

Sterling Selling and Fulfillment Foundation



Installation Guide

Release 9.10.16

Sterling Selling and Fulfillment Foundation



Installation Guide

Release 9.10.16

Note

Before using this information and the product it supports, read the information in "Notices" on page 191.

Copyright

This edition applies to the 9.1 Version of IBM Sterling Selling and Fulfillment Foundation and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1999, 2012.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Getting Started 1

Before You Begin	1
Installation Checklist	1
Documentation Library	2

Chapter 2. Creating a Security Plan 5

About Creating a Security Plan	5
Security-Planning Your Deployment Architecture	5
Security Planning-Current Security Infrastructure Analysis	6
Security Planning-Authentication and Authorization	6
Security Planning-Data Encryption	6
Security Planning-Network Topology	6
Security Planning-Java Protocol Security Measures	7
Disabling Java Protocols-EJB	8
Disabling Java Protocols-HTTP	8
Disabling Java Protocols-JMS	8
Securing Java Protocols	9
Web Security Planning	10
Web Security Planning-Post installation Recommendations	11
Web Security Planning-Session Security	11
Web Security Planning-Operating System Permissions	11
Web Security Planning-Documentation	11
Web Security Planning-Routing	11
Web Security Planning-Web Server Executables	11
Database Security	12
Database Security-Credit Card Encryption	12
Internet Explorer Security Settings	12
Configuring Browser Security Settings	12
Adding Sterling Selling and Fulfillment Foundation as a Trusted Web site	14

Chapter 3. Installing and Configuring Application Tier Software 15

Installing Your Application Server	15
Installing and Configuring a Proxy Server	15
About Configuring a Proxy Server for SSL or HTTPS Configuration	15
Configuring a Proxy Server for SSL or HTTPS-Example	15
Setting Up an Image Server for RCP	17
Setting Up Apache Image Server-Example	17

Chapter 4. Before Installing and Configuring a Database 19

Overview	19
Database Sizing	19
Database Capacity Planning	19
Disk Estimation for IBM Sterling Distributed Order Management	19

Disk Estimation for the Networked IBM Sterling Warehouse Management System Module	21
Tracking and Estimating Future Disk Requirements	25

Chapter 5. Installing and Configuring Database Tier Software on UNIX or Linux 27

Database Configuration on UNIX/Linux-Overview	27
Overview of Oracle Database Configuration	27
Configuring Oracle with Single or Multiple Byte Characters	27
Oracle Database User Privileges	28
Manually Creating Views on Oracle	30
Setting Up Scripts When Using Locally Managed Tablespace or Other Utility to Size Database	30
Setting Database Parameters in Oracle	30
Rolling Back or Undoing Changes in Oracle	30
Enabling Failover in a Multiple Node Oracle RAC Database Cluster (UNIX/Linux)	31
Configuring the NLS_LANG Parameter for an Oracle Client	31
Configuring DB2 on UNIX or Linux	32
Installing Client Components, Compilers, and Fix Pack	32
DB2 Parameter Settings	33
DB2 JDBC Drivers	33
Tuning Considerations for DB2	33
DB2 User Privileges	34
Manually Creating Objects on DB2-Setting Up Scripts	34
Manually Creating Objects on DB2-Running Scripts	34
Manually Creating Views on DB2	35
Enabling the Text Search Feature on DB2	35

Chapter 6. Installing and Configuring Database Tier Software on Windows 37

Database Configuration on Windows-Overview	37
Configuring a Database Server on Windows	37
Configuring a Database for a Multiple Schema Environment	37
Generating Data Backup and Restore Scripts	37
Configuring MSSQL Server on Windows	38
MSSQL Server User Privileges	39
Running Scripts for an MSSQL Server Database	39
Enabling the Text Search Feature on MSSQL Server	40
Configuring an Oracle Database on Windows	41
Setting Database Parameters in Oracle	41
Rolling Back or Undoing Changes in Oracle	41
Configuring Oracle for Single or Multiple Byte Characters	41
Oracle Database User Privileges	42
Manually Creating Views on Oracle	43

Enabling Failover in a Multiple Node Oracle RAC Database Cluster in Windows	44
---	----

Chapter 7. Installing Sterling Selling and Fulfillment Foundation in a Windows Environment 45

Windows Preinstallation Tasks	45
Before You Begin Installation in a Windows Environment	45
Windows Preinstallation Checklist	47
Key Terms for Installation	49
About Applying Database Definition Language (DDL) Statements After a Windows Installation	49
Types of Installation Available on Windows	50
Differences Between Single and Multiple Schema Installations	51
Running an Installation on Windows	51
Running a GUI-Based Installation on Windows	51
Creating a Silent Installation File - Windows	54
Windows Silent Install File Parameters	55
Sample Silent Installation File - Windows	58
Running a Silent Installation on Windows	59
Running a Silent Installation in Multischema Mode	59
Post Installation Tasks for Windows	62
Upgrade Information for Windows	62
Upgrade Installation Considerations	63
Upgrading in a Windows Environment-Important Database Considerations	63
Running a Silent Upgrade on Windows	63
Sample Silent Upgrade File - Windows	63
Post Upgrade Tasks for Windows	65

Chapter 8. Installing Sterling Selling and Fulfillment Foundation in UNIX and Linux Environments 67

UNIX/Linux Preinstallation Tasks	67
Before You Begin Installation in a UNIX or Linux Environment	67
UNIX/Linux Preinstallation Checklist	68
Setting Up X Windows for a GUI Installation	70
Including Memory Parameter Settings in the Installation Command	70
About the JDK for UNIX/Linux	71
Before AIX Installations Only	71
Before Linux Installations Only	72
Before RedHat Enterprise Linux Installations Only	72
IBM and JRockit JDK Only	72
Types of Installation Available on UNIX/Linux	72
Differences Between Single and Multiple Schema Installations	73
Running an Installation in a UNIX or Linux Environment	73
Running the Interactive GUI Installation in UNIX or Linux	74
Running the Text-Based Interactive Installation Program in UNIX or Linux	77
Creating a Silent Installation File - UNIX/Linux	80
UNIX/Linux Silent Install File Parameters	81

Sample Silent Installation File - UNIX/Linux	84
Running a Silent Installation in UNIX or Linux	85
Running a Silent Installation in Multischema Mode - UNIX/Linux	85
Post Installation Tasks for UNIX/Linux	88
Upgrade Information for UNIX/Linux	89
Running a Silent Upgrade in UNIX or Linux	89
Sample Silent Upgrade File - UNIX/Linux	89
Post Upgrade Tasks for UNIX/Linux	90

Chapter 9. Installing the IBM Sterling Sensitive Data Capture Server 91

Installing the IBM Sterling Sensitive Data Capture Server	91
---	----

Chapter 10. Installing the Sterling Selling and Fulfillment Foundation Language Pack 93

Installing the Sterling Selling and Fulfillment Foundation Language Pack	93
Installing the Sterling Selling and Fulfillment Foundation Context-Sensitive Help Language Pack	93
Loading the Sterling Selling and Fulfillment Foundation Language Pack Factory Defaults-Overview	93
Loading the Sterling Selling and Fulfillment Foundation Language Pack Factory Defaults for Windows	94
Loading the Sterling Selling and Fulfillment Foundation Language Pack Factory Defaults for UNIX/Linux	94
Loading the Language Pack Translations in Windows	94
Loading the Language Pack Translations in UNIX/Linux	95
Switching the Sterling Selling and Fulfillment Foundation Base Language	96
Creating and Deploying the Enterprise Archive	96
Configuring Locales in UNIX/Linux	96
Configuring Locales in Windows	97

Chapter 11. Installing a Print Server 99

Installing a Print Server - Overview	99
Installing Software Components	99
Defining Printers on Software	100
Defining Printers for the Sterling Warehouse Management System Installation	100
Copying the Sterling Warehouse Management System Standard Label Formats	100
Installing JasperReports	100

Chapter 12. Installing a Weighing Scale 101

Installing the Weighing Scale - Overview	101
Installing the Weighing Scale	101

Chapter 13. Installing the Mobile Application 103

Installing the Mobile Application - Overview . . .	103
Installing on Mobile Terminals	103
Ensuring Reinstallation on Cold Boot	105
Installing on VT220 Mobile Terminals - Overview	107
Installing the Mobile Application on VT220	
Mobile Terminals	108
Setting Up a Terminal for Use with the Mobile	
Application	109

Chapter 14. Configuring Utilities . . . 111

Configuring Sterling Selling and Fulfillment	
Foundation Utilities - Overview	111
Installation Utilities	111
Creating Database Schemas and Loading	
Factory Defaults After Installation	112
Enabling the Oracle Database Text Search	
Feature - Overview	112
Installing Third-Party JAR Files	119
Development Utilities - Overview	121
About the Configuration Deployment Tool	121
Truncating Transaction Data	121
Run-Time Utilities	122
Setting Up the Runtime Utilities	123

Chapter 15. Deploying Sterling Selling and Fulfillment Foundation 127

Sterling Selling and Fulfillment Foundation	
Deployment - Overview	127
Configuring an Oracle WebLogic Application	
Server - Overview	127
Setting Up the WebLogic Script File	127
Configuring Oracle WebLogic XML Registry	128
Disabling Instrumented Stack Traces in	
WebLogic.	129
Setting Up WebLogic to Display Barcodes and	
Graphs	129
Setting Up WebLogic to Use HTTP In-Memory	
Session Replication	130
Building the Enterprise Archive (EAR) Package on	
WebLogic.	130
Deploying Context-Sensitive Help on WebLogic	131
Building Web Services on WebLogic.	131
Defining an EJB Web Service with WebLogic	132
Defining a JAX-WS Web Service on WebLogic	133
Example of webservicesbeans.xml file -	
WebLogic.	136
Exposing an SDF Service as a Web Service on	
WebLogic.	136
Configuring a JAX-WS Web Services Client	
(UNIX/Linux Only) on WebLogic	137
Including Custom Classes on WebLogic	138
Creating the EAR on WebLogic	139
Precompiling the WAR File on WebLogic	140
Deploying the Enterprise Archive (EAR) on	
WebLogic.	140
Verifying Installation on WebLogic	141
Configuring a WebSphere Application Server -	
Overview.	141

Configuring WebSphere to Avoid Direct	
Datasource Lookup Warning Messages at	
Runtime	141
Configuring the WebSphere Classloader	142
Setting Up Application Clients to Invoke	
Sterling Selling and Fulfillment Foundation EJBs	142
Configuring WebSphere JVM Settings	143
Setting Up WebSphere to Display Barcodes and	
Graphs	143
Building the Enterprise Archive (EAR) Package on	
WebSphere	144
Deploying Context-Sensitive Help on	
WebSphere	145
Building Web Services on WebSphere	145
Defining an EJB Web Service with WebSphere	146
Defining a JAX-WS Web Service with	
WebSphere	147
Example of webservicesbeans.xml file -	
WebSphere	150
Exposing an SDF Service as a Web Service on	
WebSphere	150
Configuring a JAX-WS Web Services Client	
(UNIX/Linux Only) on WebSphere	151
Including Custom Classes on WebSphere	152
Creating Enterprise Archive Files on WebSphere	153
Precompiling the WAR File on WebSphere.	154
Deploying the Enterprise Archive (EAR) Using the	
WebSphere Admin Console.	154
Verifying the Installation on WebSphere	156
Configuring a JBoss Application Server.	156
Setting Up JBoss to Display Barcodes and	
Graphs	157
Building the Enterprise Archive (EAR) Package on	
JBoss	158
Deploying Context-Sensitive Help on JBoss	158
Building Web Services on JBoss	159
Defining an EJB Web Service with JBoss	159
Defining a JAX-WS Web Service with JBoss	160
Example of webservicesbeans.xml file - JBoss	163
Exposing an SDF Service as a Web Service on	
JBoss	164
Configuring a JAX-WS Web Services Client	
(UNIX/Linux Only) on JBoss	165
Including Custom Classes on JBoss	166
Creating the EAR on JBoss	166
Precompiling the WAR File on JBoss	168
Deploying the Enterprise Archive (EAR) on JBoss	168
Configuring DataSource Connection Pooling on	
WebLogic, WebSphere, and JBoss.	169
Configuring Data Source Connection Pooling for	
Single Schema Installations.	169
Configuring Data Source Connection Pooling for	
Multiple Schema Installations	170
Configuring a Restrictive Cookie Path - Overview	171
Configuring a Restrictive Cookie Path on	
WebLogic.	171
Configuring a Restrictive Cookie Path on	
WebSphere	172
Configuring a Restrictive Cookie Path on JBoss	172
Setting the Client Character Display.	172
Clearing Browser Caches	172

Clearing the Java Plugin Cache	173
Statistics Monitoring	173
Setting an Enterprise for Logging In to IBM Sterling Business Center	173

Chapter 16. Deploying and Updating the Rich Client Platform Applications . 175

Before You Begin RCP Deployment	175
Deploying RCP - Creating the RCP_EXTN_FOLDER Folder	176
Deploying RCP - Configuring Locations	177
Deploying RCP - Localizing Bundle and Theme Files	179
Deploying RCP - Enabling HTTPS	179
Deploying RCP - Applying Updates	179
Deploying RCP - Running the Ant Script	181
Deploying RCP - Deploying Clients through a Remote Terminal	182

Deploying RCP - Configuring Location Settings	183
Deploying RCP - Connection Settings	184
Deploying RCP - Connection Settings for Fetching Images from the Server	185
Deploying RCP - Configuring Connection Settings for HTTPS Connection	187
Deploying RCP - Configuring Connection Settings for Context-Sensitive Help	187
Deploying RCP - Security Certificates	188
Deploying RCP - Compression in the Rich Client Platform	188
Deploying RCP - Creating Server-Side Commands Without Running the Application.	189

Notices 191

Index 195

Chapter 1. Getting Started

Before You Begin

Before you begin installing IBM® Sterling Selling and Fulfillment Foundation, read this guide thoroughly. Then define your processes for handling the following:

- Development and Test Environments
- Security Strategy
- Change Management Strategy
- Development and Test Procedures
- Rollback Strategy
- Upgrades and Maintenance Strategy

In addition, before starting the installation process, read the *Sterling Selling and Fulfillment Foundation: Performance Management Guide* which contains information that helps you optimize the performance of your Sterling Selling and Fulfillment Foundation.

Installation Checklist

When installing the components used by Sterling Selling and Fulfillment Foundation, follow the sequence of tasks provided in the following checklist and additional instructions in the chapters of this guide.

During the installation and setup processes, you should also refer to the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*, which is a companion guide and should be used during each step of the process. Doing so can eliminate future problems and help you to troubleshoot errors.

Installation Checklist

1. Ensure that you have the necessary system requirements to install and run Sterling Selling and Fulfillment Foundation. See the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* for information.
2. Determine whether you want to enable an Online Documentation Library (recommended) or a Local Documentation Library.
3. Set up your security infrastructure.
4. Install and configure your application server.
5. Install and configure your WebServer or Proxy Server.
6. Install and configure your database software on UNIX/Linux or Windows.
7. Size your database on UNIX or Windows.
8. Install the Sterling Selling and Fulfillment Foundation application on UNIX/Linux or Windows.
9. Install and configure the Sterling Sensitive Data Capture Server (recommended).
10. Install the Sterling Selling and Fulfillment Foundation language pack (optional).
11. Optionally install the print server.
12. Optionally install the weighing scale software.

13. Optionally install the Mobile Application.
14. Configure the Sterling Selling and Fulfillment Foundation properties to use with the database, agent servers, LDAP servers, logging, and so forth. See the *Sterling Selling and Fulfillment Foundation: Properties Guide* for more information.
15. Configure the Sterling Selling and Fulfillment Foundation utilities for installation, runtime, migration, and production.
16. Set up the application server for use with Sterling Selling and Fulfillment Foundation on WebLogic, WebSphere®, or JBoss.
17. Build your Enterprise Archive (EAR) on WebLogic, WebSphere, or JBoss.
18. Deploy the EAR to your application server as appropriate on WebLogic, WebSphere, or JBoss.
19. Optionally deploy and update the Sterling Rich Client applications.
20. Optionally run the configuration deployment tool to migrate your configuration data. For information about this tool, see the *Sterling Selling and Fulfillment Foundation: Configuration Deployment Tool Guide*.
21. Optionally log into the IBM Sterling Business Center application, which is part of the Sterling Selling and Fulfillment Foundation installation. For more information about IBM Sterling Business Center, see the *Business Center: Item Administration Guide* and the *Business Center: Pricing Administration Guide*.

Documentation Library

During installation, you can select whether you want to enable an Online Documentation Library or a Local Documentation Library. You can also change this selection after installation.

The Online Documentation Library is hosted by IBM. It provides online access to the IBM Sterling Selling and Fulfillment Foundation documentation in HTML and PDF formats. The combination of Google mini-search capability with HTML format enables users to search the entire documentation set and narrow their results quickly. Because the Online Documentation Library is hosted by IBM, the documentation set is updated dynamically with changes and hot fixes.

The Local Documentation Library is hosted locally by your enterprise. It provides access to the IBM Sterling Selling and Fulfillment Foundation documentation in HTML format, as well as index and search capability on a book-by-book basis. It does not provide updates for changes and hot fixes except through product upgrades. This documentation is always accessible, even if you choose to implement the Online Documentation Library.

Users can access either of the libraries by selecting Documentation Library in the Help Menu, as shown in Figure 1.

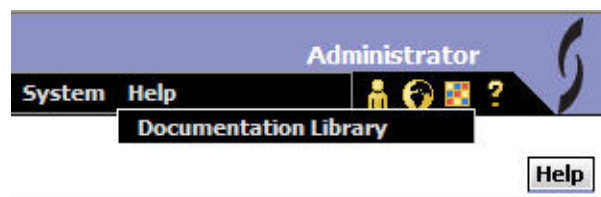


Figure 1. Product Documentation Menu

The Online Documentation Library is also available through the Customer Center at the following URL:

<https://cn.sterlingcommerce.com/login.jsp>

For information about how to enable either of the documentation libraries, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

The Context-Sensitive Help that you can access by clicking the Help button shown in Figure 1 on page 2 provides a single page of help documentation, while the Online Documentation and Local Documentation Libraries provide the entire documentation set. For more information about enabling Context-Sensitive Help, see “Deploying Context-Sensitive Help on WebLogic” on page 131, or “Deploying Context-Sensitive Help on WebSphere” on page 145, or “Deploying Context-Sensitive Help on JBoss” on page 158, as appropriate for your installation.

Chapter 2. Creating a Security Plan

About Creating a Security Plan

This chapter provides security recommendations and guidelines for running Sterling Selling and Fulfillment Foundation. It is intended to help you create a reasonably secure implementation of the application.

Because we recognize that you may have unique business or operational requirements, Sterling Selling and Fulfillment Foundation does not provide a specific set of instructions that you can follow to completion for creating a security plan. Typically, it is not possible to configure a system solely for security at the detriment of other engineering or business realities.

Given the complicated nature of security, it is recommended that you refer to the following documents:

- <http://www.nsa.gov/snac> for tips on how to harden your operating system, database, and network
- http://download.oracle.com/docs/cd/E12840_01/wls/docs103/security/index.html for tips on how to secure Oracle WebLogic 10.3
- <http://support.microsoft.com/ph/2855> for tips on how to secure Microsoft SQL Server 2005 and 2008
- The Rhino9 Team, *The Modern Hackers Desk Reference*; available at <http://www.f4.ca/text/mhdr.html>.
- Tom Bialaski and Michael Haines, *Solaris and LDAP Naming Services, Deploying LDAP in the Enterprise*; Prentice Hall PTR, 2001.

Security-Planning Your Deployment Architecture

Prior to procuring and implementing the hardware and software that make up Sterling Selling and Fulfillment Foundation, you need to plan your deployment architecture by completing the following tasks:

- Conduct an analysis of the current security infrastructure in your organization. For more information on identifying the correct security infrastructure in Sterling Selling and Fulfillment Foundation, see “Security Planning-Current Security Infrastructure Analysis” on page 6.
- Conduct an analysis of authentication and authorization mechanisms in your organization to identify the steps needed to incorporate them into Sterling Selling and Fulfillment Foundation. For more information on the mechanism used for authentication in Sterling Selling and Fulfillment Foundation, see “Security Planning-Authentication and Authorization” on page 6.
- Conduct an analysis of your data encryption mechanisms for deploying Sterling Selling and Fulfillment Foundation over the internet. For more information on the different variations of the data encryption mechanisms, refer to “Security Planning-Data Encryption” on page 6.
- Conduct an analysis of your organization's network topology required to deploy Sterling Selling and Fulfillment Foundation. For more information on the various methods to deploy the application, refer to “Security Planning-Network Topology” on page 6.

Completing these tasks enables you to:

- Estimate your server requirements.
- List the major security software and hardware needed to implement Sterling Selling and Fulfillment Foundation.

Security Planning-Current Security Infrastructure Analysis

In order to ensure that your Sterling Selling and Fulfillment Foundation is a secure Web application, there are many factors involved. Be sure to answer the following questions before you start your Sterling Selling and Fulfillment Foundation implementation.

- Does your organization have security personnel? If not, you may wish to seek input from an Internet security company in your area.
- Do you own a network scanner such as Internet Security Systems System Scanner or Internet Scanner[®]? Products like these help you identify common problems with servers that are exposed to the Internet.
- Do you own an intrusion detection system such as Symantec Intruder Alert? This type of product works with your firewall to stop an intrusion before mission-critical data or systems are tampered with.

Security Planning-Authentication and Authorization

Authentication and authorization are vital to security. Due to the constantly changing authentication methodologies, including biometrics, public key infrastructure (PKI), and ever-increasing encryption algorithms, Sterling Selling and Fulfillment Foundation provides documentation on implementing a lightweight directory access protocol (LDAP) or any Java Authentication and Authorization Service (JAAS) compliant security module for authentication. With LDAP user and password management can be centralized. For information on deploying Sterling Selling and Fulfillment Foundation and integrating with LDAP, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*. The default authentication mechanism is implemented against the Sterling Selling and Fulfillment Foundation database.

Security Planning-Data Encryption

Due to the differences in the nature of businesses, you may implement different security measures when implementing a web application. How you plan to deploy the application and what security measures are taken are unique to each business. Most security measures come at a cost of performance. The Internet is a public network. Sensitive data should be encrypted while traveling across it. Encrypting information that travels across the Internet has an associated cost. If Sterling Selling and Fulfillment Foundation is not to be deployed on the Internet, encryption may not be necessary and the cost is thereby negated.

The data encryption mechanisms recommended for Sterling Selling and Fulfillment Foundation are:

- SSL - 128-bit encryption is the recommended encryption level.
- VPN - 3DES or AES is the recommended encryption algorithm.

Security Planning-Network Topology

Where is Sterling Selling and Fulfillment Foundation being accessed from?

- Public Internet?
- Virtual private network (VPN)?
- Internal Local area network (LAN)?

Sterling Selling and Fulfillment Foundation is typically implemented as an internal application that is accessible from an Internal Network or across from VPN.

Regardless of which network, we strongly recommend that you use SSL to encrypt all the Sterling Selling and Fulfillment Foundation screen requests. SSL processing can be expensive and can add an additional 30% or more processing overhead to each application server transaction. Depending on your transaction volumes, you may want to offload your SSL processing to specialized devices such as an F5 load-balancer with built-in hardware SSL engines.

Accessing Over the Public Internet

If you are accessing Sterling Selling and Fulfillment Foundation over the Public Internet you have to also consider additional security concerns such as denial of service attacks.

Deploying Over a Virtual Private Network

If you are deploying Sterling Selling and Fulfillment Foundation over a virtual private network (VPN), the major factor in security and performance is the VPN encryption. Many firewall providers offer encryption and decryption accelerators that can be added directly to their firewalls. Checkpoint's FireWall-1, VPN-1 Accelerator Card II, is an example of this. However, one consideration for purchasing accelerator cards is how many VPN tunnels are needed. You also need to determine if the VPN is being set up for site-to-site implementation or if each individual user opens their own tunnel. If you decide on a site-to-site VPN, typically memory in the firewall is the greatest concern. If each user opens their own tunnel, processor speed is the largest concern.

In many cases the deciding factor is the speed at which your VPN is connected. If you have a T1 line, a single processor machine may suit your needs. If you plan to deploy over a T3 line, you may wish to consider a multiple-processor machine. Most firewall and VPN vendors can help you size the machine you purchase from them for optimal security and performance.

Deploying Over a Local Area Network

If you are deploying Sterling Selling and Fulfillment Foundation over a local area network (LAN), performance should not be an issue. We strongly recommend you SSL all Sterling Selling and Fulfillment Foundation screens even on an Internal Network.

Security Planning-Java Protocol Security Measures

As with the usage of any protocol technology there are certain associated risks. The Sterling Selling and Fulfillment Foundation APIs are exposed over various protocols. Therefore, IBM strongly recommends that you disable protocols that you do not use.

Disabling Java Protocols-EJB

About this task

To disable Enterprise JavaBeans (EJB) from Sterling Selling and Fulfillment Foundation, comment out the “session” element in the XML descriptor file, <INSTALL_DIR>/repository/eardata/platform/descriptors/<App_Server>/EJB/META-INF/ejb-jar.xml.

Important: To avoid an error when deploying the `ejb-jar.xml` for WebLogic, you must comment out the following session bean of the XML file:

```
<session>
  <display-name> The Sterling Selling and
  Fulfillment Foundation DOM API Session bean
  </display-name>
  <ejb-name> interop.services.ejb.InteropEJBApi </ejb-name>
  <home> com.yantra.interop.services.ejb.InteropEJBHome </home>
  <remote> com.yantra.interop.services.ejb.InteropEJBApi </remote>
  <ejb-class> com.yantra.interop.services.ejb.InteropEJBImpl </ejb-class>
  <session-type> Stateless </session-type>
  <transaction-type> Bean </transaction-type>
</session>
```

This session bean is deprecated as of Release 7.7.

Disabling Java Protocols-HTTP

About this task

To disable Hypertext Transfer Protocol (HTTP) as the means to enter API information in Sterling Selling and Fulfillment Foundation, the deployment descriptor needs to be modified. The deployment descriptor, `web.xml`, is defined by the servlet specification from Sun Microsystems. This deployment descriptor can be used to deploy a web application on any J2EE-compliant application server. The deployment descriptors for Sterling Selling and Fulfillment Foundation are stored in the <INSTALL_DIR>/repository/eardata/smcfs<application_name>/descriptors/<App_Server>/WAR/WEB-INF directory. The deployment descriptor for the `InteropHttpServlet` needs to be removed from the `web.xml` file to disable the servlet. Remember to remove both the `servlet-name` and the `servlet-mapping` entries from this file.

Disabling Java Protocols-JMS

About this task

In order to use the Java Messaging Service (JMS) features of Sterling Selling and Fulfillment Foundation, there must be a JMS server. There must be queues set up both on the JMS Server and within Sterling Selling and Fulfillment Foundation.

To ensure that JMS is not used without authorization there should be appropriate permissions on the JMS server and in Sterling Selling and Fulfillment Foundation. You can limit the ability of users to enable JMS by disabling permissions using Process Modeling in the Applications Manager. For more information about enabling and disabling permissions, see the *Sterling Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

Securing Java Protocols

About this task

Protocols are specified in the `yifclient.properties` file as LOCAL. To specify a different protocol, use the `<INSTALL_DIR>/properties/customer_overrides.properties` file to override the `yif.apifactory.protocol=<protocol_type>` property. Other valid values for `<protocol_type>` are HTTP, HTTPS, and EJB. For additional information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Note: If you use an EJB protocol, you must also add the following property entries to the `<INSTALL_DIR>/properties/customer_overrides.properties` file based on your application server:

For WebLogic:

```
yif.java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory
yif.java.naming.provider.url=t3://<ipaddress>:<port>
```

For WebSphere:

```
yif.java.naming.factory.initial=com.ibm.websphere.naming.WsnInitialContextFactory
yif.java.naming.provider.url=iiop://<ipAddress>:<port>
```

For additional information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Securing Java Protocols-EJB

When the Sterling Selling and Fulfillment Foundation APIs are deployed through EJB, they use a Java Naming and Directory Interface (JNDI) lookup for a context to call the EJB Objects. JNDI looks up a context that is a handle to the EJB Object or API. The APIs do not have authentication or authorization. However, security principal and credentials can be supplied by specifying them in the `yifclient.properties` configuration file. The server can be set up to validate the passed security credentials.

The Sterling Selling and Fulfillment Foundation HTTP/HTTPS Interface uses JavaServer Pages (JSPs) installed on the application server and does not need access to JNDI. There are two ways to protect the Sterling Selling and Fulfillment Foundation APIs over EJB:

- WebLogic allows JNDI and remote method invocation (RMI) to be tunneled over HTTP. In your architecture there should be a proxy to inspect all the requests for Sterling Selling and Fulfillment Foundation. This ensures that all the requests are for HTML, and not tunneled RMI or JNDI over HTTP.
- If Sterling Selling and Fulfillment Foundation is deployed on WebLogic, a security realm should be set up to protect JNDI resources. This does not affect any screens that are packaged with Sterling Selling and Fulfillment Foundation or any screens that extend Sterling Selling and Fulfillment Foundation.

If the application is deployed on WebSphere or JBoss, you must set up permissions for EJB method. This does not affect any standard screens that are packaged with Sterling Selling and Fulfillment Foundation or the custom screens you create.

Important: If you attempt to run Sterling Selling and Fulfillment Foundation using HTTPS, the Applications Manager does not open.

If a custom user interface is being built using the Sterling Selling and Fulfillment Foundation APIs through EJB and not by extending the Sterling Selling and Fulfillment Foundation Presentation Framework, you cannot use the client wrapper supplied with Sterling Selling and Fulfillment Foundation because it currently is incapable of passing credentials. This also applies to any use of the YIFAPIFactory class.

Securing Java Protocols-HTTP API Tester

The HTTP API tester is provided *only* to test APIs in development mode. If you plan to provide access to this page in production, you should secure access to it.

You can use the HTTP API tester to test the upload and download of binary large objects (BLOBs). To upload a BLOB, user information (user ID and password) should already be present in the session. If a session is not already open, you can make a dummy API call so that user information gets stored in the session. You do not need to make a dummy API call to download a BLOB.

To secure access to the Sterling Selling and Fulfillment Foundation `httpapitester`, the deployment descriptor needs to be modified. The deployment descriptor's `web.xml` is defined by the servlet specification from Sun Microsystems. This deployment descriptor can be used to deploy a web application on any J2EE-compliant application server. The deployment descriptor for Sterling Selling and Fulfillment Foundation are stored in the `<INSTALL_DIR>/repository/eardata/smcfs<application_name>/descriptors/<App_Server>/WAR/WEB-INF` directory. By using the security-constraint element with the web-resource-collection element, you can set up authorization to protect this page from unauthorized access. For more information about the `web.xml` deployment descriptor, see the documentation for your application server.

Note: After `buildear.sh` is run, a `web.xml.sample` file is generated in the `<INSTALL_DIR>/repository/eardata/smcfs/extn` folder. To perform any changes to the `web.xml` file, copy the `web.xml.sample` file to the same folder (`<INSTALL_DIR>/repository/eardata/smcfs/extn`) and rename it to `web.xml`. Now perform changes to the `web.xml` file in the `<INSTALL_DIR>/repository/smcfs/extn` folder.

Alternatively, you can simply remove the `yfshttpapi` directory under `<INSTALL_DIR>/repository/eardata/platform/war` and secure the `/interop/InteropHttpServlet` servlet using the security features provided by your application server.

Specify the following URL to access the HTTP API tester: `http://<ipaddress>:<port>/smcfs/yfshttpapi/yantrahttpapitester.jsp`

Securing Java Protocols-COM+

The extended Component Object Model (COM+) specification covers security in great detail. Any COM+ object deployed on a server complies with this standard. For information on setting up security for COM+ objects, see *The Microsoft Developers Network* article available at: <http://msdn.microsoft.com/en-us/library/ms681314.aspx>

Web Security Planning

IBM highly recommends that a security audit is made prior to deployment.

IBM also recommends that you write log files to several servers. There are several applications that do this with no specific need for Sterling Selling and Fulfillment Foundation to duplicate their efforts. Additionally, products like Symantec's Intruder Alert monitor log files for authentication failures and alert an administrator if a threshold is exceeded.

Web Security Planning-Post installation Recommendations

About this task

After the installation of Sterling Selling and Fulfillment Foundation, be sure to complete the following for ensured security:

Procedure

1. Change the password of the default user (admin).
2. Change permissions on <INSTALL_DIR>/bin/migrator.* files to non-executable.
3. API security is enabled during installation. After installation, you may want to reset the property `api.security.mode` and carefully consider your API security configuration. For more information about API security modes, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.
4. Ensure that the required permissions to access the API resources are defined for an application. For more information about configuring API security, refer to the *Sterling Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

Web Security Planning-Session Security

Session security is handled by the application server, and is stored in a non-persistent cookie on the client. You should ensure that all transactions with the application server are protected with SSL to prevent session hijacking attacks.

Web Security Planning-Operating System Permissions

The following files contain confidential information, such as user name and password combinations stored in clear text. These files should be secured through operating system permissions:

- `sandbox.cfg`
- `<appserver>.log`
- `jdbc.properties.in`
- `yfs.properties.in`
- `yifclient.properties.in`

Web Security Planning-Documentation

All the documentation files for Sterling Selling and Fulfillment Foundation and third-party software should be removed from any production servers.

Web Security Planning-Routing

Routing should not be enabled on a production web server.

Web Security Planning-Web Server Executables

Web servers should *not* be run as root. This ensures that if someone compromises any software associated with the deployment through a bug, they don't have root privileges to damage the server. Web servers allow you to access files on their host

machines and as root any of those files can be modified for a deeper attack or deleted to make your web servers unavailable.

It is acceptable, although not recommended, to start the web server as root. A proxy server can be used to accept HTTP traffic and redirect it to a port above 1024 on a UNIX system. If a proxy is not available and the web server must be started on port 80 it is necessary to start the web server as root. The web server then calls `setuid` to transfer root privileges to a generic unprivileged account. The web server's configuration file should allow you to specify what user it runs as. Any user may own the binary. The `setuid` bit should not be set on the web server binary.

Database Security

Set up separate accounts on the database server for installing the Sterling Selling and Fulfillment Foundation schema and for accessing the application database.

If using an Oracle database on the production database server, the Oracle parameter `DBLINK_ENCRYPT_LOGIN` in your `init.ora` file should be set to `TRUE`. This ensures that all connections to the database are not sent as clear text.

Database Security-Credit Card Encryption

IBM provides an application, the IBM Sterling Sensitive Data Capture Server, that captures and tokenizes credit card numbers and store value card numbers. IBM recommends that you review the *Sterling Selling and Fulfillment Foundation: Secure Deployment Guide* for the IBM approach to meeting PCI DSS and PA-DSS requirements.

If you want to ensure that credit card numbers are encrypted at the database level, you configure that functionality when setting Hub attributes in the Applications Manager. When setting Hub attributes, make sure that the credit card number encrypting option is checked. For more information and specific instructions for setting up security, see the *Sterling Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

Sterling Selling and Fulfillment Foundation also supplies APIs and user exits to encrypt credit card and other secure information. For more information about these APIs, user exits, and other data encryption, see the *Sterling Selling and Fulfillment Foundation: Extending Transactions* and the *Sterling Selling and Fulfillment Foundation: Javadocs*.

Internet Explorer Security Settings

When using IBM Sterling Selling and Fulfillment Foundation without any customizations, you may need to set security or privacy settings for your Internet Explorer in order to obtain the best browser performance.

Configuring Browser Security Settings

About this task

To configure the Internet Explorer security and privacy settings:

Procedure

1. From the Internet Explorer menu, select **Tools > Internet Options > Security**.

2. Select the Web content zone from which Sterling Selling and Fulfillment Foundation is accessed.
3. Choose Default Level and set the security level to High.
4. Depending on the version of Internet Explorer you have installed, choose Custom Level and set your security settings according to one of the following tables.

Table 1. Internet Explorer Version 7 Security Settings for Sterling Selling and Fulfillment Foundation

Internet Explorer Security Setting	Sterling Selling and Fulfillment Foundation
ActiveX Controls and Plug-ins	
Download signed ActiveX controls	Prompt/Enable
Download unsigned ActiveX controls	Disable
Initialize and script ActiveX controls not marked as safe	Disable
Run ActiveX controls and plug-ins	Prompt/Enable
Script ActiveX controls marked as safe for scripting	Enable
Downloads	
File download	Enable
Font download	Prompt
Microsoft VM	
Java permissions	High Safety
Miscellaneous	
Access data sources across domains	Disable
Allow META REFRESH	Disable
Display mixed content	Prompt
Do not prompt for client certificate selection when no certificates or only one certificate exists	Disable
Drag and drop or copy and paste files	Prompt
Installation of desktop items	Disable
Launching programs and files in an IFRAME	Disable
Navigate sub-frames across different domains	Disable
Software channel permissions	High Safety
Submit non-encrypted form data	Prompt
Userdata persistence	Disable
Scripting	
Active scripting	Enable
Allow paste operations via script	Enable
Scripting of Java applets	Enable
User Authentication	
Logon	Prompt for user name and password

Table 2. Internet Explorer Version 7 Privacy Settings for Sterling Selling and Fulfillment Foundation

Advanced Privacy Setting	Sterling Selling and Fulfillment Foundation
Cookies	
Override automatic cookie handling	Yes
First-Party Cookies	Block
Third-Party Cookies	Block
Always allow session cookies	Yes

Adding Sterling Selling and Fulfillment Foundation as a Trusted Web site

About this task

You should set Sterling Selling and Fulfillment Foundation to be recognized as a trusted Web site. Not doing so could cause certain pop-up windows such as date and time selection to display a status bar, thereby hiding certain action buttons.

To add Sterling Selling and Fulfillment Foundation to the list of trusted Web sites:

Procedure

1. In the Internet Explorer menu bar, select Tools > Internet Options. The Internet Options pop-up window is displayed.
2. In the Internet Options pop-up window, select the Security tab.
3. Click the Trusted Sites icon.
4. Click the Sites action button. The Trusted Sites pop-up window is displayed.
5. In the 'Add this Web site to the zone' text box, enter the server address where the Application Console is installed. The port number does not need to be specified.
6. Clear the 'Require server verification (HTTPS:) for all sites in this zone' checkbox.
7. Click OK. This takes you back to the Internet Options pop-up window.
8. Click OK.

Chapter 3. Installing and Configuring Application Tier Software

Installing Your Application Server

Before installing your application server, check the requirements in the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* to make sure you have the applicable hardware and software versions installed.

Install your application server according to the vendor instructions on the product CD-ROM disk.

You should install the Java Development Kit (JDK) that is shipped with your application server (unless otherwise specified in the *Sterling Selling and Fulfillment Foundation: System Requirements Guide*. When upgrading the JDK, be sure to set the correct JAVA_HOME environment variable and update the PATH.

Installing and Configuring a Proxy Server

Installing a proxy web server on dedicated hardware provides:

- Additional network security layers.
- Additional processing power for data encryption protocols.
- Additional options for high availability for your application.

You can install a proxy or Web server to avoid any bottlenecks that may occur when systems try to access Sterling Selling and Fulfillment Foundation installed on your application server. Install and configure the Web server version specified by your application server provider.

About Configuring a Proxy Server for SSL or HTTPS Configuration

You can configure a Web server as an SSL proxy and a load balancer. For an example of an Apache and WebLogic, see "Configuring a Proxy Server for SSL or HTTPS-Example." For specific instructions about configuring the proxy server for SSL or HTTPS using the Apache HTTP Server and IBM WebSphere, refer to the appropriate IBM documentation. Information about configuring the Apache HTTP server as a proxy server and a load balancer using SSL or HTTPS on JBoss can be found at: <http://community.jboss.org/wiki/UsingModproxyWithJBoss>

The SSL proxy allows the web server to manage the SSL encryption load and pass clear text back to application servers. It also divides the workload among the available application servers using the "round-robin load balancing" algorithm. This reduces the network traffic between the web server and application server. The web server enables you to use one secure URL to access any number of application servers that run IBM Sterling Selling and Fulfillment Suite applications.

Configuring a Proxy Server for SSL or HTTPS-Example About this task

The following procedure explains how to configure a proxy server for WebLogic. Refer to the WebLogic example provided here as a general guideline for

configuring a Web server. For information about how to configure a proxy server for WebSphere and JBoss, refer to the documentation pertaining to those products.

Before you begin: The Oracle WebLogic 10.3 installation does not include the Apache HTTP server plug-ins. You must download these plug-ins in a separate compressed file from the Oracle download and support sites.

To configure a proxy server for SSL or HTTPS using the Apache HTTP Server and Oracle WebLogic:

Procedure

1. Install and run Sterling Selling and Fulfillment Foundation on the application servers.

2. Copy the appropriate plug-in to the `/etc/apache2/modules` directory.

For WebLogic 10.3.2, this is:

```
<WL_HOME>/wlserver_10.3/server/plugin/<OS>/<processor type>
```

Here, `<WL_HOME>` refers to the WebLogic installation directory.

Example:

```
<WL_HOME>/wlsl03Linux/wlserver_10.3/server/plugin/linux/x86_64
```

- For `i686`, copy the WLS plug-in.
 - For `x86_64`, copy the 64-bit plug-in. The 64-bit plug-in must be requested from Oracle Customer Service.
 - Ensure the plugin is executable.
3. To enable the WebLogic plug-in for load-balancing using HTTP or HTTPS, modify the `httpd.conf` file and add the following. To enable an SSL, add `include conf.d/ssl.conf` as instructed by Apache. By default, `RHAS3` has `include conf.d/*`, which includes `ssl.conf`.

```
LoadModule weblogic_module /etc/apache2/modules/<appserver_plugin_file>
```

For an HTTP proxy, outside any `VirtualHost`, add the following section. The `context_root` value is the `context_root` for the web application being proxied:

```
<IfModule mod_weblogic.c>
    WebLogicCluster
    <managed_server1_hostname/IP_address>,<managed_server2_hostname/IP_address>
    DynamicServerList OFF
    Debug ON
    Idempotent OFF
</IfModule>
<Location /context_root>
    SetHandler weblogic-handler
</Location>
```

4. Modify the `ssl.conf` file and add the following lines to the `<VirtualHost _default_:443>` section. The `context_root` value is the `context_root` for the web application being proxied.

```
<IfModule mod_weblogic.c>
    WebLogicCluster
    <managed_server1_hostname/IP_address>,<managed_server2_hostname/IP_address>
    DynamicServerList OFF
    Debug ON
    Idempotent OFF
</IfModule>
<Location /context_root>
    SetHandler weblogic-handler
</Location>
```

5. Create security or SSL certificate, if necessary. If you do not have a CA-signed certificate, you can get one from the Certificate Authority companies such as

VeriSign. For more information about security or SSL certificates, see “Deploying RCP - Security Certificates” on page 188.

- Restart Apache, and verify access with any browser.
Continue and complete steps 7-10 if using an RCP application only.
- Copy the security certificate to the <RCP_EXTENSIONS_FOLDER>/truststore directory.
- Build the RCP client.
- Edit the `locations.ycfg` file and modify the protocol, server, and port attributes of the Config element. Ensure that these attributes point to the proxy.
- Start the client.

