wizard begins. Follow the online instructions in the Installation Wizard to complete the installation.

Note: The eMPC application automatically installs as a service. If you do not want the eMPC application to install as a service, un-check the checkboxes when the Installation Wizard asks you if you want to install the application as a service.

Starting the eMPC Application

Starting the eMPC Application as a Service

To start the eMPC application as a service, simply reboot your machine. The services are configured to start automatically on startup. You can also start them manually from the Service Control Panel.

Starting the eMPC Application as a Program

If you did not install the eMPC application as a service, start it by starting its individual components: the Catalog Server and Upload Server components. The order in which you start these components does not matter.

Starting the Upload Server Component

Start the Upload Server component by selecting:
Start | **Programs** | **Commerce One** | **eMarketPlace Catalog** | **UploadServer**

The Content Engine Console window appears, displaying server initialization messages. The Upload Server is ready for use when the "Listening on port 2001" message displays.

Note: If you configure the Upload Server to run on a port other than 2001, that port number appears in the "Listening on port" message.

To access the Upload Server administrator interface directly, point a web browser to the following URL:

```
http://<hostname>:<port>/imerge/administrator/multipane?
```

in which <hostname> is the name of the machine hosting the Upload Catalog Server component and <port> is the port on which the Upload Server component is running. The default port for the Upload Server component is 2001.

If you want to access the Upload Server administrator interface through a proxy server, substitute <hostname> with the name of the machine hosting the proxy server, and <port> with the port the proxy server is redirecting requests for 2001.

Note: Commerce One recommends that you always access the Upload Server component through the proxy server.

Starting the Catalog Server Component

Start the Catalog Server component by selecting:
Start | **Programs** | **Commerce One** | **eMarketPlace Catalog** | **CatalogServer**

The Content Engine Console window appears, displaying server initialization messages. The Catalog Server is ready for use when the "Listening on port 2500" message displays.

Note: If you configure the Upload Server to run on a port other than 2500, that port number appears in the "Listening on port" message.

Access the Catalog Server administrator interface by pointing a web browser to the following URL:

```
http://<hostname>:<port>/imerge/administrator/multipane?
```

in which <hostname> is the name of the machine hosting the eMPC Catalog Server component and <port> is the port on which the Catalog Server component is running. The default port for the Catalog Server component is 2500.

If you want to access the Catalog Server administrator interface through a proxy server, substitute <hostname> with the name of the machine hosting the proxy server, and <port> with the port the proxy server is redirecting requests for 2500.

Note: Commerce One recommends that you always access the Catalog Server component through the proxy server.

Accessing the Administrator Interfaces

Accessing the Upload Server Administrator Interface

To access the Upload Server administrator user interface, first start the Upload Server program or service, then point a web browser to the following URL:

```
http://<hostname>:<port>/imerge/administrator/multipane?
```

in which <hostname> is the name of the machine hosting the Upload Catalog Server component and <port> is the port on which the Upload Server component is running. The default port for the Upload Server component is 2001.

If you want to access the Upload Server administrator interface through a proxy server, substitute <hostname> with the name of the machine hosting the proxy server, and <port> with the port the proxy server is redirecting requests for 2001.

Note: Commerce One recommends that you always access the Upload Server component through the proxy server.

Accessing the Catalog Server Administrator Interface

To access the Catalog Server administrator user interface, first start the Catalog Server program or service, then point a web browser to the following URL:

```
http://<hostname>:<port>/imerge/administrator/multipane?
```

in which <hostname> is the name of the machine hosting the eMPC Catalog Server component and <port> is the port on which the Catalog Server component is running. The default port for the Catalog Server component is 2500.

If you want to access the Catalog Server administrator interface through a proxy server, substitute <hostname> with the name of the machine hosting the proxy server, and <port> with the port the proxy server is redirecting requests for 2500.

Note: Commerce One recommends that you always access the Catalog Server component through the proxy server.

Accessing the User Interface

To view the interface accessible to end-users (such as buyers), enter the following URL:

```
http://<hostname>:<port>/imerge/empc/multipane?
```

Substitute <hostname> with the name of the machine hosting the proxy server, and <port> with the port the proxy server is redirecting requests for 2500.

3 Initial Configuration

This chapter describes the preparation of the SQL database for use by the eMPC application, as well as mandatory configuration procedures for the eMPC application. You only need to perform these configuration procedures one time after installing the eMPC application.

Prepare the SQL Server Database

The eMPC application stores and accesses data in a SQL Server 2000 database running on a separate machine.

Note: The eMPC application only supports SQL Server 2000; it does not support SQL Server 7. Refer to the SQL Server documentation for SQL Server-specific information.

1. Install the SQL Server software on the machine you want to use to host your database. This machine must be accessible by the machine hosting the eMPC application.
2. Create a new SQL database.
3. Create a sub-directory in the file system of the SQL server machine to serve as your Transfer Directory, such as:

   ```
   d:/temp/eMPCTransfer
   ```

   The Transfer Directory must be accessible by the eMPC application. For more information on the Transfer Directory, see Application Architecture on page -6.

Note: If the eMPC processes and the SQL Server application are on separate machines, enable sharing so that this directory may be accessed from the eMPC machine.

Set the Virtual Memory

Perform the following steps on the machine hosting the eMPC application:

1. Go to **Start** | **Settings | Control Panel**.
2. Double-click the **System** icon. The **System Properties** window appears.
3. Click on the **Advanced** tab and click on **Performance Options**.
4. Click the **Change** button on the performance options window.
5. In the **Virtual Memory** window, select the eMPC installation drive and set the minimum virtual memory to 2000 MB and Max to 4000 MB.
6. Click the **Set** button, then click **OK**.

Edit the eMPCDiskServer and eMPCSQLConnector Agents

When you install the eMPC application, all of its agents are automatically configured for practical purposes except for the eMPCDiskServer and eMPCSQLConnector agents. Perform the following procedures on the machine hosting the Upload Server component to configure these agents.

1. Start the Upload Server component and access its administrator interface, as described in Starting the Upload Server Component on page -5.
2. The eMPC administrator interface appears in your browser. It is divided into three panes. Click the **Agent** link in the pane on the left side. A list of agents appears in the pane on the right side.
3. Click the **eMPCDiskServer** link in the pane on the right side. The eMPCDiskServer agent's properties appear in the middle pane.
4. Click the **Edit** button in the middle pane. The middle pane displays editable versions of the eMPCDiskServer agent's properties.
5. In the **TransferDirectory** and **OSNativeTransferDirectory** fields, enter the pathnames of the Transfer Directory you created in step 3 in Prepare the SQL Server Database on page 3-1. Commerce One recommends that you use a pathname that conforms to Universal Naming Convention (UNC) standards. The pathname you enter in the **OSNativeTransferDirectory** field should conform to the operating system running on the machine hosting the SQL Server.

Note: If you are performing an HA setup, enter a UNC path for the transfer directory.

For example:

- **TransferDirectory** — `\\<sqlhost>\temp\eMPCTransferDirectory\`
- **OSNativeTransferDirectory** — `d:\temp\eMPCTransferDirectory\`

Caution: You must include a backslash at the end of each field.

6. Scroll the middle pane down and click the **Update** button.
7. Click the **eMPCSQLConnector** link in the pane on the right side. The eMPCSQLConnector agent's properties appear in the middle pane.
8. Click the **Edit** button in the middle pane. The middle pane displays editable versions of the eMPCSQLConnector agent's properties.

Caution: Use extreme care when editing the eMPCSQLConnector agent's properties. The eMPCSQLConnector agent creates a Data Source Name (DSN) on your system, which is usually placed under `c:\iMergeDSNs\<datasourcename>.dsn`. If you want to make a correction or change to the eMPCSQLConnector agent after you click the **Update** button, you must uninstall the eMPC application, delete the DSN, and start over from scratch.

9. Fill in only the following fields. Do not change any of the other fields.
   - **DataSourceName** — Enter the ODBC Data Source Name (DSN) representing this SQL database connection. You may choose any name, but avoid any name that might have previously been chosen.
   - **LoginID** — Enter the name you use to log into the SQL database.
   - **Password** — Enter the password you use to log into the SQL database.
   - **HostName** — Enter the name of the machine on which the database resides.
   - **Database** — Enter the name of the SQL database you created in step 2 in Prepare the SQL Server Database on page 3-1.

Note: A database with the name you enter in the **Database** field must already exist on the host machine you specified in the **HostName** field.

10. Double-check your edits. If you are sure they are correct, scroll the middle pane down and click the **Update** button. A message in the right-side pane should appear, indicating that no tables are found. This message appears because the connection is successful, but the database you created is currently empty. You can safely ignore this message.
11. Run the SaveApplication Action to save your changes. For more information, see SaveApplication on page -2.
12. Start the Catalog Server component and access its administrator interface, as described in Starting the Catalog Server Component on page -6.
13. The eMPC administrator interface appears in your browser. It is divided into three panes. Click the **Agent** link in the pane on the left side. A list of agents appears in the pane on the right side.
14. Follow 3 through 11 to configure the eMPCSQLConnector in the Catalog Server. Since you already configured the DSN and it exists on disk, simply enter the name of that DSN and click update.

Configuring the TempDirectory Attribute (Optional)

The eMPC application uses a spooling mechanism to store incoming xCBL catalogs before moving them to the xCBL Content Library directory. By default, this mechanism uses the directory `<install_dir>\eMPC\UploadServer\xcbl-spool` directory. If you are operating the eMPC application on a different drive, or if you simply want the eMPC application to use an alternate directory for this temporary storage, configure the **TempDirectory** attribute found on the **ApplicationConfiguration** object in the Upload Server component. To do so, follow these instructions:

1. On the hard drive on which you are running the eMPC application, create a directory in which you want to temporarily store incoming xCBL catalogs before they are sent to the xCBL Content Library directory.
2. Access the **Administrator** agent on the UploadServer component.
3. Expand the **OtherObjects** parent directory in the left pane.
4. Click on **ApplicationConfiguration** in the left pane.
5. Click on **LocalApplicationConfiguration** in the right pane.
6. Click the **Edit** button near the bottom of the middle pane.
7. In the **TempDirectory** field, specify the path of the directory you created in step 1. You must specify an absolute path. For example: `D:\eMPCtemp\`
8. Click the **Update** button.

Automatic Database Configuration

Changes occur automatically to the SQL Server database configuration the first time you process a catalog in the eMPC application. These changes are described below. Commerce One strongly recommends that you do not change these configurations once they are set.

In the SQL Server properties, the following configurations automatically occur:

- A "-x" is added to the **Startup** parameter.
- The **Boost SQL Server** option is selected.
- The **User Windows NT Fibers** option is selected.
- Physical memory for the SQL Server application is reserved.

In the Database you create, the following configurations automatically occur:

- The **Recovery/Model** parameter is set to Simple.
- The **Auto Shrink** option is deselected.
- The **Auto Create Statistics** option is selected.

- The **Auto Update Statistics** option is selected.
- The **Data Device** size is set to 1 GB and the **Transaction Log Device** size is set to 500 MB.

4 JRun eMPC Proxy Web Server Installation and Integration

This chapter describes how to install an eMPC Proxy Web Server and integrate it with Microsoft's Internet Information Services (IIS).

Overview

The eMPC Proxy can be used to provide additional security protection for your eMPC Catalog and Upload Servers. For each of these 2 servers, you can setup 1 or more eMPC Proxy applications, which can limit which users can access the server and which agents (paths) of the server they can access. For example, for your eMPC Catalog Server, you may want to setup 2 eMPC Proxy applications: one for the administrator, which permits access to all agents of the Server, and a second for the client users, which can only see the catalog ('empc') agent.

The eMPC Proxy can be used in two modes:

1. A standard proxy mode.
2. A proxy/authentication mode that verifies user credentials passed via an OCI request.

In the standard proxy mode, the eMPC Proxy is just a conduit of all traffic between the client and the respective server. It does no user authentication, but can restrict which agents of the server can be seen. If the URL of a request references an agent not in the proxy's "subpath filter" list, the proxy will return a NOT FOUND response.

In the proxy/authentication mode, the proxy does everything that it does in the proxy mode, but additionally, authenticates each user's session based on OCI credentials, which must be included in the original request of the session.

Prerequisites

You must install the following prerequisistes for the eMPC Proxy:

- Windows 2000 with IIS
- JRun 3.1

eMPC Proxy Installation and Setup

Note: You can optionally install the eMPC Proxy on the same machine in your MarketSite environment that is hosting the IIS and JRun applications.

This section describes how to install and configure the eMPC Proxy, which ships on a separate eMPC Proxy CD-ROM.

The eMPC Proxy installation takes two steps:

1. Install the eMPC Proxy by running its Installation Wizard.
2. Set the properties for each eMPC Proxy Server application.

These steps are covered in detail in the following sections:

Install the eMPC Proxy

1. Uninstall any previous versions of the JRun eMPC Proxy, as well as any folders and files created by the previous installation.
2. Close all applications.
3. Place the eMPC Proxy CD in the CD-ROM drive of the Web Server machine.
4. Start the Installation Wizard (setup.exe) by double-clicking on it in Windows Explorer.
5. Follow the on-screen instructions in the Installation Wizard. After selecting the **Program Folder**, you will be asked if you want to configure JRun. On selecting **Yes**, a separate CMD window appears. This window shows the log of the commands to add and install a new JRun Server for the eMPC Proxy. Watch carefully for any errors.
6. After the server addition and install, you are prompted in the CMD window to **press any key to continue.** Press the space bar. This initiates the commands to connect the new JRun eMPC Proxy Server to the Internet Information Server (IIS) using the JRun Connector Wizard.
7. When the **JRun Connector Wizard** starts, you are prompted to verify that the IIS Server has been stopped. Using the Windows Services Control Panel, stop the IIS Admin Server, if it is running. Press the **Enter** key to continue.

8. You are prompted to enter the values needed by the **JRun Connector Wizard**. Be careful here because you do not always want the default! Select the values as follows:
   - Containing Server = empcProxy
   - Web Server = Internet Information Server
   - Web Server Version = 5.0
   - Platform = intel-win
   - Proxy Host = 127.0.0.1, the default
   - Proxy Port = 8101 if no other JRun servers, else you will have to check the other JRun servers to find the next unused port number.
   - IIS's SCRIPTS directory = C:\inetpub\scripts, unless you installed IIS in a location other than the default.
   - Install as a global filter = yes, the default
   - Install = yes, the default

Figure 4-8 shows an example of a JRun CMD Window. The correct answers are bolded and enlarged.

Figure 4-8 Log File of the JRun CMD Window

```
E:\Program Files\Commerce One\eMPCProxy 2.0\tools\config>echo "Adding JRun eMPC
Proxy Server..."
"Adding JRun eMPC Proxy Server..."

E:\Program Files\Commerce One\eMPCProxy 2.0\tools\config>java -classpath "E:\Pro
gram Files\Allaire\JRun\lib\jrun.jar" allaire.jrun.tools.Server add "empcProxy"
"empcProxy.properties" false "E:\Program Files\Allaire\JRun"
The server 'empcProxy' was added.

E:\Program Files\Commerce One\eMPCProxy 2.0\tools\config>"E:\Program Files\Allai
re\JRun\bin\jrun.exe" -install "empcProxy" "empcProxy" -quiet

E:\Program Files\Commerce One\eMPCProxy 2.0\tools\config>pause
Press any key to continue . . .

E:\Program Files\Commerce One\eMPCProxy 2.0\tools\config>echo "Connecting JRun t
o IIS Web Server..."
"Connecting JRun to IIS Web Server..."

E:\Program Files\Commerce One\eMPCProxy 2.0\tools\config>java -classpath "E:\Pro
gram Files\Allaire\JRun\servers\admin\jmc-app\web-inf\lib\jmc-app.jar;E:\Program
 Files\Allaire\JRun\lib\jrun.jar;E:\Program Files\Allaire\JRun\lib\ext\servlet.j
ar" allaire.jrun.admin.ConnectorInstaller "E:\Program Files\Allaire\JRun"
Welcome to JRun Connector Wizard.
Please stop your target web server before proceeding.
Press 'Enter' to continue.
Available web application container(s):

1. admin
2. default
3. empcProxy

Select a web application container [default=1]: 3
Supported web server(s):

1. Apache Web Server
2. Netscape Enterprise Server
3. Netscape FastTrack Server
4. Internet Information Server
5. Personal Web Server
6. WebSite Pro
7. Zeus Web Server

Select a web server: 4
Supported web server version(s):

1. 3.0
2. 4.0
3. 5.0

Select a web server version: 3
Supported platform(s):
```

```
1. intel-win

Select a platform: 1
Choose Proxy Host [default=127.0.0.1]:
Choose Proxy Port [default=8081]:
IIS's SCRIPTS directory: C:\inetpub\scripts
JRun Connector Filter for IIS:

1. Yes
2. No

Install as a global filter: [default=1]:

1. Yes
2. No

Install? [default=1]:
```

9. At this point, the CMD window closes and you are returned to the **Installation Wizard** window.
10. Click the **Finish** button.
11. Copy the file **iaik.jar** from the directory `<empc_proxy_install_dir>\web-inf\lib\` to the directory `<jrun_install_dir>\lib\ext\`

Set the eMPC Proxy Application Properties

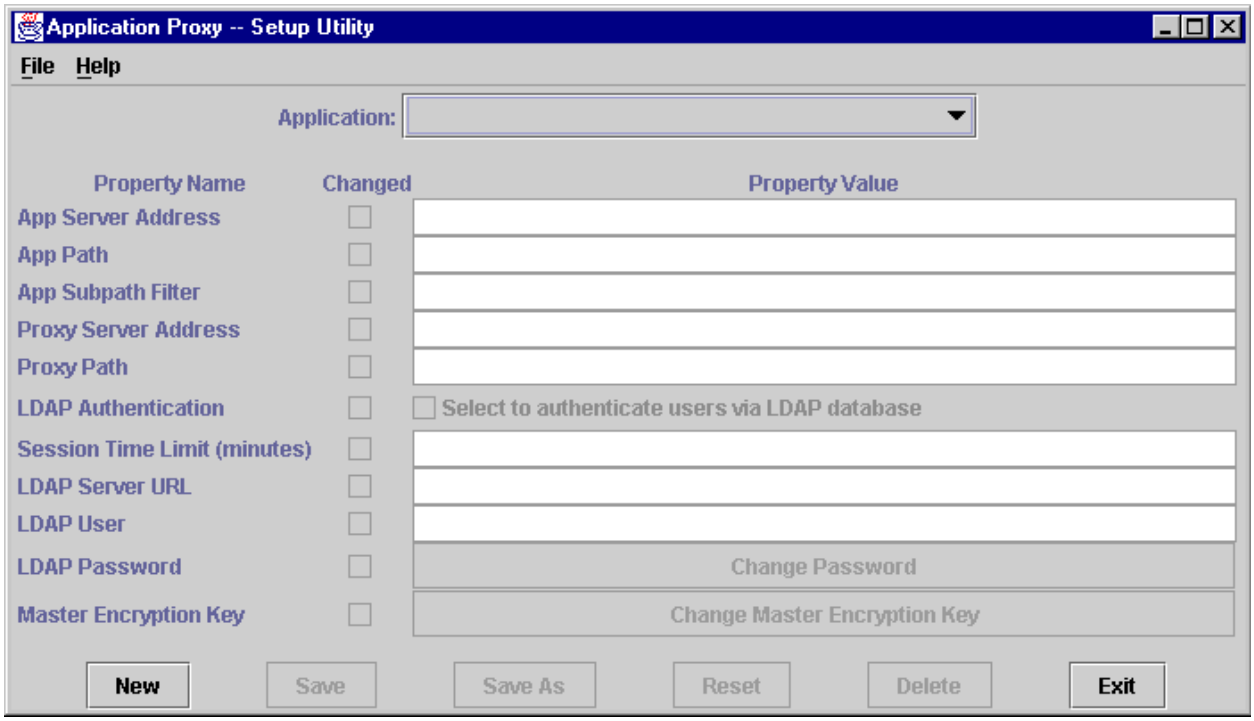
For each application deployed in the previous step, you must specify a set of properties, which are the operating parameters for the eMPC Proxy Servlet for that application. The applications deployed were:

1. eMPC — For end users of eMPC Catalog Server
2. eMPCCatalogAdmin — For administrator of eMPC Catalog Server
3. eMPCUploadAdmin — For administrator of eMPC Upload Server

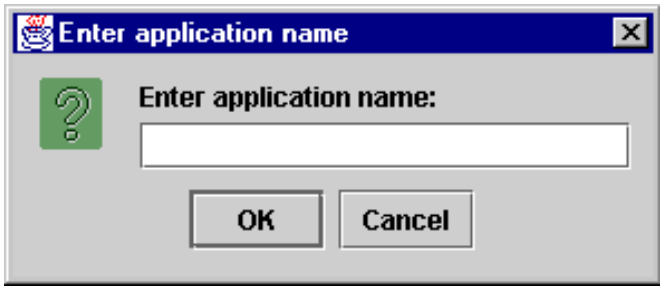
Specify eMPC Proxy Properties by using the eMPC Proxy Property Editor as follows:

1. In Explorer, double-click its command file, located in the eMPC Proxy Installation directory:
   `<empc_proxy_install_dir>\Web-inf\bin\eMPC_proxyPropEditor.bat`.

   The following window appears:



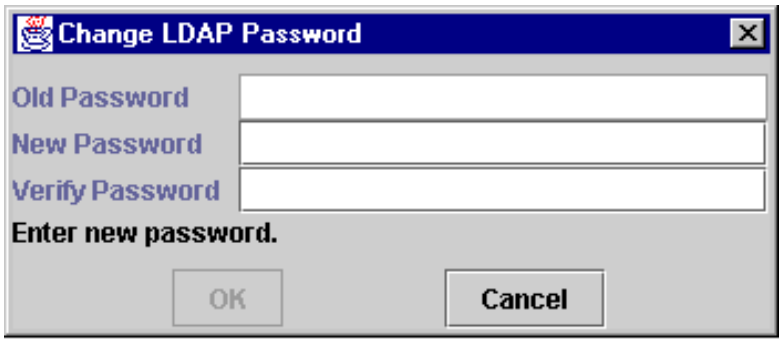
2. At the top center of the window is a choice list of existing applications, whose properties files can be edited. Initially, this list is empty and you must click the **New** button to start a properties file for an application. If you want to edit the properties of an existing application, you would first select the respective application from this list.



3. After clicking the **New** button, you are prompted to enter an application name. The value to enter is the same name, **including case** (see list above), as was specified for the URL of the application when you deployed the application under JRun, for example, eMPCCatalogAdmin. It will be the prefix of the URL path for the proxy (`/<appName>/servlet/EMPCServlet/imerge/...`).
4. Enter the address of the eMPC Server in the **App Server Address** field. The format is `<protocol>://<hostname>:<port>`. For example, the address for an eMPC Catalog Server running on host 'george' at the default eMPC Catalog Server port (2500) would be `http://george:2500`.
5. Enter the path to the application in the **App Path** field. For any eMPC application, this will always be `/imerge`.
6. Enter the subpath filter for the application in the **App Subpath Filter** field. If no value is entered in this field, the users of the proxy application will be able to access all agents (paths) of the eMPC Server. Commerce One advises leaving this value blank only for an administrator (super user) application. For a proxy application for standard catalog users, enter `empc`, which will limit the users to access only the catalog agent. In a request URL, the agent name follows the application name, for example, the URL path to the administrator agent would be `/eMPCCatalogAdmin/servlet/EMPCServlet/imerge/administrator/...` Separate multiple agent names in the filter list with spaces or commas.
7. Enter the address of the proxy server in the **Proxy Server Address** field. The value entered should be the address a client user would use to access the proxy. It is used to translate URLs in responses from the eMPC Server before sending them back to the client's browser, so that all future requests will also go through the proxy and not directly to the eMPC Server. The format is `<protocol>://<proxyHost>[:<port>]`, where the port is optional, such as when using the IIS default web server as the initial handler of requests.
8. Check the value in the **Proxy Path** field, which is automatically set depending

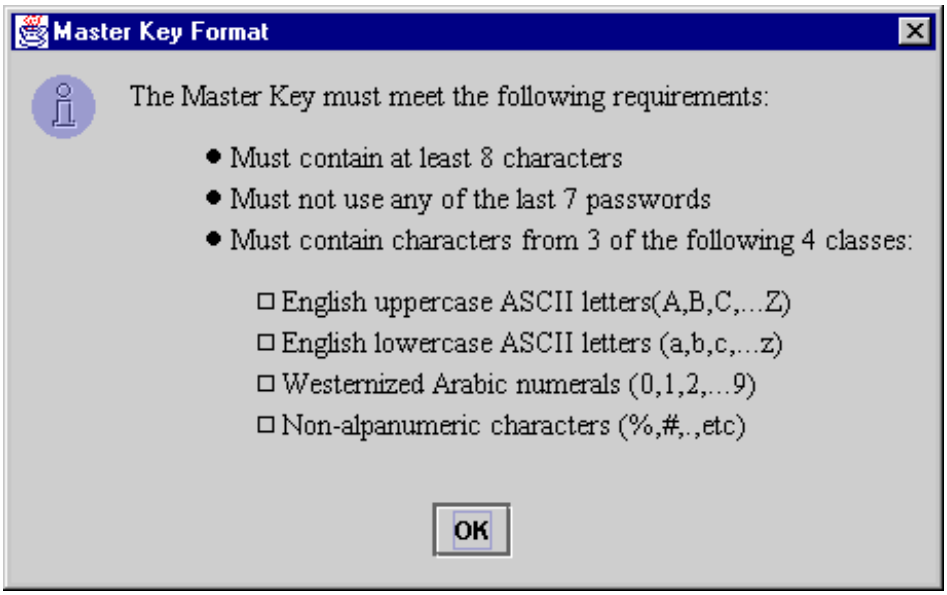
on the proxy application name. This value must be at the start of the path of any URL which is to go to the proxy application you are defining.