Setting Up an Image Server for RCP

About this task

If fetching images for RCP-based PCAs, you must set up an image server. You can set up any server (such as Apache) as your image server.

To set up the image server:

Procedure

- Install a web server on any system on which you intend to host the images. For example, you can install an Apache web server on a Windows system.
- Use the default port # 80 (or any available port #) while installing the Image Server & exclude this port from the OS firewall, if required.
- Store the images in any convenient location under the <IMAGE_SERVER_HOME> directory. For example, you can store the images under the following directory:

```
<IMAGE_SERVER_HOME>/icons/rcp
```

Here, <IMAGE_SERVER_HOME> refers to the name of the directory to which the web server that you have installed points.

For more information about configuring connection settings to fetch images from the server, see “Deploying RCP - Connection Settings for Fetching Images from the Server” on page 185.

Setting Up Apache Image Server-Example

About this task

This example illustrates the procedure given in “Setting Up an Image Server for RCP.” If you install Apache as the web server, then to configure it as the image server, do the following:

Procedure

- Edit the `httpd.conf` file to define an alias directive. You can find this file under the following directory structure:

```
<APACHE_HOME>/conf/httpd.conf
```

Here, <APACHE_HOME> refers to the name of the directory where you have installed Apache.

The following is a sample entry from the `httpd.conf` file:

```
Alias /icons "<INSTALL_DIR>/COM/images/icons/"
<Directory "<INSTALL_DIR>/COM/images/icons">
AllowOverride None
Order allow, deny
Allow from all
</Directory>
```

where /icons is the <virtual dir path> that points to the <INSTALL_DIR>/COM/images/icons/ directory.

For more information about how to define alias directives, go to http://httpd.apache.org/docs/2.2/mod/mod_alias.html#alias. This link provides information about alias directives for Apache version 2.2.

2. Add a new entry or edit the existing entry for configuring the port. For example, add a new entry: Listen 80 in the httpd.conf file. This sets up the server to listen to port number 80 (default setting).
3. Restart the web server.

When you apply the above configuration, the URL `http://<IMAGE_SERVER_HOST_NAME>:<port>/icons` points to the local directory `<INSTALL_DIR>/COM/images/icons/` and the contents in the local directory are served by the web server.

Results

Finally, test to ensure that the images are accessible through the browser; for example, `http://<IMAGE_SERVER_HOST_NAME>:<port>/<virtual dir path>/rcp/<IMAGE_FILE_NAME>` from any system. If the images are not displayed, the image server is not configured properly.

Chapter 4. Before Installing and Configuring a Database

Overview

Use the following database sizing exercises to help you estimate the requirements for your database.

Database Sizing

Database sizing is designed to give you estimates of the database growth and to assist in planning the disk requirements. The planning of the capacity required in your company and the steps to estimate the disk size are described in “Database Capacity Planning,” “Disk Estimation for IBM Sterling Distributed Order Management,” and “Disk Estimation for the Networked IBM Sterling Warehouse Management System Module” on page 21.

Database Capacity Planning

There are many factors to consider when estimating the amount of disk space that is required for Sterling Selling and Fulfillment Foundation. As a result, trying to consider all growth factors is impractical because the user may not know the answers to many questions that are required to do a detailed forecast. Over the years the cost of disks has dramatically decreased, and the capacity and speed of disks has increased. The method of how information system managers order disk capacity has also changed from purchasing disk arrays that are dedicated to a particular database server and project to the concept of SANS.

The Sterling Selling and Fulfillment Foundation provides a methodology to estimate your initial disk requirements. Consider the confidence that you have in your data estimates when making the final purchase decision and adjust accordingly. After the initial purchase and production deployment, disk growth should be tracked for future purchase forecasts.

- If you use or are planning to use IBM Sterling Distributed Order Management, see “Disk Estimation for IBM Sterling Distributed Order Management.”
- If you use or are planning to use both Sterling Distributed Order Management and the networked IBM Sterling Warehouse Management System, see “Disk Estimation for IBM Sterling Distributed Order Management” and “Disk Estimation for the Networked IBM Sterling Warehouse Management System Module” on page 21.
- If you are planning to use only Sterling Warehouse Management System see “Disk Estimation for the Networked IBM Sterling Warehouse Management System Module” on page 21.

Disk Estimation for IBM Sterling Distributed Order Management

About this task

Estimate the disk space you will need for the Sterling Distributed Order Management module of Sterling Selling and Fulfillment Foundation.

The estimation methodology consists of three parts:

1. Estimate the number of orders and order lines you expect to keep in the database.
2. Multiply the number obtained in the previous step by a storage usage factor.
3. Add a minimum base amount.

Keep the following information in mind before calculating the estimated disk space:

- Gather information about the amount of time required to maintain the database, such as:
 - How long do you plan to keep data in the main transactional database before orders are purged to the history database?
 - How long are orders kept in the history database before they are purged?
 - Are you purchasing the storage for the first few years into the implementation?

Use the following examples to help answer these questions:

- **Case 1** - You need to purchase storage for the first 3 years of the implementation, and your company's data retention policy says that you have to keep data online in the main transactional database for 1 year and in the history database for another 5 years. Orders that are older than 6 years are purged from the system.

The following solution lets you achieve this goal:

If you need to purchase storage to cover the first 3 years of implementation, that storage has to be sufficient for 3 years worth of data. At the end of year 3, your database has the data for the third year in the main transactional database while the data for the first and second years is in the history. In this example, you should enter the number 3 as the number of years worth of orders that you expect to keep in the database.

- **Case 2** - Sterling Selling and Fulfillment Foundation has been in production for 10 years and your company's data retention policy says that you have to keep data online in the main transactional database for 1 year and in the history database for another 5 years. Orders that are older than 6 years are purged from the system. Given the same data retention policy as above, how much storage is required?

At the end of the tenth year, the database has the data for the tenth year in the main transactional database and the data for the fifth, sixth, seventh, eighth and ninth years in the history. Therefore, the database has six years (as dictated by the data retention policy) in the database. In this example, you should enter the number 6 as the number of years worth of orders that you expect to be kept in the database.

- The order in Table 3 on page 21 includes sales, transfer, return, and work orders.
- This storage estimate is for work-in-progress tables that are used as part of order processing. When the orders are processed, the records in these tables can be purged from the system. These tables include the YFS_IMPORT, YFS_EXPORT, and so forth. You are strongly urged to aggressively purge data from these tables.
- When procuring your storage, ensure that the storage device has at least the amount of usable space specified in Step 8 of Table 3 on page 21. This table provides an idea of the usable space for the storage device in your company. However, the actual amount you might require is a factor of Redundant Array of Inexpensive Disks (RAID) set up. This disk subsystem is composed of more than one disk drive to provide improved reliability, response time, and storage capacity.

Use the following worksheet to help you estimate required disk space.

Table 3. Steps for Disk Space Estimation for the Sterling Distributed Order Management Module

Step	Description	Disk Space
1.	Enter the number of years' worth of information to be kept in the system (retention time).	_____
2.	Enter the number of orders you expect to be in the system during the time period specified in Step 1.	_____
3.	Enter the number of order lines present in a typical order.	_____
4.	Enter the number of order lines that are to be stored in the database (multiply the values provided in Step 2 and Step 3).	_____
5.	Enter the order line multiplier: Choose one of the following storage factors that most closely approximates a description of your Sterling Selling and Fulfillment Foundation system: (a) 30 KB - This is primarily used for order management with very little customization. (b) 35 KB - This is primarily used for order management with moderate amount of customization.	_____
6.	Multiply the expected number of order lines from Step 4 and the storage factor from Step 5.	_____
7.	The minimum base storage requirement.	150 MB
8.	The minimum operational storage requirements for Sterling Selling and Fulfillment Foundation.	500 MB
9.	Enter the total estimated storage obtained by adding the values from Step 6, Step 7, and Step 8.	_____

Disk Estimation for the Networked IBM Sterling Warehouse Management System Module

About this task

Estimate the database disk storage needed for the networked Sterling Warehouse Management System module of Sterling Selling and Fulfillment Foundation.

This estimation methodology consists of three parts:

Procedure

1. Estimate the number of shipment lines you expect to keep in the database.
2. Multiply the number obtained in Step 1 by a storage usage factor depending on the specifics of your implementation.
3. Add a minimum base amount for each warehouse or stockroom that you have defined.

Results

If you are planning to use both the Sterling Distributed Order Management and Sterling Warehouse Management System modules, use Table 4 on page 22.

If you are planning to use only the Sterling Warehouse Management System module, use Table 5 on page 24.

Keep the following information in mind before calculating the estimated disk space:

- Gather information about maintaining the database, such as:
 - How long do you plan to keep data in the main transactional database before shipment data is purged to the history database?
 - How long is the shipment data kept in the history database before it is purged?
 - Are you purchasing the storage for the first few years into the implementation?

Example

You need to purchase storage for the first 2 years of the implementation, and your company's data retention policy says that you have to keep data online in the main transactional database for 1 year and in the history database for another year. Shipments that are older than 2 years are purged from the system.

The following solution lets you achieve this goal:

If you need to purchase storage to cover the first 2 years of implementation, that storage has to be sufficient for 2 years worth of data. At the end of year 2, your database has data from the second year in the main transactional database while the data from the first year is in the history. In this example, you should enter the number 2 as the number of years worth of shipment-related data that you expect to keep in the database.

- The shipment lines discussed in Table 4 and Table 5 on page 24 include space requirements for demand-based replenishment.
- This storage estimate is for work-in-progress tables that are used as part of the shipment and receipt processing. When the shipments are processed, the records in these tables can be purged from the system. These tables include the YFS_IMPORT, YFS_EXPORT, YFS_TASK, YFS_TASK_STATUS_AUDIT, and so forth. You are strongly urged to aggressively purge data from these tables.
- When procuring your storage, ensure that the storage device has at least the amount of usable space specified in the last step of Table 4 or Table 5 on page 24. These tables provide an idea of the usable space for the storage device in your company. However, the actual amount you may need is a factor of Redundant Array of Inexpensive Disks (RAID) set up. This disk subsystem is composed of more than one disk drive to provide improved reliability, response time and storage capacity.

Use the worksheet in Table 4 or Table 5 on page 24 to help you estimate required disk space.

Table 4. Disk Space Estimation for the Networked Sterling Warehouse Management System Module if Using Sterling Distributed Order Management and Sterling Warehouse Management System

Step	Description	Disk Space
1.	Enter the number of years worth of information to be kept in the system (retention time).	_____
2.	Enter the number of shipment lines you expect to be in the system during the time period specified in Step 1.	_____

Table 4. Disk Space Estimation for the Networked Sterling Warehouse Management System Module if Using Sterling Distributed Order Management and Sterling Warehouse Management System (continued)

Step	Description	Disk Space
3.	<p>Enter the shipment line multiplier. This includes demand-based replenishment. Choose one of the following storage factors that most closely approximates a description of your Sterling Selling and Fulfillment Foundation system:</p> <p>(a) 10 KB - For warehouses using no tag-controlled items, no serial tracking, no LNPNs or only Pallet LPNs, and more than 80% PARCEL shipping.</p> <p>(b) 12 KB - For warehouses using no tag-controlled items, no serial tracking, no LNPNs or only Pallet LPNs, and PARCEL as well as TL - LTL shipping.</p> <p>(c) 15 KB - For warehouses using tag-controlled items, no serial tracking, no LNPNs or only Pallet LPNs, and more than 80% TL - LTL shipping.</p> <p>(d) 20 KB - For warehouses using no tag-controlled items, no serial tracking, largely CASE LPNs, and more than 80% PARCEL shipping.</p> <p>(e) 25 KB - For warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and PARCEL as well as TL - LTL shipping.</p> <p>(f) 30 KB - For warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and more than 80% TL - LTL shipping.</p>	_____
4.	Multiply the expected number of shipment lines from Step 2 and the storage factor from Step 3.	_____
5.	Enter the number of receipt lines you expect to be in the system during the time period specified in Step 1.	_____
6.	<p>Enter the receipt line multiplier. Choose one of the following storage factors that most closely approximates a description of your Sterling Selling and Fulfillment Foundation system:</p> <p>(a) 25 KB - For warehouses using no tag-controlled items, no serial tracking, and no LNPNs.</p> <p>(b) 27 KB - For warehouses using no tag-controlled items, no serial tracking, and no LNPNs or only Pallet LPNs.</p> <p>(c) 35 KB - For warehouses using no tag-controlled items, no serial tracking, and more than 80% CASE LPNs.</p> <p>(d) 40 KB - For warehouses using tag-controlled items, serial tracking, and more than 80% CASE LPNs.</p>	_____
7.	Multiply the expected number of receipt lines from Step 5 and the storage factor from Step 6.	_____
8.	<p>Enter the number of warehouses planned:</p> <p>(a) Enter the number of stores or stock rooms planned.</p> <p>(b) Enter the number of other warehouses planned.</p>	_____ _____

Table 4. Disk Space Estimation for the Networked Sterling Warehouse Management System Module if Using Sterling Distributed Order Management and Sterling Warehouse Management System (continued)

Step	Description	Disk Space
9.	Calculate the minimum space required for your set up based on the data in Step 8 and the minimum storage requirement given below: (a) 20 MB for each store or stock room. (b) 50 MB for each other warehouse.	_____ _____
10.	The minimum operational storage requirements for Sterling Selling and Fulfillment Foundation.	500 MB
11.	Enter the total estimated storage obtained by adding the values from Step 4, Step 7, Step 8, Step 9, and Step 10.	_____
12.	Enter the value of Step 9 from Table 3 on page 21.	_____
13.	Enter the total estimated storage obtained by adding the values from Step 11 and Step 12.	_____

Table 5. Estimating Disk Space for Networked Sterling Warehouse Management System Module - If Using Sterling Warehouse Management System Only

Step	Description	Disk Space
1.	Enter the number of years worth of information to be kept in the system (retention time).	_____
2.	Enter the number of shipment lines you expect to be in the system during the time period specified in Step 1.	_____
3.	Enter the shipment line multiplier. This factor includes demand-based replenishment. Choose from one of the following storage factors that most closely approximates a description of your Sterling Selling and Fulfillment Foundation system: (a) 25 KB - For warehouses using no tag-controlled items, no serial tracking, no LNPNS or only Pallet LPNs, and more than 80% PARCEL shipping. (b) 27 KB - For warehouses using no tag-controlled items, no serial tracking, no LNPNS or only Pallet LPNs, and PARCEL as well as TL - LTL shipping. (c) 30 KB - For warehouses using tag-controlled items, no serial tracking, no LNPNS or only Pallet LPNs, and more than 80% TL - LTL shipping. (d) 35 KB - For warehouses using no tag-controlled items, no serial tracking, largely CASE LPNs, and more than 80% PARCEL shipping. (e) 40 KB - For warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and PARCEL as well as TL - LTL shipping. (f) 50 KB - For warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and more than 80% TL - LTL shipping.	_____

Table 5. Estimating Disk Space for Networked Sterling Warehouse Management System Module - If Using Sterling Warehouse Management System Only (continued)

Step	Description	Disk Space
4.	Multiply the expected number of shipment lines from Step 2 and the storage factor from Step 3.	_____
5.	Enter the number of receipt lines you expect to be in the system during the time period specified in Step 1.	_____
6.	Enter the receipt line multiplier. Choose from one of the following storage factors that most closely approximates a description of your Sterling Selling and Fulfillment Foundation system: (a) 25 KB - For warehouses using no tag-controlled items, no serial tracking, and no LNPNs. (b) 27 KB - For warehouses using no tag-controlled items, no serial tracking, and no LNPNs or only Pallet LPNs. (c) 35 KB - For warehouses using no tag-controlled items, no serial tracking, and more than 80% CASE LPNs. (d) 40 KB - For warehouses using tag-controlled items, serial tracking, and more than 80% CASE LPNs.	_____
7.	Multiply the expected number of receipt lines from Step 5 and the storage factor from Step 6.	_____
8.	Enter the number of warehouses planned: (a) Enter the number of stores or stock rooms planned. (b) Enter the number of other warehouses planned.	_____ _____
9.	Calculate the minimum space required for your set up based on the data in Step 8 and the minimum storage requirement given below: (a) 20 MB for each store or stock room. (b) 50 MB for each other warehouse.	_____ _____
10.	The minimum operational storage requirements for Sterling Selling and Fulfillment Foundation.	500 MB
11.	Enter the total estimated storage obtained by adding the values from Step 4, Step 7, Step 8, Step 9, and Step 10.	_____

Tracking and Estimating Future Disk Requirements

You should track your actual database storage usage and the number of database records regularly. Correlating these two metrics enabled you to plan your future disk requirements. Moreover, determining the average amount of space used for each order line or shipment line, enables you to accurately predict your future growth requirements.

Chapter 5. Installing and Configuring Database Tier Software on UNIX or Linux

Database Configuration on UNIX/Linux-Overview

This chapter describes how to install and configure the database tier software to run Sterling Selling and Fulfillment Foundation in a UNIX or Linux environment.

Before installing your database server, verify that you have the applicable software versions. For more information, see *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.

If you are planning a multischema installation, you must deploy the same database vendor and version across all deployments.

Overview of Oracle Database Configuration

About this task

You need to configure your Oracle database for running in a production environment with Sterling Selling and Fulfillment Foundation. To configure an Oracle database for a production environment, you must:

- Size the database by estimating the required disk space.
- Create views and db_link or synonyms for integrating with the IBM Sterling Warehouse Management System installation.
- Set the database connection properties.

To create the Oracle database to handle multiple byte characters

Procedure

1. Choose the correct data encoding format for your language. See "Configuring Oracle with Single or Multiple Byte Characters" for more information.
2. Choose the character set suitable for your language. See "Configuring Oracle with Single or Multiple Byte Characters" for specific settings to ensure the database field sizes.

Results

You can use an Oracle database for maintaining information on Sterling Selling and Fulfillment Foundation.

Configuring Oracle with Single or Multiple Byte Characters

About this task

To configure Oracle with single or multiple byte characters:

Procedure

1. After installing Oracle, run the create instance procedure. Use a character set appropriate for your desired language. For example:
`CHARACTER SET "UTF8"`
2. Configure the INIT<INSTANCE_NAME>.ORA file for Oracle as follows:

open_cursors= <set to appropriate value>

For example, the minimum value for WebLogic equals number of threads (across all application servers) + (connection pool size X prepared statement pool size)

cursor_sharing=FORCE

compatible=<set to appropriate value, or remove to default to current release>

timed_statistics=true

db_block_size=8192

optimizer_mode=ALL_ROWS

If you are using a multi-byte character set, set the following and restart Oracle:

nls_length_semantics=CHAR

Alternatively you can run the following prior to running any create table scripts:

```
alter session set nls_length_semantics = CHAR
```

Setting this attribute ensures that the field sizes are not impacted by the number of bytes a data type can store. For example, Varchar(40) would now be able to store 40 Japanese characters instead of 40/3 bytes in the UTF-8 character set.

For the Japanese locale, the AL32UTF-8 character set or the UTF-16 character set must be used.

When you change the multi-byte character set to CHAR by setting nls_length_semantics = CHAR, Oracle reserves space equivalent to 'n' chars, which is more than 'n' bytes. Therefore, when you run the dbverify.sh command, the reduced entries in table columns are printed in the EFrame_Drops.lst file.

3. Download the Oracle JDBC driver for the version of Oracle you are using from the Oracle Web site and copy it to a well known location for reference during installation.

The Oracle JDBC driver can be found in the JDBC Driver Downloads section at:

http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

See the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* for Oracle database supported version information.

Results

If configuring for multibyte, do not modify the Sterling Selling and Fulfillment Foundation DDL.

Oracle Database User Privileges

Unless specifically stated for a given task, the Sterling Selling and Fulfillment Foundation user does not require database administrator (DBA) privileges.

Oracle Administrator Privileges

The following list includes some of the basic privileges that should be granted to the Sterling Selling and Fulfillment Foundation administrative user who creates or modifies the Oracle database:

- Alter any sequence
- Alter session
- Create any sequence
- Create procedure

- Create sequence
- Create session
- Create synonym
- Create table
- Create trigger
- Create type
- Create view
- Delete any table
- Execute any procedure
- Execute any type
- Connect
- Insert any table
- Select any dictionary
- Select any sequence
- Select any table
- Select catalog role
- Update any table

Assign the user responsible for creating and modifying the Oracle database a specified quota (extent) in the tablespace, even if that user was assigned an unlimited tablespace when created. Otherwise, the installer will generate a "ORA-01950: no privileges on tablespace name" error.

Oracle Privileges for Application Users

The following list contains basic privileges that should be granted to users who will only run the application.

The Privileges/Grant is only for the tables owned by the user. The following is an example script to create table update grants:

```
select 'grant update on '||table_name||' to [user];'
from user_tables
```

You can write additional scripts to create similar permissions for the following privileges:

- Alter session
- Create session
- Delete table
- Execute procedure
- Insert table
- Select sequence
- Select table
- Update table
- If you are using text indexes, you must also have privileges for CTXAPP or CTXCAT, depending on the type of text indexes you are using.

Manually Creating Views on Oracle

To configure your Oracle database for your production environment, you must set up and run two scripts, `Interop_Views.sql` and `ImportExport_View.sql`, to create the views for your schema.

These script files reside in the `install_dir/database/oracle/scripts/CustomDBViews/transaction` directory.

If you used the silent install method and set the `-nodbverify` option to true, you will also need to create the tables, indexes, sequences, and so forth for your schema.

Table, index, and sequence create DDLs are created during installation. These reside in `install_dir/repository/scripts` directory.

Setting Up Scripts When Using Locally Managed Tablespaces or Other Utility to Size Database

About this task

To configure your Oracle database for your production environment, you must set up and run a series of scripts to create the tables, indexes, sequences, and so forth for your schema.

These script files reside in the `<INSTALL_DIR>/database/oracle/scripts/` directory. The `yfs_master_db_script.sql` script is the master script that calls all view scripts required for creating views. Table, index, and sequence creation DDLs are created during installation. These reside in the `<INSTALL_DIR>/repository/scripts` directory.

You must run these scripts only if you are manually creating the views after installation (`REINIT_DB=no`). In the normal installation mode (`REINIT_DB=yes`), the views will be applied automatically.

Refer to “Enabling the Oracle Database Text Search Feature - Overview” on page 112 for information about text search features on the Oracle database.

Setting Database Parameters in Oracle

For information about required parameter settings in your Oracle database, see the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

Rolling Back or Undoing Changes in Oracle

About this task

You can roll back or undo changes in Oracle by using the `AUTO UNDO` management option. It is recommended that you use this option. This avoids manual monitoring of UNDO segments.

If a server is upgraded from a previous version of Oracle, set the `UNDO_MANAGEMENT=AUTO` parameter in `init<SID>.ora`. Your database administrator needs to determine the `UNDO_RETENTION` setting. Ensure that the file system which has the `UNDOTBS1` tablespace has enough space to use the `AUTOGROW` setting.

Enabling Failover in a Multiple Node Oracle RAC Database Cluster (UNIX/Linux)

About this task

To enable failover in a multiple node Oracle RAC database cluster in UNIX/Linux, do the following:

Procedure

1. Navigate to the <INSTALL_DIR>/properties directory, where you will modify the `sandbox.cfg` and `customer_overrides.properties` files.

Note: You might need to create the `customer_overrides.properties` file, which is just for customizations and is not automatically created during an installation. For additional information about the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

2. In the `sandbox.cfg` file, add a new property for `ORACLE_JDBC_URL` which contains the Oracle RAC connection URL.

The following example shows the suggested URL form. This example shows how the information is organized, but the property value must be one string of text, starting with `ORACLE_JDBC_URL=`. Your database administrator (DBA) can modify this URL as needed.

```
jdbc:oracle:thin:@
  (DESCRIPTION=
    (ADDRESS_LIST=
      (FAILOVER=ON)
      (LOAD_BALANCE=ON)
      (ADDRESS=(PROTOCOL=TCP)(HOST=myhost1)(PORT=1521))
      (ADDRESS=(PROTOCOL=TCP)(HOST=myhost2)(PORT=1521))
    )
    (CONNECT_DATA = (SERVER = DEDICATED)(SERVICE_NAME = myservicename))
  )
```

3. In the `customer_overrides.properties` file, add the `readTimeout` property to all Oracle database pools. These values override the corresponding values in the `jdbc.properties` file.

- `jdbcService.oraclePool.prop_jdbc.readTimeout=90000`
- `jdbcService.oraclePool_local.prop_jdbc.readTimeout=90000`
- `jdbcService.oraclePool_NoTrans.prop_jdbc.readTimeout=90000`
- `jdbcService.oraclePool_NoTrans.prop_jdbc.readTimeout=90000`

The `readTimeout` value will require tuning. If the value is too low, long-running queries in the system will be interrupted. If the value is too high, recovery when a RAC node fails will be delayed.

4. Run the `setupfiles.sh` command from the <INSTALL_DIR>/bin directory.
5. Set the propagation delay on the RAC server to 0.

Configuring the NLS_LANG Parameter for an Oracle Client

About this task

To ensure the compatibility of character sets between the Oracle client and the server, the value of the `NLS_LANG` parameter that is set in the client must match the value in the server.

The entire set of NLS settings pertaining to the database is provided in the NLS_DATABASE_PARAMETERS table.

Procedure

1. Run the following queries to get the corresponding values:
 - SELECT VALUE as Language FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER='NLS_LANGUAGE';
 - SELECT VALUE as Territory FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER='NLS_TERRITORY';
 - SELECT VALUE as Characterset FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER='NLS_CHARACTERSET';

The NLS_LANG parameter is set as: <Language>_<Territory>.<Characterset> (for example, set NLS_LANG = AMERICAN_AMERICA.UTF8)
2. To set the value of the NLS_LANG parameter in Windows, verify the HKEY_LOCAL_MACHINE/SOFTWARE/ORACLE/NLS_LANG entry in the registry.
3. To set the value of the NLS_LANG parameter in UNIX, NLS_LANG is set as a local environment variable.

Configuring DB2 on UNIX or Linux

Procedure

1. Create the database. Refer to the DB2 documentation for information about creating the database, including creating a schema repository, login, and tablespace. Be sure to install the correct version and fix pack. Be sure to install the client components and compilers before you install the fix pack.
2. Size the database by estimating the required disk space.
3. Set the database connection properties.

Note: The installation script creates tables and indexes. Certain tables require a page size of 32K. You should have a tablespace to accommodate such tables. DB2 automatically places tables and indexes in the available tablespaces using its internal logic. You can move the tables to a different tablespace after the installation is complete.

4. Install Client Components, Compilers, and Fix Pack.
5. Set Parameters for DB2.
6. Grant Permissions for DB2.
7. Install JDBC Drivers for DB2.

Installing Client Components, Compilers, and Fix Pack

About this task

Sterling Selling and Fulfillment Foundation uses stored procedures for DB2®. You must install or set up the following components:

Procedure

1. Install the Administration client.
2. Install the Sterling Selling and Fulfillment Foundation Development clients.
3. Install the necessary fix pack after you install the client components and compilers. Otherwise, the clients will overwrite the fix pack binaries. See the

Sterling Selling and Fulfillment Foundation: System Requirements Guide for information about the fix pack level supported.

4. Set the path for the compiler by using the `db2set` command.

Results

For more information about these tasks, see the relevant IBM documentation.

DB2 Parameter Settings

For information about required parameter settings in your DB2 database, see the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

DB2 JDBC Drivers

About this task

For DB2, install the appropriate DB2 JDBC Type 4 driver and any correlating patches. See System Requirements for supported version information.

You can obtain these files from the IBM Web site. After you obtain this JDBC driver, record the absolute path to its location on your system. You must supply this absolute path during installation.

If the JDBC driver provided by your database vendor is distributed among multiple files, you must place all the files that comprise the JDBC driver into one .jar file. Follow these steps to create one .jar file:

Procedure

1. Identify all the vendor database jar files for the JDBC driver.
2. Create a temporary working directory (`mkdir wd; cd wd`).
3. Extract the contents of each file used for the JDBC driver using the jar utility into the temporary working directory (`jar xvf <jdbc.jar>` for each supplied jar file).

Note: Various IBM Sterling Selling and Fulfillment Foundation scripts, such as the one used for loading the factory defaults, specify a `DB_DRIVER`. The `DB_DRIVER` specified must include all of these JAR files. The `DB_DRIVER` setting is located in `sandbox.cfg`. To make changes to the `DB_DRIVER` setting, edit and save the file, then run `setupfiles.sh`.

4. Bundle the files in the temporary working directory into one file using the jar utility (`jar cvf new.jar *`).
5. Record the absolute path to this .jar file.

Results

The type-4 driver does not require a separate Java listener running on the database server. Instead, connect directly to the DB2 port.

Tuning Considerations for DB2

When setting up DB2, the Self-Tuning Memory Manager (STMM) should initially be enabled. As the database grows, these settings may be adjusted to tune the system loads. Refer to the *Sterling Selling and Fulfillment Foundation: Performance Management Guide* and the IBM documentation regarding these settings.

Registry Settings for DB2 on UNIX or Linux

For Sterling Selling and Fulfillment Foundation, set the registry variables as follows (the variables with nothing following the equal sign ensure the default setting):

```
db2set DB2_MMAP_WRITE=OFF
db2set DB2_MMAP_READ=OFF
db2set DB2_PINNED_BP=
db2set DB2MEMMAXFREE=
```

```
db2set DB2_ENABLE_BUFDPD=
db2set DB2_USE_ALTERNATE_PAGE_CLEANING=ON
db2set DB2_EVALUNCOMMITTED=ON
db2set DB2_SKIPDELETED=ON
db2set DB2_SKIPINSERTED=ON
db2set DB2_SELECTIVITY=ALL
db2set DB2LOCK_TO_RB=STATEMENT
db2set DB2_NUM_CKPW_DAEMONS=0
```

The *Sterling Selling and Fulfillment Foundation: Performance Management Guide* section on recommended UDB dbset registry variables describes the effects of these parameters with Sterling Selling and Fulfillment Foundation.

DB2 User Privileges

The DBADM role is required for performing administrative operations in the DB2 database.

Manually Creating Objects on DB2-Setting Up Scripts

About this task

To set up the scripts:

Procedure

1. Create tablespaces where the Sterling Selling and Fulfillment Foundation tables and indexes reside.
2. Only complete this step if you are manually creating database tables after installation (instead of having installation create them automatically). Modify the <INSTALL_DIR>/repository/scripts/EFrame_TableChanges.sql file to reference your newly created tablespaces.

Results

The DDLs in the Sterling Selling and Fulfillment Foundation scripts create a standard set of indexes. You may need to create additional indexes according to your business practice.

Manually Creating Objects on DB2-Running Scripts

About this task

To run the scripts:

Procedure

1. Log into the DB2 server manager as the database administrator.
2. Create the user that is the designated schema owner.

3. Grant the privileges listed in “DB2 User Privileges” on page 34 to the newly created user.
4. Log out of the DB2 Server Manager and log back in as the newly created user.
5. Verify the database as described in “Verifying the Database Schema” on page 114.
6. Load the Sterling Selling and Fulfillment Foundation database factory defaults as described in “Loading the Sterling Selling and Fulfillment Foundation Database Factory Defaults After Installation” on page 119.
7. Check for the degree of parallelism, using information from the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

Manually Creating Views on DB2

About this task

To configure your DB2 database for your production environment, you must set up and run a series of scripts to create the tables, indexes, sequences, and so forth for your schema.

You must run these scripts only if you are manually creating the views after installation (REINIT_DB=no). In the normal installation mode (REINIT_DB=yes), the views will be applied automatically.

These script files reside in the <INSTALL_DIR>/database/db2/scripts/ directory.

This is the list of scripts to be edited using a SQL tool:

- CustomDBViews/transaction/ImportExport_View.sql
- CustomDBViews/transaction/Interop_Views.sql
- CustomDBViews/transaction/InvSnapshot_vw.sql
- CustomDBViews/transaction/yfs_cross_reference_vw.sql
- CustomDBViews/transaction/yfs_iba_ord_demand_vw.sql
- CustomDBViews/transaction/yfs_iba_resv_demand_vw.sql
- CustomDBViews/transaction/yfs_invtmddtl_vw.sql
- CustomDBViews/transaction/yfs_noPendMove_nodeInventoryDtl_vw.sql
- CustomDBViews/transaction/yfs_onlyLPN_nodeInventoryDtl_vw.sql
- CustomDBViews/transaction/
yfs_onlyLPN_noPendMove_nodeInventoryDtl_vw.sql
- CustomDBViews/transaction/yfs_order_release_line.sql
- CustomDBViews/transaction/yfs_order_release_line_vw.sql
- CustomDBViews/transaction/yfs_nodeInventoryDtl_vw.sql
- CustomDBViews/configuration/yfs_wave_item_volume_vw.sql
- CustomDBViews/master/ycm_pricelist_vw.sql
- CustomDBViews/master/ypm_category_item_vw.sql
- CustomDBViews/master/ypm_item_vw.sql

Enabling the Text Search Feature on DB2

About this task

To enable the text search feature on DB2 database:

Procedure

1. Make sure that the DB2 database is configured with the Net Search Extender plug-in.
2. Log in to the DB2 server using the Command Editor or Command Line Processor with a user ID having DBA privileges.
3. Verify that the text search index creation was successful.
Use the `customer_overrides.properties` file that is located in the `<INSTALL_DIR>/properties` directory to set the `yfs.db.textsearch` property to `Y`.

Results

The text search indexes that are created on DB2 database using the `<INSTALL_DIR>/repository/scripts/EFrame_TextIndexAdds.sql` script are automatically updated every 6 hours. The DBA can modify this script to change this frequency, if necessary. Before running the `EFrame_TextIndexAdds.sql` script, the DBA must update the `"/*Database*/"` string in the `EFrame_TextIndexAdds.sql` script and specify the database name.

For information on how to create the text search indexes, see the *Sterling Selling and Fulfillment Foundation: Customizing Console JSP Interface for End-User* .

Chapter 6. Installing and Configuring Database Tier Software on Windows

Database Configuration on Windows-Overview

This chapter describes how to install and configure the database tier software to run Sterling Selling and Fulfillment Foundation in a Windows environment.

Before installing your database server, verify that you have the applicable software versions. For more information, see *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.

If you are planning a multischema installation, you must deploy the same database vendor and version across all deployments.

Configuring a Database Server on Windows

You must install, create, and configure a database so that each Sterling Selling and Fulfillment Foundation instance has a dedicated schema and login for the database.

Configuring a Database for a Multiple Schema Environment

If you are planning a multischema installation, you must deploy the same database vendor and version across all deployments.

Generating Data Backup and Restore Scripts

Procedure

To generate the backup and restore scripts, run the backupScriptGen.xml script located in the <INSTALL_DIR><INSTALL>/bin directory using the following command:

```
sci_ant.cmd -f backupScriptGen.xml -DbType=<database_type>  
<INSTALL><INSTALL>This script generates sample backup and restore scripts in the  
<INSTALL_DIR><INSTALL>/bin/sample directory.
```

You can rename and customize the scripts to suit your business needs. For example, you can modify the script to add your custom configuration tables and modify the path where the data files are stored. These scripts depend on utilities provided by the database vendors.

The backupScriptGen.xml script accepts the following arguments:

Table 6. backupScripGen.xml Arguments

Argument	Purpose	Accepted Values
-Dos=	<p>Determines what kind of script is generated.</p> <p>If "windows" is selected, a .cmd file is created.</p> <p>If "linux" or "unix" is selected, a .sh file is created.</p> <p>If "all" is selected, both .cmd and .sh files are created.</p>	<ul style="list-style-type: none"> • windows • unix • linux • all
-DbType=	<p>Determines for which databases the scripts will be generated.</p>	<ul style="list-style-type: none"> • oracle • db2 • sqlserver • iseries • all
-DtableType=	<p>Determines which entities to generate scripts for. Valid values are any TableType attribute defined for an entity.</p>	<ul style="list-style-type: none"> • ALL • CONFIGURATION • MASTER • METADATA • STATISTICS • TRANSACTION <p>Note: The value ALL is supported only for single-schema mode.</p>

Results

Running the backupScriptGen.xml script creates both backup and restore scripts for the selected operating systems and databases.

- Oracle scripts depend on export, import, or sqlplus utilities. You can modify and use the following scripts:
 - backup_config_oracle.cmd
 - restore_config_oracle.cmd
 - delete_configuration_oracle.sql
- MSSQL Server scripts depend on bcp or osql utilities. You can modify and use the following scripts:
 - backup_config_sqlserver.cmd
 - restore_config_sqlserver.cmd
 - delete_configuration_oracle.sql

Configuring MSSQL Server on Windows

You can use an MSSQL Server database for maintaining information on Sterling Selling and Fulfillment Foundation. See the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* for specific version information.

Install MSSQL Server before installing Sterling Selling and Fulfillment Foundation.

Follow the installation procedures in your MSSQL Server installation manual. Refer to the MSSQL Server documentation for information about creating the database, including creating a schema repository, login, and tablespace. Be sure to install the correct version and service pack.

Special Settings and Considerations for MSSQL Server Configuration

When configuring MSSQL Server, follow these directions:

- Ensure that Named Pipes & TCP/IP protocols are enabled in the network utility of the MSSQL Server.
- For MSSQL Server, do not use case-sensitive column names in the database. Case-sensitive names will prevent the MSSQL Server System Management Console from loading.

Database Parameter Setting in MSSQL Server

To prevent deadlocks, set the parameter `READ_COMMITTED_SNAPSHOT` to `ON` at the database level.

For more information about required parameter settings in your MSSQL Server database, see the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

MSSQL Server User Privileges

The following roles are required for creating objects in the Microsoft SQLServer database:

- `db_accessadmin`
- `db_datareader`
- `db_datawriter`
- `db_ddladmin`
- `db_owner`

Running Scripts for an MSSQL Server Database

About this task

To run the scripts:

Procedure

1. Make sure you have an MSSQL Server client installed on your computer.
2. From the `<INSTALL_DIR>\database\sqlserver\scripts` directory, run the `yfssqlserver_master_db_script.cmd` script. This runs all of the required scripts using an MSSQL Server command-line utility.
3. Examine the log files for errors.
4. If the application is installed on a non-Windows machine, all sql scripts under `<INSTALL_DIR>\database\<dbtype>\scripts\CustomDBViews` and `yfs_seq_sqlserver.sql` must be applied manually.
5. When running DBVerify-generated SQL scripts for MSSQL Server, you may see warnings about maximum row size. You can ignore these warnings.
6. However, when adding indexes on MSSQL Server, you may see warnings similar to the following for indexes containing large, variable-length columns:

- Warning! The maximum key length is 900 bytes. The index 'YFS_PROPERTY_I1' has maximum length of 910 bytes. For some combination of large values, the insert/update operation will fail.
- Warning! The maximum key length is 900 bytes. The index 'YFS_CATEGORY_I1' has maximum length of 976 bytes. For some combination of large values, the insert/update operation will fail.
- Warning! The maximum key length is 900 bytes. The index 'YFS_COUNT_STRATEGY_I1' has maximum length of 1136 bytes. For some combination of large values, the insert/update operation will fail.
- Warning! The maximum key length is 900 bytes. The index 'YFS_INVENTORY_SUPPLY_TMP_I4' has maximum length of 1044 bytes. For some combination of large values, the insert/update operation will fail.
- Warning! The maximum key length is 900 bytes. The index 'YFS_OBJECT_LOCK_I1' has maximum length of 1096 bytes. For some combination of large values, the insert/update operation will fail.
- Warning! The maximum key length is 900 bytes. The index 'YFS_XREF_VALUE_UI1' has maximum length of 1008 bytes. For some combination of large values, the insert/update operation will fail.

Results

If the data entered into these columns is under 900 bytes, you can ignore these messages. If the data is over 900 bytes in the indexed columns, drop the indexes. API calls to insert data into these tables must not violate the unique columns.

Enabling the Text Search Feature on MSSQL Server

About this task

To enable the text search feature on the MSSQL Server database:

Procedure

1. Make sure that the Microsoft Search service is running on the machine on which the MSSQL Server is installed.

By default, the full-text engine automatically runs as a service named Microsoft Search on Microsoft Windows, NT[®] Server 4.0, and Windows[®] 2000.

2. Log in to the MSSQL Server manager with a user ID that has DBA privileges.

Note: The text search indexes that are created on the MSSQL Server using the <INSTALL_DIR>\repository\scripts\EFrame_TextIndexAdds.sql script are not automatically updated. Before running the EFrame_TextIndexAdds.sql script, the DBA must update the "*Database*" string in the EFrame_TextIndexAdds.sql script and specify the database name.

3. Verify that the text search index creation was successful.
4. From the <INSTALL_DIR>\repository\scripts directory, run the EFrame_TextIndexModify.sql script to enable the text search indexes to be incrementally updated when a text search enabled column is modified.
5. Edit the customer_overrides.properties file that is located in the <INSTALL_DIR>\properties directory to add the following entries:

```
yfs.yfs.db.textsearch=Y
```


Results

For information on how to create the text search indexes, see the *Sterling Selling and Fulfillment Foundation: Customizing Console JSP Interface for End-User* .

Configuring an Oracle Database on Windows

About this task

You can use an Oracle database for maintaining information on Sterling Selling and Fulfillment Foundation. To configure an Oracle database for use in a Sterling Selling and Fulfillment Foundation production environment, you must:

Procedure

1. Create the database. Refer to the Oracle documentation for information about creating the database, including creating a schema repository, login, and tablespace.
2. Size the database by estimating the required disk space.
3. Set the database connection properties.
4. Set the Oracle database parameters.
5. Grant Permissions in Oracle.
6. Create views and db_link or synonyms for integrating with the IBM Sterling Warehouse Management System installation.

Setting Database Parameters in Oracle

For information about required parameter settings in your Oracle database, see the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

Rolling Back or Undoing Changes in Oracle

You can roll back or undo changes in Oracle using the AUTO UNDO management option. It is recommended that you use this option. This avoids manual monitoring of UNDO segments.

Configuring Oracle for Single or Multiple Byte Characters

About this task

To set up Oracle for use with single or multiple byte characters:

Procedure

1. Run the Oracle create instance procedure. Use a character set appropriate for your desired language. For example:

```
CHARACTER SET "UTF8"
```

2. Configure the INIT<INSTANCE_NAME>.ORA file for Oracle as follows:

```
open_cursors= <set to appropriate value>
```

```
For example, the minimum value for WebLogic equals number of threads  
(across all application servers) + (connection pool size X prepared statement  
pool size)
```

```
cursor_sharing=FORCE
```

```
compatible=<set to appropriate value, or remove to default to current release>
```

```
timed_statistics=true
```

```
db_block_size=8192
```

```
optimizer_mode=ALL_ROWS
```

If you are using multi-byte character set, set the following and restart Oracle:

```
nls_length_semantics=CHAR
```

Alternatively you can run the following prior to running any create table scripts:

```
alter session set nls_length_semantics = CHAR
```

Setting this attribute ensures that the field sizes are not impacted by the number of bytes a data type can store. For example, Varchar(40) would now be able to store 40 Japanese characters instead of 40/3 bytes in the UTF-8 character set.