9. At this point you have specified all the proxy properties that are needed for a standard, pass through, proxy application. If you do not want this proxy application to authenticate users against an LDAP database using OCI parameters, verify that the **LDAP Authentication** field is not checked and skip ahead to Step 17 to save your properties. If you do want to use LDAP Authentication, select the checkbox in this field, which will enable the next 5 fields.
10. Enter the session time limit, in minutes. When using LDAP Authentication, the eMPC Proxy will not accept any more requests for a session after the session timeout period has been exceeded. If no value is entered, the default, 120 minutes, will be used.
11. Enter the URL of the LDAP server in the **LDAP Server URL** field. This value should start with the 'ldap' protocol. For example, `ldap://myldaphost:456/o=mycompany.com`.
12. Enter the user name to use in connecting to the LDAP server in the **LDAP User** field. For example, `cn=Directory Manager`.
13. Enter the password for accessing the LDAP server. To do this, click the **Change Password** button. This will result in the display of the following window. Follow the instructions just below the **Verify Password** field:



   a. Since this is the first time, the **Old Password** field is disabled. To change this password in the future, you will have to first correctly enter the old password before being able to enter the new value.
   b. Enter the LDAP Password in the **New Password** field.
   c. Re-enter the LDAP Password in the **Verify Password** field.
   d. Click the **OK** button, which will be enabled when a valid new password has been entered. At this point, the new password will be encrypted and all **clear text** versions of the password in memory will be deleted.

14. Enter a master encryption key for your site. The master key you enter will be used on encrypting all passwords and session cookies. Entering a site-specific master key value is not required, but is recommended to add addional protection to your sensitive information. To do this, click the **Change Master Encryption Key** button. This will result in the display of the following window, which lists the requirements for a master key.



15. Read and then click the **OK** button. This results in the display of the following window.

Follow the instructions just below the **Verify Password** field:

a. Since this is the first time, the **Old Master Key** field is disabled. To change this master key in the future, you will have to first correctly enter the old master key before being able to enter the new value.

b. Enter your master key in the **New Master Key** field.

c. Re-enter your master key in the **Verify Master Key** field.

16. Click the **OK** button, which will be enabled when a valid new master key has been entered. At this point, the new master key will be encrypted and all "clear text" versions of the master key in memory will be deleted.
17. Save the properties for the application by clicking the **Save** button. The properties will then be written to the following file in your eMPC Proxy installation directory:

    ```
    <empc_proxy_install_dir>\lib\<appName>_proxy.properties
    ```

If you have multiple eMPC proxy applications for which you need to create a properties file, you can repeat the above steps, or if they are similar to a previously entered set of properties, you can enter just the changes to the previous set and then click the **Save As** button, which will prompt you for the name of the new application.

Click the **Reset** button to reset all values for the current application to the last saved values.

To delete the properties for an application, select the application using the top center choice list and then click the **Delete** button.

Optional IIS Configurations

Creating a Separate eMPC Web Site for Multiple Application Hosting in IIS

The MarketSite installation may be such that more than one application is served by the IIS web server on the eMPCProxy machine. In this circumstance, it will be necessary in IIS to create a separate Web Site exclusively for eMPC.

To create a separate web site for eMPC, do the following in the Internet Service Manager on the eMPCProxy machine.

1. Right click on the server and select New/Web Site. This will invoke the Web Site Creation Wizard.
2. Follow the wizard prompts, specifying the following and otherwise accepting the default values.
   - Description — eMPC Web Site
   - IP Address — An IP address that has been created specifically for eMPC. Additional IP addresses may be created using the "Network and Dial-up Connections" facility. Edit the appropriate Local Area Connection, right click for Properties, choose Internet Protocol (TCP/IP) and right click for Properties, push the Advanced button, and then use Add to add an additional IP address. Separate steps will be necessary to register this IP address with the DNS server.
   - Web Site Home Directory — Create a directory such as `c:\inetpub\wwwroot_empc` that may serve as the home directory for the eMPC web site
3. Create a Virtual Directory within the web site for the Scripts by doing the following:
   a. In the Internet Service Manager, right click on the new eMPC Web Site and choose New/Virtual Directory.
   b. Set the Alias equal to "Scripts."
   c. Set the directory equal to some new directory that you create, such as `c:\inetpub\Scripts_empc.`

Note: The JRun Connection Wizard should be run (or rerun) to point to this web site/IP address and virtual scripts directory. This will allow the eMPCProxy server to interact only with the eMPC web site, and to not interfere with other web sites that may be hosted by IIS on the same machine.

Configuring IIS for HTTPS

To provide for secure SSL communication between users and the eMPC application, it is necessary to obtain and configure the IIS web server with a "certificate."

1. Invoke the IIS Web Server Certificate Wizard by following these steps:
   a. Open the Internet Service Manager and right click the eMPC Web Site for Properties.
   b. Choose the Directory Security tab.
   c. Press the Server Certificate button in the Secure communications frame.
   d. Follow the wizard prompts to create a new certificate request. Upon completion, the wizard will write an encoded certificate request certreq.txt to disk.
2. Obtain a certificate. Send the certificate request to a commercial certificate authority, such as VeriSign, or generate ones own certificate using third-party cryptographic tools.
3. Upon receipt of an encoded/encrypted certificate, install the certificate into IIS by the following steps.
   a. Open the Internet Service Manager and right click the eMPC Web Site for Properties.
   b. Choose the Directory Security tab.
   c. Press the Server Certificate button in the Secure communications frame.
   d. The Certificate Wizard will prompt for the location of the certificate, a file with extension *.cer.
   e. Press Okay and/or Apply several times to apply and accept the changes.
4. Restart the eMPC Web Site, and test by connecting to it with the HTTPS protocol, e.g., `https://<machine>/eMPC/servlet/EMPCServlet/imerge/empc/multipane?`
5. Invoke the Certificate. The certificate may be obtained from a commercial certificate authority such as VeriSign, or may be created using third-party cryptographic tools.

5 Configuring MarketSiteBuilder (MSB) for eMPC

This chapter describes everything you need to know to integrate the eMPC application with the MarketSite Builder (MSB) application. You can skip this chapter if you do not have MSB in your environment.

In general, the steps to integrate eMPC with MSB are as follows:

1. Install and configure SiteMinder Web Agent on the IIS/JRun/eMPCProxy machine.
2. Login to MSB as the Portal Administrator and define three new services, one for each of the following:
   a. eMPC end-user
   b. eMPC Catalog Admin
   c. eMPC Upload Admin
3. Create a role for the eMPC administration, containing the two administrative services.
4. Enable the SiteMinder WebAgent.

Overview

In general, the steps to integrate eMPC with MSB are as follows:

1. Install and configure SiteMinder Web Agent on the IIS/JRun/eMPCProxy machine.
2. Login to MSB as the Portal Administrator and define three new services, one for each of the following:
   a. eMPC end-user
   b. eMPC Catalog Admin
   c. eMPC Upload Admin
3. Create a role for the eMPC administration, containing the two administrative

services.

4. Enable the SiteMinder WebAgent.

Always configure the two eMPC administrative services with SiteMinder single-Sign-on (SSO) authentication, in order to limit access to the portal administrator. Configure the end-user eMPC service to also use SiteMinder SSO unless you are deploying a MarketSite 3.x environment, or your eMPC catalog will be predominantly accessed from remote procurement applications not hosted at your MarketPlace; see section Security on page 1-5 for more information about SiteMinder versus OCI-based security.

Implementation

Install and Configure SiteMinder Web Agent

Install and configure the SiteMinder Web Agent by executing the SiteMinder Web self-extracting installer Web-Agent-4.51-NT.exe. Accept all of the installation defaults, and at the conclusion of the installer, approve the checkbox option to invoke the SiteMinder WebAgent Configuration Wizard.

The WebAgent Configuration Wizard will ask a series of questions. Provide answers and/or accept the defaults as follows:

- Web Server — IIS
- PolicyServer IP Address — Eenter the IP address of the SiteMinder Policy Server.
- Shared secret — Enter a secret to be shared and to help authenticate the policy server and web agent to each other. The same shared secret will also be entered when creating the eMPC services in MSB.
- Default Agent ID Name — Accept the default, which will be the machine name.
- Proxy user — Provide a local user ID and password for a user with local system administrative rights. This can be the local administrator user or a special user created solely for this purpose and imbued with system administrative rights.
- SingleSignon cookies — Choose the default, “Require Cookies.”
- Cookie domain — Verify that the value corresponds to the MarketSite network domain.

After completing the configuration wizard, you may verify the values entered as well as enable logging by executing the SiteMinder IIS Web Agent Management Console from the Programs/SiteMinder from the task bar. To turn on logging, select the “Logging” tab, check “Record a log file,” and provide a path and log file name. Then choose **OK**.

Note: After making changes to the Web Agent configuration via the Web Agent Management Console, it is always recommended that you

restart the web server to apply the changes.

Create the eMPC Service for End-user Catalog Access

1. Login to the MSB application, select **Manage Services**, and click **Add**.
2. Provide the information for the new service as follows:
   - Service Display Name — eMPC
   - … Available to — Portal, TP, Guest (choose Guest if desired)
   - User Interface Based Service — <select>
3. Provide the User Interface Service information as follows:
   - Access URL — `http://<empcproxy>/eMPC/servlet/EMPCServlet/imerge/empc/multipane?`

   Note: If your IIS server has been configured for SSL, you can optionally use the HTTPS protocol.

   - Description — eMPC Catalog Service
   - Launch Service — Inherit portal navigation
   - Use Single Signon — <select>
   - SiteMinder Agent Name — Populate with the agent identity created in the SiteMinder WebAgent configuration step; generally the machine name of the eMPCProxy machine, but sometimes an alias
   - Web Agent Host IP Address — IP address of the eMPC web site
   - Web Agent Shared Secret — Password entered at time of SiteMinder WebAgent configuration
   - Resource Realm — /eMPC/servlet/ (This is an URL fragment that uniquely relates to the eMPC catalog service. The realm for the admin services will differ.)
   - TP Auto Subscription: — check this box unless TP subscription is desired
4. Click **Finish** several times to return to the **Manage Services** page.

Create the eMPC Catalog Admin Service

1. Click the **Add** button again and provide the following information:
   - Service Display Name — eMPC Catalog Admin
   - Service Available To — Portal
   - User Interface Based Service — <select>
   - Access URL — `http://<empcproxy>/eMPCCatalogAdmin/servlet/EMPCServlet/imerge/administrator/multipane?`
   - Description — eMPC Catalog Admin
   - Launch Service — Inherit portal navigation
   - Use Single Signon — <select>
   - SiteMinder Agent Name — <populate with the agent identity>
   - Web Agent Host IP Address — <IP address of the eMPC web site>
   - Web Agent Shared Secret — <password entered at time of SiteMinder WebAgent configuration>
   - Resource Realm — /eMPCCatalogAdmin/servlet/
2. Click **Finish** several times to return to the **Manage Services** page.