For the Japanese locale, the AL32UTF-8 character set or the UTF-16 character set must be used.

When you change the multi-byte character set to CHAR by setting `nls_length_semantics = CHAR`, Oracle reserves space equivalent to 'n' chars, which is more than 'n' bytes. Therefore, when you run the `dbverify.cmd` command, the reduced entries in table columns are printed in the `EFrame_Drops.lst` file.

3. Download the Oracle JDBC driver for the version of Oracle you are using from the Oracle Web site and copy it to a well known location for reference during installation.

The Oracle JDBC driver can be found in the JDBC Driver Downloads section at: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

See the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* for Oracle database supported version information.

Results

To create the Oracle database to handle multiple byte characters, do not modify the Sterling Selling and Fulfillment Foundation DDL.

Oracle Database User Privileges

Unless specifically stated for a given task, the Sterling Selling and Fulfillment Foundation user does not require database administrator (DBA) privileges.

Administrator Privileges

The following list includes some of the basic privileges that should be granted to the Sterling Selling and Fulfillment Foundation administrative user who creates or modifies the Oracle database:

- Alter any sequence
- Alter session
- Create any sequence
- Create procedure
- Create sequence
- Create session
- Create synonym
- Create table
- Create trigger
- Create type
- Create view
- Delete any table

- Execute any procedure
- Execute any type
- Connect
- Insert any table
- Select any dictionary
- Select any sequence
- Select any table
- Select catalog role
- Update any table

Assign the user responsible for creating and modifying the Oracle database a specified quota (extent) in the tablespace, even if that user was assigned an unlimited tablespace when created. Otherwise, the installer will generate a "ORA-01950: no privileges on tablespace name" error.

Application User Privileges

The following list contains basic privileges that should be granted to users who will only run the application.

The Privileges/Grant is only for the tables owned by the user. The following is an example script to create table update grants:

```
select 'grant update on '||table_name||' to [user];'
from user_tables
```

You can write additional scripts to create similar permissions for the following privileges:

- Alter session
- Create session
- Delete table
- Execute procedure
- Insert table
- Select sequence
- Select table
- Update table
- If you are using text indexes, you must also have privileges for CTXAPP or CTXCAT, depending on the type of text indexes you are using.

Manually Creating Views on Oracle

To configure your Oracle database for your production environment, you must set up and run two scripts, Interop_Views.sql and ImportExport_View.sql, to create the views for your schema.

These script files reside in the <INSTALL_DIR>\database\oracle\scripts\CustomDBViews\transaction directory.

Table, index, and sequence create DDLs are created during installation. These reside in the install_dir\repository\scripts directory.

If you plan to use the silent install method and set the `-nodbverify` option to true, you will also need to create the tables, indexes, sequences, and so forth for your schema.

Refer to “Enabling the Oracle Database Text Search Feature - Overview” on page 112 for information about text search features on the Oracle database.

Enabling Failover in a Multiple Node Oracle RAC Database Cluster in Windows

About this task

To enable failover in a multiple node Oracle RAC database cluster in Windows, do the following:

Procedure

1. Navigate to the `<INSTALL_DIR>/properties` directory, where you will modify the `sandbox.cfg` and `customer_overrides.properties` files.

Note: You might need to create the `customer_overrides.properties` file, which is just for customizations and is not automatically created during an installation. For additional information about the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

2. In the `sandbox.cfg` file, add a new property for `ORACLE_JDBC_URL` which contains the Oracle RAC connection URL.

The following example shows the suggested URL form. This example shows how the information is organized, but the property value must be one string of text, starting with `ORACLE_JDBC_URL=`. Your database administrator (DBA) can modify this URL as needed.

```
jdbc:oracle:thin:@
  (DESCRIPTION=
    (ADDRESS_LIST=
      (FAILOVER=ON)
      (LOAD_BALANCE=ON)
      (ADDRESS=(PROTOCOL=TCP)(HOST=myhost1)(PORT=1521))
      (ADDRESS=(PROTOCOL=TCP)(HOST=myhost2)(PORT=1521))
    )
    (CONNECT_DATA = (SERVER = DEDICATED)(SERVICE_NAME = myservicename))
  )
```

3. In the `customer_overrides.properties` file, add the `readTimeout` property to all Oracle database pools. These values override the corresponding values in the `jdbc.properties` file.

- `jdbcService.oraclePool.prop_jdbc.readTimeout=90000`
- `jdbcService.oraclePool_local.prop_jdbc.readTimeout=90000`
- `jdbcService.oraclePool_NoTrans.prop_jdbc.readTimeout=90000`
- `jdbcService.oraclePool_NoTrans.prop_jdbc.readTimeout=90000`

The `readTimeout` value will require tuning. If the value is too low, long-running queries in the system will be interrupted. If the value is too high, recovery when a RAC node fails will be delayed.

For additional information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

4. Run the `setupfiles.sh` command from the `<INSTALL_DIR>/bin` directory.
5. Set the propagation delay on the RAC server to 0.

Chapter 7. Installing Sterling Selling and Fulfillment Foundation in a Windows Environment

Use the topics in this section to prepare for installation, run the installation, and as a reference for postinstallation tasks.

- “Windows Preinstallation Tasks”
- “Running an Installation on Windows” on page 51
- “Post Installation Tasks for Windows” on page 62
- “Upgrade Information for Windows” on page 62

Windows Preinstallation Tasks

Complete the tasks in this section prior to installing Sterling Selling and Fulfillment Foundation

- “Before You Begin Installation in a Windows Environment”
- “Windows Preinstallation Checklist” on page 47
- “Key Terms for Installation” on page 49
- “About Applying Database Definition Language (DDL) Statements After a Windows Installation” on page 49
- “Types of Installation Available on Windows” on page 50
- “Differences Between Single and Multiple Schema Installations” on page 51

Before You Begin Installation in a Windows Environment

- If you are upgrading from a prior release, see the *Sterling Selling and Fulfillment Foundation: Upgrade Guide* before continuing with the installation of Sterling Selling and Fulfillment Foundation.
- Before installing Sterling Selling and Fulfillment Foundation, ensure that you already have installed the appropriate software listed in the *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.
- Throughout this document, <INSTALL_DIR> refers to the directory where you install Sterling Selling and Fulfillment Foundation, for example, <INSTALL_DIR>=C:/Supply_Chain_Apps.
- During the installation, various messages are displayed, including some warning messages. These warning messages require no action on your part and are included so that helpful data is recorded in the log file.
- A root user cannot install Sterling Selling and Fulfillment Foundation.
- The following instructions assume that you received an installation CD for Sterling Selling and Fulfillment Foundation. If you downloaded Sterling Selling and Fulfillment Foundation or a Service Pack (SP) from the Electronic Software Distribution (ESD) Portal, extract the downloaded file to an empty directory. Do not change the directory structure of the newly extracted files. The directory containing the extracted files is an electronic image of an installation CD. Use this directory wherever there is a reference to the installation CD in the following instructions. Ignore any instructions to place the installation CD in a drive.
- The SMCFS_9.1.jar must be in a directory on the Windows server where you will install Sterling Selling and Fulfillment Foundation. If you are using FTP to copy the file, verify that your session is set to binary mode.

- In silent installation mode, the directory path to SMCFS_9.1.jar cannot include any spaces.
- If using silent installation, record the path to your silent installation file.
- To install more than one instance of Sterling Selling and Fulfillment Foundation on the same Windows server, you must install the second instance in a different directory and use a different initial port number. This second port number must be at least 100 higher than the first port number.
- During the installation, various messages are displayed, including some warning messages. These warning messages require no action on your part and are included so that helpful data is recorded in the log file.
- The GUI installation creates the following installation directory structure: <INSTALL_DIR>\Foundation. The Sterling Selling and Fulfillment Foundation files and subdirectories are installed under <INSTALL_DIR>\Foundation.
- The silent installation process creates an <INSTALL_DIR>, and the Sterling Selling and Fulfillment Foundation files and subdirectories are installed directly under it.
- In windows, if using silent installation, the directory path to SMCFS_9.1.jar cannot include any spaces.
- About creating database views: Views must be created manually. Instructions to create views vary depending on which database is used. All database view related scripts are located at <INSTALL_DIR>\database\<db_type>\scripts. For more information, see the information about configuring your database type in Chapter 6, "Installing and Configuring Database Tier Software on Windows," on page 37 and Chapter 14, "Configuring Utilities," on page 111.

Including Memory Parameter Settings in the Installation Command

When invoking the install script, you will need to add memory parameters. During installation, these parameter values are written to the sandbox.cfg property file, where you can tune them after installation.

The example parameter values that follow are recommended for the stack shown; however, your memory requirements may differ. See the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* for your specific supported stack requirements.

Example Memory Parameter Settings for Installation

The -XX:MaxPermSize parameter indicates the maximum amount of memory. In the command, the memory parameter values **512m**, **256m**, and **1408m** were tested on the following stack:

Operating System: Windows VISTA and RHL 5.3 64-bit

Data Base: Oracle 11g

JDK: SUN JSDK 1.5.0_11 and above

Application Server: WebLogic 10.3

Example based on Windows:

```
<JAVA_HOME>/bin/java -Djavataskargs="-XX:MaxPermSize=512m"
-Dcomptaskargs="-J-Xms1024m -J-Xmx2048m" -jar SCIInstallWizard.jar
```

- When creating a name, such as an account name, permissions name, profile name, or database name, follow these conventions:
 - Use any valid alphanumeric characters and -, :, \$, &, or _.
 - Do not use spaces or apostrophes.

About the JDK for Windows

- To determine which JDK version you need, see the *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.
- In Windows, the directory name where the JDK resides cannot include a space.
- Edit the JAVA_HOME environment variable to point to the directory where you installed the JDK.
- After you install the JDK, record the absolute path to its location on your system. You must supply the absolute path when you install Sterling Selling and Fulfillment Foundation.
- During the installation, you will be prompted whether to have the installer create a copy of the JDK under your install folder. If you do not want the JDK copied, download version 2.7.1 of Xerces and Xalan from Apache prior to running the installation. Copy the following files into the jre/lib/endorsed directory of your JDK:
 - serializer.jar
 - xalan.jar
 - xercesImpl.jar
 - xml-apis.jar

Windows Preinstallation Checklist

The preinstallation checklist for the Windows environment identifies the tasks you must complete before installing Sterling Selling and Fulfillment Foundation.

You may want to make a copy of the following checklist and use it to record the information you collect.

Table 7. Preinstallation Checklist for Windows

Step	Description	Your Notes
1	Verify that your system meets the hardware and software requirements specified for the current release. See the <i>Sterling Selling and Fulfillment Foundation: System Requirements Guide</i> for detailed information.	
2	If you are using a non-English environment, confirm that you are using the appropriate character set.	

Table 7. Preinstallation Checklist for Windows (continued)

Step	Description	Your Notes
3	<p>Determine and record information about the JDK.</p> <ul style="list-style-type: none"> • Version of the JDK • Absolute path to the JDK. The path name can not include spaces. <p>During the installation, you will be prompted whether to have the installer create a copy of the JDK under your install folder. If you do not want the JDK copied, do the following prior to install:</p> <p>Download Version 2.7.1 of Xerces and Xalan from Apache. Copy the following files into the <code>jre/lib/endorsed</code> directory of your JDK:</p> <ul style="list-style-type: none"> • <code>serializer.jar</code> • <code>xalan.jar</code> • <code>xercesImpl.jar</code> • <code>xml-apis.jar</code> 	
4	<p>Determine and record information about your Oracle, MSSQL Server or DB2 database server. Be aware that this information may be case sensitive.</p> <ul style="list-style-type: none"> • Database vendor • Database user name and associated password • Database catalog name • Database host name (or IP address) • Database host port number • (Oracle and MSSQL Server) Absolute path and file name for one JDBC driver. • (DB2 only) Absolute paths and file names for two JDBC driver files. • If you are planning a multischema deployment, determine and record this database information for four separate schemas: Metadata, Statistics, System Configuration, and Transaction/Master Data. 	
5	<p>Decide if you are going to manually or automatically apply database definition language (DDL) statements (schema) to the database.</p> <p>For more information, refer to “About Applying Database Definition Language (DDL) Statements After a Windows Installation” on page 49.</p>	

Table 7. Preinstallation Checklist for Windows (continued)

Step	Description	Your Notes
6	<p>Determine and record the directory in which you plan to install Sterling Selling and Fulfillment Foundation.</p> <ul style="list-style-type: none"> • There must be a minimum of 5GB free disk space on the drive specified for installation • The name of the directory can not include spaces and must be less than 30 characters long. • If you use the silent installation method, you cannot install into a pre-existing directory. The silent installation process will fail if a pre-existing directory is specified. See “Types of Installation Available on Windows” on page 50 for more information about using the silent installation method. • If you use the GUI installation method, you can install into either a pre-existing directory or a new directory to be created by the installation process. 	
7	<p>If you are running a silent installation, create a silent installation text file, using a text editor. For more information, refer to “Types of Installation Available on Windows” on page 50.</p>	

Key Terms for Installation

The following terms and definitions will assist you in understanding installation concepts.

- Database catalog name – This is typically the database name; also known as SERVICE_NAME or SID in some versions of Oracle.
- Java Software Development Kit (JDK) – Software development kit (SDK) for producing Java programs. Produced by Sun Microsystems, Inc., the JDK includes JavaBeans component architecture and support for JDBC.

About Applying Database Definition Language (DDL) Statements After a Windows Installation

When you install Sterling Selling and Fulfillment Foundation, you can manually apply database definition language (DDL) statements to your database tables instead of requiring the installation process to do it directly. This enables you to apply DDL statements for database creation separately from the installation.

This feature increases database security by reducing the database permissions of the Sterling Selling and Fulfillment Foundation database user. The rights to create tables, indexes, etc. can be reserved for a secure user like a database administrator (DBA). A business can require that only a DBA with the proper permissions can make database changes.

If you choose to manually apply the DDL, the installation process will provide the location of the DDL scripts. The installation process will then continue the rest of

the installation. The installation process may validate the database with DBVerify and warn you if there are differences, but it will not exit. It will allow the processing of the packages to continue normally.

If you do not choose to manually apply the DDL, the installation will apply both the DDL and the resources.

During the GUI installation, you indicate whether or not you want to set up DDLs and run factory setup on the Upgrade/Database Options screen. Here, you choose one of the following:

- If you are upgrading from a previous release, or if you do not want the database DDLs and Factory Setup installation done as part of this installation process, click Yes for the option *Do you want to upgrade from a previous installation or perform new installation with no Database DDLs and Factory setup?* Click Next. If you are not upgrading and you select this option, you must manually create your database tables and load factory setup after the installation process. However, if you are upgrading, the DDLs are applied as part of the upgrade process.
- If you are not upgrading from a previous release, or if you do want the installation process to apply the database DDLs and install factory setup, click No, and click Next.

See the section about configuring your database in Chapter 6, “Installing and Configuring Database Tier Software on Windows,” on page 37 for information about running view scripts after the initial installation and Chapter 14, “Configuring Utilities,” on page 111 for information about manually installing the database DDLs and factory setup.

Types of Installation Available on Windows

Interactive Installation using GUI

You can run the installation from a script, `SCIInstallWizard.jar`, that starts an interactive install wizard program. It enables you to step through a GUI wizard to enter the necessary information about your database, JDK driver, selected install folder, and so on. After you've entered this information, you start the installation by clicking the Install button. A status screen remains open, and displays messages about the progress of the installation. Once Sterling Selling and Fulfillment Foundation is installed, the status screen displays an install complete message. See “Running a GUI-Based Installation on Windows” on page 51.

The Windows GUI installer installs Sterling Selling and Fulfillment Foundation in the Foundation subdirectory, unlike the silent installation, which installs the product directly into the folder you specify.

Silent Installation

Instead of using the interactive method and being prompted to enter all the information needed by the install, you can create a silent installation text file that contains this information. You invoke the installation from the command line and reference the text file, which is then used by the installer. See “Running a Silent Installation on Windows” on page 59.

The silent installation installs the product directly into the folder that you specify in the silent install text file.

Differences Between Single and Multiple Schema Installations

In a multischema installation, you must deploy the same database vendor and version across deployments.

Running an Installation on Windows

Use the topics in this section to install Sterling Selling and Fulfillment Foundation.

- “Running a GUI-Based Installation on Windows”
- “Creating a Silent Installation File - Windows” on page 54
- “Windows Silent Install File Parameters” on page 55
- “Sample Silent Installation File - Windows” on page 58
- “Running a Silent Installation on Windows” on page 59
- “Running a Silent Installation in Multischema Mode” on page 59

Running a GUI-Based Installation on Windows

About this task

To install Sterling Selling and Fulfillment Foundation in a Windows environment, refer to your preinstallation checklist and follow the steps below.

Procedure

1. Close all open Windows programs and any command prompt windows.
2. Copy the SCIIInstallWizard.jar and SMCFS_9.1.jar files to a Windows directory on the server where Sterling Selling and Fulfillment Foundation is to be installed.
3. At a command prompt, browse to the directory where SCIIInstallWizard.jar is located. Enter the following command, using memory values you entered in the preinstallation checklist: `<JAVA_HOME>/bin/java -Djavataskargs="-XX:MaxPermSize=<value_1>" -Dcomtaskargs="-J-Xms<value_2>-J-Xmx<value_3>" -jar SCIIInstallWizard.jar`

The installation dialog box is displayed.

4. Click Next to start the installation program.
5. Type the path of your JDK directory, or search for it using **Select Folder**.
If you want a local copy of the JDK to be created in your installation, click Yes. Otherwise, click No. Click Next.
6. On the Upgrade/Database Options screen, do one of the following:

- If you are upgrading from a previous release, or if you do not want the database DDLs and Factory Setup installation done as part of this installation process, click Yes for the option *Do you want to upgrade from a previous installation or perform new installation with no Database DDLs and Factory setup?* Click Next.

If you are not upgrading and you select this option, you must manually create your database tables and load factory setup after the installation process. See the section about configuring your database for production in Chapter 6, “Installing and Configuring Database Tier Software on Windows,” on page 37 for information about running view scripts after the initial installation and Chapter 14, “Configuring Utilities,” on page 111 for information about manually installing the database DDLs and factory setup. However, if you are upgrading, the DDLs are applied as part of the upgrade process.

- If you are not upgrading from a previous release, or if you do want the installation process to apply the database DDLs and install factory setup, click No, and click Next.
7. Choose an installation directory for Sterling Selling and Fulfillment Foundation. Enter the directory name, or browse to the directory. For Windows 2000/NT server, the path to this directory must be 8 characters or less and cannot include any spaces. This directory is referred to as <INSTALL_DIR> in subsequent steps. After selecting the folder, click Next.
If the directory does not exist, a message asking if the directory should be created is displayed. Click Yes to create the directory or No to return to the previous screen.
 8. The Please Confirm Install Jar Location screen opens and displays the path for the install jar file. To confirm the location, click Next. Or you can search for the file using the Select File button, select the file, and then click Next.
 9. Select the database vendor that you want to use (Oracle, MSSQL Server, or DB2) and click Next.
 10. If do not want to enable MultiSchema support, click No, click Next, and skip to Step 11 on page 53. To enable MultiSchema support, click Yes, click Next, and proceed as follows:
 - a. If you are performing an installation in upgrade mode, skip to Step B. However, if you are performing a complete installation, one of the following screens is displayed:
 - Database Setup for Oracle: enter the Oracle JDBC driver file and click Next. Or, you can search for the JDBC driver file using the Select File button, select the file, and then click Next.
 - Database Setup for SQLServer: enter the SQLServer JDBC driver file. You can search for the JDBC driver file using the Select File button, and then select the file. Click Next.
 - Database Setup for DB2: specify whether you want to enable multibyte support by clicking Yes or No. Enter the DB2 JDBC driver file and the DB2 License file. To search for the JDBC driver file or the DB2 License file, click the corresponding Select File button to navigate to the file location, and then select the file. Click Next.

The following series of screens ask for and then confirm database account information for four multischema data tables: Metadata, Statistics, System Configuration, and Transaction/Master.

- b. Enter database account information for your Metadata schema and click Next:
 - Database user name
 - Database password
 - Confirm database password
 - Database catalog name
 - Database host name (or IP address)
 - Database port
 - (Optional) JDBC driver file. If you are performing a complete installation, you already specified the JDBC driver file in Step A. However, if you are performing an installation in upgrade mode, enter a JDBC driver file or use the Select File button to navigate to the JDBC file location and select the file.

Note: If you want to create an alternate DB_SCHEMA_OWNER, consult your database administrator, as this is considered an expert installation scenario and can be performed only through the silent installation. For more information, see “Creating a Silent Installation File - Windows” on page 54.

- c. After you enter the Metadata Schema information and click Next, the Confirm Database Information screen displays the database account information you entered on the previous screen. This screen is read-only. If the information is correct, click Next. If any information needs to be changed, click Back to return to the previous screen and make the changes.
- d. (Optional) If you are performing the installation in upgrade mode, skip to Step 14. If you are performing a complete installation, the installation guides you through entering database information for the remaining three database schemas: Statistics, System Configuration, and Transaction/Master. When finished, click Next.

The database information you entered for Multischema Support tables is saved in <INSTALL_DIR>\multischema.xml. A sample of this file is shown in “Running a Silent Installation in Multischema Mode” on page 59. Skip to Step 14.

11. If you selected Oracle or SQLServer as your database, skip to Step 12. If you selected DB2, select whether to enable database multibyte support: click Yes to enable; otherwise, click No. Click Next.

12. Enter the following database account information and click Next:

- Database user name
- Database password
- Confirm database password
- Database catalog name
- Database host name (or IP address)
- Database port
- (Oracle and MSSQL Server) Absolute path and file name for the JDBC driver file.
- (DB2 only) Absolute paths and file names for the JDBC driver file and the DB2 license file.

For DB2, use the Type-4 JDBC driver. This type of driver converts JDBC calls into the network protocol used directly by DB2, allowing a direct call from the application to the DB2 server.

13. After you enter the database information and click Next, the Confirm Database Information screen displays the database account information you entered on the previous screen. This screen is read-only. If the information is correct, click Next. If any information needs to be changed, click Back to return to the previous screen and make the changes.
14. The Documentation Access screen is displayed. Select Online (default) or Local for the type of Documentation Library you want to enable, and click Next.
15. The Select JDK screen is displayed. Indicate whether the JDK specified earlier is 32-bit or 64-bit. The default is 64-Bit. Click Next to continue.
16. The Checklist for Installation Process screen is displayed. The screen displays a read-only checklist of activities that the installation program will perform. The following tasks and prompt are shown:
 - Verification of Sufficient Disk Space
 - Verify the selected JDK is supported
 - Perform Installation of Foundation Components

- Backup install files
- The installation location is also shown on the screen. This is the directory you chose earlier, followed by the subdirectory name "Foundation". The Windows GUI installer installs Sterling Selling and Fulfillment Foundation in the Foundation subdirectory, unlike the silent installation, which installs the product directly into the folder you specify.

Click Next.

17. On the Installation Progress screen, click Install to proceed with the installation. If you want to see additional information about the progress of the installation, click Show Details, then click Install. This information will also be available after installation in the <INSTALL_DIR>\PreInstallSI.log file.
18. When the installation is complete, the message *Finished* is displayed. Click OK to close the message box. The Installation Progress screen displays the status Complete as its heading and the message *BUILD SUCCESSFUL* in the **Output** box.

Note: You can delete the installed_data directory if it remains after installation. This is not needed after installing the product.

19. See "Post Installation Tasks for Windows" on page 62 for configuration procedures that you must complete prior to running Sterling Selling and Fulfillment Foundation.

Creating a Silent Installation File - Windows

About this task

If you prefer not to run the installer in the interactive install mode, you can create a text file that contains all of the information the installer needs and run the install from a command line in silent mode instead.

Procedure

Create the silent installation text file using a text editor. Follow these guidelines:

- The file must contain the entries marked as required in the following table. See "Sample Silent Installation File - Windows" on page 58 for an example of a silent install file.
- You can create a silent upgrade text file using the parameters in the following table. See "Sample Silent Upgrade File - Windows" on page 63 for an example of a silent upgrade file.
- On Windows: Use UNIX slashes ("/") when specifying paths in the file. For example, to specify the installation directory to be created, you might enter something similar to one of the following sample paths:
 - C:/Foundation
 - or
 - D:/Applications/Myinstall
- When you are finished creating the file, record its name and location. You will use this information later to start the installation.
- If you are going to do a silent installation or upgrade in multiple schema mode, you must also create a multischema.xml file.

Windows Silent Install File Parameters

Example Entry	Description
DB_VENDOR=<db_vendor>	(Required) The database vendor to use (Oracle, DB2, MSSQL). Note: In a multischema installation, you must deploy the same database vendor and version across deployments.
MSSQL2005=<TRUE/FALSE>	If you are running on MSSQL 2005 or later, this attribute must be set to TRUE. If you are not running on MSSQL, do not include this attribute. Note: This attribute is case-sensitive.
DB_USER=<db_user_name>	(Required) Database login ID with which to connect. In a multischema deployment, this must be the username for the Metadata schema.
DB_DATA=<db_dat_catalog>	(Required) Database name to connect with. (For more information, see “Key Terms for Installation” on page 49.)
DB_PASS=<db_password>	(Required) Database password with which to connect. In a multischema deployment, this must be the password for the Metadata schema.
DB_HOST=<db_host>	(Required) Host for database (for example, server or IP address).
DB_PORT=<db_listener_port>	(Required) Database listener port to which to connect.
DB_DRIVERS=<absolute path to driver jar>	(Required for Oracle, MSSQL, and DB2. Full path to the JDBC driver file(s): <ul style="list-style-type: none"> • Oracle and MSSQL Server require one file. • DB2 requires two files, the license file and the driver file. When specifying more than one file, use semicolons (;) for separators as necessary. Examples: <ul style="list-style-type: none"> • <JDBC_driver_dir>/jdbc.jar • <JDBC_driver_dir>/db2_1_jdbc.jar;<JDBC_driver_dir>/db2_2_jdbc.jar
DB_DRIVERS_VERSION=<db_driver_version>	(Required) Free form version string for JDBC driver. This is informational only.
DB_SCHEMA_OWNER	(Required for multischema mode) Default schema/schema-owner for the provided login ID. If you wish to change this value to an alternate schema, consult your database administrator, as this is considered an expert installation scenario and can be performed only through the silent installation.
LOAD_FACTORY_SETUP=true	Indicates whether you want to load factory setup defaults during installation (true) or manually after installation (false). If you are performing an installation in upgrade mode, set this property to false. For information about manually loading the factory defaults, see “Loading the Sterling Selling and Fulfillment Foundation Database Factory Defaults After Installation” on page 119.

Example Entry	Description
NO_DBVERIFY	Valid values are true or false. When set to true during installation and installservice, dbverify will not be run. This means that Sterling Selling and Fulfillment Foundation will not generate DDL to make the database like the XML entity repository. If you are performing an installation in upgrade mode, set this property to true.
REINIT_DB=true	Valid values are true or false. By default, the value is set to true. If the value is set to false, the Sterling Selling and Fulfillment Foundation installation will complete successfully, but no database operation will be performed as part of the installation process. If you are performing an installation in upgrade mode, set this value to false.
multischema.applyddl	Valid values are true/false. If set to true (default), enables the DBVerify script to generate and apply database DDLs automatically. If set to false, allows DBVerify script to generate the DDLs but does not apply them. This property is set to false in the GUI installer, by default. If you are using the GUI Installer and do not want to apply DDLs, ensure that this property to set to false in the sandbox.cfg file. If this property is set to true or is absent in the sandbox.cfg file, the DBVerify script generates the DDLs and applies them. Otherwise, the scripts are generated and not applied.
multischema.enabled=<true/false>	If true, this attribute indicates that this is a multischema installation. The installation then looks for a customer-created multischema.xml file, which specifies database information for the Configuration, Metadata, Transaction, and Statistics schemas. Note: This attribute is case-sensitive. See “Running a Silent Installation in Multischema Mode” on page 59 for more information.
multischema.version=<version_number>	(Required) This attribute indicates which version is being installed. For the Sterling Selling and Fulfillment Foundation Release 9.1, you must enter 9.1. Note: This attribute is case-sensitive.
multischema.file=<filename>.xml	(Required if you enable multischema.) This attribute indicates the name of the user-defined XML file that contains multischema database information. Note: This attribute is case-sensitive.
STERLING_FOUNDATION_PRODUCT_LABEL=smcfs	(Required) Specifies the product label.
STERLING_FOUNDATION_PRODUCT_VERSION=9.1	(Required) Specifies the product version you are installing.
ACCEPT_LICENSE=Y	Because the silent installer does not bring up an explicit license dialog, please specify your acceptance of the licensing terms by including this property in your silent install file. Note: To review the license file prior to installation, browse to the ProductFiles folder on your product CD and open Readme.htm.

Example Entry	Description
ACTIVE_DOC_URL= <ONLINE/LOCAL>	<p>When set to ONLINE (default), the URL set for ONLINE_DOC_URL is used.</p> <p>When set to LOCAL, the URL set for LOCAL_DOC_URL is used.</p> <p>The ONLINE_DOC_URL or the LOCAL_DOC_URL is required when ACTIVE_DOC_URL is set to ONLINE or LOCAL, respectively.</p> <p>If you want to switch between the Online and Local Documentation Libraries after installation, refer to the <i>Sterling Selling and Fulfillment Foundation: Properties Guide</i> for more information about these properties.</p>
ONLINE_DOC_URL=<url>	The value of this property is the URL for the Web-based Online Documentation Library that is in .js and PDF format: http://help.sterlingcommerce.com/SFSF91/index.jsp
LOCAL_DOC_URL=<url>	The value of this property is the URL for the Local Documentation Library that is in HTML format: /smcfsdocs/yfscommon/online_help/en_US/wwhelp/wwhimpl/js/html/wwhelp.htm
JAVADOC_PRODUCT_LABEL=Sterling Selling and Fulfillment Foundation 9.1	(Required) Specifies the Javadocs to be installed.
JDK64BIT=<true/false>	<p>Specifies whether you are using a 32-bit JDK or a 64-bit JDK.</p> <p>Default=true</p>
ADDITIONAL_ANT_JAVA_TASK_ARGS= -Xms1024m -Xmx1408m -XX:MaxPermSize=512m ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms256m -J-Xmx1408m	<p>See the <i>Sterling Selling and Fulfillment Foundation: System Requirements Guide</i> for memory parameter values based on your operating system.</p> <p>These parameter values are written to the sandbox.cfg file during installation. After installation, you can tune them if you are seeing Out-of-Memory errors.</p> <p>Note: The ADDITIONAL_ANT_JAVA_TASK_ARGS property must not be set for IBM and JRockit JDK.</p> <p>See the <i>Sterling Selling and Fulfillment Foundation: Properties Guide</i> for more information about these parameters.</p>
ENTITY_GEN_LOGLEVEL=<VERBOSE/INFO>	<p>Specify VERBOSE to enable logging of verbose messages during entity class generation.</p> <p>Default=INFO</p>
GENERATE_ALL_DBCLASSES=<true/false>	<p>Specify true to generate all DBClasses.</p> <p>Default=false</p>
SUPPORT_MULTIBYTE=<Y/N>	<p>If you are installing on DB2 and need to localize your database using a multibyte character set, set this flag to Y. This ensures that the database column sizes are large enough to handle the multibyte characters correctly.</p> <p>Default is N.</p>
ORACLE_JDBC_URL=<url>	Required only if Oracle RAC is being used as the database. You must mention the complete URL without any blank space.

Sample Silent Installation File - Windows

Following is a sample silent installation file:

```
INSTALL_DIR=C:/<INSTALL_DIR>

DB_VENDOR=Oracle

DB_USER=joe_smith

DB_DATA=110n

DB_PASS=joes_password

DB_HOST=00.00.00.00

DB_PORT=1521

DB_DRIVERS=C:/Oracle_Drivers/ojdbc6.jar

DB_DRIVERS_VERSION=11

DB_SCHEMA_OWNER=DB0

multischema.applyddl=false

multischema.enabled=true

multischema.version=9.1

multischema.file=multischema.xml

STERLING_FOUNDATION_PRODUCT_LABEL=smcfs

STERLING_FOUNDATION_PRODUCT_VERSION=9.1

ACCEPT_LICENSE=Y

ACTIVE_DOC_URL=ONLINE

ONLINE_DOC_URL=http://help.sterlingcommerce.com/SFSF91/index.jsp

LOCAL_DOC_URL=URL=/smcfsdocs/yfsccommon/online_help/en_US/wwhelp/wwhimpl/js/html/wwhelp.htm

JAVADOC_PRODUCT_LABEL=Sterling Selling and Fulfillment Foundation 9.1

JDK64BIT=true

ADDITIONAL_ANT_JAVA_TASK_ARGS=-Xms1024m -Xmx1408m -XX:MaxPermSize=512m

ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms256m -J-Xmx1408m

GENERATE_ALL_DBCLASSES=false
```

ENTITY_GEN_LOGLEVEL=INFO

Running a Silent Installation on Windows

About this task

Use the following instructions to install in a Windows environment from a command line, using the silent installation text file that you have already created. See “Creating a Silent Installation File - Windows” on page 54. You can also use the following instructions to upgrade in a Windows environment from a command line, using a silent upgrade text file that you have already created. See “Sample Silent Upgrade File - Windows” on page 63.

To install Sterling Selling and Fulfillment Foundation, refer to your preinstallation checklist and follow the steps below:

Procedure

1. At a command prompt, type one of the following commands, which include paths to the JDK, the SMCFS_9.1.jar file, and the silent installation text file:

```
<JAVA_HOME>\bin\java -jar SMCFS_9.1.jar -f install.silent
```

Or, if you do not want the JDK copied into the application's local directories, but prefer to use a JDK that is in a different location, enter:

```
<JAVA_HOME>/bin/java -jar SMCFS_9.1.jar -f install.silent -nocopyjvm
```

After the installation process begins, you can follow the progress of your installation through the <INSTALL_DIR>\InstallSI.log file.

The installation displays the message *Finished* when done.

Note: You can delete the installed_data directory if it remains after installation. This is not needed after installing the product.

2. See “Post Installation Tasks for Windows” on page 62 for configuration procedures that you must complete prior to running Sterling Selling and Fulfillment Foundation, as well as optional tasks, and post installation tips.

Running a Silent Installation in Multischema Mode

If you are planning to run a silent installation or upgrade and plan to use multischema mode, you must create a multischema.xml file for the installer to use.

The multischema.xml file is invoked by the installation process if multischema is enabled in the silent installation file. The file must specify the database account information for each of the four allowable four data tables that you use: Metadata, Statistics, System Configuration, and Transaction/Master.

It also shows parameters for specifying multiple passwords and their effective dates so that you can predefine passwords for a given pool months in advance. These passwords will change automatically without a server restart.

In a multischema installation, you must deploy the same database vendor and version across deployments.

Multischema Parameters

You can change only the following parameters:

Multischema.xml File Properties

Parameter

Definition

<jdbc_url>

Specify the URL to connect to the database.

- If using Oracle, set to:

```
jdbc:oracle:thin:@<DatabaseServerHostnam  
e/IPAddress>:<TNSListenerPortNumber>:<Da  
tabaseSID>
```

- If using MSSQL Server, set to:

```
jdbc:sqlserver://<Database  
ServerHostname>:<PortNumber>;DatabaseNam  
e=<Database name>
```

- If using DB2, set to:

```
jdbc:db2://<Database  
ServerHostname>:<Port Number>/<Database  
name>.<db_user> Database user name
```

<db_user>

Specify the user name associated with the database.

<db_password>

Specify the password associated with the database.

<db_driver_class>

Specify the class name of your database driver as follows.

- If using Oracle, set to:

```
oracle.jdbc.driver.OracleDriver
```

- If using MSSQL Server, set to:

```
com.microsoft.sqlserver.jdbc.SQLServerDriver
```

- If using DB2, set to:

```
com.ibm.db2.jcc.DB2Driver
```

<db_schema>

Specify the schema name associated with the database if it is different from the <db_user> name you entered.

Note: This parameter is case-sensitive and you must specify it in UPPERCASE.

<datasource>

Specify the datasource that you created on the application server for use with this schema.

<password.1>

Specify a database password for the effectivity date shown in the <effective.1> parameter shown in this table.

<password.2>

Specify a database password for the effectivity date shown in the <effective.2> parameter shown in this table.

<effective.1>

Specify an effective date for the <password.1> parameter shown in this table.

Example: 2009-07-16T15:00:00

<effective.2>

Specify an effective date for the <password.2> parameter shown in this table.

Example: 2009-08-25T12:00:00

Sample Multischema.xml File

Unlike the silent installation, the GUI and text-based installations automatically create the multischema.xml file for you. The following example is for a DB2 installation.

See the *Sterling Selling and Fulfillment Foundation: Multi-Tenant Enterprise Guide* for more information about the various schema types mentioned here, and about colonies and other multischema information.

```
<?xml version="1.0" encoding="UTF-8"?>
<colonyconfig>
  <colonies>
    <colony name="DEFAULT" pkprefix="20" version="9.1">
      <schema poolid="DEFAULT_METADATA" tabletype="METADATA"/>
      <schema poolid="DEFAULT_STATISTICS_91" tabletype="STATISTICS"/>
      <schema poolid="DEFAULT_CONFIGURATION_91" tabletype="CONFIGURATION"/>
      <schema poolid="DEFAULT_TRANSACTION_91" tabletype="TRANSACTION"/>
      <schema poolid="DEFAULT_TRANSACTION_91" tabletype="MASTER"/>
    </colony>
  </colonies>
  <pools>
    <pool id="DEFAULT_METADATA">
      <jdbc>
        <param name="url" value="jdbc:db2://00.00.00.00:50000/devdb2"/>
        <param name="user" value="metadata_user"/>
        <param name="password" value="metadata_user"/>
        <param name="driver" value="com.ibm.db2.jcc.DB2Driver"/>
        <param name="schema" value="METADATA_USER"/>
        <param name="datasource" value="metadata_datasource_name"/>
        <param name="password.1" value="password_1"/>
        <param name="password.2" value="password_2"/>
        <param name="effective.1" value="2009-07-16T15:00:00"/>
        <param name="effective.2" value="2009-08-16T15:00:00"/>
      </jdbc>
    </pool>
    <pool id="DEFAULT_STATISTICS_91">
      <jdbc>
        <param name="url" value="jdbc:db2://00.00.00.00:50000/devdb2"/>
        <param name="user" value="statistics_user"/>
        <param name="password" value="statistics_user"/>
        <param name="driver" value="com.ibm.db2.jcc.DB2Driver"/>
        <param name="schema" value="STATISTICS_USER"/>
        <param name="datasource" value="statistics_datasource_name"/>
        <param name="password.1" value="password_1"/>
        <param name="password.2" value="password_2"/>
        <param name="effective.1" value="2011-07-16T15:00:00"/>
        <param name="effective.2" value="2011-08-16T15:00:00"/>
      </jdbc>
    </pool>
    <pool id="DEFAULT_CONFIGURATION_91">
      <jdbc>
        <param name="url" value="jdbc:db2://00.00.00.00:50000/devdb2"/>
        <param name="user" value="configuration_user"/>
        <param name="password" value="configuration_user"/>
        <param name="driver" value="com.ibm.db2.jcc.DB2Driver"/>
        <param name="schema" value="CONFIGURATION_USER"/>
        <param name="datasource" value="configuration_datasource_name"/>
        <param name="password.1" value="password_1"/>
      </jdbc>
    </pool>
  </pools>
</colonyconfig>
```

```

        <param name="password.2" value="password_2"/>
        <param name="effective.1" value="2011-07-16T15:00:00"/>
        <param name="effective.2" value="2011-08-16T15:00:00"/>
    </jdbc>
</pool>
<pool id="DEFAULT_TRANSACTION_91">
    <jdbc>
        <param name="url" value="jdbc:db2://00.00.00.00:50000/devdb2"/>
        <param name="user" value="transaction_user"/>
        <param name="password" value="transaction_user"/>
        <param name="driver" value="com.ibm.db2.jcc.DB2Driver"/>
        <param name="schema" value="TRANSACTION_USER"/>
        <param name="datasource" value="transaction_datasource_name"/>
        <param name="password.1" value="password_1"/>
        <param name="password.2" value="password_2"/>
        <param name="effective.1" value="2011-07-16T15:00:00"/>
        <param name="effective.2" value="2011-08-16T15:00:00"/>
    </jdbc>
</pool>
</pools>
</colonyconfig>

```

Post Installation Tasks for Windows

Security

- See “Web Security Planning-Post installation Recommendations” on page 11 for information about the security measures that are recommended for your consideration.

Database

- Views must be created manually. Instructions to create views vary depending on what database is used. All database view related scripts are located at <INSTALL_DIR>\database\<db_type>\scripts. For more information, refer to the section about configuring your database type (MSSQL Server, Oracle, or DB2) for production in Chapter 6, “Installing and Configuring Database Tier Software on Windows,” on page 37 and Chapter 14, “Configuring Utilities,” on page 111.

Changing to Multischema Mode after Installation

- If you did not install or upgrade in multischema mode but you want to create the appropriate tables for multischema mode and update these tables at a later time, you can:
 - Set the following properties in the sandbox.cfg file:


```
multischema.enabled=true
multischema.version=9.1
```
 - Run the dbverify script on multischema colonies, as described in “Verifying the Database Schema” on page 114.

If you want to create or add colonies to your deployment, see the *Sterling Selling and Fulfillment Foundation: Multi-Tenant Enterprise Guide* for instructions.

Upgrade Information for Windows

If installing in upgrade mode, use the topics in this section for reference.

- “Upgrade Installation Considerations” on page 63
- “Upgrading in a Windows Environment-Important Database Considerations” on page 63
- “Running a Silent Upgrade on Windows” on page 63

- “Sample Silent Upgrade File - Windows”
- “Post Upgrade Tasks for Windows” on page 65

Upgrade Installation Considerations

Silent Installation

You can run an upgrade using the silent installation process. For information about running the silent installation in upgrade mode, refer to “Running a Silent Installation on Windows” on page 59. For information about creating the silent upgrade file, refer to “Sample Silent Upgrade File - Windows.”

Multischema Environments

- In a multischema installation, you must deploy the same database vendor and version across deployments.
- An installation in upgrade mode for multischema is supported only if you are upgrading from a previous release of multischema.

Upgrading in a Windows Environment-Important Database Considerations

If you are reinstalling Sterling Selling and Fulfillment Foundation, be aware that data in your existing database will be deleted. To prevent this, either back up the existing database or save it under a different name. For more information about backing up and restoring data, see “Generating Data Backup and Restore Scripts” on page 37.

After creating and configuring your database, recycle the database. Then stop and restart Sterling Selling and Fulfillment Foundation to apply the changes.

Running a Silent Upgrade on Windows

Procedure

1. Ensure that you have set `LOAD_FACTORY_SETUP=false`, `REINIT_DB=false`, and `NO_DBVERIFY=true` in your silent upgrade text file.

2. Enter the following:

```
<JAVA_HOME>/bin/java -jar SMCFS_9.1.jar -f install.silent
```

After the installation process begins, you can follow the progress of your installation through the `<INSTALL_DIR>\InstallSI.log` file.