Create the eMPC Upload Admin Service

1. Click the Add button and provide the following information:
   - Service Display Name — eMPC Upload Admin
   - Service Available To — Portal
   - User Interface Based Service — <select>
   - Access URL — `http://<empcproxy>/eMPCUploadAdmin/servlet/EMPCServlet/imerge/administrator/multipane?`
   - Description — eMPC Upload Admin
   - Launch Service — Inherit portal navigation
   - Use Single Signon — <select>
   - SiteMinder Agent Name — `<populate with the agent identity>`
   - Web Agent Host IP Address — `<IP address of the eMPC web site>`
   - Web Agent Shared Secret — `<password entered at time of SiteMinder WebAgent configuration>`
   - Resource Realm — `/eMPCUploadAdmin/servlet/`
2. Click **Finish** several times to return to the **Manage Services** page.

Create a Role to Contain eMPC Administrator Services

Services in MSB are made available to users by assigning users and services to different roles. In this step, create a role for the eMPC administrative services and add the Portal Administrator as the only user with access to the role.

1. Click the **Manage Roles** link.
2. Click the **Add** button.
3. Enter `eMPC Admin` for the name of the role.
4. In the **Available Privileges** box, find and select the **eMPC Catalog Admin** service and the **eMPC Upload Admin** service. You can use shift+click to select both services simultaneously.
5. Hit the **Add Privilege** button to add the services to the role.
6. Click the **Finish** button to return to the **Manage Roles** page.
7. Click the **Modify** link in the **Add Member** column for the **eMPC Admin** role.
8. Select the **admin: Portal Admin** user, and click the **Add Member** button.
9. Click **Back**. Verify that the eMPC admin services are available to the portal administrator by selecting the **Use Services** link under **Member Services**.

Expose the eMPC (End-user) Service to Trading Partner Users

You can expose the eMPC service to end-users or buyers by doing the following:

1. **Make the service available to the trading partner.** When defining the eMPC service, the portal administrator may declare it as "TP Auto Subscription," meaning that it will be available to all trading partners automatically. Alternatively, the portal administrator may declare the service to require TP Subscription. If a subscription is required to the eMPC, then the portal administrator must develop a work process for regular review and approval of trading partner subscription requests.
2. **Make the service available to trading partner users.** Trading partner administrators may control who within their organizations should have access to different services. To make the eMPC available to users within an organization, the trading partner administrator must define a role (or edit an existing role, such as the Users role), add the eMPC service to it, and add users.

Enable SiteMinder WebAgent

On the machine hosting the eMPCProxy, IIS, JRun, execute the

Programs | SiteMinder | IIS Web Agent Management Console from the task bar. Choose the **Settings** tab, and check both the **Enable Web Agent** and **Enforce Policies** checkboxes. Then, restart the web server with the system services utility.

6 Procurement Application Integration

Integrating the eMPC application with one or more procurement applications enables buyers to access an eMPC catalog from a procurement application, search for products, add products to an electronic shopping cart, and have their selections automatically returned to the requisition form within the procurement application upon checkout. This process is called a "roundtrip."

The eMPC application uses SAP's Open Catalog Interface (OCI) protocol to complete roundtrips. This allows for easy integration with various hosted and external procurement applications.

This chapter describes the procedures for integrating the eMPC application with the following e-procurement applications:

- Enterprise Buyer Desktop (EBD)
- Enterprise Buyer Professional (EBP)

This chapter also discusses the eMPC application's OCI Mapping Table, which you can configure so that the eMPC application sends customized data to the procurement application.

Note: The EBP application uses OCI 2.0c, which utilizes the HTTP POST method. The EBD application uses OCI 2.0b, which utilizes the GET method. The eMPC application can integrate with either version of OCI and use either method without requiring the administrator to make any additional changes.

Integrating with Enterprise Buyer Desktop

Integrating the eMPC and EBD applications involves creating and defining a Supplier in the EBD application (as described in Creating and Defining the eMPC Supplier in EBD on page 6-2), and enabling the On-Demand Supplier Registration feature in the EBD application (as described in On Demand Supplier Registration on page 6-4).

Creating and Defining the eMPC Supplier in EBD

To integrate the eMPC application with the EBD application, do the following:

1. Define a new Supplier in EBD.

   Note: Commerce One recommends that you name the Supplier you create something obvious like “eMPC” to differentiate it from other Suppliers in the EBD application. For information on creating Suppliers in the EBD application, see *Enterprise Buyer Desktop Administration Guide*.

2. After you create a new Supplier, the Supplier Definition page appears. Select the “Intermediary” check-box on this page. This signals the EBD application that the eMPC application is a multi-supplier catalog, and that each individual SKU returned via OCI may have a different supplier MPID.

3. Click on the **Supplier Round-Trip** link and fill out the fields as follows:
   - **Round Trip Name** — The name that end-users see when clicking the **RoundTrip** link in the EBD application. Commerce One recommends that you populate this field with "eMPC" or "eMarketPlace Catalog."
   - **Round Trip Description** — A longer, more descriptive name for the roundtrip, such as "Product Exchange eMarketPlace Catalog."
   - **Login ID** — This field is used to pass catalog login credentials in the OCI request. Leave this field blank if you are using SiteMinder Security SSO. If you are using OCI-based security, populate this field with the string "admin." The username lives in the MarketSite/MarketSet platform database.
   - **Set/New Password** — This field is used to pass catalog login credentials in the OCI request. Leave this field blank if you are using SiteMinder Security SSO. If you are using OCI-based security, populate this field with the password for the trading partner system account. The password lives in the MarketSite/MarketSet platform database.
   - **Catalog URL** — The stub URL for accessing the eMPC catalog. The form of the URL is as follows:
     - If you have not configured the eMPC Proxy, enter the following:
       ```
       http://eMPCmachine:2500/imerge/empc/
       multipane?MainstripCommand=FrontDoor&WholeStrip=Yes&
       CurrentFrame=mainstrip__
       ```
     - If you have configured the eMPC Proxy, enter the following:
       ```
       http://<empc_proxy_host>.<domain>/eMPC/servlet/
       EMPCServlet/imerge/empc/
       multipane?MainstripCommand=FrontDoor&WholeStrip=Yes&
       CurrentFrame=mainstrip__
       ```

Note: Do not leave out the two trailing underscores at the end of the URL.

Catalog Parameters — Commerce One recommends that you leave this field blank; however, you can use it to allow additional catalog parameters to be sent to the eMPC application in the OCI request. The eMPC application does not utilize these parameters unless you customize it to do so.

On Demand Supplier Registration

When you attempt to access the eMPC application from an EBD application, it is possible that the Supplier trading partner identifier (MPID) that accompanies an incoming line item may not be 'known' to the Enterprise Buyer Desktop Edition. That is, the Supplier has not been entered in the Enterprise Buyer Desktop Edition database and there is no agreement between the Buyer and Supplier. If you enable the On Demand Supplier Registration in the EBD application, it attempts to acquire the Supplier's trading partner information from its associated MarketSite Trading Partner Directory. The EBD application then uses this information in conjunction with a set of defaults to register that Supplier within the EBD application. This allows the EBD application to create an order, however, the Buyer may wish or need to update some of the Supplier information through the EBD Admin interface. In addition, the Buyer may need to contact the Supplier to acquire contract/account specific information.

If the EBD application with which you are integrating is not hosted in the MarketSet/MarketSite environment, Commerce One strongly recommends that you set up the On Demand Supplier Registration feature in EBD application. If you do not set up the On Demand Supplier Registration feature, you must pre-define all of the Suppliers with whom you expect to do business in the EBD database.

Integrating with Enterprise Buyer Professional

To integrate the eMPC and EBP applications, do the following:

- Configure the EBP application with the connection information to the eMPC catalog. For information, see your EBP documentation.
- Populate the SAPSupplierLookupTable in the eMPC application such that SAPSupplier IDs (not MarketSite IDs) are returned to the EBP application upon checkout from the eMPC application. For information, see SAPSupplierLookupTable on page 6-6.

EBP Configuration

Configure an external catalog call structure in the EBP application with the information in the table below.

Caution: These parameters are order-dependent; changing the order of them could make the integration not work.

Seq	Name	Contents	Type

1		If you have not configured the eMPC Proxy, enter the following:
		`http://eMPCmachine:2500/imerge/empc/multipane?`
		If you have configured the eMPC Proxy, enter the following:
		`http://<empc_proxy_host>.<domain>/eMPC/servlet/EMPCServlet/imerge/empc/multipane?`
			URL

2	MainstripCommand	FrontDoor	Fixed Value
3	WholeStrip	Yes	Fixed Value
4	CurrentFrame	mainstrip__?	Fixed Value
40	Buyer_mpid	46baa3b6-783b-1000-9b9b-cdc58f900001	Fixed Value
50	HOOK_URL		Return URL
60	~OkCode	_top	Fixed Value
70	~Target	_top	Fixed Value

Seq	Name	Contents	Type
80	~caller	CTLG	Fixed Value
90	~ForceTarget	YES	Fixed Value

SAPSupplierLookupTable

The EBP and eMPC applications have different ways of numbering Suppliers. In addition, the EBP application associates other values with each Supplier that the eMPC application does not. To solve this problem, the eMPC application has a rudimentary lookup table called the **SAPSupplierLookupTable**. This table allows you to define SupplierID/SAPSupplierID mappings on a BuyerMPID basis in the eMPC application, allowing the eMPC application to return those associated values to the EBP application. The EBP application expects these values and may not be able to complete transactions if this table is not filled correctly.

The **SAPSupplierLookupTable** has columns for the following values:

- **SupplierID** — MarketSite Supplier ID
- **SAPSupplierID** — SAP Supplier ID, which is associated directly with Buyer in the SAP application.
- **BuyerMPID** — The MPID for the Buyer.
- **SAPMaterialGroup** — Another value that can be returned to EBP in the cart. This field needs a value, so if you do not have a value, enter the null string with a pair of double-quotation marks ("").

To edit the **SAPSupplierLookupTable**, do the following:

1. Start the Catalog Server component.
2. Select the **Warehouse** agent from the **Agent** menu.
3. In the left pane, click the scissors icon next to the **SAPSupplierLookupTable** link. An editable version of the **SAPSupplierLookupTable** appears.
4. Fill in the fields and click the **Update** button.
5. Repeat these steps as many times as necessary.

OCI Mapping Table

The eMPC application contains an OCI Mapping Table called EMPCVIEW.OCIMappingTable that allows you to specify the data that the eMPC application returns to the procurement application. It also enables you to map fields in the eMPC application to fields in your procurement application. In most cases, you do not need to change the values in this table; the defaults are set to work with the EBD and EBP applications "out of the box." You may, however, want to change the values if:

- You are using a customized EBD or EBP application that has been configured to accept additional data.
- You are using a procurement application other than EBD or EBP.
- You wish to pass non-standard values from the eMPC application to a procurement application.

Note: Having your procurement application receive customized data from the eMPC application usually requires extensive modifications to your procurement application.

To access the OCI Mapping Table:

1. Start your Catalog Server.
2. In the **Thing** hierarchy, maximize **OtherObjects**.
3. Click on **OCIMapping**.
4. In the right-side pane, click on **EMPCVIEW.OCIMappingTable**. The OCI Mapping Table appears.

To edit the OCI Mapping Table, click the **Edit** button at the bottom of the table. Customize the output of the eMPC application by changing the values for the five New_Item-Cust_Fields.

7 Actions and Chores

This chapter describes the most commonly used predefined Actions and Chores that you can access through the eMPC administrator interfaces. These Actions and Chores offer different ways of performing various administration tasks, some of which are mandatory. Read this entire chapter to learn about the different Actions and Chores available to you through the eMPC administrator interfaces, and use the ones that are appropriate for your setup.

Note: For a more in-depth understanding of Actions and Chores, including creating your own, see *Content Engine Administration Guide*.

Actions

This section contains an alphabetical listing of the most commonly used predefined Actions in the eMPC application.

MonitorEvents

The eMPC application has a window that lets you view a listing of jobs it is performing, as well as a list of jobs it has completed. This window is called the Events window.

To view the Events window, select the MonitorEvents action from the Action pull-down menu, and click the **Submit** button. The Events window appears, continuously refreshing itself every five to ten seconds. If you have executed any actions or chores, they appear in the Events window.

The Event window lists jobs, Actions, and Chores, and it is important not to confuse them. For example, a Chore may involve the initiation of several Actions that take longer to process than the Chore itself. In such a case, the eMPC application may indicate that the Chore is complete, even though it is still processing the Actions initiated by the Chore.

Note: The **Current Jobs Progress** field in the Events window is not indicative of the actual percentage of the job being done; it simply indicates that the job is progressing.

UpdateCurrencyConversionMatrix

The eMPC Installation Wizard creates a file called currency_rates.tdt. This file contains a table that lists countries, their associated currency codes, and the value of the each currency in US dollars. Exchange rates fluctuate, so it is necessary for organizations with international customers to change this table on a regular basis to reflect the current exchange rates. Every time you change the `currency_rates.tdt` file, you must run the RefreshCurrencyRates Action to load the new currency rates into the eMPC application. Alternatively, you can schedule the AutoMaterializeCanonicalPrices Chore, which automatically invokes the RefreshCurrencyRates Action at regular intervals or times that you specify.

The RefreshCurrencyRates Action exists on both the Catalog Server and Upload Server components, so you must execute the action in both the administrator interfaces.

SaveApplication

Anytime you make any changes to your eMPC application (such as scheduling Chores or configuring Agents), you must save these changes using the **SaveApplication** Action. To execute the **SaveApplication** Action, select the **SaveApplication** Action from the **Select Action**... drop down menu. Then, enter the following parameters:

- **DirectoryName** — Specify the path of the folder in which you want the eMPC application to save the current application. If the folder you specify does not exist, the eMPC application creates it. Typically, the path should be `C:/Program Files/commerceone/empc2.0/catalogserver` for edits you are making to the Catalog Server component, or `C:/Program Files/commerceone/empc2.0/uploadserver` for edits you are making to the Upload Server component.

  It is important that you enter the pathname accurately. Also, if you are using a drive letter other than C, be sure to specify that letter.

- **SaveToSeparateFiles** — Indicate whether or not you want the eMPC application to dump out a separate file for each Agent being saved. Commerce One recommends you set the value of this field to **Yes**.

- **SaveOnlyIfChanged** — Indicate whether or not you want the eMPC application to dump only to the file (or respective files) if the Agents to be dumped to that file have changed, or if no file has been dumped. Commerce One recommends you set the value of this field to **Yes**.

You now need to save the settings you just specified for the SaveApplication action. Enter a name in the box next to **Save**, such as "save.app," and click the **Save** button. Then click the **Execute** button.

Note: The application is not saved until you click the **Execute** button.

xCBLConnectUnconnectedCatalogs

The **xCBLConnectUnconnectedCatalogs** Action establishes a connection between the Catalog Server component and any catalogs that are in the SQL database that have not been connected to the Catalog Server component. You must run this Action on the Catalog Server component after transferring a new catalog (or a file containing new attributes for that catalog) from the Upload Server component to the SQL database in order for users to be able to access the new catalog. Also, run this action on the Catalog Server component each time you send Buyer View rules from the Catalog Staging Area application to the eMPC Upload Server component. For more information on Buyer View rules, see *Catalog Staging Area Administration Guide*.

You do not, however, need to run this action if you are modifying prices in a catalog to which the Catalog Server component has established a connection. For example, if you transfer a price update file for an existing catalog from the Upload Server component to the SQL database, users accessing that catalog immediately see the new prices as defined in the price update file.

To execute the **xCBLConnectUnconnectedCatalogs** Action, select the **xCBLConnectUnconnectedCatalogs** Action from the **Select Action**... drop down menu. Commerce One recommends that you execute the xCBLConnectUnconnectedCatalogs Action during off-peak hours.

As an alternative to manually executing the xCBLConnectUnconnectedCatalogs Action, you can configure the AutoConnectCatalogs Chore to automatically execute the action on a regular basis.

xCBLTransferCatalogs

The **xCBLTransferCatalogs** Action transfers any catalogs that have not been transferred from the Upload Server component to the SQL database. You must run this Action anytime you load a catalog from the CSA application into the Upload Server component in order for the eMPC application to store the catalog and make it available to users accessing the Catalog Server component.

To execute the **xCBLTransferCatalogs** Action, select the **xCBLTransferCatalogs** Action from the **Select Action**... drop down menu. Commerce One recommends that you execute the xCBLConnectUnconnectedCatalogs Action during off-peak hours.

As an alternative to manually executing the xCBLTransferCatalogs Action, you can configure the AutoTransferCatalogs Chore to automatically execute the action on a regular basis.

Chores

Scheduling Chores

The configuration procedures are similar for each predefined Chore in the eMPC application. To configure a Chore, do the following:

1. Access the eMPC administrator interface that has the Chore you want to configure. For example, if you want to configure a Chore that exists on the Upload Server administrator interface, start the Upload Server and use a web browser to access the Upload Server administrator interface, as described in Starting the eMPC Application on page 2-3.
2. Click the **Chore** link in the Class Hierarchy in the left-side frame. A list of Chores appears in the frame on the right-side.
3. Click the link for the Chore that you want to configure. The properties for the that Chore appear in the middle pane.
4. Click the **Edit** button. The middle pane displays editable versions of the properties for the Chore.
5. Set in the following parameters:
   - **Basetime** — Specify the starting date and time for the Chore. If the Chore is repeating, specify the date and time of the first run. If the Chore is not repeating, specify the one time you want the Chore to run and leave the rest of the fields blank.
   - **Period** — If the Chore is periodic, specify the number of seconds between runs. If you do not specify a period, the period defaults to 86,400 seconds (24 hours), and the Chore runs once a day on the specified days at the time specified in the **Basetime** field.
   - **RunOnDay** — If the Chore is repeating and always runs on the same days of the week, specify the days of the week you want the Chore to run. Specifying values for both Period and RunOnDay fields causes the Chore to run on each specified day at the interval specified by the **Period** field.
   - **RunAtLoadTime** — Choose **Yes** if you want this Chore to run each time you start the eMPC application.
   - **Activity** — Choose the Action invocations you want to run as part of this Chore.
6. Click the **Update** button when you are done.

Types of Chores

This section contains an alphabetical listing of the most commonly used predefined Chores in the eMPC application.

AutoConnectCatalogs

The AutoConnectCatalogs Chore automatically invokes the xCBLConnectUnconnectedCatalogs Action at regular intervals or times that you specify. It exists on the Catalog Server component.

AutoTransferCatalogs

The AutoTransferCatalogs Chore automatically invokes the xCBLTransferCatalogs Action at regular intervals or times that you specify. It exists on the Upload Server component. Commerce One recommends that you configure this Chore to execute at the same rate the Upload Server component uploads catalogs from the CSA application. For example, if the Upload Server component uploads a catalog from the CSA application in an hour, configure the AutoTransferCatalogs Chore to execute on an hourly basis.

AutoMaterializeCanonicalPrices

The AutoMaterializeCanonicalPrices Chore exists in the Upload Server. It invokes the UpdateCurrencyConversionMatrix and MaterializeCanonicalPrices actions, using the currency_rates.txt from disk. The AutoMaterializeCanonicalPrices Chore updates the materialized canonical prices for all products in the eMPC database.

AutoUpdateCurrencyConversionMatrix

The AutoUpdateCurrencyConversionMatrix Chore exists in the Catalog Server and needs the currency_rates.txt file from disk. It provides the Catalog Server with current exchange rates for “on-the-fly” eliciting and presenting of localized pricing information.

8 Additional Administrative Tasks

This chapter describes the common eMPC administrative tasks that you may or may not have to perform, depending on your configuration.

Update Currency Exchange Rates

To update currency exchange rates, do the following steps.

Caution: Some of the steps below require the use of either actions or chores. It is important to execute the actions or chores on both eMPC processes at about the same time, with a single copy of the `currency_rates.tdt` file. This is so both processes can make conversions to and from the canonical currency using a consistent set of exchange rates. Failure to update both processes together can result in inconsistent and unexpected query results.

1. Using a text editor, open the `currency_rates.tdt` file located the **eMPC 2.0/empc/currencydata**/ directory.
2. Update the `currency_rates.tdt` file with the current exchange rates. The exchange rates in the file are denominated as USD (United States dollars) per currency. For example:

   ```
   CANADA   CAD   0.65192
   ```

   where `0.65192` is the USD/CAD rate.
3. Refresh the currency exchange rates in eMPC and recompute the canonical representation of all stored catalog prices by doing one of the following:
   - In the UploadServer, run the following two actions:
     - **Update.CurrencyConversionMatrix**
     - **Materialize.CanonicalPrices**
   - Execute the following **AutoMaterializeCanonicalPrices** chore.

4. In the CatalogServer, either run the **Update.CurrencyConversionMatrix** action, or execute the **AutoUpdateCurrencyConversionMatrix** single chore.

9 Localization

This chapter describes how to localize the text on the eMPC user interface and on-line help screens.

Note: Although the eMPC application can handle text in multiple languages simultaneously, international text is only intelligible in the eMPC application when viewed using a browser with the correct character set and/or language pack installed. Both eMPC end-users and administrators must install the necessary language packs on their machines before attempting to access a localized version of the eMPC application. Failure to do so generally results in the display of unintelligible characters on the screen, even if the eMPC application correctly interprets the file.

Localizing the User Interface

Spelling Parameters

Every word and phrase in the eMPC user interface has an internal spelling parameter that controls its textual manifestation. For example, the **delete-button** spelling parameter controls the text for the **Delete** button on the eMPC application user interface. If you change the **delete-button** spelling parameter to "Remove," the text on all the **Delete** buttons on the eMPC user interface changes to **Remove**.

Note: Changing the internal spelling parameter does not alter the functionality of the eMPC application. For example, a **Delete** button still deletes data even if you change the **delete-button** spelling parameter to "Add."

Language Files

Spelling parameters and their values are stored in Language files. Language files are mapping tables that specify values for all the spelling parameters in the eMPC application. You can edit Language files to alter the values of spelling parameters, thereby changing the appearance of the words and phrases on the eMPC application user interface.

Caution: When editing the spelling parameters in the Language files, be careful not to delete leading or trailing spaces that are intentionally placed within the quotes for screen display purposes.

Language File Syntax

Language files are organized so that a language expert who is not familiar with the eMPC application can use a text editor to perform the localization. The Language files syntax contain the spelling parameters followed by their associated value. For example, the association of the spelling of the **Update** button in the USEnglish Language files appears as follows:

```
update-button “Update”
```

Associated values are either atomic values or phrases. Atomic values are simple spellings, and are used to place words like “Delete” on the delete button. Phrases are specifications of how to parametrically construct a phrase of text with embedded references to objects.

Atomic Values

Atomic values can be defined in two ways:

- As quoted strings of ISO-latin characters, such as “Delete”
- As the eMPC application extended character specifications of the form #U(x1 x2 x3...xn), where x[i] can be one of the following:
  - A quoted string of ISO-latin characters
  - A character constant (such as #\A (the upper-case "A" character)
  - The decimal value of a unicode character, such as 22235

Thus, the following is a legal eMPC application atomic language association:

#U("Hello " #\T "his has mixed-language content, like: " 22235 26446)

Phrases

Phrase specifications are used whenever the eMPC application needs to construct a sentence or phrase that contains context-specific references. Phrases in a language association are always a list of values surrounded by parentheses, and can consist of any one of the following kinds of entities:

- Atomic values, as detailed previously in this section.
- Integers indicating argument substitutions to perform (for example, 1 and 2).
- Article specifiers of the form "Articlexxx," where "xxx" is an integer corresponding to one of the arguments. Article specifiers insert an indefinite article appropriate to argument in question.

Note: The agreement of articles is language-specific, and special-purpose code must be written for each new language that uses articles. In French, for example, an article reference would need to know how to agree with the gender of the noun following it, and would therefore require a dictionary.

The arguments used in the phrases have an internal ordering that is specified by the order in which the arguments are naturally printed in US English.