The installation displays the message *Installation has completed successfully* when done.

3. See “Post Upgrade Tasks for Windows” on page 65 for configuration procedures that you must complete prior to running Sterling Selling and Fulfillment Foundation, as well as optional tasks, and post upgrade tips.

Sample Silent Upgrade File - Windows

Following is a sample silent upgrade file:

```
INSTALL_DIR=C: /<INSTALL_DIR>
```

```
LOAD_FACTORY_SETUP=false
```

```
REINIT_DB=false
```

```
NO_DBVERIFY=true
DB_VENDOR=Oracle
DB_USER=joe_smith
DB_DATA=110n
DB_PASS=joes_password
DB_HOST=00.00.00.00
DB_PORT=1221
DB_DRIVERS=C:/Oracle_Drivers/ojdbc6.jar
DB_DRIVERS_VERSION=11
DB_SCHEMA_OWNER=DB0
multischema.applyddl=false
multischema.enabled=true
multischema.version=9.1
multischema.file=multischema.xml
STERLING_FOUNDATION_PRODUCT_LABEL=smcfs
STERLING_FOUNDATION_PRODUCT_VERSION=9.1
ACCEPT_LICENSE=Y
ACTIVE_DOC_URL=ONLINE
ONLINE_DOC_URL=http://help.sterlingcommerce.com/SFSF91/index.jsp
LOCAL_DOC_URL=URL=/smcfsdocs/yfscommon/online_help/en_US/wwhelp/wwhimpl/js/
html/wwhelp.htm
JAVADOC_PRODUCT_LABEL= Sterling Selling and Fulfillment Foundation 9.1
JDK64BIT=true
ADDITIONAL_ANT_JAVA_TASK_ARGS=-Xms1024m -Xmx1408m -XX:MaxPermSize=512m
ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms256m -J-Xmx1408m
GENERATE_ALL_DBCLASSES=false
ENTITY_GEN_LOGLEVEL=INFO
```


Post Upgrade Tasks for Windows

If you have installed in upgrade mode and you want to create the appropriate tables for multischema mode and update these tables at a later time, you can:

- Set the following properties in the `sandbox.cfg` file:

```
multischema.enabled=true  
multischema.version=9.1
```

- Run the `dbverify` script on multischema colonies, as described in “Verifying the Database Schema” on page 114.

If you want to create or add colonies to your deployment, see the *Sterling Selling and Fulfillment Foundation: Multi-Tenant Enterprise Guide* for instructions.

Chapter 8. Installing Sterling Selling and Fulfillment Foundation in UNIX and Linux Environments

Use the topics in this section to prepare for installation, run the installation, and as a reference for postinstallation tasks.

- “UNIX/Linux Preinstallation Tasks”
- “Running an Installation in a UNIX or Linux Environment” on page 73
- “Post Installation Tasks for UNIX/Linux” on page 88
- “Upgrade Information for UNIX/Linux” on page 89

UNIX/Linux Preinstallation Tasks

Complete the tasks in this section prior to installing Sterling Selling and Fulfillment Foundation

- “Before You Begin Installation in a UNIX or Linux Environment”
- “UNIX/Linux Preinstallation Checklist” on page 68
- “Setting Up X Windows for a GUI Installation” on page 70
- “Including Memory Parameter Settings in the Installation Command” on page 70
- “About the JDK for UNIX/Linux” on page 71
- “Before AIX Installations Only” on page 71
- “Before Linux Installations Only” on page 72
- “Before RedHat Enterprise Linux Installations Only” on page 72
- “IBM and JRockit JDK Only” on page 72
- “Types of Installation Available on UNIX/Linux” on page 72
- “Differences Between Single and Multiple Schema Installations” on page 73

Before You Begin Installation in a UNIX or Linux Environment

- You can install Sterling Selling and Fulfillment Foundation on a UNIX or Linux system locally in an X Windows environment, or remotely, in a text-based console environment.
- If you are upgrading from a prior release, see the *Sterling Selling and Fulfillment Foundation: Upgrade Guide* **before** continuing with the installation of Sterling Selling and Fulfillment Foundation.
- An installation in upgrade mode for multischema is supported only if you are upgrading from a previous release of multischema. Before installing Sterling Selling and Fulfillment Foundation, ensure that you have installed the appropriate software listed in the *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.
- Throughout this document, <INSTALL_DIR> refers to the directory where you install Sterling Selling and Fulfillment Foundation. For example, <INSTALL_DIR>=/Supply_Chain_Apps.
- On Linux, do not use any soft/symbolic links in the path to the jar file. Make sure that you specify the full path to the jar file.
- You **must** be in the directory where SCInstallWizard.jar resides when issuing the installation command.

- During the installation, various messages are displayed, including some warning messages. These warning messages require no action on your part and are included so that helpful data is recorded in the log file.
- Create a UNIX Account
In a UNIX or Linux environment, you must create a UNIX administrative account on the host server for each installation of Sterling Selling and Fulfillment Foundation. For example, if you want to create a test environment and a production environment, you need to create two UNIX accounts on the host server, one for the test and one for the production environment. For more information about creating UNIX accounts, see your operating system documentation.
- A root user cannot install Sterling Selling and Fulfillment Foundation.
- The installation instructions assume that you received an installation CD for Sterling Selling and Fulfillment Foundation. If you downloaded Sterling Selling and Fulfillment Foundation or a Service Pack (SP) from the Electronic Software Distribution (ESD) Portal, extract the downloaded file to an empty directory. Do not change the directory structure of the newly extracted files. The directory containing the extracted files is an electronic image of an installation CD. Use this directory wherever there is a reference to the installation CD in the following instructions. Ignore any instructions to place the installation CD in a drive.
- If using silent installation, record the path to your silent installation file.
- The GUI installation creates the following installation directory structure: <INSTALL_DIR>/Foundation. The Sterling Selling and Fulfillment Foundation files and subdirectories are installed under <INSTALL_DIR>/Foundation.
- The silent installation process creates an <INSTALL_DIR>, and the Sterling Selling and Fulfillment Foundation files and subdirectories are installed directly under it.
- About creating database views: Views must be created manually. Instructions to create views vary depending on which database is used. All database view related scripts are located at <INSTALL_DIR>/database/<db_type>/scripts. For more information, refer to the section about configuring your database in Chapter 5, "Installing and Configuring Database Tier Software on UNIX or Linux," on page 27 and Chapter 14, "Configuring Utilities," on page 111.
- When creating a name, such as an account name, permissions name, profile name, or database name, follow these conventions:
 - Use any valid alphanumeric characters and _ (underscore).
 - Do not use spaces or apostrophes.
- Copy the SCIInstallWizard.jar and SMCFS_9.1.jar files to your home directory or base directory and change to that directory.

UNIX/Linux Preinstallation Checklist

The preinstallation checklist contains the items you need to gather, and tasks you need to complete prior to installing Sterling Selling and Fulfillment Foundation.

You may want to make a copy of the following checklist and use it to record the information you collect. The sections following the checklist include detailed explanations of some of the items on the checklist.

Table 8. Preinstallation Checklist for UNIX or Linux

Step	Description	Your Notes
1	Verify that your system meets the hardware and software requirements specified for this release. For more information, see the <i>Sterling Selling and Fulfillment Foundation: System Requirements Guide</i> .	
2	If you are using a non-English environment, confirm that you are using the appropriate character set.	
3	Determine and record information about the JDK. <ul style="list-style-type: none"> • Version of the JDK • Absolute path to the JDK 	
4	Decide whether you want the installer to create a copy of the JDK beneath the install directory.	
4	Set umask to 002.	
5	Determine and record information about your Oracle, MSSQL Server, or DB2 database server; determine and record information about your database server. Be aware that this information may be case sensitive. <ul style="list-style-type: none"> • Database user name and associated password • Database catalog name • Database host name (or IP address) • Database host port number • (Oracle and MSSQL Server. Absolute path and file name for one JDBC driver. • (DB2 only) Absolute paths and file names for two JDBC files: the driver file and the license file. • If you are planning a multischema deployment, determine and record this database information for four separate schemas: Metadata, Statistics, System Configuration, and Transaction/Master Data. 	
6	Decide if you are going to manually or automatically apply database definition language (DDL) statements (schema) to the database.	

Table 8. Preinstallation Checklist for UNIX or Linux (continued)

Step	Description	Your Notes
7	<p>Determine and record the directory in which you plan to install Sterling Selling and Fulfillment Foundation.</p> <ul style="list-style-type: none"> • The file system must have adequate free disk space. • The name of the directory is case-sensitive. • If you use the silent installation method, you cannot install into a pre-existing directory. The silent installation process will fail if a pre-existing directory is specified. See “Types of Installation Available on UNIX/Linux” on page 72 for more information about using the silent installation method. • If you use the GUI or text-based installation methods, you can install into either a pre-existing directory or a new directory to be created by the installation process. 	
8	<p>If you are running a silent installation, create a silent installation text file. For more information, refer to “Types of Installation Available on UNIX/Linux” on page 72.</p>	

Setting Up X Windows for a GUI Installation

About this task

Using a GUI-supported operating system, perform the following actions:

Procedure

1. Use a connectivity client to connect to your UNIX/Linux account.
2. Set the display to use your X server as a client using the following command:

```
export DISPLAY=<server>:0.0
```

 (or the appropriate Display identifier)

In the above command, :0.0 can be a different value, such as :8.0.

Including Memory Parameter Settings in the Installation Command

When invoking the install script, you will need to add memory parameters. During installation, these parameter values are written to the sandbox.cfg property file, where you can tune them after installation.

The example parameter values that follow are recommended for the stack shown; however, your memory requirements may differ. See the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* for your specific supported stack requirements.

Example Memory Parameter Settings for Installation

The `-XX:MaxPermSize` parameter indicates the maximum amount of memory. In the command, the memory parameter values **512m**, **256m**, and **1408m** were tested on the following stack:

Operating System: Windows VISTA and RHL 5.3 64-bit

Data Base: Oracle 11g

JDK: SUN JDK 1.60_19 and above

Application Server: WebLogic 10.3

About the JDK for UNIX/Linux

- To determine which JDK version and patches you need, see the System Requirements.
- Edit the `JAVA_HOME` environment variable to point to the directory where you installed the JDK.
- After you install the JDK, record the absolute path to its location on your system. You must supply the absolute path when you install IBM Sterling Selling and Fulfillment Foundation.
- During the installation, you will be prompted whether to have the installer create a copy of the JDK under your install folder. If you do not want the JDK copied, download version 2.7.1 of xerces and xalan from Apache prior to running the installation. Copy the following files into the `jre/lib/endorsed` directory of your JDK:
 - `serializer.jar`
 - `xalan.jar`
 - `xercesImpl.jar`
 - `xml-apis.jar`

Before AIX Installations Only

About this task

During the installation process, you specify the name of the directory to be created for Sterling Selling and Fulfillment Foundation. The installation process creates the directory and uses it as the Home folder for the Sterling Selling and Fulfillment Foundation files and subdirectories. Throughout this book, this directory is referred to as `<INSTALL_DIR>`.

To ensure that `<INSTALL_DIR>` has the necessary permissions, AIX® users must run the following command on the parent directory of `<INSTALL_DIR>` before installation:

```
chmod -R a-s <absolute path>/install_dir_parent
```

Here, `install_dir_parent` is the directory in which `<INSTALL_DIR>` will be created.

For example, to specify `AIX_1/applications/test1/my_install` as your installation directory, you could run the command from the `AIX_1/applications` directory (directly above the `test1` directory):

```
chmod -R a-s test1
```

or from another location on the file system:

```
chmod -R a-s /AIX_1/applications/test1
```

This ensures that when the `my_install` directory is created during installation, it inherits the correct permissions from `test1`.

Before Linux Installations Only

About this task

This task applies only to application systems that use a Linux operating system.

Procedure

If the base locale for the system is English, make the following system change: set the `LANG` environment variable to `en_US`.

Before RedHat Enterprise Linux Installations Only

About this task

Perform this procedure for application systems that use the RedHat Enterprise Linux operating system only.

Make the following system changes:

Procedure

1. If the base locale for the system is English, edit the `/etc/sysconfig/i18n` file by changing the `SUPPORTED` variable from `en_US.utf8` to `en_US`. You can also allow multiple support using the following format:

```
en_US.utf8:en_US
```

Save and close the `/etc/sysconfig/i18n` file.

2. Edit the `/etc/security/limits.conf` file by adding the following lines:

```
* hard  nofile  8196
* soft  nofile  2048
* hard  memlock 3000000
* soft  memlock 4000000
* hard  nproc  16000
* soft  nproc  16000
* hard  stack  512000
* soft  stack  512000
```

This updates the system ulimits.

Save and close the `/etc/security/limits.conf` file.

3. Reboot the system.

IBM and JRockit JDK Only

The `-Djavataskargs` property must not be set for IBM and JRockit JDK.

Types of Installation Available on UNIX/Linux Interactive Installation using X Windows GUI

You can run the installation from a script, `SCIInstallWizard.jar`, that starts an interactive install wizard program. It enables you to step through a GUI wizard to enter the necessary information about your database, JDK driver, selected install folder, and so on. After you've entered this information, you start the installation

by clicking the Install button. A status screen remains open, and displays messages about the progress of the installation. Once Sterling Selling and Fulfillment Foundation is installed, the status screen displays an install complete message. See “Running the Interactive GUI Installation in UNIX or Linux” on page 74.

The GUI installer installs Sterling Selling and Fulfillment Foundation in the Foundation subdirectory, unlike the silent installation, which installs the product directly into the folder you specify.

Text-Based Interactive Installation

You can run the installation from a script, `SCIIInstallWizard.jar`, that starts an interactive text-based install wizard program. It enables you to step through a wizard to enter the necessary information about your database, JDK driver, selected install folder, and so on. After you've entered this information, you start the installation by clicking the Install button. Once Sterling Selling and Fulfillment Foundation is installed, an "install complete" message is displayed. See “Running the Text-Based Interactive Installation Program in UNIX or Linux” on page 77.

The GUI installer installs Sterling Selling and Fulfillment Foundation in the Foundation subdirectory, unlike the silent installation, which installs the product directly into the folder you specify.

Silent Installation

Instead of using the interactive method and being prompted to enter all the information needed by the install, you can create a silent installation text file that contains this information. You invoke the installation from the command line and reference the text file, which is then used by the installer. See “Running a Silent Installation in UNIX or Linux” on page 85.

The silent installation installs the product directly into the folder that you specify in the silent install text file.

Differences Between Single and Multiple Schema Installations

In a multischema installation, you must deploy the same database vendor and version across deployments.

Running an Installation in a UNIX or Linux Environment

Use the topics in this section to install Sterling Selling and Fulfillment Foundation

- “Running the Interactive GUI Installation in UNIX or Linux” on page 74
- “Running the Text-Based Interactive Installation Program in UNIX or Linux” on page 77
- “Creating a Silent Installation File - UNIX/Linux” on page 80
- “UNIX/Linux Silent Install File Parameters” on page 81
- “Sample Silent Installation File - UNIX/Linux” on page 84
- “Running a Silent Installation in UNIX or Linux” on page 85
- “Running a Silent Installation in Multischema Mode - UNIX/Linux” on page 85

Running the Interactive GUI Installation in UNIX or Linux

About this task

Use the following instructions to install in a UNIX or Linux environment from a command line, using a graphical user interface (GUI) in an X Windows client.

To install Sterling Selling and Fulfillment Foundation, refer to your preinstallation checklist and follow the steps below:

If you are upgrading from a previous release, stop any running instances of the previous installation.

Procedure

1. For either a new Sterling Selling and Fulfillment Foundation system, or for an upgrade from a previous release, enter the following command:

```
<JAVA_HOME>/bin/java -Djavataskargs="-XX:MaxPermSize=<value_1>"  
-Dcomtaskargs="-J-Xms<value_2> -J-Xmx<value_3>" -jar  
SCIInstallWizard.jar
```

See the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* for memory parameter values based on your operating system.

If you are doing an upgrade, at this point, refer to the *Sterling Selling and Fulfillment Foundation: Upgrade Guide* to complete the installation.

The installation dialog box appears in a GUI.

2. Click Next to start the installation program.
3. Type the full path of your JDK directory, or search for it using the **Select Folder** button.

If you want a local copy of the JDK created in your installation, click Yes. Otherwise, click No. Click Next.

4. On the Upgrade/Database Options screen, do one of the following:
 - If you are upgrading from a previous release, or if you do not want the database DDLs and Factory Setup installation done as part of this installation process, click Yes for the option *Do you want to upgrade from a previous installation or perform new installation with no Database DDLs and Factory setup?* Click Next.

Note: If you are not upgrading and you select this option, you must manually create your database tables and load factory setup after the installation process. See the section about configuring your database for production in Chapter 5, "Installing and Configuring Database Tier Software on UNIX or Linux," on page 27 for information about running view scripts after the initial installation and Chapter 14, "Configuring Utilities," on page 111 for information about manually installing the database DDLs and factory setup. However, if you are upgrading, the DDLs are applied as part of the upgrade process.

- If you are not upgrading from a previous release, or if you do want the installation process to apply the database DDLs and install factory setup, click No, and click Next.
5. Choose an installation directory for Sterling Selling and Fulfillment Foundation. Enter the directory or click Select Folder to open a window where you can navigate to the folder that you want to use as the installation directory. This directory is referred to as <INSTALL_DIR> in subsequent prompts. After selecting the folder, click Next.

If the directory does not exist, a message asking if the directory should be created is displayed. Click Yes to create the directory or No to return to the previous screen. When you enter the directory path name, it cannot contain spaces.

6. Choose the installation jar location. Click **Select File** to navigate to the folder that contains the installation jar file. If the file does not exist, an error message is displayed. Click **Next** to proceed.
7. Select the database vendor that you want to use (Oracle, MSSQL Server, or DB2) and click **Next**.
8. If you do not want to enable multischema support, click **No**, click **Next**, and skip to step 9 on page 76. However if you would like to enable multischema support, click **Yes**, click **Next**, and proceed as follows:
 - a. (Optional). If you are performing an installation in upgrade mode, skip to step b. However, if you are performing a complete installation, one of the following Database Setup screens opens, where you must specify a JDBC driver file for the database vendor you selected:
 - In the Database Setup for Oracle screen, enter the Oracle JDBC driver file and click **Next**. Or, you can search for the JDBC driver file using the **Select File** button, select the file, and then click **Next**.
 - In the Database Setup for SQLServer screen, enter the SQLServer JDBC driver file. You can search for the JDBC driver file using the **Select File** button, and then select the file. Click **Next**.
 - In the Database Setup for DB2 screen, specify whether you want to enable multibyte support by clicking **Yes** or **No**, enter the DB2 JDBC driver file, and enter the DB2 License file. To search for the JDBC driver file or the DB2 License file, click the corresponding **Select File** button to navigate to the file location, and then select the file. Click **Next**.

The following series of screens ask for and then confirm database account information for four multischema data tables: Metadata, Statistics, System Configuration, and Transaction/Master.

- b. Enter database account information for your Metadata schema and click **Next**:
 - Database user name
 - Database password
 - Confirm database password
 - Database catalog name
 - Database host name (or IP address)
 - Database port
 - (Optional) JDBC driver file. If you are performing a complete installation, you already specified the JDBC driver file in step a. However, if you are performing an installation in upgrade mode, enter a JDBC driver file or use the **Select File** button to navigate to the JDBC file location and select the file.

Note: To create an alternate DB_SCHEMA_OWNER, consult your database administrator. This is considered an expert installation scenario and must be done using the silent installation method. For more information, see “Running a Silent Installation in UNIX or Linux” on page 85.

- c. After you enter the Metadata Schema information and click **Next**, the **Confirm Database Information for Metadata Schema** screen displays the database account information you entered on the previous screen. This

screen is read-only. If the information is correct, click **Next**. If any information needs to be changed, click **Back** to return to the previous screen and make the changes.

- d. (Optional) If you are performing the installation in upgrade mode, skip to step 12. If you are performing a complete installation, the installation guides you through entering database information for the remaining three database schemas: Statistics, System Configuration, and Transaction/Master. When finished, click **Next**.

The database information you entered for multischema support tables is saved in <INSTALL_DIR>/multischema.xml. A sample of this file is shown in “Running a Silent Installation in Multischema Mode - UNIX/Linux” on page 85.

- e. Skip to step 12.

9. If you selected Oracle or SQLServer as your database, go to step 10. However, if you selected DB2, specify whether or not to enable database multibyte support. To enable database multibyte support, select **Yes**. Otherwise, select **No**. Click **Next** to continue.

10. Configure your database by entering the following information and click **Next**:

- (DB2 only) Select whether to support multibyte character sets.
- Database user name
- Database password
- Confirm database password
- Database catalog name
- Database host name (or IP address)
- Database port
- (Oracle and MSSQL Server) Absolute path and file name for one JDBC driver.
- (DB2 only) Absolute paths and file names for the JDBC driver and the DB2 License file.

For DB2, use the Type-4 JDBC driver. This type of driver converts JDBC calls into the network protocol used directly by DB2, allowing a direct call from Sterling Selling and Fulfillment Foundation to the DB2 server.

Note: It is not possible to pass multiple database drivers when running the GUI installation program in UNIX or Linux. However, DB2 requires you to provide the JDBC driver file and the DB2 license file. Extract the two available jars and place them under a single jar that you will provide as input if using GUI mode.

11. After you enter the database information and click **Next**, the **Confirm Database Information** screen displays the database account information you entered on the previous screen and the path to the install jar. This screen is read-only. If the information is correct, click **Next**. If any information needs to be changed, click **Back** to return to the previous screen and make the changes.
12. The **Documentation Access** screen opens. Select **Online** (default) or **Local** for the type of Documentation Library you want to enable. Click **Next**.
13. The **Select JDK** screen opens. Click the appropriate button to specify whether the JDK you specified earlier is 32-Bit or 64-Bit. The default is 64-Bit. Click **Next**.
14. The **Checklist for Installation Process** screen opens. The screen displays a read-only checklist of activities that the installation program will perform. The following tasks and prompt are shown:

- *Verification of Sufficient Disk Space*
- *Verify the selected JDK is supported*
- *Perform Installation of Foundation Components*
- *Backup install files*
- The *Installation location* is also shown on the screen. This is the directory you chose earlier, followed by the subdirectory name "Foundation".

Click Next.

15. On the Installation Progress screen, click Install to proceed with the installation. If you want to see detailed information about the progress of the installation, click Show Details, then click Install. This information will also be available after installation in the <INSTALL_DIR>/PreInstallSI.log file.
16. When the installation is finished, the message *Finished* is displayed. Click OK to close the message box. The Installation Progress screen displays the status Complete as its heading and the message *BUILD SUCCESSFUL* in the **Output** box.

Note: You can delete the installed_data directory if it remains after installation. This is not needed after installing the product.

17. See "Post Installation Tasks for UNIX/Linux" on page 88 for configuration procedures that you must complete prior to running Sterling Selling and Fulfillment Foundation, as well as optional tasks, and post installation tips.

Running the Text-Based Interactive Installation Program in UNIX or Linux

About this task

Use the following instructions to install in a UNIX or Linux environment from a command line, using a text-based (non-GUI) interface.

If you are upgrading from a previous release, stop any running instances of the previous installation.

Procedure

1. To begin the installation, type the absolute path to the JDK directory followed by the command: <JAVA_HOME>/bin/java -Djavataskargs="-XX:MaxPermSize=<value_1>" -Dcomtaskargs="-J-Xms<value_2>" -J-Xmx<value_3>" -jar SCIIInstallWizard.jar
2. At the "Welcome to the IBM Sterling Selling and Fulfillment Suite installer" prompt, press Enter.
3. When prompted, type the path of the JDK directory and press Enter. This <JDK_PATH> should point to the root folder of the JDK installation. For example, the java executables should be present under the <JDK_PATH>/bin folder.
4. At "Do you want a local copy of the JDK to be created in your Installation?" enter 1 for yes (or press Enter) or enter 2 for no.
5. On the Upgrade/Database Options screen, you must respond to the following question, "Do you want to upgrade from a previous installation or perform new installation with no Database DDLs and Factorysetup?" Use the following bullets as a guide for your answer:
 - If you are upgrading from a previous release, enter 1. See the *Sterling Selling and Fulfillment Foundation: Upgrade Guide* for more specific information about performing an upgrade.

- If you are not upgrading from a previous release, and you want the installation program to install the database DDLs and factory setup for you, enter 2.
 - If you are not upgrading from a previous release, and you need to do the database DDL and factory setup installation manually after this installation program runs, enter 1. You might need to choose this option if your DBA is the only one who can access or make changes to databases, but you are responsible for doing the rest of the installation, for example.
6. At the "Please choose an installation directory" prompt, type the absolute path and name of the installation directory with no spaces and press Enter. This can be either an existing or a new directory. If this is a new directory, the path name cannot contain spaces. You will be prompted, "The directory does not exist, create it?" Type Y for yes or N for no, then press Enter.
 7. At the prompt for "Please confirm install jar location. Ensure the following path correctly points to the install jar," press Enter to confirm the installation jar presented, or enter a different installation jar.
 8. At the "Please select a database vendor" prompt, choose one of the following:
 - (1) Oracle
 - (2) MS SQL 2005/2008
 - (3) DB2

Type the appropriate number and press Enter.

9. At the prompt, "Would you like to enable MultiSchema Support?", enter 1 for Yes or 2 for No. If you enter 2 for No, skip to step 10 on page 79. If you would like to enable multischema support, enter 1 for Yes, and proceed as follows:
 - a. (Optional). If you are performing an installation in upgrade mode, skip to step b. However, if you are performing a complete installation, one of the following prompts appears, where you must specify a JDBC driver file for the database vendor you selected earlier:
 - At the "Database Setup for Oracle" prompt, enter the absolute path and file name for the Oracle JDBC driver file.
 - At the "Database Setup for SQLServer" series of prompts, enter the absolute path and file name for the SQLServer JDBC driver file.
 - At the "Database Setup for DB2" series of prompts, indicate whether or not to enable multibyte support by entering 1 for Yes (enable), or 2 for No (do not enable). Press Enter. Next, enter the absolute path and file name for the DB2 JDBC driver file, and press Enter. Finally, enter the absolute path and file name for the DB2 License file and press Enter. For DB2, use the Type-4 JDBC driver. This type of driver converts JDBC calls into the network protocol used directly by DB2, allowing a direct call from the application to the DB2 server.

The following series of prompts ask for and then confirm database account information for four multischema data tables: MetaData, Statistics, System Configuration, and Transaction/Master.
 - b. Starting with Metadata, you are prompted separately for each of the following items. Enter a value and press Enter for each item.
 - Database user name
 - Database password
 - Confirm database password
 - Database catalog name
 - Database host name (or IP address)

- Database port
- (Upgrade only) JDBC driver file: If you are performing a complete installation, you already specified the JDBC driver file in step a. However, if you are performing an installation in upgrade mode, enter the absolute path and file name for the JDBC driver for your database vendor. For DB2, enter the absolute path and file name for the JDBC driver, then for the DB2 License file. For DB2, use the Type-4 JDBC driver. This type of driver converts JDBC calls into the network protocol used directly by DB2, allowing a direct call from the application to the DB2 server.

Note: To create an alternate DB_SCHEMA_OWNER, consult your database administrator. This is considered an expert installation scenario and must be done using the silent installation method. For more information, see “Creating a Silent Installation File - UNIX/Linux” on page 80.

- c. After you enter the Metadata schema information, the Confirm Database Information screen (read-only) displays the information you entered at the previous prompts and the path to the install jar. Read each item's value and press Enter to accept each one.

The installation program verifies the database connection. If a connection cannot be established, you receive an error and can re-enter the database information to make more connection attempts. If you still cannot make a connection, consult with your database administrator (DBA).

- d. (Optional) If you are performing the installation in upgrade mode, skip to step 13 on page 80. If you are performing a complete installation, the installation guides you through entering database information for the remaining three database schemas: Statistics, System Configuration, and Transaction/Master. When finished, press Enter and skip to step 13 on page 80.

Note: Once you have entered and verified the information for all four schemas, the installer verifies the database connection. If a connection cannot be established, you receive an error and can re-enter the database information to make more connection attempts. If you still cannot make a connection, consult with your database administrator (DBA).

The database information you entered for Multischema support tables is saved in <INSTALL_DIR>/multischema.xml. A sample of this file is shown in “Running a Silent Installation in Multischema Mode - UNIX/Linux” on page 85.

10. If you selected Oracle or MSSQL Server as your database, go to step 11. If you selected DB2 as your database, enter 1 for multibyte support or 2 for no multibyte support.
11. On the "Please enter your database account information" screen, you are prompted separately for each of the following items. Enter a value for the first item, then press Enter. The second item will be displayed. Enter a value and press Enter. Repeat for each item.
 - Database user name
 - Database password
 - Confirm database password
 - Database catalog name
 - Database host name (or IP address)
 - Database host port number (For DB2, use port 50000)

- (Oracle and MSSQL Server) Absolute path and file name for one JDBC driver.
 - (DB2 only) Absolute paths and file names for the JDBC driver and DB2 License file. For DB2, use the Type-4 JDBC driver. This type of driver converts JDBC calls into the network protocol used directly by DB2, allowing a direct call from the application to the DB2 server.
12. After you enter the database information, the Confirm Database Information screen (read-only) displays the information you entered at the previous prompts and the path to the install jar. Review each item's value, and press Enter to accept each one.
- The installation program verifies the database connection. If a connection cannot be established, you receive an error and can re-enter the database information to make more connection attempts. If you still cannot make a connection, consult with your database administrator (DBA).
13. At the "Select how you want to access product documentation" prompt, specify the type of Documentation Library you want to enable by entering 1 for Online or 2 for Local.
14. At the "Is this a 32 or 64 - bit JDK?" prompt, enter 1 if using a 32-Bit JDK or enter 2 if using a 64-Bit JDK.
- The installation process begins.
15. The system displays the message BUILD SUCCESSFUL when the installation is complete. For specific information about the installation, view the <INSTALL_DIR>/PreInstallSI.log file.
- Note:** You can delete the installed_data directory if it remains after installation. This is not needed after installing the product.
16. See "Post Installation Tasks for UNIX/Linux" on page 88 for configuration procedures that you must complete prior to running Sterling Selling and Fulfillment Foundation, as well as optional tasks, and post installation tips.

Creating a Silent Installation File - UNIX/Linux

About this task

If you prefer not to run the installer in the interactive install mode, you can create a text file that contains all of the information the installer needs and run the install from a command line in silent mode instead.

Procedure

Create the silent installation text file using a text editor. Follow these guidelines:

- The file must contain the entries marked as required in the following table. See "Sample Silent Installation File - UNIX/Linux" on page 84 for an example of a silent install file.
- You can create a silent upgrade text file using the parameters in the following table. See "Sample Silent Upgrade File - UNIX/Linux" on page 89 for an example of a silent upgrade file.
- When you are finished creating the file, record its name and location. You will use this information later to start the installation.
- If you are going to do a silent installation or upgrade in multiple schema mode, you must also create a multischema.xml file.

UNIX/Linux Silent Install File Parameters

UNIX and Linux Silent Install File Properties

Example Entry	Description
INSTALL_DIR=<INSTALL_DIR>	(Required) Full path of your installation directory. Note: This property cannot point to a pre-existing directory, and the path name cannot contain spaces, or the installation will fail.
DB_VENDOR=<db_vendor>	(Required) The database vendor to use (Oracle, DB2, MSSQL). Note: In a multischema installation, you must deploy the same database vendor and version across deployments.
MSSQL2005=<TRUE/FALSE>	If you are running on MSSQL 2005 or a later version, this attribute must be set to TRUE. If you are not running on MSSQL, do not include this attribute. Note: This attribute is case-sensitive.
DB_USER=<db_user_name>	(Required) User of database (system or user name). In a multischema deployment, this must be the username for the Metadata schema.
DB_DATA=<db_dat_catalog>	(Required) Database name to connect with.
DB_PASS=<db_password>	(Required) Database password. In a multischema deployment, this must be the password for the Metadata schema.
DB_HOST=<db_host>	(Required) Host for database (for example, server or IP address).
DB_PORT=<db_listener_port>	(Required) Database listener port.
DB_DRIVERS=<absolute path to driver jar>	(Required for Oracle, MSSQL Server, or DB2.) Full path to the JDBC driver file(s): <ul style="list-style-type: none"> • Oracle and MSSQL Server require one file. • DB2 requires two files: the license file and the driver file. When specifying more than one file, use colons (:) for separators as necessary. Examples: <ul style="list-style-type: none"> • <JDBC_driver_dir>/jdbc.jar • <JDBC_driver_dir>/db2_1_jdbc.jar:<JDBC_driver_dir>/db2_2_jdbc.jar
DB_DRIVERS_VERSION=<db_driver_version>	(Required) Free form version string for JDBC driver. This is informational only.
DB_SCHEMA_OWNER	(Required for multischema mode) Default schema/schema-owner for the provided login ID. If you wish to change this value to an alternate schema, consult your database administrator, as this is considered an expert installation scenario and can be performed only through the silent installation.

Example Entry	Description
LOAD_FACTORY_SETUP=true	<p>Indicates whether you want to load factory setup defaults during installation (true) or manually after installation (false). If you are performing an installation in upgrade mode, set this value to false.</p> <p>For information about manually loading the factory defaults, see “Loading the Sterling Selling and Fulfillment Foundation Database Factory Defaults After Installation” on page 119.</p>
NO_DBVERIFY=false	<p>Valid values are true/false. When set to true during installation and installservice, DBVerify will not be run. This means that Sterling Selling and Fulfillment Foundation will not generate DDL to make the database like the XML entity repository. If you are performing an installation in upgrade mode, set this value to true.</p>
REINIT_DB=true	<p>Valid values are true/false. By default, the value is set to true. If the value is set to false, the Sterling Selling and Fulfillment Foundation installation will complete successfully, but no database operation will be performed as part of the installation process. If you are performing an installation in upgrade mode, set this value to false.</p>
multischema.applyddl=true	<p>Valid values are true/false. If set to true (default), enables the DBVerify script to generate and apply database DDLs automatically. If set to false, allows DBVerify script to generate the DDLs but does not apply them.</p> <p>This property is set to false in the GUI installer, by default. If you are using the GUI Installer and do not want to apply DDLs, ensure that this property to set to false in the sandbox.cfg file. If this property is set to true or is absent in the sandbox.cfg file, the DBVerify script generates the DDLs and applies them. Otherwise, the scripts are generated and not applied.</p>
multischema.enabled=<true/false>	<p>If true, this attribute indicates that this is a multischema installation. The installation then looks for a customer-created multischema.xml file, which specifies database information for the Configuration, Metadata, Transaction, and Statistics schemas.</p> <p>Note: This attribute is case-sensitive.</p> <p>See “Running a Silent Installation in Multischema Mode - UNIX/Linux” on page 85 for more information.</p>
multischema.version=<version_number>	<p>(Required) This attribute indicates which version is being installed. For the Sterling Selling and Fulfillment Foundation Release 9.1, you must enter 9.1.</p> <p>Note: This attribute is case-sensitive.</p>

Example Entry	Description
multischema.file=<filename>.xml	(Required if you enable multischema.) This attribute indicates the name of the user-defined XML file that contains multischema database information. Note: This attribute is case-sensitive.
STERLING_FOUNDATION_PRODUCT_LABEL=smcfs	(Required) Specifies the product label.
STERLING_FOUNDATION_PRODUCT_VERSION=9.1	(Required) Specifies the product version you are installing.
ACCEPT_LICENSE=Y	Because the silent installer does not bring up an explicit license dialog, specify your acceptance of the licensing terms by including this property in your silent install file. Note: To review the license file prior to installation, browse to the ProductFiles folder on your product CD and open Readme.htm.
ACTIVE_DOC_URL= <ONLINE/LOCAL>	When set to ONLINE (default), the URL set for ONLINE_DOC_URL is used. When set to LOCAL, the URL set for LOCAL_DOC_URL is used. The ONLINE_DOC_URL or the LOCAL_DOC_URL is required when ACTIVE_DOC_URL is set to ONLINE or LOCAL, respectively. If you want to switch between the Online and Local Documentation Libraries after installation, refer to the <i>Sterling Selling and Fulfillment Foundation: Properties Guide</i> for more information about these properties.
ONLINE_DOC_URL=<url>	The value of this property is the URL for the Web-based Online Documentation Library that is in .js and PDF format: http://help.sterlingcommerce.com/SFSF91/index.jsp
LOCAL_DOC_URL=<url>	The value of this property is the URL for the Local Documentation Library that is in HTML format: /smcfsdocs/yfcommon/ online_help/en_US/wwhelp/wwhimpl/js/html/wwhelp.htm
JAVADOC_PRODUCT_LABEL=Sterling Selling and Fulfillment Foundation 9.1	(Required) Specifies the Javadocs to be installed.
JDK64BIT=<true/false>	Specifies whether you are using a 32-bit JDK or a 64-bit JDK. Default=true

Example Entry	Description
ADDITIONAL_ANT_JAVA_TASK_ARGS=- XX:MaxPermSize=512m ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms256m -J-Xmx1408m	See the <i>Sterling Selling and Fulfillment Foundation: System Requirements Guide</i> for memory parameter values based on your operating system. These parameter values are written to the sandbox.cfg file during installation. After installation, you can tune them if you are seeing Out-of-Memory errors. Note: The ADDITIONAL_ANT_JAVA_TASK_ARGS property must not be set for IBM and JRockit JDK. See the <i>Sterling Selling and Fulfillment Foundation: Properties Guide</i> for more information about these parameters.
ENTITY_GEN_LOGLEVEL=<VERBOSE/INFO>	Specify VERBOSE to enable logging of verbose messages during entity class generation. Default=INFO
GENERATE_ALL_DBCLASSES=<true/false>	Specify True to generate all DBClasses. Default=false
SUPPORT_MULTIBYTE=<Y/N>	If you are installing on DB2 and need to localize your database using a multibyte character set, set this flag to Y. This ensures that the database column sizes are large enough to handle the multibyte characters correctly. Default is N.
ORACLE_JDBC_URL=<url>	Required only if Oracle RAC is being used as the database. You must mention the complete URL without any blank space.

Sample Silent Installation File - UNIX/Linux

Following is a sample silent installation file:

```

INSTALL_DIR=home/jdoe/<INSTALL_DIR>
DB_VENDOR=Oracle
DB_USER=joe_smith
DB_DATA=110n
DB_PASS=joes_password
DB_HOST=00.00.00.00
DB_PORT=1521
DB_DRIVERS=home/jdoe/Oracle_Drivers/ojdbc6.jar
DB_DRIVERS_VERSION=11
DB_SCHEMA_OWNER=DB0
multischema.applyddl=false
multischema.enabled=true
multischema.version=9.1
multischema.file=multischema.xml
STERLING_FOUNDATION_PRODUCT_LABEL=smcfs
STERLING_FOUNDATION_PRODUCT_VERSION=9.1
ACCEPT_LICENSE=Y
ACTIVE_DOC_URL=ONLINE
ONLINE_DOC_URL=http://help.sterlingcommerce.com/SFSF91/index.jsp
LOCAL_DOC_URL=URL=/smcfsdocs/yfsccommon/online_help/en_US/wwhelp/wwhimpl/js/html/wwhelp.htm
JAVADOC_PRODUCT_LABEL=Sterling Selling and
Fulfillment Foundation 9.1
JDK64BIT=true

```

```
ADDITIONAL_ANT_JAVA_TASK_ARGS=-Xms1024m -Xmx1408m -XX:MaxPermSize=512m
ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms256m -J-Xmx1408m
GENERATE_ALL_DBCLASSES=false
ENTITY_GEN_LOGLEVEL=INFO
```

Running a Silent Installation in UNIX or Linux

About this task

Use the following instructions to install in a UNIX or Linux environment from a command line, using a silent installation file. See “Creating a Silent Installation File - UNIX/Linux” on page 80 for information about creating a silent install or upgrade file.

Procedure

1. Type the following command, which include paths to the JDK, the application jar file, and the silent installation file:

```
<JAVA_HOME>/bin/java -jar SMCFS_9.1.jar -f install.silent
```

If you prefer to use a JDK that is downloaded to an external location and is not copied into your application's local directories, enter:

```
<JAVA_HOME>/bin/java -jar SMCFS_9.1.jar -f install.silent -nocopyjvm
```

After the installation process begins, you can follow the progress of your installation through the <INSTALL_DIR>/InstallSI.log.

The installation displays the message *Installation has completed successfully* when done.

Note: You can delete the installed_data directory if it remains after installation. This is not needed after installing the product.

2. See “Post Installation Tasks for UNIX/Linux” on page 88 for configuration procedures that you must complete prior to running Sterling Selling and Fulfillment Foundation, as well as optional tasks, and post installation tips.

Running a Silent Installation in Multischema Mode - UNIX/Linux

If you are planning to run a silent installation or upgrade and plan to use multischema mode, you must create a multischema.xml file for the installer to use.

The multischema.xml file is invoked by the installation process if multischema is enabled in the silent installation file. The file must specify the database account information for each of the four allowable four data tables that you use: Metadata, Statistics, System Configuration, and Transaction/Master.

It also shows parameters for specifying multiple passwords and their effective dates so that you can predefine passwords for a given pool months in advance. These passwords will change automatically without a server restart.

In a multischema installation, you must deploy the same database vendor and version across deployments.

Multischema Parameters

You can change only the following parameters:

Parameter

Definition

<jdbc_url>

Specify the URL to connect to the database.

- If using Oracle, set to:

```
jdbc:oracle:thin:@<DatabaseServerHostnam  
e/IPAddress>:<TNSListenerPortNumber>:<Da  
tabaseSID>
```

- If using MSSQL Server, set to:

```
jdbc:sqlserver://<Database  
ServerHostname>:<PortNumber>;DatabaseNam  
e=<Database name>
```

- If using DB2, set to:

```
jdbc:db2://<Database  
ServerHostname>:<Port Number>/<Database  
name>.<db_user> Database user name
```

<db_user>

Specify the user name associated with the database.

<db_password>

Specify the password associated with the database.

<db_driver_class>

Specify the class name of your database driver as follows.

- If using Oracle, set to:

```
oracle.jdbc.driver.OracleDriver
```

- If using MSSQL Server, set to:

```
com.microsoft.sqlserver.jdbc.SQLServerDriver
```

- If using DB2, set to:

```
com.ibm.db2.jcc.DB2Driver
```

<db_schema>

Specify the schema name associated with the database if it is different from the <db_user> name you entered.

Note: This parameter is case-sensitive and you must specify it in UPPERCASE.

<datasource>

Specify the datasource that you created on the application server for use with this schema.

<password.1>

Specify a database password for the effectivity date shown in the <effective.1> parameter shown in this table.

<password.2>

Specify an effective date for the <password.1> parameter shown in this table.

Example: 2009-07-16T15:00:00

<effective.2>

Specify an effective date for the <password.2> parameter shown in this table.

Example: 2009-08-25T12:00:00

Sample Multischema.xml File

Unlike the silent installation, the GUI and text-based installations automatically create the multischema.xml file for you. The following example is for a DB2 installation.