The following is an example of a phrase in a Language file:

```
find-every-1-such-that("Find every " 1 " such that:<br>")
```

Character Set Support for Language Files

The eMPC application supports several character sets natively in its Language files. Just as ISO-Latin text can be entered within a quoted string, as in "Example String," any character from the sets supported by the eMPC application may be entered similarly, without needing to utilize the #U(...) syntax as detailed above.

A user can utilize any text editor that supports one of the character sets and type the translated strings natively, within quotes. Special care should be taken to name the file appropriately. The naming conventions detailed above must be adhered to, for example, using lower-case ASCII letters for names. Furthermore, any character set must be referenced in the file extension. For example, `spanish.lang-ucs2`, or `japanese.lang-utf8`. If no character set is referenced in the file extension, the eMPC application assumes the ISO-Latin character set, for example `usenglish.lang`.

Character Code Convention

The eMPC application supports the import of content in multiple languages by handling a number of different character code conventions used to encode the characters in the file. The eMPC application currently supports seven different character code conventions, which are sufficient to represent characters in every natural language.

The character code conventions currently supported by the eMPC application are as follows:

- ASCII: 8-bit wide characters with the most significant bit not set.
- ISOLatin: 8-bit wide characters with all of the ASCII characters and Latin/European characters represented with the high order bit set. This is the eMPC application's default.
- UCS2: A 16-bit wide representation for extended characters with the low order byte coming first (little endian). This character code convention is sufficient to represent almost all characters in common use around the world, and is commonly used by Microsoft software.
- UCS2BigEndian: A 16-bit wide representation exactly like UCS2, only with the bytes in big endian order.
- UCS4: A 32-bit wide representation for extended characters with the low order byte coming first (little endian). This character code convention is sufficient to represent all characters in all languages, but is rarely used because it is much less space efficient than other formats.
- UCS4BigEndian: A 32-bit wide representation exactly like UCS4 only with the bytes in big endian order.
- UTF8: A variable length encoding for extended characters that represents ISOLatin/ASCII characters with one byte, and all other characters with a larger number of bytes depending on the unicode value of the character in question. Like UCS4, this representation is able to represent all characters in all languages, and is generally much more space efficient than UCS4. UTF8 is emerging as a common transfer format.

The eMPC application may not be able to read every possible encoding of international text, but it usually is possible to save text out into one of the above formats using the software that originates the text.

Language Files Types

The eMPC application uses three types of language files:

- `administrator` — Contains spelling parameters that effect administrator pages.
- `basic` — Contains spelling parameters that effect pages visible to the end-user.
- `empc` — Contains spelling parameters that effect the basic schema, product classes, and attributes.

You must have one instance of each type of Language file for each language to which you localize the eMPC application.

Table 9-1 shows the format for the name of each type of Language file.

Table 9-1 Language File Naming Formats

Language File Type	Language File Name	Example
administrator	<language>-administrator.lang	german-administrator.lang
basic	<language>-basic.lang	german-basic.lang
empc	<language>-empc.lang	german-empc.lang

If you want to have your Language files in one of the wide character formats, add a "-<character code convention>" suffix to the name of the Language files, substituting <character code convention> with the name of the character code convention. For example, if you want to use the utf8 character code convention for your Korean `basic` Language file, change the name of that file to:

```
korean-basic.lang-utf8
```

You can use multi-byte text editing tools to then edit these files.

US English Language Files

The eMPC application ships with the `administrator`, `basic`, and `empc` Language files for US English, which is the default language of the eMPC application. These files are as follows:

- usenglish-administrator.lang
- usenglish-basic.lang
- usenglish-empc.lang

When you install the eMPC application, these files install on to your system in a folder called `usenglish`, which exists in the `empc/local/` directory. The eMPC application is preconfigured to use these Language files.

Creating Language Files

If you do not have the Language files for the language to which you want to localize the eMPC application, you must create those Language files. To create the necessary Language files, do the following:

1. Make copies of the US English Language files that ship with the eMPC application.
2. Change the filenames of the copies appropriately, using only lowercase letters. For example, if you are creating Italian Language files, do the following:
   - Change the filename of the copy of the US English `administrator` Language file to `italian-administrator.lang`.
   - Change the filename of the copy of the US English `basic` Language file to `italian-basic.lang`.
   - Change the filename of the copy of the US English `empc` Language file to `italian-empc.lang`.
3. In the copies of the empc and `basic` Language files, change the English values of the spelling parameters to their appropriate translations.

Note: Commerce One strongly recommends that you do not alter the spelling parameters in the administrator Language file; however, you must have a version of the `administrator` Language file that has the name of the language to which you are localizing in the filename, even though the values of the spelling parameters in the file itself are in US English.

Preconfigured Languages

In addition to US English, the eMPC application comes preconfigured to use Language files for the languages listed in Table 9-2 . The Language files for these languages do not ship with the eMPC application; however, the eMPC Installation Wizard creates a folder in the `empc/local/` directory for each preconfigured language.

Table 9-2 Additional Preconfigured Languages

Language	Corresponding Folder
Chinese Simplified	empc/local/chinesesimplified
Chinese Traditional	empc/local/chinesetraditional
French	empc/local/french
German	empc/local/german
Italian	empc/local/italian
Japanese	empc/local/japanese
Korean	empc/local/korean
Portuguese Brazilian	empc/local/portuguesebrazilian
Spanish International	empc/local/spanishinternational

Localizing to a Preconfigured Language

For an eMPC application to render a catalog in a language for which it is preconfigured, you must have all three types of Language files (`administrator`, `basic`, and `empc`) for each language to which you are localizing, and those files must exist in a folder that is named after the language in the `empc/local/` directory. If you do not have the Language files for the language to which you are localizing, you must create those files.

For information on creating Language files, see Creating Language Files on page 9-6. To locate the folder in which to place the Language files, see Preconfigured Languages on page 9-6. If you are unsure if the eMPC application is preconfigured to use the language to which you are localizing, check Table 9-2 .

After you create the Language files and place them in their corresponding folder in the `empc/local/` directory, they automatically load the next time you start the eMPC application. If you want to the Language files to load without restarting the eMPC application, execute the LoadPatches Action on the eMPC administrator interface.

Localizing to a Non-Preconfigured Language

To setup your eMPC application to render a catalog in a language for which it is not preconfigured, do the following:

1. Create or obtain the Language files for the language to which you want to localize the eMPC application. (For information on creating Language files, see Creating Language Files on page 9-6.)
2. In the `empc/local/` directory, create a folder and name it after the language

to which you are localizing. Be sure to use all lowercase letters in the folder name. For example, if you are localizing to Tagalog, name the folder:

```
tagalog
```

3. Transfer the language files you created in step 1 to the folder you created in step 2.
4. Start the eMPC application if you have not already done so.
5. Create the following objects in the eMPC application:
   - A Tongue named after the language you are creating. For example: tagalog.
   - A UIConfigurationPolicy that is named in the following format:
     `eMPCUIConfiguration-<two letter language code>`
     For example, if you are creating a UIConfigurationPolicy for Tagalog, name it:
     `eMPCUIConfiguration-tl`
   - An ApplicationViewAgent that is named in the following format:
     `eMPC-<two letter language code>`
     For example, if you are creating a ApplicationViewAgent for Tagalog, name it:
     `eMPC-tl`
6. Configure the ApplicationViewAgent to use the UIConfigurationPolicy you created step 5.
7. Configure the UIConfigurationPolicy to use the Tongue you created in step 5.

Note: For information on Tongues, UIConfigurationPolicies, and ApplicationViewAgents, see *Content Engine Administration Guide*.

Localizing the Online Help

The eMPC Installation Wizard creates the following directory on your system:

```
eMPC/help
```

The files for the eMPC on-line help screens install into this directory. System administrators with a strong HTML background can edit these files manually to create localized help messages.

Note: The `index.html` file is the top level page that controls the frameset for the user online help.

A Setting up eMPC in a High Availability Environment

This chapter describes how to set up the eMPC application in a high availability (HA) environment. Prior to following the steps in this chapter, read the chapters on HA setup in *Solution Deployment Guide*.

Overview

In an HA environment, you install the Catalog Server and Upload Server components on separate nodes. Place the Catalog Server and CartState on one machine and the Upload Server on a different machine. The database and transfer directory can exist on a separate cluster.

Prerequisites

Software Requirements

- Windows 2000 Advanced Server with MSCS
- One of the following browsers:
  - Internet Explorer version 5 or higher
  - Netscape version 4.7.x
- Microsoft Data Access Components 2.6 RTM (2.60.6526.3)
- Microsoft Jet 4.0 SP3
- Java Development Kit (JDK) 1.3

Hardware Requirements

Compaq's CL 380

Additional Requirements

This sections assumes that you are familiar with clustering concepts and MSCS. It also assumes that you have already built a cluster and that you have prepared your SQL database or database cluster. You must also have four IP addresses available for use.

Catalog Server Installation

Perform the following steps on the first node in your setup. Use the MSCS Cluster Administrator tool unless otherwise indicated.

Create a New Cluster Group for the eMPC Catalog Server

1. Go to **File | New | Group**. The **New Group** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the group. For example, eMPC Catalog Server.
   - **Description** — Optionally enter a description of the group.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Select the node that you want to be primary node for this group, and add it to preferred owners. You will use the other node for the eMPC Upload Server.
5. Click **Finish**. The group is created and is in an offline state.

Create a Physical Disk Resource for the eMPC Catalog Server

Add a physical disk resource to the group either by moving an existing one from a a disk group created by MSCS, or by creating a new physical disk resource if MSCS has not already created one. Bring the disk online. Commerce One recommends that you label the disk for identification purposes.

Note: You may have to take the disk offline in order to be moved into the group.

Install the eMPC Catalog Server

Install the eMPC Catalog Server by doing the following:

1. Insert the eMPC CD-ROM.
2. Go to the `\nt` directory and double click the **Setup.exe** file. The installation program begins.
3. As you follow the on-screen instructions to install the eMPC Catalog Server, be sure to do the following:
   - On the **Choose Destination Location** screen, click **Browse** and change the drive letter to the disk in the group you created above.
   - Perform a **Typical** install.
   - Select both the eMPC Catalog Server and eMPC Upload Server options on the **Install eMPC Servers as Services** screen.
4. When the installation is done, go to **Start | Settings | Control Panels | Administrative Tools | Services**.
5. If the eMPC Catalog and/or eMPC Upload services are started, stop them.
6. Change the startup type of each service from automatic to manual. To do this, double-click on each service and select **Manual** in the **Startup Type** field. This allows the services to be started and stopped by MSCS, as opposed to the system.

Create a Generic Service Resource for the eMPC Catalog Server

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the service. For example, eMPC Catalog Service.
   - **Description** — Optionally enter a description of the service.
   - **Resource Type** — Select **Generic Service**.
   - **Group** — Select the name of the group you created for the Catalog Server if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk to the **Resource Dependency** box.
7. Click **Next**. The **Generic Service Parameters** screen appears.

8. In the **Service Name** field, enter the exact name of the service (eMPC Catalog Server). Leave the **Start Parameters** field blank.

Note: Be sure to use the actual service name and not its display name. To do this, go to **Start | Settings | Control Panels | Administrative Tools | Services**, and view the service's properties. Use the value for **Service Name**, not **Display Name** (although they may be identical).

9. Click **Next**. The **Registry Replication** screen appears.
10. Leave the **Registry Replication** screen blank and click **Finish**. The service is created and is in an offline state.
11. Bring the service online.

Create an IP Address Resource for the eMPC Catalog Server

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the IP Address. For example, eMPC Catalog Server IP.
   - **Description** — Optionally enter a description, or the IP Address itself.
   - **Resource Type** — Select **IP Address**.
   - **Group** — Select the name of the group you created for the Catalog Server if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk and generic service to the **Resource Dependency** box.
7. Click **Next**. The **TCP/IP Parameters** screen appears.
8. In the **Address** field, enter the a unique IP Address for the eMPC Catalog Server. Enter its subnet mask in the **Subnet Mask** field. Make sure that the **Network** field is pointing to the network that you have designated as the client-access network in MSCS, and leave the **Enable NetBios for this address** checkbox selected.
9. Click **Finish**. The IP Address resource is created and is in an offline state.
10. Bring the IP Address online.

Create a Network Name Resource for the eMPC Catalog Server

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a network name for the eMPC Catalog Server. For example, eMPC Catalog Server.
   - **Description** — Optionally enter a description.
   - **Resource Type** — Select **Network Name**.
   - **Group** — Select the name of the group you created for the Catalog Server if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk, generic service, and IP Address to the **Resource Dependency** box.
7. Click **Next**. The **Network Name Parameters** screen appears.
8. Enter the virtual hostname. **Do not** use the fully-qualified domain name.
9. Click **Finish**. The Network Name is created and is in an offline state.
10. Bring the Network Name online.

Upload Server Installation

Perform the following steps on the second node in your setup. Use the MSCS Cluster Administrator tool unless otherwise indicated.

Create a New Cluster Group for the eMPC Upload Server

1. Go to **File | New | Group**. The **New Group** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the group. For example, eMPC Upload Server.
   - **Description** — Optionally enter a description of the group.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Select the node that you did not use for the eMPC Catalog Server.
5. Click **Finish**. The group is created and is in an offline state.

Create a Physical Disk Resource for the eMPC Catalog Server

Add a physical disk resource to the group either by moving an existing one from a a disk group created by MSCS, or by creating a new physical disk resource if MSCS has not already created one. Bring the disk online. Commerce One recommends that you label the disk for identification purposes.

Note: You may have to take the disk offline in order to be moved into the group.

Install the eMPC Upload Server

Install the eMPC Upload Server by doing the following:

1. Insert the eMPC CD-ROM.
2. Go to the `eMPC\UploadServer\nt` directory and double click the **Setup.exe** file. The installation program begins.
3. As you follow the on-screen instructions to install the eMPC Upload Server, be sure to do the following:
   - On the **Choose Destination Location** screen, click **Browse** and change the drive letter to the disk in the group you created above.
   - Perform a **Typical** install.
   - Select both the eMPC Upload Server and eMPC Upload Server options on the **Install eMPC Servers as Services** screen.
4. When the installation is done, go to **Start | Settings | Control Panels | Administrative Tools | Services**.
5. If the eMPC Upload and/or eMPC Catalog services are started, stop them.
6. Change the startup type of each service from automatic to manual. To do this, double-click on each service and select **Manual** in the **Startup Type** field. This allows the services to be started and stopped by MSCS, as opposed to the system.

Create a Generic Service Resource for the eMPC Upload Server

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the service. For example, eMPC Upload Service.
   - **Description** — Optionally enter a description of the service.
   - **Resource Type** — Select **Generic Service**.
   - **Group** — Select the name of the group you created for the Upload Server if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk to the **Resource Dependency** box.
7. Click **Next**. The **Generic Service Parameters** screen appears.
8. In the **Service Name** field, enter the exact name of the service (eMPC Upload Server). Leave the **Start Parameters** field blank.

Note: Be sure to use the actual service name and not its display name. To do this, go to **Start | Settings | Control Panels | Administrative Tools | Services**, and view the service's properties. Use the value for **Service Name**, not **Display Name** (although they may be identical).

9. Click **Next**. The **Registry Replication** screen appears.
10. Leave the **Registry Replication** screen blank and click **Finish**. The service is created and is in an offline state.
11. Bring the service online.

Create an IP Address Resource for the eMPC Upload Server

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the IP Address. For example, eMPC Upload Server IP.
   - **Description** — Optionally enter a description, or the IP Address itself.
   - **Resource Type** — Select **IP Address**.
   - **Group** — Select the name of the group you created for the Upload Server if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk and generic service to the **Resource Dependency** box.
7. Click **Next**. The **TCP/IP Parameters** screen appears.
8. In the **Address** field, enter the a unique IP Address for the eMPC Upload Server. Enter its subnet mask in the **Subnet Mask** field. Make sure that the **Network** field is pointing to the network that you have designated as the client-access network in MSCS, and leave the **Enable NetBios for this address** checkbox selected.
9. Click **Finish**. The IP Address resource is created and is in an offline state.
10. Bring the IP Address online.

Create a Network Name Resource for the eMPC Upload Server

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a network name for the eMPC Upload Server. For example, eMPC Upload Server.
   - **Description** — Optionally enter a description.
   - **Resource Type** — Select **Network Name**.
   - **Group** — Select the name of the group you created for the Upload Server if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk, generic service, and IP Address to the **Resource Dependency** box.
7. Click **Next**. The **Network Name Parameters** screen appears.
8. Enter the virtual hostname. **Do not** use the fully-qualified domain name.
9. Click **Finish**. The Network Name is created and is in an offline state.
10. Bring the Network Name online.

Edit the Registry

Caution: This section contains information about editing the registry. Before you edit the registry, make sure you understand how to restore it if a problem occurs. For information about how to do this, view the "Restoring the Registry" Help topic in the Regedit.exe or the "Restoring a Registry Key" Help topic in `Regedt32.exe`.

Editing the Registry on the Second Node

Complete the following steps on the second node (i.e., the one on which you installed the Upload Server).

1. Open the **Registry Editor**.
2. Navigate to `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\eMPC Catalog Server.`
3. Double-click on the ImagePath key.
4. Edit the value for the ImagePath key by changing the drive letter to match the drive letter of the disk you used when installing on Node 1.

Editing the Registry on the First Node

Complete the following steps on the first node (i.e., the one on which you installed the Catalog Server).

1. Open the **Registry Editor**.
2. Navigate to `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\eMPC Upload Server`
3. Double-click on the ImagePath key.
4. Edit the value for the ImagePath key by changing the drive letter to match the drive letter of the disk you used when installing on Node 2.

Test the Installation

Test the installation by performing a failover on both groups. To do a failover, right-click on a group and select **Move Group**. The resources temporarily go offline. Verify that the resources come back online automatically.

Create the CartState File Share

Create a New Cluster Group for the CartState File Share

Create the following cluster group on the node on which you installed the Catalog Server.

1. Go to **File | New | Group**. The **New Group** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the File Share. For example, eMPC Cart State Share.
   - **Description** — Optionally enter a description of the group.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add the first node to the **Preferred Owner** box.
5. Click **Finish**. The group is created and is in an offline state.

Create a Physical Disk Resource for the File Share

Add a physical disk resource to the group either by moving an existing one from a a disk group created by MSCS, or by creating a new physical disk resource if MSCS has not already created one. Bring the disk online. Commerce One recommends that you label the disk for identification purposes.

Note: You may have to take the disk offline in order to be moved into the group.

After creating the physical resource, create a folder on the disk called **CartState**.

Create a File Share Resource

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a network name for the File Share. For example, eMPC_CartState.
   - **Description** — Optionally enter a description.
   - **Resource Type** — Select **File Share**.
   - **Group** — Select the name of the group you created for the File Share if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk to the **Resource Dependency** box.
7. Click **Next**. The **File Share Parameters** screen appears.
8. In the **Share Name** field, enter the name of the File Share.
9. In the **Path** field, enter the full path (including drve letter) to the folder that you created when you created the disk. This should be **CartState**.
10. In the **Comment** field, you can optionally enter a description that appears next to the network name when a user is browsing the network.
11. Click **Finish**. The Network Name is created and is in an offline state.
12. Bring the Network Name online.

Create an IP Address Resource for the File Share

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the IP Address. For example, eMPC Cart IP.
   - **Description** — Optionally enter a description, or the IP Address itself.
   - **Resource Type** — Select **IP Address**.
   - **Group** — Select the name of the group you created for the File Share if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.

5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk and the File Share to the **Resource Dependency** box.
7. Click **Next**. The **TCP/IP Parameters** screen appears.
8. In the **Address** field, enter the a unique IP Address for the File Share. Enter its subnet mask in the **Subnet Mask** field. Make sure that the **Network** field is pointing to the client-access network, and leave the **Enable NetBios for this address** checkbox selected.
9. Click **Finish**. The IP Address resource is created and is in an offline state.
10. Bring the IP Address online.

Create a Network Name Resource for the File Share

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a network name for the File Share. For example, eMPC CartState Network Name.
   - **Description** — Optionally enter a description.
   - **Resource Type** — Select **Network Name**.
   - **Group** — Select the name of the group you created for the Upload Server if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk, file share, and IP Address to the **Resource Dependency** box.
7. Click **Next**. The **Network Name Parameters** screen appears.
8. Enter the virtual hostname. **Do not** use the fully-qualified domain name.
9. Click **Finish**. The Network Name is created and is in an offline state.
10. Bring the Network Name online.

Test the File Share Creation

Test the creation of the file share by performing a failover. To do a failover, right-click on the group and select **Move Group**. The resources temporarily go offline. Verify that the resources come back online automatically.

Create the TransferDirectory File Share

Create a New Cluster Group for the TransferDirectory File Share

Create the following cluster group on the node on which you installed the Database Server.

1. Go to **File | New | Group**. The **New Group** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the File Share. For example, eMPC Cart State Share.
   - **Description** — Optionally enter a description of the group.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add the first node to the **Preferred Owner** box.
5. Click **Finish**. The group is created and is in an offline state.

Create a Physical Disk Resource for the File Share

Add a physical disk resource to the group either by moving an existing one from a a disk group created by MSCS, or by creating a new physical disk resource if MSCS has not already created one. Bring the disk online. Commerce One recommends that you label the disk for identification purposes.

Note: You may have to take the disk offline in order to be moved into the group.

After creating the physical resource, create a folder on the disk called **eMPCTransfer**. This folder will server as your Transfer Directory. See Configuring the TempDirectory Attribute (Optional) on page 3-5 for more information.

Create a File Share Resource

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a network name for the file share. For example, eMPCTransfer.
   - **Description** — Optionally enter a description.
   - **Resource Type** — Select **File Share**.
   - **Group** — Select the name of the group you created for the File Share if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk to the **Resource Dependency** box.
7. Click **Next**. The **File Share Parameters** screen appears.
8. In the **Share Name** field, enter the name of the File Share.
9. In the **Path** field, enter the full path (including drve letter) to the folder that you created when you created the disk. This should be **eMPCTransfer**.
10. In the **Comment** field, you can optionally enter a description that appears next to the network name when a user is browsing the network.
11. Click **Finish**. The Network Name is created and is in an offline state.
12. Bring the Network Name online.

Create an IP Address Resource for the File Share

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a new name for the IP Address. For example, eMPC Cart IP.
   - **Description** — Optionally enter a description, or the IP Address itself.
   - **Resource Type** — Select **IP Address**.
   - **Group** — Select the name of the group you created for the File Share if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.

5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk and the File Share to the **Resource Dependency** box.
7. Click **Next**. The **TCP/IP Parameters** screen appears.
8. In the **Address** field, enter the a unique IP Address for the File Share. Enter its subnet mask in the **Subnet Mask** field. Make sure that the **Network** field is pointing to the client-access network, and leave the **Enable NetBios for this address** checkbox selected.
9. Click **Finish**. The IP Address resource is created and is in an offline state.
10. Bring the IP Address online.

Create a Network Name Resource for the File Share

1. Go to **File | New | Resource**. The **New Resource** screen appears.
2. Fill in the fields as follows:
   - **Name** — Create a network name for the File Share. For example, eMPC CartState Network Name.
   - **Description** — Optionally enter a description.
   - **Resource Type** — Select **Network Name**.
   - **Group** — Select the name of the group you created for the Upload Server if it is not already selected.
3. Click **Next**. The **Preferred Owners** screen appears.
4. Add both nodes from the **Available Owners** box to the **Preferred Owners** box, if they are not there already.
5. Click **Next**. The **Dependencies** screen appears.
6. Add the disk, file share, and IP Address to the **Resource Dependency** box.
7. Click **Next**. The **Network Name Parameters** screen appears.
8. Enter the virtual hostname. **Do not** use the fully-qualified domain name.
9. Click **Finish**. The Network Name is created and is in an offline state.
10. Bring the Network Name online.

Test the File Share Creation

Test the creation of the file share by performing a failover. To do a failover, right-click on the group and select **Move Group**. The resources temporarily go offline. Verify that the resources come back online automatically.

Complete Your Setup

After you complete the HA part of your eMPC setup, go to Chapter 3, Initial Configuration and the chapters that follow it to complete your eMPC configuration.