See the *Sterling Selling and Fulfillment Foundation: Multi-Tenant Enterprise Guide* for more information about the various schema types mentioned here, and about colonies and other multischema information.

```
<?xml version="1.0" encoding="UTF-8"?>
<colonyconfig>
  <colonies>
    <colony name="DEFAULT" pkprefix="20" version="9.1">
      <schema poolid="DEFAULT_METADATA" tabletype="METADATA"/>
      <schema poolid="DEFAULT_STATISTICS_91" tabletype="STATISTICS"/>
        <schema poolid="DEFAULT_CONFIGURATION_91" tabletype="CONFIGURATION"/>
      <schema poolid="DEFAULT_TRANSACTION_91" tabletype="TRANSACTION"/>
        <schema poolid="DEFAULT_TRANSACTION_91" tabletype="MASTER"/>
    </colony>
  </colonies>
  <pools>
    <pool id="DEFAULT_METADATA">
      <jdbc>
        <param name="url" value="jdbc:db2://00.00.00.00:50000/devdb2"/>
        <param name="user" value="metadata_user"/>
        <param name="password" value="metadata_user"/>
        <param name="driver" value="com.ibm.db2.jcc.DB2Driver"/>
        <param name="schema" value="METADATA_USER"/>
        <param name="datasource" value="metadata_datasource_name"/>
        <param name="password.1" value="password_1"/>
        <param name="password.2" value="password_2"/>
        <param name="effective.1" value="2009-07-16T15:00:00"/>
        <param name="effective.2" value="2009-08-16T15:00:00"/>
      </jdbc>
    </pool>
    <pool id="DEFAULT_STATISTICS_91">
      <jdbc>
        <param name="url" value="jdbc:db2://00.00.00.00:50000/devdb2"/>
        <param name="user" value="statistics_user"/>
        <param name="password" value="statistics_user"/>
        <param name="driver" value="com.ibm.db2.jcc.DB2Driver"/>
        <param name="schema" value="STATISTICS_USER"/>
        <param name="datasource" value="statistics_datasource_name"/>
        <param name="password.1" value="password_1"/>
        <param name="password.2" value="password_2"/>
        <param name="effective.1" value="2011-07-16T15:00:00"/>
        <param name="effective.2" value="2011-08-16T15:00:00"/>
      </jdbc>
    </pool>
    <pool id="DEFAULT_CONFIGURATION_91">
      <jdbc>
        <param name="url" value="jdbc:db2://00.00.00.00:50000/devdb2"/>
        <param name="user" value="configuration_user"/>
        <param name="password" value="configuration_user"/>
        <param name="driver" value="com.ibm.db2.jcc.DB2Driver"/>
        <param name="schema" value="CONFIGURATION_USER"/>
        <param name="datasource" value="configuration_datasource_name"/>
        <param name="password.1" value="password_1"/>
        <param name="password.2" value="password_2"/>
        <param name="effective.1" value="2011-07-16T15:00:00"/>
        <param name="effective.2" value="2011-08-16T15:00:00"/>
      </jdbc>
    </pool>
    <pool id="DEFAULT_TRANSACTION_91">
      <jdbc>
```

```

        <param name="url" value="jdbc:db2://00.00.00.00:50000/devdb2"/>
        <param name="user" value="transaction_user"/>
        <param name="password" value="transaction_user"/>
        <param name="driver" value="com.ibm.db2.jcc.DB2Driver"/>
        <param name="schema" value="TRANSACTION_USER"/>
        <param name="datasource" value="transaction_datasource_name"/>
        <param name="password.1" value="password_1"/>
        <param name="password.2" value="password_2"/>
        <param name="effective.1" value="2011-07-16T15:00:00"/>
        <param name="effective.2" value="2011-08-16T15:00:00"/>
    </jdbc>
</pool>
</pools>
</colonyconfig>

```

Post Installation Tasks for UNIX/Linux

This topic contains configuration procedures that you must complete prior to running Sterling Selling and Fulfillment Foundation, as well as optional tasks and post installation tips.

Security

- See “Web Security Planning-Post installation Recommendations” on page 11 for information about the security measures that are recommended for your consideration.

Database

- Views must be created manually. Instructions to create views vary depending on what database is used. All database view related scripts are located at <INSTALL_DIR>/database/<db_type>/scripts. For more information, see the information about configuring your database type in “Database Configuration on UNIX/Linux-Overview” on page 27 and Chapter 14, “Configuring Utilities,” on page 111.

Changing to Multischema Mode after Installation

- If you did not install or upgrade in multischema mode but you want to create the appropriate tables for multischema mode and update these tables at a later time, you can:
 - Set the following properties in the `sandbox.cfg` file:


```
multischema.enabled=true
multischema.version=9.1
```
 - Run the `dbverify` script on multischema colonies, as described in “Verifying the Database Schema” on page 114.

If you want to create or add colonies to your deployment, see the *Sterling Selling and Fulfillment Foundation: Multi-Tenant Enterprise Guide* for instructions.

Temporary Directories Left by Installer

- After installation, you may see temporary directories similar to the following example:


```
tmpSterlingInstall.12345678/
```

You can delete these directories after the installation has completed.

Upgrade Information for UNIX/Linux

If installing in upgrade mode, use the topics in this section for reference.

- “Running a Silent Upgrade in UNIX or Linux”
- “Sample Silent Upgrade File - UNIX/Linux”
- “Post Upgrade Tasks for UNIX/Linux” on page 90

Running a Silent Upgrade in UNIX or Linux

About this task

Use the following instructions to upgrade in a UNIX or Linux environment from a command line, using a silent installation file. See “Creating a Silent Installation File - UNIX/Linux” on page 80 for information about creating a silent install or upgrade file.

- If the application is running, stop the previous installation before proceeding.
- See the Upgrade Guide for more detailed information about performing a Sterling Selling and Fulfillment Foundation upgrade.

1. Ensure that your silent upgrade file has the following settings:
LOAD_FACTORY_SETUP=false, NO_DBVERIFY=true, and REINIT_DB=false
2. Type the following command, which include paths to the JDK, the application jar file, and the silent installation file:

```
<JAVA_HOME>/bin/java -jar SMCFS_9.1.jar -f install.silent
```

If you prefer to use a JDK that is downloaded to an external location and is not copied into your application's local directories, enter:

```
<JAVA_HOME>/bin/java -jar SMCFS_9.1.jar -f install.silent -nocopyjvm
```

After the upgrade process begins, you can follow the progress through the <INSTALL_DIR>/PreInstallSI.log.

The installation displays the message *Installation has completed successfully* when done.

3. See “Post Installation Tasks for UNIX/Linux” on page 88 for configuration procedures that you must complete prior to running Sterling Selling and Fulfillment Foundation, as well as optional tasks, and post installation tips.

Sample Silent Upgrade File - UNIX/Linux

Following is a sample silent upgrade file:

```
INSTALL_DIR=home/joe_user/<INSTALL_DIR>
LOAD_FACTORY_SETUP=false
REINIT_DB=false
NO_DBVERIFY=true
DB_VENDOR=Oracle
DB_USER=joe_smith
DB_DATA=110n
DB_PASS=joes_password
DB_HOST=00.00.00.00
DB_PORT=1221
DB_DRIVERS=home/joe_user/dbdrivers/ojdbc6.jar
DB_DRIVERS_VERSION=11
DB_SCHEMA_OWNER=DB0
multischema.applyddl=false
multischema.enabled=true
multischema.version=9.1
multischema.file=multischema.xml
STERLING_FOUNDATION_PRODUCT_LABEL=smcfs
STERLING_FOUNDATION_PRODUCT_VERSION=9.1
```

```
ACCEPT_LICENSE=Y
ACTIVE_DOC_URL=ONLINE
ONLINE_DOC_URL=http://help.sterlingcommerce.com/SFSF91/index.jsp
LOCAL_DOC_URL=URL=/smcfsdocs/yfsccommon/online_help/en_US/wwhelp/wwhimpl/js/html/wwhelp.htm
JAVADOC_PRODUCT_LABEL=Sterling Selling and
Fulfillment Foundation 9.1
JDK64BIT=true
ADDITIONAL_ANT_JAVA_TASK_ARGS=-Xms1024m -Xmx1408m -XX:MaxPermSize=512m
ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms256m -J-Xmx1408m
GENERATE_ALL_DBCLASSES=false
ENTITY_GEN_LOGLEVEL=INFO
```

Post Upgrade Tasks for UNIX/Linux

Creating Tables for MultiSchema Mode After Installation

If you did not install or upgrade in multischema mode but you want to create the appropriate tables for multischema mode and update these tables at a later time, you can:

- Set the following properties in the `sandbox.cfg` file:

```
multischema.enabled=true
multischema.version=9.1
```

- Run the `dbverify` script on multischema colonies, as described in “Verifying the Database Schema” on page 114

If you want to create or add colonies to your deployment, see the *Sterling Selling and Fulfillment Foundation: Multi-Tenant Enterprise Guide* for instructions.

Temporary Directories Left by Installer

After installation, you may see temporary directories similar to the following example:

```
tmpSterlingInstall.12345678/
```

You can delete these directories after the installation has completed.

Chapter 9. Installing the IBM Sterling Sensitive Data Capture Server

Installing the IBM Sterling Sensitive Data Capture Server

Before installing the IBM Sterling Sensitive Data Capture Server (SSDCS) application, you must read the *Sterling Selling and Fulfillment Foundation: Sterling Sensitive Data Capture Server, Release 1.1: Configuration Guide* for information about how to configure IBM Sterling Sensitive Data Capture Server securely.

The IBM Sterling Sensitive Data Capture Server is an application that integrates with IBM Sterling Selling and Fulfillment Suite to ensure that credit card numbers and stored value card numbers are secure by tokenizing them. IBM Sterling Sensitive Data Capture Server is a system-critical application that must be installed, configured, and deployed before IBM Sterling Selling and Fulfillment Suite can capture payment information for credit cards and stored value cards.

The IBM Sterling Sensitive Data Capture Server application is packaged as a compressed file with Sterling Selling and Fulfillment Foundation. This compressed file is located in <INSTALL_DIR>/repository/external/ssdcs.zip.

For information about how to install, configure, extend, and deploy IBM Sterling Sensitive Data Capture Server, refer to the *Sterling Selling and Fulfillment Foundation: Sterling Sensitive Data Capture Server, Release 1.1: Configuration Guide*.

Chapter 10. Installing the Sterling Selling and Fulfillment Foundation Language Pack

Installing the Sterling Selling and Fulfillment Foundation Language Pack

Before installing the language pack, ensure that you have successfully installed Sterling Selling and Fulfillment Foundation, Release 9.1.

Language packs are compressed files that are compatible with the UNIX, Linux, and Windows operating systems. To install the Sterling Selling and Fulfillment Foundation language pack, extract the contents of the compressed language pack file into your <INSTALL_DIR> directory.

Installing the Sterling Selling and Fulfillment Foundation Context-Sensitive Help Language Pack

Procedure

1. Install the context-sensitive help language pack by running the following script from <INSTALL_DIR>\bin directory.

For Windows: `sci_ant.cmd -f csHelpUtil.xml -DlocaleCode=<LOCALE_CODE>`

For UNIX/Linux: `sci_ant.sh -f csHelpUtil.xml -DlocaleCode=<LOCALE_CODE>`

2. Include the following property in the `customer_overrides.properties`:

```
yfs.onlinehelp.path.overrideforlocale.<LOCALE_CODE>=/smcfsdocs/  
yfscommon/online_help/<LOCALE_CODE>
```

Here, <LOCALE_CODE> refers to [language_code]_[country_code].

For more information about using the `customer_overrides.properties` file, refer to the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Note: You must install the context-sensitive help language pack only after installing the language pack for all the applications.

Note: After installing the context-sensitive help language pack, ensure to re-create the resource JAR file and the EAR file.

Loading the Sterling Selling and Fulfillment Foundation Language Pack Factory Defaults-Overview

Prior to loading the Sterling Selling and Fulfillment Foundation Language Pack factory defaults, ensure that you have successfully completed all the instructions provided in Chapter 5, “Installing and Configuring Database Tier Software on UNIX or Linux,” on page 27 or Chapter 6, “Installing and Configuring Database Tier Software on Windows,” on page 37, as the case may be.

Note: The English language factory defaults must be loaded prior to loading the language-specific factory defaults.

Loading the Sterling Selling and Fulfillment Foundation Language Pack Factory Defaults for Windows Procedure

To load the language-specific factory defaults, run the loadDefaults.cmd script for Windows that is available in the <INSTALL_DIR>\bin directory. Pass the locale-specific installer file and the directory path of the associated XML files. For example:

```
loadDefaults.cmd <INSTALL_DIR> \repository\factorysetup\
complete_installation\<language>_<country>_locale_installer.xml
<INSTALL_DIR>\repository\factorysetup\complete_installation\XMLS
```

For more information about the configuration steps involved in loading the factory defaults, see “Loading the Sterling Selling and Fulfillment Foundation Database Factory Defaults After Installation” on page 119.

Note: You must manually create a currency definition for Spanish locale.

For more information about creating currency definition, see the *Sterling Selling and Fulfillment Foundation : Application Platform Configuration Guide*.

Loading the Sterling Selling and Fulfillment Foundation Language Pack Factory Defaults for UNIX/Linux Procedure

To load the language-specific factory defaults, run the loadDefaults.sh script for UNIX and Linux that is available in the <INSTALL_DIR>/bin directory. Pass the locale-specific installer file and the directory path of the associated XML files. For example:

```
loadDefaults.sh <INSTALL_DIR> /repository/factorysetup/complete_installation
/<language>_<country>_locale_installer.xml <INSTALL_DIR>/repository/
factorysetup/complete_installation/XMLS
```

For more information about the configuration steps involved in loading the factory defaults, see “Loading the Sterling Selling and Fulfillment Foundation Database Factory Defaults After Installation” on page 119.

Note: You must manually create a currency definition for Spanish locale.

For more information about creating currency definition, see the *Sterling Selling and Fulfillment Foundation : Application Platform Configuration Guide*.

Loading the Language Pack Translations in Windows About this task

Prior to loading the Sterling Selling and Fulfillment Foundation Language Pack factory defaults, ensure that you have successfully completed all the instructions provided in Chapter 7, “Installing Sterling Selling and Fulfillment Foundation in a Windows Environment,” on page 45.

Important: Verify that your locale settings, such as currency, time format, date, and so on are correct.

Note: The translation files required for IBM Sterling Business Center are also installed during the installation of the Sterling Selling and Fulfillment Foundation Language Pack. You do not have to perform any additional steps to localize the IBM Sterling Business Center.

Procedure

To load the language pack translation with custom localization literals, run the Localized String Reconciler tool in the IMPORT mode from the <INSTALL_DIR>/bin directory as follows:

```
sci_ant.cmd -f localizedstringreconciler.xml import
-Dsrc=<INSTALL_DIR>
/repository/factorysetup/complete_installation
/XMLS -Dbasename
=ycplocalizedstrings
```

where <INSTALL_DIR> refers to the installation directory.

The basename refers to the file present in the <INSTALL_DIR>//repository/factorysetup/complete_installation/XMLS directory, for which the translations are to be imported into the database.

The default value for the basename parameter is ycplocalizedstrings.

For example, to import translations for the IBM Sterling Call Center and IBM Sterling Store language pack, the base file is ycdlocalizedstrings. The base file is derived from the xx_XX_ycdlocalizedstrings_yy_YY.properties file. This file inserts the values specified in the properties file into the database.

The Localized String Reconciler tool inserts the values specified in the <from_language>_<from_country>_<basename>_<to_language>_<to_country>.properties file present in the <INSTALL_DIR>/repository/factorysetup/complete_installation/XMLS/<language>_<country> directory into the database.

Loading the Language Pack Translations in UNIX/Linux

About this task

Prior to loading the Sterling Selling and Fulfillment Foundation Language Pack factory defaults, ensure that you have successfully completed all the instructions provided in Chapter 8, “Installing Sterling Selling and Fulfillment Foundation in UNIX and Linux Environments,” on page 67.

Important: Verify that your locale settings, such as currency, time format, date, and so on are correct.

Note: The translation files required for IBM Sterling Business Center are also installed during the installation of the Sterling Selling and Fulfillment Foundation Language Pack. You do not have to perform any additional steps to localize the IBM Sterling Business Center.

Procedure

To load the language pack translation with custom localization literals, run the Localized String Reconciler tool in the IMPORT mode from the <INSTALL_DIR>/bin directory as follows:

```
sci_ant.sh -f localizedstringreconciler.xml import
-Dsrc=<INSTALL_DIR>/repository/factorysetup
/complete_installation/XMLS -Dbasename
=ycplocalizedstrings
```

where <INSTALL_DIR> refers to the installation directory.

The basefilename refers to the file present in the <INSTALL_DIR>\\repository/factorysetup/complete_installation/XMLS directory, for which the translations are to be imported into the database.

The default value for the basefilename parameter is ycplocalizedstrings.

For example, to import translations for the IBM Sterling Call Center and IBM Sterling Store language pack, the base file is ycdlocalizedstrings. The base file is derived from the xx_XX_ycdlocalizedstrings_yy_YY.properties file. This file inserts the values specified in the properties file into the database.

The Localized String Reconciler tool inserts the values specified in the <from_language>_<from_country>_<basefilename>_<to_language>_<to_country>.properties file present in the <INSTALL_DIR>/repository/factorysetup/complete_installation/XMLS/<language>_<country> directory into the database.

Switching the Sterling Selling and Fulfillment Foundation Base Language

The base language for the Applications Manager can be switched only once. For more information about switching the base language and performing the switch test, see the *Sterling Selling and Fulfillment Foundation: Localization Guide*.

Note: After switching the base locale from en_US to any other locale (for example, fr_FR), CatalogSearchConfigProperties.xml should be extended to include the switched locale code (which is fr_FR) in order to build the search index successfully. For more information about extending catalog search, see *Sterling Selling and Fulfillment Foundation: Extending the Database*. Also see the *Sterling Selling and Fulfillment Foundation: Catalog Management Concepts Guide* for more general information about catalog searches.

Creating and Deploying the Enterprise Archive

If you are installing both Sterling Selling and Fulfillment Foundation and the language pack together, it is sufficient if you create and deploy the EAR once. If you have already deployed your application and are installing the language pack after this, you need to re-create and redeploy the EAR file.

For more information about creating and deploying the EAR file for your chosen application server, see Chapter 15, “Deploying Sterling Selling and Fulfillment Foundation,” on page 127.

Configuring Locales in UNIX/Linux

About this task

Sterling Selling and Fulfillment Foundation runs on any locale that Java supports. If you want to run Sterling Selling and Fulfillment Foundation on a non-default locale, configure your environment to the specific locale that you want to use.

Note: To configure your operating system as a non-English environment, refer to your operating system's documentation.

To determine and set the locale in a UNIX or Linux environment:

Procedure

1. Enter locale -a. A list of locales is displayed.
2. Set your locale by entering:
 - export LANG=<locale>
 - export LC_ALL=<locale>Example to set the locale to Japanese (on Solaris):
 - export LANG=ja_JP
 - export LC_ALL=ja_JP

Note: Some UNIX shells require the setenv command instead of the export command.

To determine and set your locale in a Windows environment:

3. Select Control Panel > Regional Options > General tab.
4. From the Your locale (location) list, select the language and location.
5. Click Set Default and select the locale from the Set the appropriate locale list.

Configuring Locales in Windows

About this task

Sterling Selling and Fulfillment Foundation runs on any locale that Java supports. If you want to run Sterling Selling and Fulfillment Foundation on a non-default locale, configure your environment to the specific locale that you want to use.

Note: To configure your operating system as a non-English environment, refer to your operating system's documentation.

To determine and set your locale in a Windows environment:

Procedure

1. Select Control Panel > Regional Options > General tab.
2. From the Your locale (location) list, select the language and location.
3. Click Set Default and select the locale from the Set the appropriate locale list.

Chapter 11. Installing a Print Server

Installing a Print Server - Overview

This section explains how to install and configure the Loftware Label Manager (LLM) and Loftware Print Server (LPS).

For more information about configuring the Loftware Label Manager and Print Server, see the *Loftware Label Manager User's Guide* and the *Loftware Print Server User's Guide*, available from Loftware, Inc.

For more information about Performance Considerations for setting up the Loftware Print Server (LPS) see the *Loftware Print Server User's Guide*, available from Loftware, Inc.

Note: Loftware Label Manager and Loftware Print Server are third party software products that can be purchased directly from the vendor, Loftware, Inc.

Installing Loftware Components

The Loftware Print Server manages bar code label print requests between applications and hundreds of networked printers. As a general guideline, you should configure a maximum of 200 printers for each Loftware Print Server you install. For more information about server requirements and installation instructions, see the *Loftware Print Server User's Guide*. Contact your Loftware support representative for additional sizing and configuration support.

The Loftware Label Manager, used for designing labels, may be installed on any compatible PC. For more information about server requirements and installation guidelines, see the *Loftware Label Manager User's Guide*.

Sterling Selling and Fulfillment Foundation supports printing in the following modes:

- File Copy Mode
- TCP/IP Sockets Mode

The `yfs.ftware.tcpip.sockets` attribute in the `yfs.properties.in` file determines the mode used for printing. By default this boolean property is set to 'N' for File Copy Mode.

To configure the Loftware printing in the TCP/IP Sockets Mode, use the `<INSTALL_DIR>/properties/customer_overrides.properties` file to set the `yfs.ftware.tcpip.sockets` property to Y. For more information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Sterling Selling and Fulfillment Foundation requires the following settings in the Loftware Print Server Configuration Utility:

- In Directory Set up, ensure that the 'Pass Files' option is selected.
- When using File Copy Mode: In Directory Set up, ensure that the 'Enable Polling (Disable Event File Trigger)' option is selected. It is recommended that the Poll Interval value is set to 500 Milliseconds.

Note: In File Copy Mode, SAMBA should be configured when using a UNIX version of the application server.

Note: The Drop Directories of the printers configured in Loftware need to be mounted on to the UNIX server using SAMBA.

Defining Printers on Loftware

Configure printers on Loftware using the Loftware Design 32 tool. For more information about configuring printers using the Loftware Design 32 tool, see the *Loftware Label Manager User's Guide*.

Defining Printers for the Sterling Warehouse Management System Installation

For more information about configuring printers for the Sterling Warehouse Management System, see the *Sterling Selling and Fulfillment Foundation: Warehouse Management System Configuration Guide*.

Copying the Sterling Warehouse Management System Standard Label Formats

The Sterling Warehouse Management System provides Loftware Label Manager template (*.lwl) files which should be copied in the directory set up for labels using the Loftware Design 32 tool.

Also, copy the YCP_LABEL_FIELDS.LST file to the directory where the Loftware Label Manager has been installed. This file is available in the <INSTALL_DIR>/repository/xapi/template/source/<application_install>/prints/label directory.

Installing JasperReports

JasperReports is a third-party, open source Java reporting tool that delivers rich content on the screen, to the printer or in the format of a PDF, HTML etc.. You can use JasperReports with Sterling Selling and Fulfillment Foundation for printing or generating PDF objects for order reports, labels and so forth. The installation procedure and sample files are located in <INSTALL_DIR>/xapidocs/code_examples/jasperreports directory.

Note: For more information about JasperReports and supporting jars and components, see the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* and the alert_report_readme.html file, located in the <INSTALL_DIR>/xapidocs/code_examples/jasperreports directory.

Chapter 12. Installing a Weighing Scale

Installing the Weighing Scale - Overview

Weighing scales are typically used at packing or manifest stations. This chapter describes the installation of weighing scales for use with the Sterling Warehouse Management System.

For more information regarding the Mettler-Toledo PS Weighing Scale, see the *Mettler-Toledo PS Weighing Scale User Guide*.

The weighing scale is installed at each pack or manifest station requiring weighing scale integration.

Installing the Weighing Scale

About this task

To install the weighing scale, follow these steps on each client machine:

Procedure

1. Launch your Internet Explorer browser.
2. In the Address bar, type `http://<hostname>:<portnumber>/smcfs<application_name>/yfscommon/ibmcomm.dll` and press Enter.
3. The File Download window appears. Choose Save.
4. The Save As window appears. Save the file in any directory present in the System Class path. For example, on Windows NT, go to `C:/WINNT/system32`.

Results

For more information about setting up the weighing scale and associating it with a station, see the Equipment section of the *Sterling Selling and Fulfillment Foundation: Warehouse Management System Configuration Guide*.

Chapter 13. Installing the Mobile Application

Installing the Mobile Application - Overview

This section describes how to install the Mobile Application for use on PocketPC, WinCE, and VT220 mobile terminals.

For information on PocketPC and WinCE mobile terminal system requirements, see the *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.

Before installing the Mobile Application, install the Microsoft .NET Compact Framework on your local PC. You can download the Microsoft .NET Compact Framework from <http://www.microsoft.com>. For more information about the supported versions for the Microsoft .NET Compact Framework, see the *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.

Installing the Microsoft .NET Compact Framework creates multiple .CAB files on the system, for multiple operating systems and processors of the device.

Installing on Mobile Terminals

About this task

To set up the PC with the Mobile Application for the Mobile Terminal, follow these steps:

Procedure

1. Connect the Mobile Terminal to the PC that has ActiveSync installed. For more information, refer to the respective Mobile Terminal documentation for each device type.

Note: It is suggested that ActiveSync be used to copy the files.

Alternatively, you may transfer the file over the network, if the device is already configured to access the LAN, or you may serve the .CAB file through a web server and use the Internet Explorer browser on the device to download it.

If the Sterling Warehouse Management System is installed on a UNIX server, it may be required to copy the files from the UNIX server to the PC before launching ActiveSync.

2. Choose Start > Programs > ActiveSync on the PC.
3. Choose File > Get Connected, if not already connected.
4. Choose the Explore icon. This brings up the File Explorer for the Mobile Terminal.
5. Go to the <INSTALL_DIR> folder.
6. In the File Explorer window, click the Folders icon to bring up folders in the left panel.
7. Copy the YantraMobileApp_xxx.CAB files from the <INSTALL_DIR>/mobileapp folder on the PC to the \Application folder on the mobile terminal.

Here, xxx refers to PPC.ARM, PPC.ARMV4, WCE4.ARMV4, WCE4.ARMV4T, or WCE4_5.ARMV4.

Ensure that you choose the .CAB file that is relevant to the operating system and processor of your mobile terminal.

ARM, ARMV4, ARMV4T, and so on are the processors in the mobile device. The PPC CAB files are used in Pocket PC devices, while the CAB files with WCE4 are used in the WinCE devices.

Locate the CAB file relevant to the operating system on your mobile device as specified in Table 9.

Table 9. Operating Systems and CAB Files

CAB File	Operating System
YantraMobileApp_PPC.ARM.CAB	Pocket PC 2003
YantraMobileApp_PPC.ARMV4.CAB	Pocket PC 2003
YantraMobileApp_WCE4.ARMV4.CAB	Windows CE 4.1, Windows CE 5.0
YantraMobileApp_WCE4.ARMV4T.CAB	Windows CE 4.1, Windows CE 5.0
YantraMobileApp_WCE4_5.ARMV4.CAB	Windows CE 5.0

Note: All the WCE4 CAB files mentioned above are supported on Windows CE 5.0 operating system. However, the WCE4 CAB files, other than YantraMobileApp_WCE4_5.ARMV4.CAB, gives a warning during installation on the Windows CE 5.0.

8. Double-click on the YantraMobileApp_XXX.CAB file, on your mobile terminal. This installs the application on the mobile terminal.

Note: This file is automatically deleted upon successful installation.

9. The sample file <INSTALL_DIR>/mobileapp/yantrahostlist.xml includes application servers with Loopback, Production, QA and Test names. Replace these with the application servers along with their IP addresses, port numbers, and web context root, for example, <YantraHost name='QA' URL='10.10.10.40:7001' contextRoot='smcfs'/>. If the web context root is not passed, the Mobile Application will send the request to "/yantra<application_name>".

The application server names entered here are listed in the Servers drop-down list to which you can get connected when you launch the Mobile Application on the PocketPC or WinCE mobile terminals.

If you want to run the application on https, configure securemode and provide secureURL attributes in the <INSTALL_DIR>/mobileapp/yantrahostlist.xml file. The applicable values of securemode are 'all' and 'none'.

If you set securemode to:

- all, the application runs on https.
- none, secureURL is ignored and the application runs on http.

To use the above listed attributes, refer to the YantraHostList.xml file provided with the application.

Note: This step is not valid for VT220 mobile terminal.

Servers include application servers used for production, test, and other environments, if applicable.

10. Locate the properties file for your mobile device as specified in Table 10 and rename that file to `YMAProperties.xml`.

Table 10. Device and Properties File

Device	Properties File
LXE MX7 handheld	YMAProperties.MX7.xml
LXE VX3X series truck mount	YMAProperties.VX3X.xml
PocketPC mobile terminal	YMAProperties.ppc.xml
Symbol VRC7900 series truck mount	YMAProperties.vrc7900.xml

11. Stop ActiveSync.

For additional information about the replication of the Mobile Application to multiple mobile devices, see the *Mobile Terminal* documentation.

The *Mobile Terminal* documentation also provides additional details on re-installation.

Note: To display clear and appropriate error messages, a locale specific "System_SR_<locale>.cab" resource file is needed. For more information about the locale specific resource file, contact your PocketPC product support at <http://support.microsoft.com>.

Note: IBM recommends you to configure the barcode scanner such that the TAB character is suffixed with the scanned barcode data.

Ensuring Reinstallation on Cold Boot

This section provides instruction on how to automatically install Microsoft .NET Compact Framework and Mobile Application on cold-boot on various mobile terminals.

Note: Some of the RF vendors provide utilities for restoring applications upon cold boot. Use these utilities in place of the steps mentioned in the subsequent sections.

Installing On a PocketPC Mobile Terminal About this task

To ensure that the Microsoft .NET Compact Framework and the Mobile Application are installed automatically on cold boot, the following instructions must be followed as a one time measure.

Procedure

1. Install the Microsoft .NET Compact Framework on the mobile terminal, BEFORE installing the Mobile Application. This may be downloaded from <http://www.microsoft.com>.
2. Install the Mobile Application. For more information, see "Installing on Mobile Terminals" on page 103.
3. Copy the Microsoft .NET Compact Framework installation CAB file from the local PC to the \Application folder on the mobile device.
4. Copy all the files under Program Files/YantraMobileApp on the mobile device to the \Application\YantraMobileApp on the mobile device.

5. Copy the file YantraMobileApp.lnk from the \Windows\StartMenu\Programs folder on the mobile device to the \Application folder on the mobile device.
6. Edit the following line in the yantra.cpy file located in the <INSTALL_DIR>/mobileapp folder.

```

\Application\netcf.core.ppc3.ARM.cab \Windows\startup\netcf.core.ppc3.ARM.cab

```

 Here, change the occurrences of netcf.core.ppc3.ARM.cab to the CAB file name as appropriate for your handheld device
7. Copy the modified yantra.cpy file to the \Application folder on the mobile device.

Results

This ensures that the Microsoft .NET Compact Framework and the Mobile Application are installed on cold boot. When cold booting the system, the yantra.cpy file copies the installation files to the start directory.

Installing On a Symbol VRC7900 WinCE Mobile Terminal About this task

Symbol VRC7900 supports installation of software during cold boot by storing the installation .cab files in the \FlashFx\CAB folder.

All files that need to be copied to the folder <INSTALL_DIR>\mobileapp folder in the RAM file system must be placed under \FlashFx\CopyToRam\Root\
<INSTALL_DIR>\mobileapp.

To ensure that the Microsoft .NET Compact Framework and the Mobile Application are installed automatically on cold boot, the following instructions must be followed as a one time measure:

Procedure

1. Install the Microsoft .NET Compact Framework on the mobile terminal, BEFORE installing the Mobile Application. This may be downloaded from <http://www.microsoft.com>.
2. Install the Mobile Application. For more information, see “Installing on Mobile Terminals” on page 103.
3. Copy the Microsoft .NET Compact Framework installation CAB file, netcf.core.WINCE.ARMV4.cab, from the local PC to the \FlashFx\CAB folder on the mobile device.
4. Copy all the files (including the hidden file vsd_setup.dll) under the <INSTALL_DIR>\mobileapp folder to the \FlashFx\CopyToRam\Root\
<INSTALL_DIR>\mobileapp folder.
5. If a desktop shortcut has been created, copy the corresponding shortcut file (with .lnk extension) to the \FlashFx\CopyToRam\System\Desktop folder.
6. Copy the \Windows\My Company YantraMobileApp.unload file to the \FlashFx\CopyToRam\System folder.

Results

This ensures that the Microsoft .NET Compact Framework and the Mobile Application are installed on cold boot.

Installing On a Denso BHT400B Win CE 5.0 Mobile Terminal

About this task

To ensure that the Microsoft .NET Compact Framework and the Mobile Application are automatically installed on cold-boot, follow these instructions as a one time measure:

Procedure

1. Install the Microsoft .NET Compact Framework 2.0 SP1 on the mobile terminal, prior to installing the Mobile Application. This can be downloaded from <http://www.microsoft.com>.
2. Install the Mobile Application. For more information about installing the Mobile Application, see "Installing on Mobile Terminals" on page 103.

Note: Cold-booting of the mobile terminal erases the data stored in RAM. Therefore, copy the installation files into a folder whose contents are retained even after performing a cold-boot. See the Mobile Terminal Operator's guide provided by the manufacturer to identify the appropriate folder.

For Win CE 5.0 mobile terminals manufactured by Denso Corporation, contents of the '\Flash' folder are retained after cold-booting. If you are using any other mobile terminal, locate the folder and replace all occurrences of '\Flash' with the located folder in the following steps:

3. Copy the Microsoft .NET Compact Framework 2.0 SP1 installation CAB file from the local PC to the \Flash folder on the mobile terminal.
4. Copy the relevant Mobile Application CAB file from the local PC to the \Flash folder on the mobile terminal.
5. Create the \MobileApp folder under \Flash folder. Copy the following files from the directory where the Mobile Application is installed (\Program Files\MobileApp) to \Flash\MobileApp folder:
 - YantraHostList.xml
 - YMAProperties.xml
6. Edit the yantra_wce50.bat file located in the <INSTALL_DIR>/mobileapp folder for the following changes:
 - Change all the occurrences of NETCFv2.wce4.ARMV4.cab to the Microsoft .NET Compact Framework 2.0 SP1 CAB file appropriate for your mobile terminal.
 - Change all occurrences of YantraMobileApp_WCE4.ARMV4.CAB to the Mobile Application CAB file that is appropriate for your mobile terminal.
7. Save the modified yantra_wce50.bat file under the \Flash\StartUp folder on the mobile terminal. The contents of the StartUp folder are run automatically when you perform a cold boot. Refer to the Mobile Terminal Operator's guide provided by the manufacturer to identify this folder.

This ensures that the Microsoft .NET Compact Framework and the Mobile Application are installed on cold-booting the mobile terminal.

Installing on VT220 Mobile Terminals - Overview

This section describes how to install the Mobile Application for use on VT220 mobile terminals.

The Mobile Application can be accessed from any VT220 emulation terminal.

Mobile Application requires the installation of ncurses to enable you to change the function key sequence mapping.

Installing ncurses

The VT220 client requires `infocmp` and `tic` from ncurses distribution, to prepare TERMINFO for a VT220.

Installing ncurses is a multiple step process which involves ensuring that the required build utilities are already installed. Once these utilities are installed, you can compile or build ncurses.

For more information on the ncurses and the build utility versions see the *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.

Installing libiconv on HP-UX Itanium

The libiconv libraries for HP-UX Itanium B.11.23 need to be installed for running VT220 on the HP-UX Itanium platform.

Note: Libiconv library has runtime dependencies on libgcc and gettext. Therefore, you must install these packages while installing libiconv.

Installing the Mobile Application on VT220 Mobile Terminals

About this task

The Mobile Application can be installed on the following operating systems:

- AIX
- HP-UX on Intel Itanium (ia64) Processor
- Red Hat Enterprise Linux WS release 3
- Solaris on SPARC Processor
- Solaris 10 on AMD Opteron Processor

The VT220 terminal emulation software is installed along with the Mobile Application as described in this document for the respective operating systems.

To install the Mobile Application for VT220 terminal emulation:

Procedure

1. The Mobile Application for VT220 terminal emulation is installed under the directory `<INSTALL_DIR>/mobileapp/vt220/<OS-folder-name>`; where `<OS-folder-name>` is the folder applicable for operating system, shown in the table below. The installation location is referenced as `<VT220_HOME>`.

Table 11. Operating System to OS Folder Name

Operating System	OS Folder Name
AIX	aix
HP-UX on Intel Itanium (ia64) Processor	itanium
Red Hat Enterprise Linux WS release 3	linux
Solaris on SPARC Processor	solaris

Table 11. Operating System to OS Folder Name (continued)

Operating System	OS Folder Name
Solaris 10 on AMD Opteron Processor	solaris10

- Grant execute permission to `<VT220_HOME>/yantramobileapp` and `<VT220_HOME>/keyseq`.
- Set up the VT220 emulation terminal as described in "Setting Up a Terminal for Use with the Mobile Application". Using the `<VT220_HOME>/keyseq` binary, verify that the keys F1 through F12 display the respective keys on the `keyseq` program output.

Setting Up a Terminal for Use with the Mobile Application

About this task

When setting up a terminal for use with the Mobile Application for VT220 terminal emulation, you must perform the following preliminary actions:

Procedure

- Set your environment variable `TERM` to `vt220`.
- Under the `<VT220_HOME>` directory, create a directory called `terminfo`.
- From the `terminfo` directory use the `infocmp` command to define your terminal information as:

```
$<ncurses_home>/infocmp > vt220.ti,
```

where `<ncurses_home>` is the `ncurses` binaries installation location.

Note: If an installation location is not specified during the `ncurses` installation, the `ncurses` binaries are installed in the `/usr/bin` folder.

The `infocmp` command decompiles the terminal information, and the resulting file can be edited to map the keystrokes observed by running `keyseq`.

- You must define a `TERMINFO` variable (if you do not already have one) to tell the terminal where to find information on a particular terminal type. On `BASH` type systems, this is done using the following command:

```
$ export TERMINFO=<path to some directory that contains the .ti file>
```

- Now run the `ncurses tic` command as follows to compile your newly built terminal information file:

```
$ <ncurses_home>/tic vt220.ti
```

where `<ncurses_home>` is the `ncurses` binaries installation location.

Note: If an installation location is not specified during the `ncurses` installation, the `ncurses` binaries are installed in the `/usr/local/bin` folder.

The `tic` command places the compiled version in the appropriate place under the `$TERMINFO` directory.

- To ensure that all function keys are properly mapped, use the `<VT220_HOME>/keyseq` program. This shows you what key sequence is returned when a key is pressed. Run it and press the function keys when prompted.

For example, if the F1 key is pressed and:

- Press a key (Return to end): Key Value returned: 27
- Press a key (Return to end): Key Value returned: 91
- Press a key (Return to end): Key Value returned: 49

- Press a key (Return to end): Key Value returned: 49
- Press a key (Return to end): Key Value returned: 126

is printed to STDOUT. These values are decimal values.

Convert these values to their HEX equivalents. Then, using the "Hexadecimal - character" set from the `man ascii` command on UNIX, edit the `vt220.ti` file created above. Use Table 12 to decide which values to edit.

In the example above, the F1 key maps to `kf1` (from Table 12). Therefore, you must change the value of `kf1` in the `vt220.ti` file (generated in 3 on page 109) to:

```
\E[11~
```

7. Repeat 6 on page 109 for all the keys you want to map.
8. Recompile the edited `vt220.ti` file as directed in 5 on page 109.
9. Once you have edited and compiled your terminal information file, test the changes you made by running `keyseq` again. If all keys are properly defined, `keyseq` returns a string description and the numeric value of the key.

Results

The most common keys and their codes in the terminal information file are in Table 12.

Table 12. Terminal Information - Common Keys and Codes

Code	Key
<code>kcub1</code>	Left arrow
<code>kcuf1</code>	Right arrow
<code>kcuu1</code>	Up arrow
<code>kcud1</code>	Down arrow
<code>kf1 - kf12</code>	F1 - F12 keys

Launching the Mobile Application Using a VT220 Emulation Terminal

About this task

To launch the Mobile Application using the VT220 emulation terminal and access Context-Sensitive Help:

Procedure

1. Set an environment variable `VT220_HOME` pointing to the folder containing the VT220 executable. This environment variable must be set in the shell from where the `yantramobileapp` executable is invoked.
2. To launch the application, type the following command in the operating system shell:
`yantramobileapp -i <ip_address> -p <port_number> -c <contextRoot>`. If the web context root is not passed, the Mobile Application will send the request to `"/yantra"`.

Chapter 14. Configuring Utilities

Configuring Sterling Selling and Fulfillment Foundation Utilities - Overview

Sterling Selling and Fulfillment Foundation provides script files (.sh for UNIX and .cmd for Windows) that you must customize after installation.

This section describes all the utilities supplied by Sterling Selling and Fulfillment Foundation, organized in the order in which you are likely to use them. It describes generic customizations that apply to most or all utilities. Further details specific to each utility are provided throughout the rest of this guide.

Installation Utilities

Installation utilities enable you to install Sterling Selling and Fulfillment Foundation. These utilities are present in the <INSTALL_DIR>/bin directory. Some of the utilities used for installing the various configurations of Sterling Selling and Fulfillment Foundation are "loadFactoryDefaults" and "dbverify".

loadFactoryDefaults

This utility loads the standard installation database configuration, known as the "factory defaults". For detailed information, see "Loading the Sterling Selling and Fulfillment Foundation Database Factory Defaults After Installation" on page 119.

dbverify

This utility performs database schema creation, verification, and correction. Dbverify is used to ensure database schema integrity. When run, it invokes a Java class to compare a database with the entity XMLs and generates the SQL statements that would make the database match the entity repository; it generates SQL statements for any differences between the two.

During the installation process, dbverify is used to generate SQL scripts to create the database schema or tables and indexes. These SQL statements are then run against the database, unless you choose to manually create database schemas after installation. For more information about manually creating the database schemas after installation, see "Creating Database Schemas and Loading Factory Defaults After Installation" on page 112. For detailed information about setting up and running dbverify, see "Verifying the Database Schema" on page 114.

install3rdparty

This utility copies supplied resources into the Sterling Selling and Fulfillment Foundation directory structure, and can append added jar files to the global classpath, agent classpath, or application server EAR file.

installService

This utility installs programs for specific tasks, like a regression test jar file or a fix pack jar file.

setupfiles

This utility checks the various initial product settings files (*.in) files for variables and updates the corresponding files with the values defined in the `sandbox.cfg` file to create the final files that are used by the product during runtime.