B Configuring eMPC in a MarketSite 3.2 Environment

The MarketSite 3.2 operating environment did not include and was not integrated with the Netegrity SiteMinder software for authentication and single-signon. The absence of SiteMinder has several implications for the eMPC install, including the following:

- "OCI-based" security must be implemented for authenticating end-user access to the eMPC catalog.
- The eMPCCatalogAdmin and eMPCUploadAdmin applications must be removed from JRun in order to close a security hole that would otherwise allow end-users to navigate to the administrative interfaces.
- User authentication for the administrative interfaces can be implemented via an included loadable module that relies upon Content Engine "agent" security, offering user ID/password authentication.

Installation and configuration

To install and configure eMPC into a MarketSite 3.2 environment, proceed by following the standard steps outlined below and detailed in the main chapters of this document, while adhering to the exceptions and extra steps noted.

Step 1: Install & configure the eMPC application.

Follow the standard procedures detailed in Chapter 2, eMPC Installation and Chapter 3, Initial Configuration.

Step 2: Install the eMPCProxy and integrate it with JRun.

To do this, follow the steps in Chapter 4, JRun eMPC Proxy Web Server Installation and Integration, except remove the eMPCCatalogAdmin and eMPCUploadAdmin applications from JRun by doing the following:

1. Log into the JRun administrative interface, typically accessible via `http://localhost:8000`.

2. Select and expand the **Server empcProxy** server.
3. Click on **Web Applications**.
4. Choose **Remove an Application**.
5. Select **eMPC Catalog Admin.**
6. Click **Remove**.
7. Repeat these steps for the eMPC Upload Admin application.

After you complete the removal, set the **eMPCProxy Application Properties** as detailed in the chapter, except configure the properties only for the eMPC end-user application. The other two applications are deleted from JRun (although some of their files may still reside in the eMPCProxy installation directory).

Also, when configuring the properties for the eMPC application, follow the additional steps listed in Set the eMPC Proxy Application Properties on page 4-6 that detail how to set up the application to do user authentication against the MarketSite Platform LDAP database.

Step 3: Configure MSB for eMPC

Create the eMPC services in MSB generally as documented in Chapter 5, Configuring MarketSiteBuilder (MSB) for eMPC, except be sure to configure the access URL for the two administrative agents to point directly at the eMPC process running on the eMPC host. For example:

```
http://<empc_host>:<port>/imerge/administrator/
multipane?
```

The port is likely to be either 2001 for the UploadServer or 2500 for the CatalogServer.

Step 4: Enable eMPC administrative security

Enable eMPC administrative security by the following steps:

1. Connect a browser to the UploadServer administrative interface at:
   `http://<empc_host>:2001/imerge/administrator/multipane?`
2. Select the **LoadACL** action from the pull-down list in the upper left.
3. Enter the following string into the file path text box:
   `../modules/AdminAgentSecurity.acl`
4. Click the **Submit** button. A login dialog appears.
5. Enter the following:
   `UID=admin`
   `password=admin`

   You are returned to the Administrator interface.
6. Change the default admin password by the following steps:

a. Choose **AdminSecurity** from the **Agent** drop-down menu.

b. Single-click **User** in the left pane.

c. Single click **Admin** in the right pane.

d. Click **Edit.**

e. Change the password and optionally change the session timeout, which is expressed in seconds.

f. Click **Update** to update the information.

g. Return to the Administrator agent by selecting this agent from the **Agent** drop-down menu.

h. Execute the **SaveApplication** action from the **Action** drop-down menu.

i. Provide the absolute path to the UploadServer or CatalogServer in the save path.

j. Click Yes two times.

k. Click **Submit**.

7. Connect a browser to the CatalogServer administrative interface at: `http://<empc_host>:2500/imerge/administrator/multipane?`

8. Repeat steps 2 through 6 for the CatalogServer.

C Troubleshooting

The following table contains a list of potential problems you may have while using the eMPC application and eMPC Proxy, as well as solutions for each problem.

Component	Description	Solution
UploadServer	"1000047232 bytes have been tenured, next gc will be global"	This message may appear when you are loading large catalogs. It is insignificant and can be safely ignored.
UploadServer	Transfer fails — Transfer fails, and no files are created in the transfer directory.	Verify that the UploadServer process is able to create the transfer directory.
UploadServer	Transfer fails/does not complete — UploadServerTransfer fails/does not complete	Verify that the OSNativeTransferDirectory path exists and is accessible by the SQL server host.
	Bulk insert files are written to the transfer directory, but no data is uploaded into the database.	Verify the access rights to the directory. Verify that you have sufficient disc space for the transfer directory, database, or database log file. Verify that there is a lot of headroom in the file system.
UploadServer/ CatalogServer	Cannot change eMPCSQLConnector configuration.	This happens because the DSN file previously written on disk overrides any changes to eMPCSQLConnector. This is a known issue. To resolve it: 1. Kill the process 2. Delete *.acl*, *.iacl*, and *.bin* files except startup.acl.~1~ and startup.bin.~1~ 3. Rename the startup.acl.~1~ and startup.bin.~1~ files to startup.acl and startup.bin. 4. Restart the process 5. Reconfigure the eMPCSQLConnector and eMPCDiskServer.

Component	Description	Solution
UploadServer	How to verify successful upload from CSA	1. Examine the Events log for a success message along the lines of "Finished uploading catalog ..." 2. Look in the xCBLContentLibrary, at the files and their time stamps, and at the contents of the agent-map.map file that identifies each of the uploaded catalogs. **Note**: The upload step does not operate on the database, and a successful upload will not populate any tables into the database.
UploadServer	How to verify successful transfer into database	1. Examine the Events log for entries that say "Finished making transfer tables for ..." and "Finished transfering tables." 2. Look at the xcbl-transfer-status.status file in the xCBLContentLibrary, both at its time stamp and at its listing of successful transfers.
CatalogServer	How to verify successful catalog connection	Examine the Events log and verify the existence of an entry that says "Finished connecting catalogs ..." for the particular catalog that you are interested in.
CatalogServer	Multiple occurrences of an extended attribute are displayed in the product “inspect” view.	Extended attribute names are case-sensitive. Examine the ACP input files or the CSA output files to verify that the attribute is spelled and cased identically in each catalog.
CatalogServer	No items visible in end-user view	1. Verify that one or more catalogs are connected. 2. Verify that connected catalog(s) are either public or private*, and if private, then verify that the appropriate Buyer_Mpid parameter was passed in the catalog URL request. * You may use the Catalog Staging Area (CSA) product to determine whether catalogs are public or private, or alternatively examine the xCBL files output from ACP and/or CSA. 3. Ensure that buyer view rules created in CSA should allow visibility of the products in question. You can determine whether failure to see products is being caused by buyer administration problems by using the xCBLDisableBuyerAdmin action in the catalog server.

Component	Description	Solution
CatalogServer	No categories visible in end-user view	1. Verify that one or more catalogs have been connected. 2. Logon to CSA and navigate to the BuyerAdministration pages, and verify that no restrictive custom views have been defined that might be suppressing categories from view for the particular buyer, buying organization, or supplier. You can determine whether failure to see categories is being caused by buyer administration problems by using the xCBLDisableBuyerAdmin action in the catalog server.
CatalogServer	Items have “no inspectable attributes”	This problem may sometimes be cleared doing a “force” connection of the catalog. Execute the xCBLConnectCatalog action in the CatalogServer administrator view.
CatalogServer	Catalog items have "no inspectable attributes." Catalog items may be inspected in the center frame, but no attribute information is displayed.	This problem may occur if updates to a supplier catalog contain categories that were not present in the first or subsequent supplier catalogs. To refresh the category information, go to the CatalogServer Administrator interface, execute the xCBLConnectCatalog action, and do a "force" connection on the supplier catalog that contains the affected products.
eMPCProxy	CMD window for JRun configuration:Exception in Thread "main" java.lang.NoClassDefFoundError: allaire/jrun/tools/Server while trying to add empcProxy Server.	The version of JRun installed is prior to JRun 3.1, which doesn't have the class allaire/jrun/tools/Server. Uninstall the current JRun and install the supported version of JRun 3.1 then reinstall the eMPC Proxy.
eMPCProxy	Not able to access proxy via IIS (port 80), but able to access proxy via JRun port number. C:\inetpub\Scripts\jrun<yymmdd>: hh:mm:ss jrISAPI[init] could not initialize empcProxy	The JRun Connector Wizard failed during installation because the IIS Admin Service was not stopped at that time. Manually run the JRun Connector Wizard to connect the empcProxy JRun Server to IIS or uninstall and reinstall the eMPC Proxy making sure to stop the IIS Admin Service.

Component	Description	Solution
eMPCProxy	A “404 Not Found”message is displayed in the browser when attempting to access an eMPC service. <JRun_install_dir>\logs\empcProxy-err.log:WARNING: Attempt to access forbidden path:/<appName>/servlet/EMPCServlet/imerge/<agent>/multipane	The subpath filter property value for the eMPC application (<appName>) being accessed doesn't include the <agent> value specified in the message. This is probably a valid error as you don't want users of the specified eMPC application to have access to the <agent> specified. If you do want users of the specified <appName> to have access to the specified <agent>, then use the <empc_proxy_install_dir>\bin\eMPC_proxy PropEditor.bat to add the <agent> to the Subpath Filter property for the application <appName>.
eMPCProxy	A “404 Not Found” is displayed in the browser when attempting to access an eMPC service and LDAP Authentication enabled. <JRun_install_dir>\logs\empcProxy-err.log:ERROR:authenticator failed to init because LDAP connection failed: javax.naming.AuthenticationException: [LDAP: error code 49 - Invalid Credentials].	During eMPC Proxy installation the iaik.jar file wasn't copied from <empc_proxy_install_dir>\lib to <JRun_install_dir>\lib\ext. Copy iaik.jar from <empc_proxy_install_dir>\lib to <JRun_install_dir>\lib\ext and then restart the “Server empcProxy” service.
eMPCProxy	The "Could not connect to JRun Server" message is displayed in the browser when attempting to access an eMPC service. <JRun_install_dir>\logs\empcProxy-event.log: [0]javax.servlet.ServletException: Port in use by another service or process: <port number>	The control port number specified for the JRun "Server empcProxy" server is also specified for another server. Use the JRun Management Console or edit the <JRun_install_dir>\servers\empcProxy\local.properties file and change the port specified in the message (<port number>) to a unused port number. Restart the "Server empcProxy" service.

eMPC Application-related Log Files

eMPC log files appear under the eMPC installation location as the system is udes. These log files to do not exist until errors occur. The most useful log files are as follows:

- `error-log.html`
- `event-log.html`
- `notifications-log.html`
- `data-log.html`

All of these log files are viewable from the browser at:
`http:/host:port/imerge/*.log`

eMPC Proxy-related Log Files

For eMPC Proxy related problems, check the following logs for error information:

IIS Log Files

The following files show errors that occur when an eMPC request fails to be sent to JRun via IIS, such as when JRun is not running, was never connected to IIS, or was connected, but with a port number that was already in use.

- `C:\inetpub\Scripts\jrun<yymmdd>`
- `C:\winnt\system32\LogFiles\W3SVC1\ex<yymmdd>`
  where `<yymmdd>` is the date of the problem.

JRun Admin Log Files

The following log files show errors that occurred during JRun configuration when changes were made using the JRun Management Console:

- `<JRun_install_dir>\logs\admin-event.log`
- `<JRun_install_dir>\logs\admin-err.log`
- `<JRun_install_dir>\logs\admin-out.log`

`<JRun_install_dir>` is the home directory of JRun.

JRun eMPC Proxy Log Files

The following log files show errors that occurred during eMPC Proxy setup and during individual requests. They also show warnings such as unauthorized access attempts, session timeouts, and request failures due to using the wrong case in the path components of the URL.

- `<JRun_install_dir>\logs\empcProxy-event.log`
- `<JRun_install_dir>\logs\empcProxy-err.log`
- `<JRun_install_dir>\logs\empcProxy-out.log`

`<JRun_install_dir>` is the home directory of JRun.

eMPC Proxy Log Files

The following log file shows some errors that occur during the eMPC Proxy installation:

```
<eMPCProxy_install_dir>\empc_install.log
```

where `<eMPCProxy_install_dir>` is the home directory of the eMPC Proxy.