Creating Database Schemas and Loading Factory Defaults After Installation

By default, the database schemas are created and factory defaults are automatically loaded during installation. However, you can tell the installation process to skip these tasks, then perform the tasks manually after installation. To have the installation process skip creating the database schemas and loading factory defaults, do one of the following, depending on the type of installation you choose:

- If you are using the GUI Installer on UNIX/Linux or on Windows, check the option *Do you want to upgrade from a previous installation or perform new installation with no Database DDLs and Factory setup?* For more information about the GUI installation process, see “Running the Interactive GUI Installation in UNIX or Linux” on page 74 or “Running a GUI-Based Installation on Windows” on page 51.
- If you are using the text-based installation process on UNIX/Linux, answer "yes" to the question "Do you want to upgrade from a previous Installation?" For more information about the text-based installation process, see “Running the Text-Based Interactive Installation Program in UNIX or Linux” on page 77.
- If you are using the silent installation method, set the `LOAD_FACTORY_SETUP` parameter in your silent installation file to false prior to running the installation. For more information about silent installation, see “Creating a Silent Installation File - Windows” on page 54 or “Types of Installation Available on Windows” on page 50.

Updating Properties Files After Installation Procedure

After installing Sterling Selling and Fulfillment Foundation in Upgrade mode, reset the following properties in `<INSTALL_DIR>/properties/sandbox.cfg` as shown here:

- `REINIT_DB=true`
- `LOAD_FACTORY_SETUP=true`
- `NO_DBVERIFY=false`
- `DB_SCHEMA_OWNER=<YOUR_DATABASE_SCHEMA_OWNER>` (entry is required to be all upper-case)

After setting the properties, you must re-run `setupfiles.sh/cmd` from the `<INSTALL_DIR>/bin` folder.

For more information about editing the `sandbox.cfg` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Enabling the Oracle Database Text Search Feature - Overview

Sterling Selling and Fulfillment Foundation supports two types of text search indexes on Oracle databases: CTXCAT and CONTEXT. The CTXCAT index supports automatic updating of text search indexes, whereas, the CONTEXT index does not support automatic updating of text search indexes. IBM recommends that you use the CTXCAT index.

For information on how to create the text search indexes, refer to the *Sterling Selling and Fulfillment Foundation: Extending the Database*.

Enabling the Text Search Feature for CTXCAT Index

About this task

The CTXCAT index automatically updates text search indexes. Therefore, the DBA need not manually run the `EFrame_TextIndexUpdates.sql` script to update text search indexes.

To enable the text search feature on an Oracle database using the CTXCAT index:

Procedure

1. Make sure that the Oracle database is configured with the Oracle Text feature.
2. Log in to the Oracle server with a user ID having the CTXAPP privilege.
3. Verify that the text search index creation was successful.
4. Edit the `customer_overrides.properties` file that is located in the `<INSTALL_DIR>/properties` directory to add the following entries:

```
yfs.yfs.db.textsearch=Y
yfs.yfs.db.textsearch.oracle.contexttype=ctxcat
```

For additional information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Enabling the Text Search Feature for CONTEXT Index

About this task

The CONTEXT index does not automatically update text search indexes. Therefore, the DBA has to manually update text search indexes by running the `EFrame_TextIndexUpdates.sql` script.

To enable the text search feature on Oracle database using the CONTEXT index:

Procedure

1. Make sure that the Oracle database is configured with the Oracle Text feature.
2. Log in to the Oracle server with a user ID having the CTXAPP privilege.

Note: The CONTEXT type text search indexes that are created on Oracle database using the `EFrame_TextIndexAdds.sql` script are not updated automatically. The DBA has to run the `EFrame_TextIndexUpdates.sql` script to update the CONTEXT type text search indexes whenever required using scheduled jobs. The frequency of these scheduled jobs can be decided by the DBA.

3. Verify that the text search index creation was successful.
4. Edit the `customer_overrides.properties` file that is located in the `<INSTALL_DIR>/properties/` directory to add the following entries:

```
yfs.yfs.db.textsearch=Y
yfs.yfs.db.textsearch.oracle.contexttype=context
```

For additional information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Verifying the Database Schema

About this task

You can run the dbverify utility to verify the database schema as follows:

Procedure

1. (Oracle only) If you are using the Oracle database, add an ORA_TS_CONTEXT entry to the <INSTALL_DIR>/properties/sandbox.cfg file. This entry determines the text search index type for Oracle.

Assign one of the following values to ORA_TS_CONTEXT:

- CONTEXT

For fast retrieval of unstructured text.

- CTXCAT (default)

For retrieval of structured text like numbers and dates.

2. (Oracle only) If you are using the Oracle database, add an ORACLE-NLS_LENGTH_SEMANTICS entry to the sandbox.cfg file in the properties subdirectory of your installation directory. This entry determines the type of length semantic to be used for Oracle database, when using the DBVerify tool.

Assign one of the following values to ORACLE-NLS_LENGTH_SEMANTICS:

- CHAR

- BYTE (default)

Note: If the database or the specific session in which database was created has length semantic as CHAR, this property must be set to CHAR before running the DBVerify tool.

3. If you have a single-schema deployment, skip this step and go to Step 4. If you have a multischema deployment and are installing in upgrade mode, run the dbverify script from the <INSTALL_DIR>/bin/ folder and pass the multischema.xml file, which specifies database information and sets up multischema table types:

```
dbverify.sh -colonyxml <INSTALL_DIR>/multischema.xml (on UNIX  
and Linux)
```

or

```
dbverify.cmd -colonyxml <INSTALL_DIR>/multischema.xml (on  
Windows)
```

If you run the dbverify command without passing the multischema.xml file, it runs DBVerify for all colonies.

Note: If you set the multischema.applyddl property to true during installation, this enables the dbverify script to run the associated scripts automatically. If you set it to false, the dbverify script generates DDLs but does not apply them. In this case, you must run these scripts manually.

To run DBVerify on a specific colony, pass the Colony ID in the command line as follows:

```
dbverify.sh -ColonyId <Colony_Id> (on UNIX and Linux)
```

or

```
dbverify.cmd -ColonyId <Colony_Id> (on Windows)
```

4. If you are installing in single-schema mode, run the dbverify script from the <INSTALL_DIR>/bin/ folder as follows:

dbverify.sh (on UNIX and Linux)

or

dbverify.cmd (on Windows)

5. If you have enabled the text search feature and change the text search index type in Oracle from ctxcat to context or vice-versa, the updated create and drop SQL scripts can be found in the <INSTALL_DIR>/bin/EFrame_TextIndexUpdates.sql file.

In multischema deployments, the updated create and drop SQL scripts can be found in the <INSTALL_DIR>/bin/EFrame_<Pool_Id>_<TableType>.sql file, where:

- <Pool_Id> to the name of the jdbc pool for which the sql scripts should be run as specified in the jdbc.properties or multischema.xml file.
- <FileType> refers to one of a set of hardcoded names that are used to contains different types of sql statements. Possible values include:
 - UpdateQueries
 - TableChanges
 - Sequence
 - IndexDrops
 - IndexAdds
 - TextIndexDrops
 - TextIndexAdds
 - TextIndexUpdates
 - TextIndexModify

For example, the script for table alterations for the metadata pool for an Oracle install would be called: EFrame_oraclePool_TableChanges.sql

6. The differences between the entity XMLs and the database are generated in the form of SQL scripts, which can be run against the database to rectify the differences.

For example, if there is a mismatch in the size of a datatype for a column [varchar2(20) to varchar2(40)] that has an associated index, dbverify generates SQL statements for:

- Dropping the Index
- Changing the size of the datatype for the column
- Creating the new Index

The three SQL statements described in the previous list appear in different *.sql files. The appropriate *.sql files must be run in the proper order as follows:

- a. Run the <INSTALL_DIR>/bin/EFrame_IndexDrops.sql for dropping the index.
- b. Run the <INSTALL_DIR>/bin/EFrame_TableChanges.sql for altering the size of the datatype for a column.
- c. Run the <INSTALL_DIR>/bin/EFrame_IndexAdds.sql for creating a new index.

If the SQL statements are not run in the sequence as mentioned above, it results in script failure.

The scripts shown in Table 13 on page 116 are generated.

Note: All scripts listed below can be found in the <INSTALL_DIR>/bin directory.

Table 13. Generated Scripts

Single Schema Script Name	Multischema Script Name	Description of the script
EFrame_Sequence.sql	EFrame_<Pool_Id>_<TableType>_Sequence.sql	Contains all the additional sequences that need to be created.
EFrame_TableChanges.sql	EFrame_<Pool_Id>_<TableType>_TableChanges.sql	Contains all the table column differences that need to be applied on the database schema. Modify this file to reference your tablespaces.
EFrame_Drops.lst	EFrame_<Pool_Id>_<TableType>_Drops.lst	This file contains sample and/or informational changes that are not applied to the database by the entity deployer because they could cause data loss. Review the file thoroughly and take action on the entries as necessary for your environment. Note: Sterling Selling and Fulfillment Foundation does not provide a .sql file for removing tables from the database but does provide the drop statements in the .lst file. If you want to drop these tables, you must do it manually. Review the entries in the file carefully before taking action.
EFrame_IndexAdds.sql	EFrame_<Pool_Id>_<TableType>_IndexAdds.sql	Adds all of the indexes that need to be created in the database. Modify this file to reference your tablespaces.
EFrame_IndexDrops.sql	EFrame_<Pool_Id>_<TableType>_IndexDrops.sql	Removes any extra indexes in the database.
EFrame_TextIndexAdds.sql	EFrame_<Pool_Id>_<TableType>_TextIndexAdds.sql	Adds new text search indexes that need to be created in the database.
EFrame_TextIndexDrops.sql	EFrame_<Pool_Id>_<TableType>_TextIndexDrops.sql	Removes text search indexes from the database.
EFrame_TextIndexModify.sql	EFrame_<Pool_Id>_<TableType>_TextIndexModify.sql	Updates the text search indexes in the database.
EFrame_TextIndexUpdates.sql	EFrame_<Pool_Id>_<TableType>_TextIndexUpdates.sql	When executed, updates the content of the text indexes.

Table 13. Generated Scripts (continued)

Single Schema Script Name	Multischema Script Name	Description of the script
EFrame_UpdateQueries.sql	EFrame_<Pool_Id>_<TableType>_UpdateQueries.sql	For upgrades, updates the table column values in order to apply other changes made to the columns. For example, if a table column is changed from nullable to not nullable in the installation of a previous release, the column values must be updated before the column can be made not null in the current release because the column default values for the current release may contain null values.

Note: In single-schema deployments, the <INSTALL_DIR>/bin/EFrame_Drops.lst indicates extra objects in the database. In multischema deployments, this file name is <INSTALL_DIR>/bin/EFrame_<Pool_Id>_<TableType>_Drops.lst.

These extra objects could be custom objects or objects that are dropped as the result of a schema change or an upgrade. Please look through this script carefully.

This script may also contain reduced columns. These are columns that were changed to have a smaller size in the newer version. These changes are suppressed because:

- Not all databases will allow you to apply the changes.
 - Databases that do allow you to apply the changes can behave unpredictably if the table already contains values that are longer than the new length.
7. Run the scripts specified for your database type, as shown in the following topics. You must run these scripts only if you are manually creating the views after installation (REINIT_DB=no). In the normal installation mode (REINIT_DB=yes), the views will be applied automatically.

Running Oracle Scripts in a Multischema Deployment Procedure

1. Run all of the scripts within each directory for each schema:
<INSTALL_DIR>/database/oracle/scripts/CustomDBViews/<tabletype>
where <tabletype> is configuration, transaction, and master.
2. Run the configuration table type scripts for the Configuration schema, the transaction table type scripts for the Transaction schema, and the master table type scripts for the Master/Transaction schema.
3. Additionally, for a multischema deployment, run yfs_addnl_index.sql in <INSTALL_DIR>/database/oracle/scripts/CustomDBIndexes
where <tabletype> is configuration and transaction.

Running Oracle Scripts in a Single-Schema Deployment Procedure

Run the following script: <INSTALL_DIR>/database/oracle/scripts/yfs_master_db_script.sql

Running MSSQL Server Scripts in a Multischema Deployment Procedure

1. Run all of the scripts within each directory for each schema:
<INSTALL_DIR>/database/sqlserver/scripts/CustomDBViews/<tabletype>
where <tabletype> is configuration, transaction, and master.
2. Run the configuration table type scripts for the Configuration schema, the transaction table type scripts for the Transaction schema, and the master table type scripts for the Master/Transaction schema.
3. Additionally, for a multischema deployment, run the following scripts:
 - yfs_addnl_index.sql in <INSTALL_DIR>/database/sqlserver/scripts/CustomDBIndexes where <tabletype> is configuration and transaction.
 - yfs_seq_sqlserver.sql in <INSTALL_DIR>/database/sqlserver/scripts/CustomDBProcedures where <tabletype> is configuration and transaction.

Running MSSQL Server Scripts in a Single-Schema Deployment Procedure

Run the following script: <INSTALL_DIR>/database/sqlserver/scripts/yfssqlserver_master_db_script.cmd

Running DB2 Scripts in a Multischema Deployment Procedure

1. Run all of the scripts within each directory for each schema:
<INSTALL_DIR>/database/db2/scripts/CustomDBViews/<tabletype>
where <tabletype> is configuration, transaction, and master.
2. Run the configuration table type scripts for the Configuration schema, the transaction table type scripts for the Transaction schema, and the master table type scripts for the Master/Transaction schema.
3. Additionally, for a multischema deployment, run yfs_addnl_index.sql in
<INSTALL_DIR>/database/db2/scripts/CustomDBIndexes
where <tabletype> is configuration and transaction.

Running DB2 Scripts in a Single-Schema Deployment Procedure

Run the following scripts in the <INSTALL_DIR>/database/db2/scripts directory individually:

- CustomDBViews/transaction/ImportExport_View.sql
- CustomDBViews/transaction/Interop_Views.sql
- CustomDBViews/transaction/InvSnapshot_vw.sql
- CustomDBViews/transaction/yfs_cross_reference_vw.sql
- CustomDBViews/transaction/yfs_iba_ord_demand_vw.sql
- CustomDBViews/transaction/yfs_iba_resv_demand_vw.sql
- CustomDBViews/transaction/yfs_invtdmddtl_vw.sql
- CustomDBViews/transaction/yfs_noPendMove_nodeInventoryDtl_vw.sql
- CustomDBViews/transaction/yfs_onlyLPN_nodeInventoryDtl_vw.sql
- CustomDBViews/transaction/yfs_onlyLPN_noPendMove_nodeInventoryDtl_vw.sql
- CustomDBViews/transaction/yfs_order_release_line.sql
- CustomDBViews/transaction/yfs_order_release_line_vw.sql

- CustomDBViews/transaction/yfs_nodeInventoryDtl_vw.sql
- CustomDBViews/configuration/yfs_wave_item_volume_vw.sql
- CustomDBViews/master/ycm_pricelist_vw.sql
- CustomDBViews/master/ypm_category_item_vw.sql
- CustomDBViews/master/ypm_item_vw.sql

Loading the Sterling Selling and Fulfillment Foundation Database Factory Defaults After Installation

About this task

To load the Sterling Selling and Fulfillment Foundation database factory defaults after the product installation, load the defaults using the script applicable to your operating system.

Procedure

From the command line, run the `<INSTALL_DIR>/bin/loadFactoryDefaults.sh` command on UNIX and Linux or the `<INSTALL_DIR>\bin\loadFactoryDefaults.cmd` command on Windows.

Note: If the factory default installation stops before it is finished, each package under `<INSTALL_DIR>/repository/factorysetup` contains a file named "installer.xml.restart". This file records the location where the installation was stopped, and it is used the next time the factory defaults are installed.

Results

You can also generate audits when running `loadFactoryDefaults` script by overriding the value of the `AUDIT_LOAD_DEFAULTS` property and setting it to true. By default, this property is set to false. To override the value of this property, add an entry for it in the `sandbox.cfg` file. For more information about modifying properties and `sandbox.cfg` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Populating U.S. Zip Codes and Region Schemas After Installation

To make use of U.S. zip codes and region schemas for delivery plan maps and other location-dependent tasks, run the following scripts after installation:

```
<INSTALL_DIR>/database/FactorySetup/Optional/<dbtype>/RegionSchema-US/RegionSchema-US.sql
```

This script populates the `YFS_REGION` and `YFS_REGION_DETAIL` tables.

```
<INSTALL_DIR>/database/FactorySetup/Optional/<dbtype>/ZipCodeLocation/US_ZipcodeLocation.sql
```

This script populates the `YFS_ZIP_CODE_LOCATION` table.

Installing Third-Party JAR Files

You can use the `install3rdParty` utility to add third-party custom jars to the classpath of various utilities and Enterprise Archive (EAR) files.

Following is the syntax for the `install3rdParty` script:

```
./install3rdParty.sh vendorName vendorVersion <-d | -j | -l | -p | -r > filelist  
[-targetJVM EVERY | NOWHERE | DCL | APP | AGENT | [-uninstall]]
```

Here

- <vendorName> refers to the name of the vendor such as WebLogic, WebSphere, and JBoss.
- <vendorVersion> refers to the version of the vendor's product.
- [-uninstall] is used to remove a JAR from the JAR directory or the classpath files.

For example, `./install3rdParty.sh jboss 4_2_0 -j /ais_local/share/vbhat/sandbox/fairlopmain/install/jar/jboss/4_2_0/jboss-j2ee.jar -targetJVM APP -uninstall`. In this example, the `jboss-j2ee.jar` file will be removed from both the JAR directory, and the `APPDynamicClasspath.cfg` file.

Pass the appropriate argument based on the file type. You can pass the following arguments:

- -d for database jar/compressed files
- -j for jar/compressed files
- -l for shared libraries
- -p for properties files
- -r for resource properties files

<filelist> refers to the path to your custom file.

For example, to install the `wlclient.jar` to the dynamic classpath of the agent, use the following command:

```
<INSTALL_DIR>/bin/install3rdParty.sh weblogic 10 -j
<BEA_HOME>/wlserver_10.0/server/lib/wlclient.jar -targetJVM AGENT
```

This command causes the `wlclient.jar` file to be copied from the WebLogic installation location into the product installation location (`<INSTALL_DIR>/jar/welocic/10/wlclient.jar`). The utility then updates the `<INSTALL_DIR>/properties/AGENTDynamicclasspath.cfg.in` file with the new jar file and invokes the `<INSTALL_DIR>/bin/setupfiles.sh` utility to regenerate the `AGENTDynamicclasspath.cfg` file from the modified ".in" file.

If you want to make this new JAR available to the Application Server and Agents when running the `install3rdParty` utility, pass the following arguments based on your requirements:

Argument

Description

EVERY

Adds the new JAR to all the dynamic classpath files (for example, `APPDynamicclasspath.cfg`, `AGENTDynamicclasspath.cfg`, and `dynamicclasspath.cfg`).

NOWHERE

Adds the new JAR to the `<INSTALL_DIR>/jar` directory and do not want to update any of the dynamic classpath files

DCL

Adds the new JAR to the main `Dynamicclasspath.cfg` file only

APP

Adds the new JAR to the EAR file

AGENT

Adds the new JAR to the `AgentDynamicclasspath.cfg` file

Note: Sterling Selling and Fulfillment Foundation supports only the options listed above for the `install3rdParty` utility. Any other options that are displayed with the `-help` message command are not supported.

If the argument for `-targetJVM` is not specified, the new jar file is then added to the `DynamiClasspath.cfg` file.

Keep the following in mind when using the `install3rdParty` utility to update a classpath:

- The order of lines in the dynamic classpath files determine the order of the classpath for the application server or agent.
- Whatever is in the beginning of the file is analogous to the jar being in the beginning of the classpath.

For help in using `install3rdParty`, enter the command, including the `-help` option, on the command line. The `install3rdParty` utility prints a usage message.

Development Utilities - Overview

Development utilities enable you to customize Sterling Selling and Fulfillment Foundation to suit your business needs. They are for use while running Sterling Selling and Fulfillment Foundation in development mode.

About the Configuration Deployment Tool

The Configuration Deployment Tool enables you to migrate configuration data from your development environment to your production environment.

For more information about the configuration deployment tool, see the *Sterling Selling and Fulfillment Foundation: Configuration Deployment Tool Guide*.

Truncating Transaction Data

About this task

When deploying Sterling Selling and Fulfillment Foundation to a production environment, you may not want to include all of your transaction data. Sterling Selling and Fulfillment Foundation provides a utility through which you can generate a script to remove transaction data prior to moving into your production environment.

To truncate transaction data:

Procedure

1. From the `<INSTALL_DIR>/bin` directory use the following command appropriate for your database:
For Oracle and SQL Server:

```
./sci_ant.sh(cmd) -f generateTruncateTransactionData.xml
```


For DB2:

```
./sci_ant.sh(cmd) -Ddbtype=DB2 -f generateTruncateTransactionData.xml
```
2. The `TruncateTransactionTables.sql` script is generated and placed in the current directory.
3. To truncate your transaction data, run the newly generated `TruncateTransactionTables.sql` script against your database.

Run-Time Utilities

These utilities start processes that run in the background. The setup of these utilities is described in detail in the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Integration Server

An integration Server is a process that manages asynchronous services, such as messages to and from external systems. You can run the integration server using the `<INSTALL_DIR>/bin/startIntegrationServer` script.

The Sterling Selling and Fulfillment Foundation Integration Server allows Sterling Selling and Fulfillment Foundation to collaborate with different systems, organizations, and businesses all through a standard, uniform interface to all systems. The Sterling Selling and Fulfillment Foundation Integration Server runs in its own Java Virtual Machine (JVM) environment, separate from your application server.

Agent Server

The agent server utility starts processes responsible for processing transactions generated by the time-triggered transactions (agents). You can start multiple instances of an agent server using the `<INSTALL_DIR>/bin/agentserver.sh <server_name>` script as many times as needed.

Note: If you configure agents to use TIBCO JMS, ensure the prefetch parameter for queues is set to 1.

Note: If you configure agents or integration servers to use JBoss Messaging, ensure to uncomment the following "Attribute" xml element in `remoting-bisocket-service.xml` under

```
<PATH>/jboss-eap-5.1/jboss-as/server/all/deploy/messaging directory:  
<attribute name="pingFrequency" isParam="true">214748364</attribute>
```

.

Trigger Agent

The trigger agent utility is used for scheduling time-triggered transactions.

You can override the agent criteria attributes only in the Real-time Availability Monitor and Inventory Monitor. The command for triggering the Real-time Availability Monitor and Inventory Monitor with override abilities is:

```
triggeragent.sh <criteriaID> -<AgentCriteriaAttribute> <OverriddenValue> (or  
.cmd on Windows)
```

To enable this override, you should pass the `AgentCriteriaAttribute` and `OverriddenValue` as additional parameters to the java class in the `triggeragent.sh` (or `.cmd` on Windows) file as follows:

```
java com.yantra.ycp.agent.server.YCPAgentTrigger -criteria %*
```

Therefore, when you invoke:

```
triggerAgent.sh CustomCriteria -MyOverriddenParam DynamicValue
```

all the values are passed to the java class.

However, do not modify the parameters passed to the java class in the default `triggeragent.sh` (or `.cmd` on Windows) file. Make these changes in the file that you have copied and renamed from the `triggeragent.sh` (or `.cmd` on Windows). Also, the agent criteria XML code must have the `AllowedOverriddenCriteria` flag set to `Y`.

sender.sh Utility

The `sender.sh` utility invokes `TestClientSender` and is used for testing. It enables you to execute an API or service from the command line, as follows:

```
java
com.yantra.integration.adapter.client.TestClientSender<flowName
e/systemApiName> <is firstParameter Flow (Y/N)> <xmlFileName>
```

Here,

- The first argument takes the name of a service/sdf (*flowName*) or an API name (*systemApiName*).
- The second argument (*is firstParameterFlow (Y/N)*) determines whether the first argument is a service or an API. Valid values are `Y` and `N`. If the first argument is a service, use `Y`; if it is an API, use `N`.
- The third argument (*xmlFileName*) takes the path and name of the XML input file you want to use as input to the API or service.

Setting Up the Runtime Utilities

You can use WebLogic, WebSphere, JBoss, and TIBCO for the Java Messaging Service (JMS).

The `CLASSPATH` for the `startIntegrationServer`, `agentServer` and `triggerAgent` scripts must include certain jar files in order for them to be used on WebLogic, WebSphere MQ, JBoss, or TIBCO. Use the `<INSTALL_DIR>/bin/install3rdparty` script to include these respective jar files - as they are listed in this section - in the `AGENTDynamicclasspath.cfg` dynamic classpath file.

The JDK used by the Runtime Utilities is determined by the `AGENT_JAVA_HOME` property in `<INSTALL_DIR>/properties/sandbox.cfg`. This JDK should point to the same JDK that is used to run your application server. For more information about configuring `sandbox.cfg`, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

If you have developed custom Java classes (user exits, event handlers, and so forth), see the section on "Including Custom Classes" for your application server in Chapter 15, "Deploying Sterling Selling and Fulfillment Foundation," on page 127.

Setting Up Runtime Utilities on Oracle WebLogic About this task

Note: The classpath information provided in this section is intended as a starting point for your implementation. Depending upon the JMS, JNDI, and JDK you are using you may need to add or remove jars from the lists given here.

Include the following jar files for WebLogic JMS:

Procedure

Use the <INSTALL_DIR>/bin/install3rdparty.sh (or .cmd) script to install the wlfullclient.jar file and include it in the AGENTDynamicclasspath.cfg dynamic classpath file.

Results

For more information about developing a WebLogic Full Client, see:

http://download.oracle.com/docs/cd/E12840_01/wls/docs103/client/jarbuilder.html

Note: For more information about using the install3rdparty script, see “Installing Third-Party JAR Files” on page 119.

If you are using WebLogic Server with WebSphere MQ JMS and you are binding queues in WebLogic JNDI, use install3rdparty to include the following files from MQ_HOME/java/lib in the AGENTDynamicclasspath or the APPDynamicclasspath:

- com.ibm.mq.jar
- com.ibm.mqjms.jar
- connector.jar
- jms.jar
- jta.jar
- wlfullclient.jar
- dhbcore.jar

Note: If you are using WebLogic JMS 10.0 or later, refer to the Web site http://download.oracle.com/docs/cd/E12840_01/wls/docs103/client/jarbuilder.html and add wlfullclient.jar to the AGENTDynamicclasspath.cfg, using the install3rdparty.sh (or .cmd) script.

If you are binding queues in File Bindings JNDI, use the following files, obtainable from MQ_HOME/java/lib:

- com.ibm.mq.jar
- com.ibm.mqjms.jar
- connector.jar
- jms.jar
- jta.jar
- fscontext.jar
- providerutil.jar
- dhbcore.jar

About Setting Up Runtime Utilities on IBM WebSphere

Note: The classpath information provided in this section is intended as a starting point for your implementation. Depending upon the JMS, JNDI, and JDK you are using you may need to add or remove jars from the lists given here.

Both WebSphere and MQ have jars which are required for running the agent and integration servers. You can obtain these jars from the WebSphere or MQ server.

IBM WebSphere 7.0

If you are using WebSphere Default Messaging you need to include the following jars:

- <WAS_HOME>/lib/j2ee.jar
- <WAS_HOME>/runtimes/com.ibm.ws.sib.client.thin.jms_7.0.0.jar
- <WAS_HOME>/runtimes/com.ibm.ws.ejb.thinclient_7.0.0.jar
- <WAS_HOME>/runtimes/com.ibm.ws.orb_7.0.0.jar
- <WAS_HOME>/plugins/com.ibm.ws.wccm.jar

Note: If you are using WebSphere version 7.0.0.7 or higher, set the following property to true in the WebSphere console under Application Servers > *your server name* > web container > custom properties:

com.ibm.ws.webcontainer.dispatcherrethrower=true

Specify this property in all lowercase.

IBM WebSphere MQ (7.0) Using "fscontext" jndi

When using WebSphere MQ you should obtain the following jars from MQ_HOME/java/lib:

- com.ibm.mq.jar
- com.ibm.mqjms.jar
- fscontext.jar
- dhbcore.jar
- jms.jar
- jta.jar
- providerutil.jar

About Setting Up Runtime Utilities on JBoss

Note: The classpath information provided in this section is intended as a starting point for your implementation. Depending upon the JMS, JNDI, and JDK you are using you may need to add or remove jars from the lists given here.

If you are using JBoss JMS, add the following jars in the classpath using the <INSTALL_DIR>/bin/install3rdparty script:

- javassist.jar
- jbossall-client.jar
- jboss-aop-jdk50-client.jar
- jboss-aop-jdk50.jar
- jboss-messaging-client.jar
- log4j.jar
- trove.jar

Do not include any *ui.jar files.

- If you are using the JBoss Messaging, add the jbossmq-client.jar in the classpath using the <INSTALL_DIR>/bin/install3rdparty script.
- If you are using JBoss application server, add the log4j.jar file from JBoss at the beginning of your CLASSPATH.

- If you are invoking a JSP page for the first time, there may be a short delay while the JBoss application server compiles the JSP page. To avoid this delay, precompile the JSP files before creating the smcfs.ear file.
- For information about precompiling JSP files, see the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

About Setting Up Runtime Utilities on JBoss EAP 5.1

Note: The classpath information provided in this section is intended as a starting point for your implementation. Depending upon the JMS, JNDI, and JDK you are using you may need to add or remove jars from the lists given here.

If you are using JBossEAP5.1 JMS, add the following jars in the classpath using the <INSTALL_DIR>/bin/install3rdparty script:

- javassist.jar
- jboss-aop-client.jar
- jboss-common-core.jar
- jboss-logging-log4j.jar
- jboss-logging-spi.jar
- jboss-mdr.jar
- jboss-remoting.jar
- jnp-client.jar
- trove.jar
- jboss-serialization.jar
- jboss-messaging-client.jar

Do not include any *ui.jar files.

- If you are invoking a JSP page for the first time, there may be a short delay while the JBoss application server compiles the JSP page. To avoid this delay, precompile the JSP files before creating the smcfs.ear file.
- For information about precompiling JSP files, see the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

About Setting Up Runtime Utilities on TIBCO

Note: The classpath information provided in this section is intended as a starting point for your implementation. Depending upon the JMS, JNDI, and JDK you are using you may need to add or remove jars from the lists given here.

If you are using TIBCO JMS:

- Use the <INSTALL_DIR>/bin/install3rdparty.sh(cmd)script to install the jms.jar and tibjms.jar files and include them in the AGENTDynamicclasspath.cfg dynamic classpath file.
- Some agents will require servlet classes. For these agents, use the install3rdParty script to install the <INSTALL_DIR>/jar/geronimo/2_2/geronimo-servlet_2.5_spec-1.2.jar into the AGENTDynamicclasspath.cfg file.

Chapter 15. Deploying Sterling Selling and Fulfillment Foundation

Sterling Selling and Fulfillment Foundation Deployment - Overview

After configuring Sterling Selling and Fulfillment Foundation according to your business needs, deploy it into production based on your application server. Deployment is part of the general path that you follow when installing and deploying Sterling Selling and Fulfillment Foundation:

1. Installing the application server (JBoss, WebLogic, or WebSphere). Refer to the documentation for the application server.
2. Installing Sterling Selling and Fulfillment Foundation. Refer to the installation information for the operating system (UNIX/Linux or Windows).
3. Building the Enterprise Archive (EAR).
4. Starting the application server.
5. Deploying Sterling Selling and Fulfillment Foundation.

If you need to deploy Sterling Selling and Fulfillment Foundation in a development environment using exploded (non-ear) mode, see the *Sterling Selling and Fulfillment Foundation: Customizing Console JSP Interface for End-User* .

Before deployment, verify if you have applied all the concepts that pertain to your environment, and have completed the Performance Recommendations Checklist as described in the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

Tip: To enable faster loading of a JSP page, pre-compile your JSP files. For information on how to do this, see the JSP Pre-compilation section of the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

Note: If you are planning on installing any of the Sterling Selling and Fulfillment Foundation Packaged Composite Application (PCAs), or applying any extensions, you may want to consider delaying the building of your Enterprise Archive (EAR) until all of your PCAs are installed. Building the EAR now and for each PCA or extension installation does not cause harm, but does save time if you build your EAR only once after all PCAs or extensions are installed.

Configuring an Oracle WebLogic Application Server - Overview

This section includes the tasks necessary for configuring an Oracle WebLogic application server for use with Sterling Selling and Fulfillment Foundation.

Setting Up the WebLogic Script File

About this task

If you are using HP-UX 11iv3, verify that your kernel parameters are set according to Oracle's recommendations before you set up the WebLogic application server. For these recommendations, go to: http://download.oracle.com/docs/cd/E13196_01/platform/suppconfigs/configs/hpux/hpux_11iv3_103.html

To set up the WebLogic script file:

Procedure

1. Add the following properties to the startWebLogic.sh (or .cmd) file supplied by Oracle. Each property and its proper syntax are described in Table 14.

Table 14. startWebLogic.sh Properties

Property	Description
JAVA_OPTIONS	Java command line options for running the server. Depending on your JVM vendor, specify as follows: <ul style="list-style-type: none">• For IBM, set this value to <code>-Xms768m -Xmx768m</code>• For JRockit, set this value to <code>-Xms768m -Xmx768m</code>• For HP, set this value to <code>-XX:MaxPermSize=512m -Xms768m -Xmx768m</code> For Sun, set this value to <code>-XX:MaxPermSize=512m -Xms768m -Xmx768m</code>
DBDRIVERS	Specify the paths to your data base drivers as the first item in the value of the CLASSPATH. The out-of-the-box CLASSPATH setting is: <code>CLASSPATH="{CLASSPATH}{CLASSPATHSEP}{MEDREC_WEBLOGIC_CLASSPATH}"</code> Change this so that the path to the drivers .jars is first. For example: <code>DBDRIVERS=/<code><directory_path_to_oracle_drivers>/ojdbc6.jar</code> <code>CLASSPATH="{DBDRIVERS}{CLASSPATHSEP}{CLASSPATH}{CLASSPATHSEP}{MEDREC_WEBLOGIC_CLASSPATH}"</code></code>
JITC_COMPILEOPTS	For AIX, specify as "NQCLSINIT"
-Dfile.encoding	To ensure that all the Sterling Selling and Fulfillment Foundation UI screens display UTF-8 characters, specify as follows for java commands: <code>-Dfile.encoding=UTF-8</code> This is applicable to all the Sterling Selling and Fulfillment Foundation Java start-up scripts.
-Dvendor	System property. Specify as an argument to the java command. Can be "shell" or "weblogic" depending upon whether datasource is being used or not. <code>-Dvendor=shell</code>
-DvendorFile	System property. Specify as an argument to the java command. <code>-DvendorFile=/servers.properties</code>

2. If you are using an HTTPS transport, download the Secure Socket Extension (JSSE) 1.0.3 package from <http://java.sun.com> and add the following files to the `<JAVA_HOME>/jre/lib/ext/n/` directory:
 - jnet.jar
 - jcert.jar
 - jsse.jar

Configuring Oracle WebLogic XML Registry

About this task

You must configure WebLogic to run properly with Sterling Selling and Fulfillment Foundation.

To configure WebLogic:

Procedure

1. From the WebLogic Console menu, choose **Services** → **XML Registries**.
2. Click **New**.

Note: If the XMLs have UTF-8 specified as encoding in the header, then you do not need to specify any XML registry.

3. Click **Next**. Select the WebLogic application server or cluster to which you would like to deploy this XML Registry.
4. Click **Finish**.

Disabling Instrumented Stack Traces in WebLogic

About this task

You can eliminate additional stack traces resulting from an error on an API call in EJB mode.

To eliminate stack traces:

Procedure

1. From the WebLogic System Administration Console, select each server on which Sterling Selling and Fulfillment Foundation is deployed.
2. Select **Logging**.
3. Uncheck the checkbox for **Instrument Stack Traces** and choose **Apply**.

Setting Up WebLogic to Display Barcodes and Graphs

About this task

Sterling Selling and Fulfillment Foundation uses X Window functionality to display barcodes and dynamic graphical images (such as inventory supply and demand graphs) in a UNIX environment.

The following configuration is required to enable the X Window environment in UNIX systems for a WebLogic application server:

Procedure

1. If your UNIX server is also an X Window client, edit the `startWebLogic.sh` script, and set the `DISPLAY` environment variable as follows:

```
export DISPLAY=IP_address_of_XWindows_server:0.0.
```
2. If you are using UNIX, run the `xhost +` command to remove access control for your X Window server.

You can run X server on the same server on which you run Sterling Selling and Fulfillment Foundation. However, you need to be logged into the server console.

Note: If the X Window server goes down or crashes while the inventory user interface is using the `jbchartx.jar` file, the WebLogic server also goes down.

Setting Up WebLogic to Use HTTP In-Memory Session Replication

About this task

Sterling Selling and Fulfillment Foundation supports HTTP in-memory session replication on the following configuration:

Apache 2.0.44 with the WebLogic plug-in as the proxy server with idempotent set to OFF

We advise testing session replication if you are using a different proxy.

The `weblogic.xml` file should be edited to set up WebLogic for in-memory session replication as follows:

Procedure

1. Build the EAR file.
2. Copy the `<INSTALL_DIR>/tmp/build<package_name>/WEB-INF/weblogic.xml` file to the `<INSTALL_DIR>/extensions/<package_name>` directory, where `<package_name>` is the application name; for example, `smcfs`.
3. Add the following lines to the `weblogic.xml` file:

```
<session-descriptor>
  <session-param>
    <param-name>PersistentStoreType</param-name>
    <param-value>replicated</param-value>
  </session-param>
</session-descriptor>
```
4. Rebuild the EAR file.

Building the Enterprise Archive (EAR) Package on WebLogic

Note: Sterling Selling and Fulfillment Foundation supports overriding the context root during EAR deployment.

When deploying Sterling Selling and Fulfillment Foundation on WebLogic, use the `smcfs.ear` file, which may contain:

- `smcfs.war` - Web module that contains all of the Sterling Selling and Fulfillment Foundation JSPs and other Web application components.
- `sma.war` - Web module that contains the System Management Administrator application components.
- `sbc.war` - Web module that contains all of the IBM Sterling Business Center Web application components.
- `yantrawebservices.war` - Web module that contains all of the Sterling Selling and Fulfillment Foundation Web services interface classes.
- `smcfsejb.jar` - The EJB module that contains all the Sterling Selling and Fulfillment Foundation EJBs. You can pass an alternate earfile name by using the `-Dearfile` option to the `buildear.sh` script. Doing this will result in a name change for the `ejb` jar file. For example, if you specify an EAR file as `xyz.ear`, the `ejb` jar becomes `xyzejb.jar`.
- `smcfswsbe.jar` - The backend Web services jar file. You get this file if you expose Web services. You can pass an alternate earfile name to the `ear build` script. Doing this will result in a name change for the Web

services backend jar file. For example, if you specify an EAR file as xyz.ear, the Web service backend jar becomes xyzwsbe.jar.

- Jars that contain backend business logic.
- Jars that contain third-party libraries accessed by backend logic.

Each of the third-party JAR files are left as is and in the manifest of the application each file is indicated as a dependency. For example, log4j files are represented separately as log4j-1.2.15.jar with a dependency in the application.

Deploying Context-Sensitive Help on WebLogic

The Sterling Selling and Fulfillment Foundation Context-Sensitive Help is built as a separate EAR file called smcfsdocs.ear when you build the smcfs.ear file unless you specify "-Dnodocear=Y" during the build. You can also build the doc ear separately by running the following command:

```
<INSTALL_DIR>/bin/buildear.sh create-doc-ear -Dsupport.multi.war=true  
-Dappserver=weblogic -Dwarfiles=smcfs,sma,svc -Dearfile=smcfs.ear
```

To make use of the Context-Sensitive Help files associated with Sterling Selling and Fulfillment Foundation, deploy the smcfsdocs.ear file in all of the same locations where you deploy the smcfs.ear file.

Note: By default, for the Context-Sensitive Help to be displayed, the Sterling Selling and Fulfillment Foundation application sends the corresponding request to:

```
http://<Host name>:<Port Number>/smcfsdocs/yfscommon/online_help/en_US/  
wwhelp/wwhimpl/common/html/wwhelp.htm
```

However, if the Sterling Selling and Fulfillment Foundation application is deployed in a context root other than smcfs, for example, sterling, set the yfs.onlinehelp.path property with the context root information in the yfs.properties file, for example, set yfs.onlinehelp.path=/sterlingdocs/yfscommon/online_help. If your installation locale code is not en_US, set yfs.onlinehelp.path.overrideforlocale.<your_local_code>=/sterlingdocs/yfscommon/online_help/<your_local_code>.

For information about the full Product Documentation Library, see "Documentation Library" on page 2.

Note: You must define the scui-help-url context parameter in web.xml to deploy the context-sensitive help in IBM Sterling Business Center. The URL context parameter value must follow the following format:

```
http(s)://<server>:<port>/<context>/<online help properties>
```

Building Web Services on WebLogic

When you create the EAR file for the application, web services are also created. There are two types of web services that can be created:

- EJB: The EJB web service accepts two string inputs and returns one string output. No information about the content of the strings is included in the WSDL for this web service; the caller must open the javadocs for the API to get the information about how to structure the content. This makes it very difficult to dynamically generate calls to the XAPI web services without an additional source of information beyond the WSDL. This type of web service is created using EJB beans, and is created by default with the application EAR.

- JAX-WS: The WSDL of the JAX-WS web service includes information about the input expected by the server, the output, and exceptions, which makes it easier to dynamically generate calls to the web services. JAX-WS web services are created using JAXB beans. Instead of being wrappers around EJB beans, these are wrappers around JAXB beans.

You can choose to build the application with one or both types of web services. Both types of web services require some configuration prior to running the build script to create the application EAR and web services WAR file.

Note: You can also choose to suppress the web services build by using the `-Dnowebservice=true` option on the build script command.

If you are planning to run the application components as web services, additional setup is required. The setup takes place in either the `namedwebservices.xml` file or `webservicebeans.xml` file. You can expose APIs as web services. You can also expose synchronous SDF services through web services.

Note: If API security is enabled, ensure that you expose the Login API.

Defining an EJB Web Service with WebLogic Procedure

1. In the `<INSTALL_DIR>/repository/eardata/platform/webservices` folder, locate the `namedwebservices.xml.sample` file. Rename (or copy) it to `namedwebservices.xml`. This is the file you will edit.
2. In `namedwebservices.xml`, specify each API you want to expose as a web service in an `Api/Name` attribute.

Property

Description

ServiceName

The name of the service that you configured using the Service Definition Framework (SDF).

ExposedName

The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the web service programmatically. When specifying a service name for `ExposedName`, choose a literal that does not match any of the standard application API names.

The exposed name must start with a lower case letter.

3. Set the properties necessary for your application server in `sandbox.cfg`:

Parameter

Description

BEA_DIR

Set the value to specify the `<WL_HOME>` directory. (You may need to create the property.)

Required if building EJB web services.

For WebLogic only.

WEBSERVICES_BUILDS

Takes a comma-separated list that can include `YIFWebService` and `SIXBeanXapiJaxWS`. Defaults to `YIFWebService`.

- Required for EJB and JAX-WS web services, on all application servers.
4. Save the file, navigate to the bin directory, and run the setupfiles command:
 - For UNIX/Linux: <INSTALL_DIR>/bin/setupfiles.sh
 - For Windows: <INSTALL_DIR>\bin\setupfiles.cmd
 5. Create the EAR as described in “Creating the EAR on WebLogic” on page 139. The web services defined in the file will be built when you create the application EAR.
 6. Later, if you want to add more APIs and services as EJB Web services, repeat these steps again.

Defining a JAX-WS Web Service on WebLogic

About this task

You can expose APIs as web services. Before you create the application EAR file, you need to define which APIs will be exposed as web services so that they are included in the web services WAR file in the EAR. This section provides a high-level overview of how to configure a JAX-WS web service.

The following APIs are not supported with JAX-WS web services:

- createAsyncRequest
- evaluateAdvancedCondition
- evaluateCondition
- getAgentCriteriaList
- getPage
- invokeUE
- multiApi
- printDocumentSet

Procedure

1. Set the properties necessary for your application server in sandbox.cfg:

Parameter

Description

WEBSERVICES_BUILDS

Can take a comma separated list that can include YIFWebService and SIXBeanXapiJaxWS.

Required for EJB and JAX-WS web services, on all application servers.

XBEAN_PACKAGE

By default, the xbean package names and namespace generated for JAX-WS web services include the word "documentation." Use this parameter to replace the word "documentation" with another string.

This value can be any string which would form a valid java package name. Optional.

JAXB_LOCAL_SCOPING

Valid values are true/false. Set to true.

This flag affects the way that jaxb generates beans. Everything becomes “toplevel”, so there will be no nested inner classes. If not used, results in class file names that are too long for the file system.

Required when building JAX-WS web services.

JAXB_ALWAYS_ANNOTATE_CLASSNAMES

The classes that get generated by JAXB get “1-up” numbers at the end of them to make them distinct from other classes that would otherwise have the same name.

This can be done on a case by case basis using the `webservicebeans.xml`, or globally using this flag.

If you set this flag to true, all classes get annotated. Or, to minimize the user of one-up numbers, specify the `AnnotateClassNames` attribute on the `webservicebeans.xml` as shown in the example below to indicate which apis need to have annotations.

```
<Bean BeanName="ParticipantBean"
BeanPackage="com.sterlingcommerce.jaxws.participant.
webservices" >
    <Apis>
    <Api AnnotateClassNames="true"
Name="getOrganizationHierarchy"
ExposedName="getOrganizationHierarchy" />

    <Api AnnotateClassNames="true"
Name="getOrganizationList"
ExposedName="getOrganizationList" />

    <Api AnnotateClassNames="false"
Name="getPersonInfoList"
ExposedName="getPersonInfoList" />

    <Api AnnotateClassNames="false" Name="login"
ExposedName="login" />
    </Apis>
</Bean>
```

Valid values for this flag are true/false. It is recommended that you always include this flag and set it to true.

Required when building JAX-WS web services.

2. Save the file, navigate to the bin directory, and run the `setupfiles` command:
 - For UNIX/Linux: `<INSTALL_DIR>/bin/setupfiles.sh`
 - For Windows: `<INSTALL_DIR>\bin\setupfiles.cmd`
3. In the `<INSTALL_DIR>/repository/eardata/platform/webservices` folder, locate the `webservicebeans.xml.sample` file. Copy the file to `webservicebeans.xml`. This is the file you will edit.

webservicebeans.xml attributes	Description
Bean attributes Note: Do not put the same API into two beans.	
BeanName	Enter a descriptive name for the web service. This name is used by the WSDL generator and reflected in the client code. Required. For example, a company named "Dave's BBQ" might use the following: BeanName=DavesBBQJaxWS

webservicebeans.xml attributes	Description
BeanPackage	Enter a descriptive name for the package. Each bean must have a unique package name. This name is used by the WSDL generator and reflected in the client code. Required. For example, a company named "Dave's BBQ" might use the following: BeanPackage="com.DavesBBQ"
API element attributes	
Api Name	The name of the API that you configured using the Service Builder.
ExposedName	The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the web service programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard application API names. The exposed name must start with a lower case letter.
AnnotateClassNames	Specify the AnnotateClassNames attribute on the webservicebeans.xml to indicate which apis need to have annotations.
Service element attributes	
Service Name	The name of the service that you configured using the Service Builder.
ExposedName	The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the web service programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard application service names. The exposed name must start with a lower case letter. The exposed name must start with a lower case letter.
AnnotateClassName	Specify the AnnotateClassNames attribute on the webservicebeans.xml to indicate which apis need to have annotations.
To expose an SDF service as a web service, you must also complete the procedure in "Exposing an SDF Service as a Web Service on WebLogic" on page 136.	

4. Save the file.
5. Create the EAR as described in "Creating the EAR on WebLogic" on page 139. The web services defined in the file will be built when you create the application EAR.

CAUTION: When building an ear with JAX-WS Web services, you may encounter out of memory errors. JAX-WS Web services will generate a java class for each element in the XSD, which can exceed the system memory. If the ear build runs out of memory, then unnecessary APIs should be removed from the webservicebeans.xml file. Additionally, APIs with potentially large output XMLs may be exposed through services with reduced XSDs to eliminate unwanted elements.

Tip: To help avoid out of memory issues when building the ear, group APIs into separate beans instead of exposing a large number of APIs in a single bean.

6. Later, if you want to add more APIs and services as JAX-WS web services, repeat these steps again.

Example of webservicebeans.xml file - WebLogic

The following is an example of a webservicebeans.xml file:

```
<WebServices>
  <Beans>
    <Bean BeanName="TheBeans"
BeanPackage="com.sterlingcommerce.jaxws.the.webservices" >
      <Apis>
        <Api Name="login" ExposedName="theLogin" />
        <Api Name="getLocaleList" ExposedName="theGetLocaleList" />
      </Apis>
    </Bean>
    <Bean BeanName="OtherBeans"
BeanPackage="com.sterlingcommerce.jaxws.other.webservices" >
      <Apis>
        <Api Name="getLocaleList" ExposedName="otherGetLocaleList" />
      </Apis>
      <Services>
        <Service AnnotateClassNames="true" Name="testWS" ExposedName="testWS" />
        <Service Name="testWS2" ExposedName="testWS2" />
      </Services>
    </Bean>
  </Beans>
</WebServices>
```

Exposing an SDF Service as a Web Service on WebLogic

About this task

To expose an SDF service as a web service, you first have to copy the input and output xsds for the API that is exposed in the SDF service and edit them.

Note: If you want to overwrite the input or output XSD for any API, you must expose the API as an SDF service and then expose the SDF service as a Web service.

Procedure

1. Open the SDF and create a synchronous service that has an API node.
2. Select the desired API.
3. Save the SDF service.
4. Copy the input and output XSD files for the API you exposed in the SDF from the following location:

```
<INSTALL_DIR>/xapidocs/api_javadocs/XSD
```

to:

```
<INSTALL_DIR>/extensions/webservices
```

5. Rename the files to match the name and exposed name that you give them in webservicebeans.xml.

Use the following format:

```
<name>_<exposed_name>_input.xsd
```

```
<name>_<exposed_name>_output.xsd
```


Results

This is also the procedure you would use to extend a table, with this additional step: after copying and renaming the files, edit them to reference the new columns/API inputs/outputs.

Configuring a JAX-WS Web Services Client (UNIX/Linux Only) on WebLogic

About this task

Sterling Selling and Fulfillment Foundation provides a client generator for use with JAX-WS web services.

CAUTION:

The client generator is supported on UNIX and Linux only. It is not supported on Windows.

For WebLogic, the client generator accesses the WSDL directly from the server.

Note: If you are modifying the XSD for any API or service which has been exposed as a Web service, you must rebuild the Web services client. Additionally, each time you extend the database through customization or upgrade, you must run the `xsdgenerator`, and then rebuild the ear and client. See *Sterling Selling and Fulfillment Foundation: Customizing APIs* for more information about using the `xsdgenerator`, and see *Sterling Selling and Fulfillment Foundation: Extending the Database* for more information about extending the database. See “Building the Enterprise Archive (EAR) Package on WebLogic” on page 130 for information about building the ear.

Running the JAX-WS Client Generator on WebLogic Procedure

To invoke the generator, run the `buildjaxclient.sh` script from the `<INSTALL_DIR>/bin` directory. Include the properties required for your application server type from the following list when running the script:

Property

Description

-DBEAN_NAME

Required for all application servers. Matches the beanname on the server. Note that the name of the service might be different, depending on the application server being used.

-DAPPSERVER

Required. Valid values are `websphere`, `weblogic`, or `jboss`.

-DSERVER_URL

Required for all application servers. The http URL of the server up to, but not including, the bean name.

The following is an example of a command for `WebLogic:bin/buildjaxclient.sh`

```
-DBEAN_NAME=ParticipantBeanService -DAPPSERVER=weblogic  
-DSERVER_URL=http://00.00.00.00:1975/SIXBeanXapiJaxWS
```

Results

Running the script creates a new folder, `jaxwsclient`, that contains the compiled classes.

Sample Code for BeanService Classes - WebLogic

Sample code is generated in the same directory as the bean and BeanService class. You can use this sample as a reference for invoking the bean methods.

The sample file has the same name as the service class but with "Sample" appended to the end of the name.

For example, `SMCFSBeanService` would have an accompanying sample called `SMCFSBeanServiceSample` class.

```
jaxwsclient/output/ParticipantBeanService/com/sterlingcommerce  
/jaxws/participant/webservices/ParticipantBeanServiceSample
```

The class generates a method for each API you expose. The content will require editing.

```
public void testlogin( ) throws Exception {  
    com.sterlingcommerce.documentation.ycp.login.input.Login input  
    = new com.sterlingcommerce.documentation.ycp.login.input.Login();  
    /*  
    Insert custom code here to set values on the input object.  
    */  
    com.sterlingcommerce.documentation.ycp.login.output.Login  
    returnValue = b.login(env,input);  
    /*  
    Insert custom code here to retrieve values from the return object.  
    */  
}
```

Including Custom Classes on WebLogic

About this task

When deploying Sterling Selling and Fulfillment Foundation as Web services on WebLogic, if you have developed custom Java classes (user exits, event handlers, and so forth) you need to deploy them in order for them to be available.

To ensure that your custom classes get invoked, do the following:

Procedure

1. Create a JAR file with all your custom classes.
2. Place this JAR file in a folder structure based on the package name. For more information about packaging and deploying jar files, see the *Sterling Selling and Fulfillment Foundation: Customization Basics*.
3. Rebuild the EAR as described in "Creating the EAR on WebLogic" on page 139.

Results

The custom classes are automatically included in the `smcfs.ear` file.

Creating the EAR on WebLogic

About this task

Note: Set the number of file descriptors (`ulimit -n`) for the user creating the EAR to be greater than 8192. If you are deploying on HP set `ulimit unlimited` for the user creating the EAR.

Enterprise Archives are built using an ANT (`buildEAR.xml`) that accepts the following targets:

Main Target

Description

`create-ear`

Creates the EAR files:

- `smcfs.ear` - The Sterling Selling and Fulfillment Foundation application EAR file
- `smcfsdocs.ear` - The Sterling Selling and Fulfillment Foundation documentation EAR file

`create-doc-ear`

Creates `smcfsdocs.ear` - The Sterling Selling and Fulfillment Foundation documentation EAR

Procedure

To create an application EAR file, run the following command from the `<INSTALL_DIR>/bin` directory:

```
./buildear.sh (.cmd for Windows) -Dappserver=weblogic -Dwarfiles=smcfs,sma,svc  
-Dearfile=smcfs.ear create-ear
```

Note: Sterling Selling and Fulfillment Foundation supports the RPC encoded or document literal style and usage of invocation for Web services. When choosing style and usage for WebLogic, the Web service uses the document literal only. Running this command creates the `smcfs.ear` and `smcfsdocs.ear` files in the `<INSTALL_DIR>/external_deployments/` directory. It also puts three war files into the `smcfs.ear`:

- `smcfs.war` - The Sterling Selling and Fulfillment Foundation application war file
- `sma.war` - The System Management Administrator application war file
- `svc.war` - The IBM Sterling Business Center application war file

Note: You can add the following options to the end of the above `buildear` commands:

- `-Dnowebservice=true` If you do not want to use Web services. If you do want to use Web services, see the “Building Web Services on WebLogic” on page 131.
- `-Ddevmode=true` if you want to use the HTTP API Tester in the development environment.
- `-Dnodocear=true` if you want to skip the documentation build.
- `-Dwls-10=true` if you are creating EAR for WebLogic 10
- `-Dsupport.multi.war=true` if you want to copy all the UI jars to `<WAR>/WEB-INF/lib`. The UI jars will be copied based on the entry in `DCL.xml`.

Results

For more information about the System Management Administrator (SMA) see the *Sterling Selling and Fulfillment Foundation: System Management and Administration Guide*. For more information about IBM Sterling Business Center, see the *Business Center: Item Administration Guide* and *Business Center: Pricing Administration Guide*.

For more information about WebLogic, you can access the WebLogic documentation at http://download-llnw.oracle.com/docs/cd/E12840_01/wls/docs103/index.html (the "J2EE Deployment" section).

Installing and Deploying Sterling Selling and Fulfillment Foundation on Different Servers

About this task

This section applies only to users who are installing and deploying Sterling Selling and Fulfillment Foundation on separate systems.

In order for this scenario to deploy successfully, you must identify a log directory on the system where you are deploying Sterling Selling and Fulfillment Foundation.

Procedure

1. Set the LOG_DIR property in sandbox.cfg to a value that is meaningful on the system where the EAR will be deployed.
2. Run the <INSTALL_DIR>/bin/setupfiles.sh (or setupfiles.cmd) script.
3. Rebuild the EAR file.
4. Edit sandbox.cfg to set the LOG_DIR value back to its original value.

Results

For more information about sandbox.cfg and changing properties, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Precompiling the WAR File on WebLogic

To improve the performance when initially loading UI resources, IBM recommends that you precompile the jsps that comprise the WAR file. For more information about how to pre-compile jsps, see "JSP Pre-Compilation" in the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

Deploy the new EAR file as described in "Deploying the Enterprise Archive (EAR) on WebLogic."

Deploying the Enterprise Archive (EAR) on WebLogic

Sterling Selling and Fulfillment Foundation provides support for deploying Multiple EARs (Enterprise Archives) on a single application server. On the same application server, you can:

- Deploy different customizations of the same or different versions of the application, or
- Deploy different versions of the same application

Multiple EARs or context roots require additional memory for the application server JVM. Testing has shown that the deployment of a second IBM EAR file

requires 2.5 - 3.5 times the memory of a single EAR. Supporting two deployments may require up to 2.5 GB of heap space and 1.2 GB of permanent space.

During installation, you can use JVM-specific arguments to avoid out-of-memory errors. For more information, see the *Sterling Selling and Fulfillment Foundation: Properties Guide* descriptions of `ADDITIONAL_ANT_JAVA_TASK_ARGS` and `ADDITIONAL_ANT_COMPILER_TASK_ARGS`.

For information about JVM tuning on your application server, see the general and application server-specific JVM chapters in the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

For more information about how to implement multiple EAR files on the same application server, see the *Sterling Selling and Fulfillment Foundation: Customization Basics*.

For instructions on deploying the EAR to your WebLogic application server, see your WebLogic documentation.

Verifying Installation on WebLogic

About this task

To verify the Sterling Selling and Fulfillment Foundation installation:

Procedure

1. Restart your application server.
2. Start Internet Explorer.
3. To access the Application Console:
 - a. Access `http://<hostname>:<port>/smcfs/console/login.jsp`.
 - b. When prompted, enter your Login ID and Password.
4. To access IBM Sterling Business Center:
 - a. Access the IBM Sterling Business Center login page by setting the enterprise appropriately. For more information about setting the enterprise while logging in to IBM Sterling Business Center, see "Setting an Enterprise for Logging In to IBM Sterling Business Center" on page 173.
 - b. When prompted, enter your Login ID and Password.

Configuring a WebSphere Application Server - Overview

Before configuring WebSphere, IBM recommends that you start the WebSphere administrative server with the following memory parameters:

- `-Xms768 MB` or higher
- `-Xmx768 MB` or higher

Configuring WebSphere to Avoid Direct Datasource Lookup Warning Messages at Runtime

About this task

You have the option to avoid the warning messages regarding direct datasource lookups that occur at run time. To avoid these messages, do the following:

Procedure

1. From the WebSphere Administrative Console, expand **Troubleshooting** in the left panel and click on **Logs** and **Trace**.
2. Select each server that hosts Sterling Selling and Fulfillment Foundation and choose **Change Log Detail Levels** in the **General Properties**.
3. In the Components panel, select the class, **com.ibm.ejs.j2c.ConnectionFactoryBuilderImpl**, and specify the log level as **severe**.
4. Save the changes to the Master Configuration.

Configuring the WebSphere Classloader

About this task

Ensure that the WebSphere Classloader is set correctly for Classloader policy and Class loading modes as follows:

Procedure

1. From the Administrative Console left panel, choose Servers > Application Servers.
2. Select among the servers listed.
3. Set the Classloader policy pulldown to Single and the Class loading mode pulldown to Parent first.

Setting Up Application Clients to Invoke Sterling Selling and Fulfillment Foundation EJBs

About this task

In order to make EJB calls in Sterling Selling and Fulfillment Foundation using WebSphere you need to generate EJB stubs and skeletons. The following steps outline the method for creating the JAR files using the `ejbdeploy.sh` script to generate the stubs:

Procedure

1. Set the CLASSPATH to include `xercesImpl.jar`, `xalan.jar`, and `xml-apis.jar` as provided in the `JRE/lib/endorsed` directory. Also, CLASSPATH must include the jar files specified in the `dynamicclasspath.cfg` file.
2. Invoke the `ejbdeploy.sh` command from the `<WAS_HOME>/bin` directory with the following three arguments:
 - a. Specify the full path to the `smcfsejb.jar` file in `<INSTALL_DIR>/external_deployments/` directory.
 - b. Specify the temporary directory that is used for the EJB deployment.
 - c. Specify the full path to the desired output file, for example `smcfs_ejbstubs.jar`.

Results

Additionally set the classpath on the `ejbdeploy.sh` command line following the `-cp` argument. For example:

```
$WAS_HOME/bin/ejbdeploy.sh <INSTALL_DIR>/external_deployments/smcfsejb.jar  
$WAS_HOME/temp <INSTALL_DIR>/external_deployments/smcfsejb.jar -cp  
$CLASSPATH
```

Configuring WebSphere JVM Settings

About this task

You need to use the WebSphere Administrative Console to specify the JVM settings. These JVM settings must be set on **all** servers in a cluster (if you are using a cluster).

To configure JVM setting on WebSphere, do the following:

Procedure

1. From the WebSphere Administrative Console, select the application server specified for Sterling Selling and Fulfillment Foundation.
2. For IBM servers with IBM JDK 6.0 SR8:
 - a. Select Server Infrastructure > Java and Process Management > Process Definition > Environment Entries.
 - b. Choose New and specify the following values and then choose OK:

Name	Value	Description
PSALLOC	early	PSALLOC
NODISCLAIM	true	NODISCLAIM

3. Select Server Infrastructure > Java and Process Management > Process Definition > Java Virtual Machine. Edit the generic JVM arguments dialog to include the following values:

Property	Value	Description
-Dvendor	-Dvendor=shell	System Property. If you are using App Server Connection Pooling, use -Dvendor=websphere. Otherwise, use -Dvendor=shell.
-DvendorFile	-DvendorFile=/servers.properties	System property. Specify as an argument to the java command.

4. Under the Custom Properties section, set the JVM settings to the following values:

Name	Value	Description
client.encoding.override	UTF-8	Enables the use of special characters.

5. Restart the application server to enable these changes to take effect.
6. Save the changes to the Master Configuration.

Setting Up WebSphere to Display Barcodes and Graphs

About this task

Sterling Selling and Fulfillment Foundation uses the X Window functionality to display barcodes and dynamic graphical images (such as inventory supply & demand graphs) in a UNIX environment.

The following configuration is required to enable the X Window environment in UNIX systems for the WebSphere application server:

Procedure

1. From the WebSphere Administrative Console, go to Servers > Application Server and select the application server specified for Sterling Selling and Fulfillment Foundation.
2. On the Configuration tab, select Java and Process Management under Server Infrastructure option.
3. Select Process Definition.
4. On the configuration, go to Additional Properties and select Environment Entries.
5. Select New.
6. On the General Properties enter the Name as DISPLAY and the value as *IP_address_of_XWindows_server:0.0*. Do make sure that the X Window server accepts requests from this client.
7. If you are using UNIX, run the `xhost+` command to remove access control for your X Window server.

You can run X server on the same server in which you run Sterling Selling and Fulfillment Foundation. However, you need to be logged to the server console. Restart the application server for the DISPLAY variable to take effect.

8. Save the changes to the Master Configuration.

Note: If the X Window server goes down or crashes while the inventory user interface is using the `jbchartx.jar` file, the WebSphere server also goes down.

Building the Enterprise Archive (EAR) Package on WebSphere

When deploying Sterling Selling and Fulfillment Foundation on WebSphere, use the `smcfs.ear` file, which may contain:

- `smcfs.war` - Web module that contains all of the Sterling Selling and Fulfillment Foundation JSPs and other Web application components.
- `sma.war` - Web module that contains the System Management Administrator application components.
- `sbc.war` - Web module that contains all of the IBM Sterling Business Center Web application components.
- `yantrawebsservices.war` - Web module that contains all of the Sterling Selling and Fulfillment Foundation Web services interface classes.
- `smcfsejb.jar` - The EJB module that contains all the Sterling Selling and Fulfillment Foundation EJBs. You can pass an alternate earfile name by using the `-Dearfile` option to the `buildear.sh` script. Doing this will result in a name change for the `ejb` jar file. For example, if you specify an EAR file as `xyz.ear`, the `ejb` jar becomes `xyzejb.jar`.
- `smcfswsbe.jar` - The backend Web services jar file. You get this file if you expose Web services. You can pass an alternate earfile name to the `ear build` script. Doing this will result in a name change for the Web services backend jar file. For example, if you specify an EAR file as `xyz.ear`, the Web service backend jar becomes `xyzwsbe.jar`.
- jars that contain backend business logic
- jars that contain third-party libraries accessed by backend logic

Each of the third-party JAR files is left as is and in the manifest of the application each file is indicated as a dependency. For example, log4j files are represented separately as log4j-1.2.15.jar with a dependency in the application.

Deploying Context-Sensitive Help on WebSphere

The Sterling Selling and Fulfillment Foundation Context-Sensitive Help is built as a separate EAR file called smcfsdocs.ear using the command:

```
<INSTALL_DIR>/bin/buildear.sh -Dsupport.multi.war=true -Dappserver=websphere  
-Dwarfiles=smcfs,sma,svc -Dearfile=smcfs.ear create-doc-ear
```

To make use of the documentation and help files associated with Sterling Selling and Fulfillment Foundation, deploy the smcfsdocs.ear file in all of the same locations where you deploy the smcfs.ear file.

Note: By default, for the Context-Sensitive Help to be displayed, the Sterling Selling and Fulfillment Foundation application sends the corresponding request to:

```
http://<Host name>:<Port Number>/smcfsdocs/yfscommon/online_help/en_US/  
wwhelp/wwhimpl/common/html/wwhelp.htm
```

However, if the Sterling Selling and Fulfillment Foundation application is deployed in a context root other than smcfs, for example, sterling, set the yfs.onlinehelp.path property with the context root information in the yfs.properties file, for example, set yfs.onlinehelp.path=/sterlingdocs/yfscommon/online_help. If your installation locale code is not en_US, set yfs.onlinehelp.path.overrideforlocale.<your_local_code>=/sterlingdocs/yfscommon/online_help/<your_local_code>.

For information about the full Product Documentation Library, see “Documentation Library” on page 2.

Note: You must define the scui-help-url context parameter in web.xml to deploy the context-sensitive help in Sterling Business Center. The URL context parameter value must follow the following format:

```
http(s)://<server>:<port>/<context>/<online help properties>
```

Building Web Services on WebSphere

When you create the EAR file for the application, web services are also created. There are two types of web services that can be created:

- EJB: The EJB web service accepts two string inputs and returns one string output. No information about the content of the strings is included in the WSDL for this web service; the caller must open the javadocs for the API to get the information about how to structure the content. This makes it very difficult to dynamically generate calls to the XAPI web services without an additional source of information beyond the WSDL. This type of web service is created using EJB beans, and is created by default with the application EAR.
- JAX-WS: The WSDL of the JAX-WS web service includes information about the input expected by the server, the output, and exceptions, which makes it easier to dynamically generate calls to the web services. JAX-WS web services are created using JAXB beans. Instead of being wrappers around EJB beans, these are wrappers around JAXB beans.

You can choose to build the application with one or both types of web services. Both types of web services require some configuration prior to running the build script to create the application EAR and web services WAR file.

Note: You can also choose to suppress the web services build by using the `-Dnowebservice=true` option on the `build` command.

If you are planning to run the application components as web services, additional setup is required. The setup takes place in either the `namedwebservices.xml` file or `webservicebeans.xml` file. You can expose APIs as web services. You can also expose synchronous SDF services through web services.

Note: If API security is enabled, ensure that you expose the Login API.

Defining an EJB Web Service with WebSphere Procedure

1. In the `<INSTALL_DIR>/repository/eardata/platform/webservices` folder, locate the `namedwebservices.xml.sample` file. Rename (or copy) it to `namedwebservices.xml`. This is the file you will edit.
2. In `namedwebservices.xml`, specify each API you want to expose as a web service in an `Api/Name` attribute.

Property

Description

ServiceName

The name of the service that you configured using the Service Definition Framework (SDF).

ExposedName

The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the web service programmatically. When specifying a service name for `ExposedName`, choose a literal that does not match any of the standard application API names.

The exposed name must start with a lower case letter.

3. Set the properties necessary for your application server in `sandbox.cfg`:

Parameter

Description

WAS_DIR

Set the value to specify the `<WAS_HOME>` directory. (You may need to create the property.)

Required if building simple web services.

For WebSphere only

WEBSERVICES_BUILDS

Takes a comma-separated list that can include `YIFWebService` and `SIXBeanXapiJaxWS`. Defaults to `YIFWebService`.

Required for EJB and JAX-WS web services, on all application servers.

4. Save the file, navigate to the `bin` directory, and run the `setupfiles` command:

Note: If you receive a "permission denied" error when performing this step, grant the build user write permission to the subdirectory referenced in the error message.

- For UNIX/Linux: <INSTALL_DIR>/bin/setupfiles.sh
 - For Windows: <INSTALL_DIR>\bin\setupfiles.cmd
5. Create the EAR as described in "Creating Enterprise Archive Files on WebSphere" on page 153. The web services defined in the file will be built when you create the application EAR.
 6. Later, if you want to add more APIs and services as EJB Web services, repeat these steps again.

Defining a JAX-WS Web Service with WebSphere

About this task

You can expose APIs as web services. Before you create the application EAR file, you need to define which APIs will be exposed as web services so that they are included in the web services WAR file in the EAR. This section provides a high-level overview of how to configure a JAX-WS web service.

To expose JAXB web services, you must use the Apache Ant 1.8.1.

As the Apache Ant 1.8.1 version is not installed with application, you must configure the system to point to an external instance of Apache Ant 1.8.1 that will be used to build the EAR with the web services.

The following APIs are not supported with JAX-WS web services:

- createAsyncRequest
- evaluateAdvancedCondition
- evaluateCondition
- getAgentCriteriaList
- getPage
- invokeUE
- multiApi
- printDocumentSet

Procedure

1. Download the Apache Ant 1.8.1 from the Apache Ant website (<http://ant.apache.org/index.html>) and install using instructions provided by Ant.
2. Set the ANT_HOME in your environment to point to the Ant 1.8.1 instance.
3. Set the properties necessary for your application server in sandbox.cfg:

Parameter

Description

USE_EXTERNAL_ANT

Valid values are true/false.

To point to the Ant 1.8.1 installed in step 1, set the values to true.

WEBSERVICES_BUILDS

Can take a comma separated list that can include YIFWebService and SIXBeanXapiJaxWS.

Required for EJB and JAX-WS web services, on all application servers.

XBEAN_PACKAGE

By default, the xbean package names and namespace generated for JAX-WS web services include the word "documentation." Use this parameter to replace the word "documentation" with another string.

This value can be any string which would form a valid java package name. Optional.

JAXB_LOCAL_SCOPING

Valid values are true/false. Set to true.

This flag affects the way that jaxb generates beans. Everything becomes "toplevel", so there will be no nested inner classes. If not used, results in class file names that are too long for the file system.

Required when building JAX-WS web services.

JAXB_ALWAYS_ANNOTATE_CLASSNAMES

The classes that get generated by JAXB get "1-up" numbers at the end of them to make them distinct from other classes that would otherwise have the same name.

This can be done on a case by case basis using the webservicebeans.xml, or globally using this flag.

If you set this flag to true, all classes get annotated. Or, to minimize the user of one-up numbers, specify the `AnnotateClassNames` attribute on the webservicebeans.xml as shown in the example below to indicate which apis need to have annotations.

```
<Bean BeanName="ParticipantBean"
BeanPackage="com.sterlingcommerce.jaxws.participant.
webservices" >
    <Apis>

    <Api AnnotateClassNames="true"
Name="getOrganizationHierarchy"
ExposedName="getOrganizationHierarchy" />

    <Api AnnotateClassNames="true"
Name="getOrganizationList"
ExposedName="getOrganizationList" />

    <Api AnnotateClassNames="false"
Name="getPersonInfoList"
ExposedName="getPersonInfoList" />

    <Api AnnotateClassNames="false" Name="login"
ExposedName="login" />
    </Apis>
</Bean>
```

Valid values for this flag are true/false. It is recommended that you always include this flag and set it to true.

Required when building JAX-WS web services.

4. Save the file, navigate to the bin directory, and run the setupfiles command:

Note: If you receive a "permission denied" error when performing this step, grant the build user write permission to the subdirectory referenced in the error message.

- For UNIX/Linux: `<INSTALL_DIR>/bin/setupfiles.sh`

- For Windows: <INSTALL_DIR>\bin\setupfiles.cmd
5. In the <INSTALL_DIR>/repository/eardata/platform/webservices folder, locate the webservicebeans.xml.sample file. Copy the file to webservicebeans.xml. This is the file you will edit.

webservicebeans.xml attributes	Description
Bean attributes	
Note: Do not put the same API into two beans.	
BeanName	Enter a descriptive name for the web service. This name is used by the WSDL generator and reflected in the client code. Required. For example, a company named "Dave's BBQ" might use the following: BeanName=DavesBBQJaxWS
BeanPackage	Enter a descriptive name for the package. Each bean must have a unique package name. This name is used by the WSDL generator and reflected in the client code. Required. For example, a company named "Dave's BBQ" might use the following: BeanPackage="com.DavesBBQ
API element attributes	
Api Name	The name of the API that you configured using the Service Builder.
ExposedName	The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the web service programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard application API names. The exposed name must start with a lower case letter.
AnnotateClassNames	Specify the AnnotateClassNames attribute on the webservicebeans.xml to indicate which apis need to have annotations.
Service element attributes	
Service Name	The name of the service that you configured using the Service Builder.
ExposedName	The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the web service programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard application service names. The exposed name must start with a lower case letter. The exposed name must start with a lower case letter.
AnnotateClassName	Specify the AnnotateClassNames attribute on the webservicebeans.xml to indicate which apis need to have annotations.
To expose an SDF service as a web service, you must also complete the procedure in "Exposing an SDF Service as a Web Service on WebSphere" on page 150.	

6. Save the file.

7. Create the EAR as described in “Creating Enterprise Archive Files on WebSphere” on page 153. The web services defined in the file will be built when you create the application EAR.

CAUTION: When building an ear with JAX-WS Web services, you may encounter out of memory errors. JAX-WS Web services will generate a java class for each element in the XSD, which can exceed the system memory. If the ear build runs out of memory, then unnecessary APIs should be removed from the webservicebeans.xml file. Additionally, APIs with potentially large output XMLs may be exposed through services with reduced XSDs to eliminate unwanted elements.

Tip: To help avoid out of memory issues when building the ear, group APIs into separate beans instead of exposing a large number of APIs in a single bean.

8. Later, if you want to add more APIs and services as JAX-WS web services, repeat these steps again.

Example of webservicebeans.xml file - WebSphere

The following is an example of a webservicebeans.xml file:

```
<WebServices>
  <Beans>
    <Bean BeanName="TheBeans"
      BeanPackage="com.sterlingcommerce.jaxws.the.webservices" >
      <Apis>
        <Api Name="login" ExposedName="theLogin" />
        <Api Name="getLocaleList" ExposedName="theGetLocaleList" />
      </Apis>
    </Bean>
    <Bean BeanName="OtherBeans"
      BeanPackage="com.sterlingcommerce.jaxws.other.webservices" >
      <Apis>
        <Api Name="getLocaleList" ExposedName="otherGetLocaleList" />
      </Apis>
      <Services>
        <Service AnnotateClassNames="true" Name="testWS" ExposedName="testWS" />
        <Service Name="testWS2" ExposedName="testWS2" />
      </Services>
    </Bean>
  </Beans>
</WebServices>
```

Exposing an SDF Service as a Web Service on WebSphere

About this task

To expose an SDF service as a web service, you first have to copy the input and output xsds for the API that is exposed in the SDF service and edit them.

Note: If you want to overwrite the input or output XSD for any API, you must expose the API as an SDF service and then expose the SDF service as a Web service.

Procedure

1. Open the SDF and create a synchronous service that has an API node.
2. Select the desired API.
3. Save the SDF service.

4. Copy the input and output XSD files for the API you exposed in the SDF from the following location:

```
<INSTALL_DIR>/xapidocs/api_javadocs/XSD
```

to:

```
<INSTALL_DIR>/extensions/webservices
```

5. Rename the files to match the name and exposed name that you give them in webservicebeans.xml.

Use the following format:

```
<name>_<exposed_name>_input.xsd
```

```
<name>_<exposed_name>_output.xsd
```

Results

This is also the procedure you would use to extend a table, with this additional step: after copying and renaming the files, edit them to reference the new columns/API inputs/outputs.

Configuring a JAX-WS Web Services Client (UNIX/Linux Only) on WebSphere

About this task

Sterling Selling and Fulfillment Foundation provides a client generator for use with JAX-WS web services.

CAUTION:

The client generator is supported on UNIX and Linux only. It is not supported on Windows.

For WebSphere, the client generator accesses the WSDL directly from the server.

Note: If you are modifying the XSD for any API or service which has been exposed as a Web service, you must rebuild the Web services client. Additionally, each time you extend the database through customization or upgrade, you must run the xsdgenerator, and then rebuild the ear and client. See *Sterling Selling and Fulfillment Foundation: Customizing APIs* for more information about using the xsdgenerator, and see *Sterling Selling and Fulfillment Foundation: Extending the Database* for more information about extending the database. See “Building the Enterprise Archive (EAR) Package on WebSphere” on page 144 for information about building the ear.

Results

Running the JAX-WS Client Generator on WebSphere Procedure

To invoke the generator, run the buildjaxclient.sh script from the <INSTALL_DIR>/bin directory. Include the properties required for your application server type from the following list when running the script:

Property	Description
----------	-------------

-DBEAN_NAME

Required for all application servers. Matches the beanname on the server. Note that the name of the service might be different, depending on the application server being used.

-DAPPSERVER

Required. Valid values are websphere, weblogic, or jboss.

-DSERVER_URL

Required for all application servers. The http URL of the server up to, but not including, the bean name.

The following is an example of a command for WebSphere:

```
bin/buildjaxclient.sh -DBEAN_NAME=ParticipantBeanService
-DAPPSERVER=websphere -DSERVER_URL=http://00.00.00.00:1975/SIXBeanXapiJaxWS
```

Results

Running the script creates a new folder, jaxwsclient, that contains the compiled classes.

Sample Code for BeanService Classes - WebSphere

Sample code is generated in the same directory as the bean and BeanService class. You can use this sample as a reference for invoking the bean methods.

The sample file has the same name as the service class but with "Sample" appended to the end of the name.

For example, SMCFSBeanService would have an accompanying sample called SMCFSBeanServiceSample class.

```
jaxwsclient/output/ParticipantBeanService/com/sterlingcommerce
/jaxws/participant/webservices/ParticipantBeanServiceSample
```

The class generates a method for each API you expose. The content will require editing.

```
public void testlogin( ) throws Exception {
com.sterlingcommerce.documentation.ycp.login.input.Login input
= new com.sterlingcommerce.documentation.ycp.login.input.Login();
    /*
Insert custom code here to set values on the input object.
    */
com.sterlingcommerce.documentation.ycp.login.output.Login
returnValue = b.login(env,input);
    /*
Insert custom code here to retrieve values from the return object.
    */
}
```

Including Custom Classes on WebSphere**About this task**

If you have developed custom Java classes (user exits, event handlers, and so forth) you need to deploy them in order for them to be available.

To ensure that your custom classes get invoked, do the following:

Procedure

1. Create a JAR file with all your custom classes.
2. Use `install3rdParty.sh` to include your custom jar file in the APP classpath. For more information about packaging and deploying jar files, see the *Sterling Selling and Fulfillment Foundation: Customization Basics*.
3. Rebuild the EAR as described in “Creating Enterprise Archive Files on WebSphere.”

Results

The custom classes are automatically included in the `smcfs.ear` file.

Creating Enterprise Archive Files on WebSphere

About this task

Enterprise Archives are built using an ANT (`buildEAR.xml`) that accepts the following targets:

Main Target

Description

`create-ear`

Creates the EAR files:

- `smcfs.ear` - The Sterling Selling and Fulfillment Foundation application EAR file
- `smcfsdocs.ear` - The Sterling Selling and Fulfillment Foundation documentation EAR file

`create-doc-ear`

Creates `smcfsdocs.ear` - the Sterling Selling and Fulfillment Foundation documentation EAR

Procedure

To create an application EAR file, run the following command from the `<INSTALL_DIR>/bin` directory:

```
./buildear.sh (.cmd for Windows) -Dappserver=websphere  
-Dwarfiles=smcfs,sma,sbc -Dearfile=smcfs.ear create-ear
```

Note: Sterling Selling and Fulfillment Foundation supports the RPC encoded or document literal style and usage of invocation for Web services. When choosing style and usage for WebSphere, pass the following in the ear command line:

```
-D websphere-java2wsdl-style=<rpc|document>
```

Running this command creates the `smcfs.ear` and `smcfsdocs.ear` files in the `<INSTALL_DIR>/external_deployments/` directory. It also puts three war files into the `smcfs.ear`:

- `smcfs.war` - The Sterling Selling and Fulfillment Foundation application war file
- `sma.war` - The System Management Administrator application war file
- `sbc.war` - The IBM Sterling Business Center application war file

Note: You can add the following options to the end of the above `buildear` commands:

- `-Dnowebservice=true` If you do not want to use Web services. If you do want to use Web services, see the “Building Web Services on WebSphere” on page 145.
- `-Ddevmode=true` If you want to use the HTTP API Tester in the development environment.
- `-Dnodecar=true` If you want to skip the documentation build.
- `-Dsupport.multi.war=true` If you want to copy all the UI jars to `<WAR>/WEB-INF/lib`. The UI jars will be copied based on the entry in `DCL.xml`.

For more information about the System Management Administrator (SMA) see the *Sterling Selling and Fulfillment Foundation: System Management and Administration Guide*. For more information about IBM Sterling Business Center, see the *Business Center: Item Administration Guide* and *Business Center: Pricing Administration Guide*.

For more information about WebSphere, you can access the WebSphere documentation at <http://www-01.ibm.com/software/webservers/appserv/was/library/index.html>

Precompiling the WAR File on WebSphere

To improve the performance when initially loading UI resources, IBM recommends that you precompile the jsps that comprise the WAR file. For more information about how to pre-compile jsps, see “JSP Pre-Compilation” in the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

Deploying the Enterprise Archive (EAR) Using the WebSphere Admin Console

About this task

Sterling Selling and Fulfillment Foundation provides support for deploying Multiple EARs (Enterprise Archives) on a single application server. On the same application server, you can:

- Deploy different customizations of the same or different versions of the application, or
- Deploy different versions of the same application

Multiple EARs or context roots require additional memory for the application server JVM. Testing has shown that the deployment of a second IBM EAR file requires 2.5 - 3.5 times the memory of a single EAR. Supporting two deployments may require up to 2.5 GB of heap space and 1.2 GB of permanent space.

During installation, you can use JVM-specific arguments to avoid out-of-memory errors. For more information, see the *Sterling Selling and Fulfillment Foundation: Properties Guide* descriptions of `ADDITIONAL_ANT_JAVA_TASK_ARGS` and `ADDITIONAL_ANT_COMPILER_TASK_ARGS`.

For information about JVM tuning on your application server, see the general and application server-specific JVM chapters in the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

For more information about how to implement multiple EAR files on the same application server, see the *Sterling Selling and Fulfillment Foundation: Customization Basics*.

Note: Before deploying JAX-WS web services on WebSphere, note that the WebSphere server will attempt to inspect every jar in the WEB-INF lib by default. To avoid this behavior, which can add a significant amount of time to the deployment of the JAX-WS web services, configure the WebSphere server to ignore certain jars during deployment of the JAX-WS web service. Add any jars that the server should skip to the Ignore-Scanning-Archives section in <WAS_HOME>/properties/amm.filter.properties file. The jars you include will depend on what applications you build and so forth. One way to determine what jars to filter is to extract the SIXBean war file and use the jar -tvf command on it. Put the jar files you find in the WEB-INF/lib into the amm.filter.properties file.

To deploy the EAR on WebSphere:

Procedure

1. From the WebSphere Administrative Console menu in the left pane, select Applications > Application Types > WebSphere enterprise applications.
2. The right pane is populated with a list of applications that are deployed. Click the Install button.
3. Choose Local File System or Remote File System. Click the corresponding Browse button and browse to the Enterprise Archive such as smcfs.ear you want to deploy. Click Next.
4. Choose Fast Path option. Click Next.
5. Check Deploy enterprise beans, and change the application name as follows:
 - Ensure that there are no spaces in the application name; otherwise, the WebSphere-provided jsp compiler script will fail.
 - Ensure that the application name is different from that of the documentation EAR; otherwise, accepting the default makes both names the same.

If you are using Web services, check Deploy WebServices.

Note: If you want to precompile the JSP files during deployment, check Precompile JavaServer Pages files.

Click Next.

6. The Map Modules to Servers screen displays. Select the checkbox next to each desired module (at least two entries, smcfsejb.jar and smcfs.war, should be present). Click the Cluster/Server in the Cluster and Server pane. Click **Apply**. The screen refreshes and the server field is updated with the chosen value. Click Next.
7. Accept the default JNDI names for the EJB modules on the Provide JNDI Names for Beans screen. Click Next.
8. On the Map Virtual Hosts for Web Modules screen, select your web module and its correct virtual host. Choose Next.
9. The Ensure all Unprotected 2.x Methods screen displays. Click Next.
10. The Provide Options to perform the WebServices Deployment screen displays. Leave them as is and click Next.
11. On the summary page, choose Finish.
12. If you are deploying the IBM Sterling Field Sales application, make the following additional changes in the WebSphere Administration Console:
 - a. For each one of the application servers where you deploy the IBM Sterling Selling and Fulfillment Suite application, verify that the ClassLoader Policy is set to Multiple.

- b. Navigate to Enterprise Applications > Application Name > Class Loader. Set the Class Loader Order to "Classes loaded with local class loader first (parent last)."
- c. Navigate to Enterprise Applications > Application Name > Class Loader. Set the WAR Class Loader Policy to "Class Loader for each WAR file in application."

Verifying the Installation on WebSphere

About this task

To verify the Sterling Selling and Fulfillment Foundation installation:

Procedure

1. Restart your application server.
2. Start Internet Explorer.
3. To access the Application Console:
 - a. Access `http://<hostname>:<port>/smcfs/console/login.jsp`.
 - b. When prompted, enter your Login ID and Password.
4. To access IBM Sterling Business Center :
 - a. Access the IBM Sterling Business Center login page by setting the enterprise appropriately. For more information about setting the enterprise while logging in to IBM Sterling Business Center, see "Setting an Enterprise for Logging In to IBM Sterling Business Center" on page 173.
 - b. When prompted, enter your Login ID and Password.

Configuring a JBoss Application Server

About this task

Note: The JBoss server must have the default name of "all" for the precompilation scripts to run successfully.

To set up the JBoss application server, you must set up some properties in the JBoss script file.

To set up the JBoss script file, do the following:

Procedure

1. Add the following properties to the `<JBOSS_HOME>/bin/run.conf` file supplied by JBoss. Each property and its proper syntax are described in the following list:

Property

Required Edits

JAVA_OPTS

Depending on your JVM vendor, specify as follows:

`-Xms<value> -Xmx<value>`

For example, for HP UX 11i on Itanium, set this value to

`-XX:MaxPermSize=512m -Xms768m -Xmx768m`

For information about supported JDK tiers and memory requirements for specific operating systems, see the *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.

-Dfile.encoding

To ensure that all the Sterling Selling and Fulfillment Foundation screens display UTF-8 characters for java commands, specify:

```
-Dfile.encoding=UTF-8
```

This is applicable to all the Sterling Selling and Fulfillment Foundation Java start-up scripts.

-Dvendor

System Property. If you are using App Server Connection Pooling, use `-Dvendor=jboss`. Otherwise, use `-Dvendor=shell`.

```
-Dvendor=shell
```

-DvendorFile

System property. Specify as an argument to the java command.

```
-DvendorFile=/servers.properties
```

The `-D` properties shown above can either be included in `JAVA_OPTS` or passed in a command line. For example:

```
run.sh -c yantra_domain
-Djboss.partition.name=yantraPartition -b host
-Dvendor=shell -DvendorFile=/servers.properties
-DLOGFILE=logs/JBoss_sci.log
-DSECURITY_LOGFILE=logs/JBoss_security.log
```

2. If you are using an HTTPS transport, download the Secure Socket Extension (JSSE) 1.0.3 package from <http://java.sun.com> and add the following files to the `<JAVA_HOME>/jre/lib/extn/` directory:
 - `jnet.jar`
 - `jcrt.jar`
 - `jsse.jar`

In addition to this setup, see “Setting Up JBoss to Display Barcodes and Graphs” for information about setting up the JBoss application server to display barcodes and graphs.

Setting Up JBoss to Display Barcodes and Graphs

About this task

Sterling Selling and Fulfillment Foundation uses the X Window functionality to display barcodes and dynamic graphical images (such as inventory supply & demand graphs) in a UNIX environment. The following configuration is required to enable the X Window environment in UNIX systems for JBoss application servers:

Procedure

1. If your UNIX server is also an X Window client, edit the `run.sh` script, and set the `DISPLAY` environment variable as follows:

```
export DISPLAY=<IP_address_of_XWindows_server>:0.0
```
2. If you are using UNIX, run the `xhost +` command to remove access control for your X Window server.
3. You can run `X-server` on the same server in which you run Sterling Selling and Fulfillment Foundation. However, you need to be logged in to the server console.

Note: If the X Window server goes down or crashes while the inventory user interface is using the `jbchartx.jar` file, the JBoss server also goes down.

Building the Enterprise Archive (EAR) Package on JBoss

When deploying Sterling Selling and Fulfillment Foundation on JBoss, use the `smcfs.ear` file, which may contain:

- `smcfs.war` - Web module that contains all of the Sterling Selling and Fulfillment Foundation JSPs and other Web application components.
- `sma.war` - Web module that contains the System Management Administrator application components.
- `sbc.war` - Web module that contains all of the IBM Sterling Business Center Web application components.
- `smcfsejb.jar` - The EJB module that contains all the Sterling Selling and Fulfillment Foundation EJBs. You can pass an alternate earfile name by using the `-Dearfile` option to the `buildear.sh` script. Doing this will result in a name change for the `ejb` jar file. For example, if you specify an EAR file as `xyz.ear`, the `ejb` jar becomes `xyzejb.jar`.
- `smcfswsbe.jar` - The backend Web services jar file. You get this file if you expose Web services. You can pass an alternate earfile name to the `ear` build script. Doing this will result in a name change for the Web services backend jar file. For example, if you specify an EAR file as `xyz.ear`, the Web services backend jar becomes `xyzwsbe.jar`.
- Jars that contain backend business logic.
- Jars that contain third-party libraries accessed by backend logic.

Each of the third-party JAR files are left as is and in the manifest of the application each file is indicated as a dependency. For example, `log4j` files are represented separately as `log4j-1.2.15.jar` with a dependency in the application.

Deploying Context-Sensitive Help on JBoss

The Sterling Selling and Fulfillment Foundation Context-Sensitive Help documentation is built as a separate EAR file called `smcfsdocs.ear` using the command:

```
<INSTALL_DIR>/bin/buildear.sh create-doc-ear
-Dsupport_multi.war=true -Dappserver=jboss
-Dwarfiles=smcfs,sma,sbc -Dearfile=smcfs.ear
```

To make use of the documentation and Context-Sensitive Help files associated with Sterling Selling and Fulfillment Foundation, deploy the `smcfsdocs.ear` file in all of the same locations where you deploy the `smcfs.ear` file.

Note: By default, for the Context-Sensitive Help to be displayed, the Sterling Selling and Fulfillment Foundation application sends the corresponding request to:

```
http://<Host name>:<Port Number>/smcfsdocs/yfscommon/online_help/en_US/
wwhelp/wwhimpl/common/html/wwhelp.htm
```

However, if the Sterling Selling and Fulfillment Foundation application is deployed in a context root other than `smcfs`, for example, `sterling`, set the `yfs.onlinehelp.path` property with the context root information in the `yfs.properties` file, for example, set `yfs.onlinehelp.path=/sterlingdocs/yfscommon/online_help`. If your installation locale code is not `en_US`, set

```
yfs.onlinehelp.path.overrideforlocale.<your_local_code>=/sterlingdocs/  
yfscommon/online_help/<your_local_code>.
```

For information about the full Product Documentation Library, see “Documentation Library” on page 2.

Note: You must define the `scui-help-url` context parameter in `web.xml` to deploy the context-sensitive help in Sterling Business Center. The URL context parameter value must follow the following format:

```
http(s)://<server>:<port>/<context>/<online help properties>
```

Building Web Services on JBoss

When you create the EAR file for the application, web services are also created. There are two types of web services that can be created:

- **EJB:** The EJB web service accepts two string inputs and returns one string output. No information about the content of the strings is included in the WSDL for this web service; the caller must open the javadocs for the API to get the information about how to structure the content. This makes it very difficult to dynamically generate calls to the XAPI web services without an additional source of information beyond the WSDL. This type of web service is created using EJB beans, and is created by default with the application EAR.
- **JAX-WS:** The WSDL of the JAX-WS web service includes information about the input expected by the server, the output, and exceptions, which makes it easier to dynamically generate calls to the web services. JAX-WS web services are created using JAXB beans. Instead of being wrappers around EJB beans, these are wrappers around JAXB beans.

You can choose to build the application with one or both types of web services. Both types of web services require some configuration prior to running the `buildear` script to create the application EAR and web services WAR file.

Note: You can also choose to suppress the web services build by using the `-Dnowebservice=true` option on the `buildear` command.

If you are planning to run the application components as web services, additional setup is required. The setup takes place in either the `namedwebservices.xml` file or `webservicebeans.xml` file. You can expose APIs as web services. You can also expose synchronous SDF services through web services.

Note: If API security is enabled, ensure that you expose the Login API.

Defining an EJB Web Service with JBoss

Procedure

1. In the `<INSTALL_DIR>/repository/eardata/platform/webservices` folder, locate the `namedwebservices.xml.sample` file. Rename (or copy) it to `namedwebservices.xml`. This is the file you will edit.
2. In `namedwebservices.xml`, specify each API you want to expose as a web service in an `Api/Name` attribute.

Property

Description

ServiceName

The name of the service that you configured using the Service Definition Framework (SDF).

ExposedName

The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the web service programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard application API names.

The exposed name must start with a lower case letter.

3. Set the properties necessary for your application server in sandbox.cfg:

Parameter**Description****EJB_3_ENABLED**

Set to true or false. Determines whether the EJBs are generated according to the spec version 2 or 3. JBoss supports both.

Required if building EJB web services.

For JBoss only.

JBOSS_PRECOMPILE_JSP

Precompiles pages in the WAR file.

Set to true(precompile jsps) or false(do not precompile jsps). There is no default set by installation, but jsps will not be precompiled unless you set this to true.

Required if building EJB web services.

For JBoss only.

JBOSS_DIR

Set to the absolute path of the JBoss installation directory.

Required if building EJB web services.

For JBoss only.

WEBSERVICES_BUILDS

Takes a comma-separated list that can include YIFWebService and SIXBeanXapiJaxWS. Defaults to YIFWebService.

Required for EJB and JAX-WS web services, on all application servers.

4. Save the file, navigate to the bin directory, and run the setupfiles command:
 - For UNIX/Linux: <INSTALL_DIR>/bin/setupfiles.sh
 - For Windows: <INSTALL_DIR>\bin\setupfiles.cmd
5. Create the EAR as described in “Creating the EAR on JBoss” on page 166. The web services defined in the file will be built when you create the application EAR.
6. Later, if you want to add more APIs and services as EJB Web services, repeat these steps again.

Defining a JAX-WS Web Service with JBoss

About this task

You can expose APIs as web services. Before you create the application EAR file, you need to define which APIs will be exposed as web services so that they are included in the web services WAR file in the EAR. This section provides a high-level overview of how to configure a JAX-WS web service.

To expose JAXB web services, you must use the Apache Ant 1.8.1.

As the Apache Ant 1.8.1 version is not installed with application, you must configure the system to point to an external instance of Apache Ant 1.8.1 that will be used to build the EAR with the web services.

The following APIs are not supported with JAX-WS web services:

- createAsyncRequest
- evaluateAdvancedCondition
- evaluateCondition
- getAgentCriteriaList
- getPage
- invokeUE
- multiApi
- printDocumentSet

Procedure

1. Download the Apache Ant 1.8.1 from the Apache Ant website (<http://ant.apache.org/index.html>) and install using instructions provided by Ant.
2. Set the ANT_HOME in your environment to point to the Ant 1.8.1 instance.
3. Set the properties necessary for your application server in sandbox.cfg:

Parameter

Description

USE_EXTERNAL_ANT

Valid values are true/false.

To point to the Ant 1.8.1 installed in step 1, set the values to true.

WEBSERVICES_BUILDS

Can take a comma separated list that can include YIFWebService and SIXBeanXapiJaxWS.

Required for EJB and JAX-WS web services, on all application servers.

XBEAN_PACKAGE

By default, the xbean package names and namespace generated for JAX-WS web services include the word "documentation." Use this parameter to replace the word "documentation" with another string.

This value can be any string which would form a valid java package name. Optional.

JAXB_LOCAL_SCOPING

Valid values are true/false. Set to true.

This flag affects the way that jaxb generates beans. Everything becomes "toplevel", so there will be no nested inner classes. If not used, results in class file names that are too long for the file system.

Required when building JAX-WS web services.

JAXB_ALWAYS_ANNOTATE_CLASSNAMES

The classes that get generated by JAXB get "1-up" numbers at the end of them to make them distinct from other classes that would otherwise have the same name.

This can be done on a case by case basis using the `webservicebeans.xml`, or globally using this flag.

If you set this flag to true, all classes get annotated. Or, to minimize the user of one-up numbers, specify the `AnnotateClassNames` attribute on the `webservicebeans.xml` as shown in the example below to indicate which apis need to have annotations.

```
<Bean BeanName="ParticipantBean"
BeanPackage="com.sterlingcommerce.jaxws.participant.
webservices" >
  <Apis>
    <Api AnnotateClassNames="true"
Name="getOrganizationHierarchy"
ExposedName="getOrganizationHierarchy" />

    <Api AnnotateClassNames="true"
Name="getOrganizationList"
ExposedName="getOrganizationList" />

    <Api AnnotateClassNames="false"
Name="getPersonInfoList"
ExposedName="getPersonInfoList" />

    <Api AnnotateClassNames="false" Name="login"
ExposedName="login" />
  </Apis>
</Bean>
```

Valid values for this flag are true/false. It is recommended that you always include this flag and set it to true.

Required when building JAX-WS web services.

4. Save the file, navigate to the bin directory, and run the `setupfiles` command:
 - For UNIX/Linux: `<INSTALL_DIR>/bin/setupfiles.sh`
 - For Windows: `<INSTALL_DIR>\bin\setupfiles.cmd`
5. In the `<INSTALL_DIR>/repository/eardata/platform/webservices` folder, locate the `webservicebeans.xml.sample` file. Copy the file to `webservicebeans.xml`. This is the file you will edit.

webservicebeans.xml attributes	Description
Bean attributes Note: Do not put the same API into two beans.	
BeanName	Enter a descriptive name for the web service. This name is used by the WSDL generator and reflected in the client code. Required. For example, a company named "Dave's BBQ" might use the following: BeanName=DavesBBQJaxWS
BeanPackage	Enter a descriptive name for the package. Each bean must have a unique package name. This name is used by the WSDL generator and reflected in the client code. Required. For example, a company named "Dave's BBQ" might use the following: BeanPackage="com.DavesBBQ
API element attributes	

webservicebeans.xml attributes	Description
Api Name	The name of the API that you configured using the Service Builder.
ExposedName	The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the web service programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard application API names. The exposed name must start with a lower case letter.
AnnotateClassNames	Specify the AnnotateClassNames attribute on the webservicebeans.xml to indicate which apis need to have annotations.
Service element attributes	
Service Name	The name of the service that you configured using the Service Builder.
ExposedName	The name that is used in the Web Services Description Language (WSDL) file. This is the name that is used to call the web service programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard application service names. The exposed name must start with a lower case letter.The exposed name must start with a lower case letter.
AnnotateClassName	Specify the AnnotateClassNames attribute on the webservicebeans.xml to indicate which apis need to have annotations.
To expose an SDF service as a web service, you must also complete the procedure in “Exposing an SDF Service as a Web Service on JBoss” on page 164.	

6. Save the file.
7. Create the EAR as described in “Creating the EAR on JBoss” on page 166. The web services defined in the file will be built when you create the application EAR.

CAUTION: When building an ear with JAX-WS web services, you may encounter out of memory errors. JAX-WS web services will generate a java class for each element in the XSD, which can exceed the system memory. If the ear build runs out of memory, then unnecessary APIs should be removed from the webservicebeans.xml file. Additionally, APIs with potentially large output XMLs may be exposed through services with reduced XSDs to eliminate unwanted elements.

Tip: To help avoid out of memory issues when building the ear, group APIs into separate beans instead of exposing a large number of APIs in a single bean.

8. Later, if you want to add more APIs and services as JAX-WS web services, repeat these steps again.

Example of webservicessbeans.xml file - JBoss

The following is an example of a webservicebeans.xml file:

```

<WebServices>
  <Beans>
    <Bean BeanName="TheBeans"
BeanPackage="com.sterlingcommerce.jaxws.the.webservices" >
      <Apis>
        <Api Name="login" ExposedName="theLogin" />
        <Api Name="getLocaleList" ExposedName="theGetLocaleList" />
      </Apis>
    </Bean>
    <Bean BeanName="OtherBeans"
BeanPackage="com.sterlingcommerce.jaxws.other.webservices" >
      <Apis>
        <Api Name="getLocaleList" ExposedName="otherGetLocaleList" />
      </Apis>
      <Services>
        <Service AnnotateClassNames="true" Name="testWS" ExposedName="testWS" />
        <Service Name="testWS2" ExposedName="testWS2" />
      </Services>
    </Bean>
  </Beans>
</WebServices>

```

Exposing an SDF Service as a Web Service on JBoss

About this task

To expose an SDF service as a web service, you first have to copy the input and output xsds for the API that is exposed in the SDF service and edit them.

Note: If you want to overwrite the input or output XSD for any API, you must expose the API as an SDF service and then expose the SDF service as a Web service.

Procedure

1. Open the SDF and create a synchronous service that has an API node.
2. Select the desired API.
3. Save the SDF service.
4. Copy the input and output XSD files for the API you exposed in the SDF from the following location:

```
<INSTALL_DIR>/xapidocs/api_javadocs/XSD
```

to:

```
<INSTALL_DIR>/extensions/webservices
```

5. Rename the files to match the name and exposed name that you give them in webservicebeans.xml.

Use the following format:

```
<name>_<exposed_name>_input.xsd
```

```
<name>_<exposed_name>_output.xsd
```

Results

This is also the procedure you would use to extend a table, with this additional step: after copying and renaming the files, edit them to reference the new columns/API inputs/outputs.

Configuring a JAX-WS Web Services Client (UNIX/Linux Only) on JBoss

Sterling Selling and Fulfillment Foundation provides a client generator for use with JAX-WS web services.

CAUTION:

The client generator is supported on UNIX and Linux only. It is not supported on Windows.

For JBoss, the client generator accesses the WSDL directly from the server.

Note: If you are modifying the XSD for any API or service which has been exposed as a web service, you must rebuild the web services client. Additionally, each time you extend the database through customization or upgrade, you must run the xsdgenerator, and then rebuild the ear and client. See *Sterling Selling and Fulfillment Foundation: Customizing APIs* for more information about using the xsdgenerator, and see *Sterling Selling and Fulfillment Foundation: Extending the Database* for more information about extending the database. See “Building the Enterprise Archive (EAR) Package on JBoss” on page 158 for information about building the ear.

Running the JAX-WS Client Generator on JBoss Procedure

To invoke the generator, run the buildjaxclient.sh script from the <INSTALL_DIR>/bin directory. Include the properties required for your application server type from the following list when running the script:

Property

Description

-DBEAN_NAME

Required for all application servers. Matches the beanname on the server. Note that the name of the service might be different, depending on the application server being used.

-DAPPSERVER

Required. Valid values are websphere, weblogic, or jboss.

-DSERVER_URL

Required for all application servers. The http URL of the server up to, but not including, the bean name.

The following is an example of a command for JBoss:

```
bin/buildjaxclient.sh -DBEAN_NAME=ParticipantBean -DAPPSERVER=jboss  
-DSERVER_URL=http://00.00.00.00:8080/SIXBeanXapiJaxWS
```

Results

Running the script creates a new folder, jaxwsclient, that contains the compiled classes.

Sample Code for BeanService Classes - JBoss

Sample code is generated in the same directory as the bean and BeanService class. You can use this sample as a reference for invoking the bean methods.

The sample file has the same name as the service class but with "Sample" appended to the end of the name.

For example, SMCFSBeanService would have an accompanying sample called SMCFSBeanServiceSample class.

```
jaxwsclient/output/ParticipantBeanService/com/sterlingcommerce  
/jaxws/participant/webservices/ParticipantBeanServiceSample
```

The class generates a method for each API you expose. The content will require editing.

```
public void testlogin( ) throws Exception {  
    com.sterlingcommerce.documentation.ycp.login.input.Login input  
    = new com.sterlingcommerce.documentation.ycp.login.input.Login();  
    /*  
    Insert custom code here to set values on the input object.  
    */  
    com.sterlingcommerce.documentation.ycp.login.output.Login  
    returnValue = b.login(env,input);  
    /*  
    Insert custom code here to retrieve values from the return object.  
    */  
}
```

Including Custom Classes on JBoss

About this task

When deploying Sterling Selling and Fulfillment Foundation as Web service on JBoss, if you have developed custom Java classes (user exits, event handlers, and so forth) you need to deploy them in order for them to be available.

To ensure that your custom classes get invoked, do the following:

Procedure

1. Create a JAR file with all your custom classes.
2. Place this JAR file in a folder structure based on the package name. For more information about packaging and deploying jar files, see the *Sterling Selling and Fulfillment Foundation: Customization Basics*.
3. Rebuild the EAR as described in "Creating the EAR on JBoss."

Results

These classes are automatically included in the smcfs.ear file.

Creating the EAR on JBoss

About this task

Enterprise Archives are built using an ANT (buildEAR.xml) that accepts the following targets:

Main Target

Description

create-ear

Creates the EAR files:

- smcfs.ear - the Sterling Selling and Fulfillment Foundation application EAR file

- `smcfsdocs.ear` - the Sterling Selling and Fulfillment Foundation documentation EAR file

create-doc-ear

Creates `smcfsdocs.ear` - the Sterling Selling and Fulfillment Foundation documentation EAR

Note: To successfully build the application EAR file in 64-bit JBoss, use the following memory parameters:

```
-XX:MaxPermSize=768m -Xmx2048m -Xms2048m
```

You can set these memory parameters during installation or in the `sandbox.cfg` file after installation. For more information about setting them during installation, refer to “Creating a Silent Installation File - Windows” on page 54. For setting them after installation, refer to the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

Procedure

To create an application EAR file for JBoss 5.x, run the following command from the `<INSTALL_DIR>/bin` directory:

```
./buildear.sh (.cmd for Windows) -Dappserver=jboss  
-Dwarfiles=smcfs,sma,sbc -Dearfile=smcfs.ear -Djbossmajorversion=5 create-ear
```

To create an application EAR file for JBoss 4.x, run the following command from the `<INSTALL_DIR>/bin` directory:

```
./buildear.sh (.cmd for Windows) -Dappserver=jboss  
-Dwarfiles=smcfs,sma,sbc -Dearfile=smcfs.ear -Djbossmajorversion=4 create-ear
```

Note: While creating the EAR for JBoss, if the parameter `'jbossmajorversion'` is not provided, then the default value for parameter will be set as 5.

Note: Sterling Selling and Fulfillment Foundation supports the RPC encoded or document literal style and usage of invocation for Web services. When choosing style and usage for JBoss, pass the following in the ear command line:

```
-Djboss-java2wsdl-style=<rpc|document>
```

Running this command creates the `smcfs.ear` and `smcfsdocs.ear` files in the `<INSTALL_DIR>/external_deployments/` directory. It also puts three war files into the `smcfs.ear`:

- `smcfs.war` - the Sterling Selling and Fulfillment Foundation application war file
- `sma.war` - the System Management Administrator application war file
- `sbc.war` - the IBM Sterling Business Center application war file

Note: You can add the following options to the end of the above `buildear` commands:

- `-Dnowebservice=true` if you do not want to use Web services. If you do want to use Web services, see the “Building Web Services on JBoss” on page 159.
- `-Ddevmode=true` if you want to use the HTTP API Tester in the development environment.
- `-Dnodocear=true` if you want to skip the documentation build.
- `-Dsupport.multi.war=true` if you want to copy all the UI jars to `<WAR>/WEB-INF/lib`. The UI jars will be copied based on the entry in `DCL.xml`.

Results

For more information about the System Management Administrator (SMA) see the *Sterling Selling and Fulfillment Foundation: System Management and Administration Guide*. For more information about IBM Sterling Business Center, see the *Business Center: Item Administration Guide* and *Business Center: Pricing Administration Guide*.

For more information about JBoss, you can access the JBoss documentation at <http://www.jboss.org/community/wiki/jbossapplicationserverofficialdocumentationpage>.

Precompiling the WAR File on JBoss

See the *Sterling Selling and Fulfillment Foundation: Performance Management Guide* for instructions on how to precompile the WAR on JBoss. There are settings that must be configured before you create the EAR file.

Deploying the Enterprise Archive (EAR) on JBoss

Sterling Selling and Fulfillment Foundation provides support for deploying Multiple EARs (Enterprise Archives) on a single application server. On the same application server, you can:

- Deploy different customizations of the same or different versions of the application, or
- Deploy different versions of the same application

Multiple EARs or context roots require additional memory for the application server JVM. Testing has shown that the deployment of a second IBM EAR file requires 2.5 - 3.5 times the memory of a single EAR. Supporting two deployments may require up to 2.5 GB of heap space and 1.2 GB of permanent space.

During installation, you can use JVM-specific arguments to avoid out-of-memory errors. For more information, see the *Sterling Selling and Fulfillment Foundation: Properties Guide* descriptions of `ADDITIONAL_ANT_JAVA_TASK_ARGS` and `ADDITIONAL_ANT_COMPILER_TASK_ARGS`.

For information about JVM tuning on your application server, see the general and application server-specific JVM chapters in the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

For more information about how to implement multiple EAR files on the same application server, see the *Sterling Selling and Fulfillment Foundation: Customization Basics*.

Deploy your newly created `smcfs.ear` file as described in your application server documentation, using the `deployEARjboss.xml` ant script.

1. Stop the application server.
2. Copy the ear file to the deployment directory on the application server (the JBoss installation directory is `<servername>/deploy`).
3. Restart the application server.
4. Log in.

To verify the Sterling Selling and Fulfillment Foundation installation:

1. Restart your application server.

2. Start Internet Explorer.
3. To access the Application Console:
 - a. Access `http://<hostname>:<port>/smcfs/console/login.jsp`.
 - b. When prompted, enter your Login ID and Password.
4. To access IBM Sterling Business Center:
 - a. Access the IBM Sterling Business Center login page by setting the enterprise appropriately. For more information about setting the enterprise while logging in to IBM Sterling Business Center, see “Setting an Enterprise for Logging In to IBM Sterling Business Center” on page 173.
 - b. When prompted, enter your Login ID and Password.

Configuring DataSource Connection Pooling on WebLogic, WebSphere, and JBoss

You can specify an external datasource and connection pool instead of the default connection pooling supplied with Sterling Selling and Fulfillment Foundation to manage your database connections. The configuration process includes these high-level tasks:

- Create and configure any datasources and connection pools necessary on your application server.
- Define the datasource in Sterling Selling and Fulfillment Foundation using the `jdbc` datasource parameter.
- Edit the `-dvendor` option in the application server configuration. This is done in the application server software.

Configuring Data Source Connection Pooling for Single Schema Installations

About this task

To configure a datasource and connection pool for an instance of Sterling Selling and Fulfillment Foundation that was installed to use a single database schema:

Procedure

1. Create the connection pool and datasource on your application server. See your application server documentation for specific configuration and operating instructions.
2. Install Sterling Selling and Fulfillment Foundation.
3. In a text editor, open (or create) the `customer_overrides.properties` file in the `<install_dir>\properties` folder.
4. In `customer_overrides`, add the line for your database from the following list:
 - For DB2, add `jdbcService.db2Pool.datasource=<datasourceName>`
 - For Oracle, add `jdbcService.oraclePool.datasource=<datasourceName>`
 - For Microsoft SQL Server, add `jdbcService.mssqlPool.datasource=<datasourceName>`
 where `<datasourceName>` is the logical name of the external datasource.
5. Save the `customer_overrides` file and run `setupfiles.sh/cmd`.
6. Build and deploy the EAR.
7. Update the application server configuration to use `-Dvendor=<appserver>`
 – where the value of `<appserver>` is `weblogic`, `jboss`, or `websphere`.

8. Start the application server.

Results

Depending on your application server, you can find more information about -Dvendor and other properties in the application server startup scripts in “Configuring an Oracle WebLogic Application Server - Overview” on page 127, “Configuring a WebSphere Application Server - Overview” on page 141, or “Configuring a JBoss Application Server” on page 156.

Configuring Data Source Connection Pooling for Multiple Schema Installations

About this task

In a multiple schema installation, you can have four types of schema: Metadata, Statistics, System Configuration, and Transaction/Master.

- You must configure a datasource on your application server for each schema and colony you have. For example, you could have a datasource named `metadata_ds` for the METADATA schema; a datasource named `config_ds` for the CONFIG schema, and so on.
- When multiple schemas are defined, the schemas must all use either datasource or direct JDBC calls - they cannot use both. Sterling Selling and Fulfillment Foundation does not support 2-mode (datasource & direct JDBC calls). For example, you could not have schema "DEFAULT_METADATA" use a datasource pool, and schema "DEFAULT_BI_STAGING" use a direct JDBC call. They would both have to use the same method.
- Datasources for the Statistics, System Configuration, and Transaction/Master schema can be configured using either the System Management Administrator (SMA) or the `manageDBpool` API. To configure the datasource for a Metadata schema, you must use the `manageDBpool` API.
- At least three data sources must be specified for multischema installations.
- If not set, the datasource name defaults to pool id.
- For more information about Multischema database parameters, see “Running a Silent Installation in Multischema Mode” on page 59

To configure datasources for multiple schema:

Procedure

1. Create the connection pools and datasources for each schema on your application server. See your application server documentation for specific configuration and operating instructions.
2. Install the application.
3. In a text editor, open (or create) the `customer_overrides.properties` file in the `<install_dir>\properties` folder.
4. In `customer_overrides`, add the line for your database from the following list:
 - For DB2, add `jdbcService.db2Pool.datasource=<datasourceName>`
 - For Oracle, add `jdbcService.oraclePool.datasource=<datasourceName>`
 - For Microsoft SQL Server, add `jdbcService.mssqlPool.datasource=<datasourceName>`
where `<datasourceName>` is the logical name of the external datasource.
5. Save the `customer_overrides` file and run `setupfiles.sh/cmd`.

6. Use the System Management Administrator (SMA) or manageDBpool API to add the datasource attribute to the selected pools.

See the *Sterling Selling and Fulfillment Foundation: Multi-Tenant Enterprise Guide* for more information about the SMA.

Note: You cannot add the datasource attribute for the metadata pool in the SMA. You must use the manageDBpool API instead.

7. Build and deploy the EAR.
8. Update the application server configuration to use `-Dvendor=<appserver>`
– where the value of `<appserver>` is `weblogic`, `jboss`, or `websphere`.
9. Start the application server.

Results

Depending on your application server, you can find more information about `-Dvendor` and other properties in the application server startup scripts in “Configuring an Oracle WebLogic Application Server - Overview” on page 127, “Configuring a WebSphere Application Server - Overview” on page 141, or “Configuring a JBoss Application Server” on page 156.

Configuring a Restrictive Cookie Path - Overview

When multiple applications are deployed on the same domain and the restrictive cookie path is not set, the user may be automatically logged off from the application when one application sends information to another application. For example, in IBM Sterling Business Center, you will automatically be logged off in the following scenarios:

- The IBM Sterling Web application and the IBM Sterling Business Center application are deployed on the same domain, and you click the **Preview Item Details** link in the IBM Sterling Business Center application.
- The image server and the IBM Sterling Business Center application are deployed on the same domain, and you click the **Related Tasks** link in the IBM Sterling Business Center application.
- The Visual Modeler and the IBM Sterling Business Center application are deployed on the same domain, and you click the **Launch Visual Modeler** link in the IBM Sterling Business Center application.

Configuring a Restrictive Cookie Path on WebLogic

About this task

To set the restrictive cookie path in Oracle WebLogic, complete the following:

Procedure

1. Extract the `weblogic.xml` file from the war package where you want to add the restrictive cookie path.
2. Copy the extracted `weblogic.xml` file to the following location:
`<INSTALL_DIR>/extensions/<WAR Package>`
where `<WAR Package>` is the package for the deployed application. For example, this would typically be `smcfs`, `swc`, `sbc`, and `sfs` for the Sterling Selling and Fulfillment Foundation, IBM Sterling Web, IBM Sterling Business Center, and IBM Sterling Field Sales applications, respectively.

3. Add the following to the `weblogic.xml` file:

```
<session-descriptor>
  <session-param>
    <param-name>CookiePath</param-name>
    <param-value>.<context-path></param-value>
  </session-param>
</session-descriptor>
```

where `<context-path>` is the context path for the deployed application. For example, this would typically be `/smcfs`, `/swc`, `/sbc`, and `/sfs` for the Sterling Selling and Fulfillment Foundation, IBM Sterling Web, IBM Sterling Business Center, and IBM Sterling Field Sales applications, respectively.

4. Rebuild the EAR file.

Configuring a Restrictive Cookie Path on WebSphere Procedure

1. In the WebSphere Administration Console, navigate to the **Session Manager** → **Cookie** tab.
2. In the **Cookie** tab, set **Cookie Path** to the context path of your application.
For example, this would typically be `/smcfs`, `/swc`, `/sbc`, and `/sfs` for the Sterling Selling and Fulfillment Foundation, IBM Sterling Web, IBM Sterling Business Center, and IBM Sterling Field Sales applications, respectively.

Configuring a Restrictive Cookie Path on JBoss

By default, JBoss sets the restrictive cookie path; therefore, no additional configuration is required.

Setting the Client Character Display

When displaying special characters, such as for various languages, the client computer must be configured to display these characters.

For Unicode characters to display correctly in the Application Console, each Windows client must be configured.

1. To configure a client machine, select Control Panel > Regional and Language Options.

Clearing Browser Caches

About this task

Once Sterling Selling and Fulfillment Foundation is ready for deployment, each user must clear the browser caches on their client machines before launching Sterling Selling and Fulfillment Foundation.

To clear the browser cache:

Procedure

1. From the Windows start menu, select Settings > Control Panel > Internet Options. Choose the General tab, and in the Temporary Internet Files inner panel, choose the Delete Files button. The Delete Files dialog displays.
2. Enable the Delete All Offline Content option. Then click OK, and click OK once more.
3. Close the Internet Properties window.

Clearing the Java Plugin Cache

About this task

Once Sterling Selling and Fulfillment Foundation is ready for deployment, each user must clear the Java Plugin caches on their client machines before launching Sterling Selling and Fulfillment Foundation.

To clear the Java plugin cache:

Procedure

1. From the Windows start menu, select Settings > Control Panel > Java Plugin and choose the Cache tab.
2. Click Clear JAR Cache.
3. Click OK.
4. Close the Java Plugin Control Panel window.

Statistics Monitoring

In order to measure throughput performance, runtime statistics can be gathered. Note that this feature and the data gathered by it in the YFS_STATISTICS_DETAILS table are only for use by IBM personnel, as any metric can change without notice.

In a production environment, you should leave statistics generation enabled to collect statistics data in 10 minute intervals (the default). You should also schedule statistic purges on a regular basis (for example, every two weeks).

Setting an Enterprise for Logging In to IBM Sterling Business Center

IBM Sterling Business Center supports enterprise-specific login. Therefore, when you log in to IBM Sterling Business Center, it is mandatory that along with your User ID and password, you set the enterprise code of the enterprise that you want to administer.

You should set the enterprise code as a request parameter in the login URL. Therefore, to log in to IBM Sterling Business Center, you should use the following URL:

```
http://<server>:<port>/sbc/sbc/launch.jsp?EnterpriseCode=<Enterprise_Code>
```

For example, if the enterprise code of the enterprise you want to administer is XYZ-123, use the following URL to log in to IBM Sterling Business Center:

```
http://<server>:<port>/sbc/sbc/launch.jsp?EnterpriseCode=XYZ-123
```

Chapter 16. Deploying and Updating the Rich Client Platform Applications

Before You Begin RCP Deployment

Before you start deploying a Rich Client Platform application you must have installed Sterling Selling and Fulfillment Foundation. For more information about installing Sterling Selling and Fulfillment Foundation, see Chapter 8, “Installing Sterling Selling and Fulfillment Foundation in UNIX and Linux Environments,” on page 67 or Chapter 7, “Installing Sterling Selling and Fulfillment Foundation in a Windows Environment,” on page 45.

Review the *Sterling Selling and Fulfillment Foundation: System Requirements Guide* for information about supported software versions specifically for RCP, including operating systems and browsers.

The components that are shipped as part of Sterling Selling and Fulfillment Foundation (that is, what is available when Sterling Selling and Fulfillment Foundation is installed) are:

- RCP Infrastructure plug-in compressed file
- RCP Foundation plug-in compressed file
- JREs for each of the supported operating systems. For more information about supported JREs and operating systems, see the *Sterling Selling and Fulfillment Foundation: System Requirements Guide*.
- Eclipse dependencies for each of the supported operating systems

After you install Sterling Selling and Fulfillment Foundation, you can view the directory structure, which contains:

- The <INSTALL_DIR>/platformrcp/<PLATFORM_VERSION> folder—This contains the Rich Client Platform files, plug-ins, or JREs.
- The <INSTALL_DIR>/platformrcp/<PLATFORM_VERSION>/rcpclient folder—This contains the Rich Client Platform plug-in and tools plug-in compressed files.
- The <INSTALL_DIR>/platformrcp/<PLATFORM_VERSION>/rcpdependencies folder—This contains the Rich Client Platform dependency directories for the supported operating systems. For example, linux-gtk, windows, and so forth. Each of these directories contains the supported JREs and Eclipse plug-ins, features, or files. Also, each of these directories contain the osversion.properties text file which provides information about the supported versions of the operating system.
- The <INSTALL_DIR>/repository/xapi/template/merged/api folder—This contains the API XML templates used by the Rich Client Platform.
- The <INSTALL_DIR>/properties folder—This is the location of the customer_overrides.properties file, which is used when enabling auto updates for the individual PCA. For more information about enabling auto updates, see “Deploying RCP - Applying Updates” on page 179

What is available when a Rich Client Platform-based PCA is installed?

When you install a Rich Client Platform-based PCA client, a compressed file that contains the Rich Client Platform application plug-ins or features is provided. For example, when you install the IBM Sterling Call Center application, the <INSTALL_DIR>/rcp/COM/rcpclient directory is automatically created. The com.zip file is stored in this directory, which contains the IBM Sterling Call Center-specific plug-ins or features.

Note: If deploying an application, such as IBM Sterling Call Center or IBM Sterling Store Inventory Management, also see the application Deployment Guide for information specific to that application for configuring the Rich Client Platform.

Deploying RCP - Creating the RCP_EXTN_FOLDER Folder

About this task

To maintain all SSL certificates, you must create a <RCP_EXTN_FOLDER> folder in which to store any new plug-ins and resource files that you create while extending Rich Client Platform-based PCA client application. The environment variable for this folder is RCP_EXTN_FOLDER.

The <RCP_EXTN_FOLDER> folder structure is better explained with an example as follows:

Procedure

1. Create an appropriate <RCP_EXTN_FOLDER> folder for storing the RCP extensions that you create when extending the Rich Client Platform-based PCA client application. For example, rcpextnworkarea folder.

Note: You can create the <RCP_EXTN_FOLDER> folder in any directory outside the <INSTALL_DIR> directory.

2. Under the <RCP_EXTN_FOLDER> folder, create the following directories:
 - libs
 - plugins
 - resources
 - truststore
3. In the <RCP_EXTN_FOLDER>/plugins directory, store all new plug-ins that you created for extending the screens.
4. In the <RCP_EXTN_FOLDER>/resources directory, store the locations.ycfg file, rcpssecureapis.xml file (if necessary), localized bundle and theme files, and localized icons. The ant script creates the resources.jar file and copies the contents of the resources folder into this jar file. After copying the contents, the resources.jar file is copied into the Rich Client Platform plug-in.
5. In the <RCP_EXTN_FOLDER>/truststore directory, store the SSL trust certificates that needs to be used when the client application is communicating with the server in secure mode. The SSL certificates are automatically copied by the ant script to the correct folder in the Rich Client Platform plug-in.
6. Create the jasper folder within the <RCP_EXTN_FOLDER>/libs directory.
7. Copy the following jasper libs needed for JasperReports to the <RCP_EXTN_FOLDER>/libs/jasper folder:
 - barbecue-1.1.jar

- commons-beanutils-1.5.jar
- commons-collections-3.2.jar
- commons-digester-1.7.jar
- commons-logging-1.0.2.jar
- iReport.jar
- itext-1.3.1.jar
- jasperreports-3.6.1.jar

These are third party files, which must be downloaded from a third party site. They are not provided with Sterling Selling and Fulfillment Foundation. For instructions on downloading jasper libs, see the <INSTALL_DIR>/xapidocs/code_examples/jasperreports/alert_readme.html file for instructions.

Deploying RCP- Caching Data Types Locally

About this task

To improve the system performance when logging into the Rich Client Platform-based PCA application, you must cache data locally in the client.

To cache data types locally:

Procedure

1. Copy the datatypes.xml file from the <INSTALL_DIR>/repository/datatypes folder, and yfsdatatypepemap.xml files from the <INSTALL_DIR>/repository/xapi/template/merged/resource directory to the <RCP_EXTN_FOLDER>/resources directory.
2. Create the extn directory under the <RCP_EXTN_FOLDER>/resources directory.
3. Copy the extended datatypes.xml and yfsdatatypepemap.xml from the extensions directory to the <RCP_EXTN_FOLDER>/resources/extn directory.

Deploying RCP - Configuring Locations

A location is synonymous to a geographic location. For example, store location, call center location, and so forth. Each location has an identifier associated with it, which uniquely identifies the appropriate geographical location.

To configure locations, you must define locations in the locations.ycfg file. For Sterling Selling and Fulfillment Foundation, the locations.ycfg.sample file is located in the <INSTALL_DIR>/platformrcp/<PLATFORM_VERSION>/rcpclient/com.yantra.yfc.rcp_<version> folder.

To configure locations, copy the locations.ycfg.sample file to create a new locations.ycfg file.

Note: If deploying an application, such as IBM Sterling Call Center or IBM Sterling Store Inventory Management, see the application Deployment Guide for information specific to that application for configuring locations.

Deploying RCP - Creating and Configuring a New locations.ycfg XML File

About this task

To configure a new locations.ycfg file:

Procedure

1. Copy the sample locations.ycfg file from the <INSTALL_DIR>/platformrcp/<PLATFORM_VERSION>/rcpclient/com.yantra.yfc.rcp_<version> folder and store it in the <RCP_EXTN_FOLDER>/resource directory.

where <RCP_EXTN_FOLDER> refers to the folder that you created for storing Rich Client Platform-based PCA client application extensions. For more information about creating the <RCP_EXTN_FOLDER> folder, see “Deploying RCP - Creating the RCP_EXTN_FOLDER Folder” on page 176.

2. Define new locations in the locations.ycfg file by using the information provided in the locations.ycfg.sample file, which contains proxy server and application server URL settings for various geographical locations.

Following is sample configuration data from the locations.ycfg.sample file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Locations>
  <Location id = "DEFAULT"
    proxyServer="yourproxyserver.com"
    proxyPort="8080"
    updateType = "pull">
    <Config Name = "DEFAULT"
      Protocol = "http"
      BaseUrl = "localhost"
      PortNumber = "7001"
      ApiUrl = "<WEB_APP_NAME>smcfs/RcpServlet"
      CompressionEnabled = "N"
    </Config>
  </Location>
  <Location id = "REMOTE"
    proxyServer="yourproxyserver.com"
    proxyPort="8080"
    updateType = "client">
    <Config Name = "IMAGE"
      Protocol = "http"
      BaseUrl = "localhost"
      PortNumber = "7001"
      ApiUrl = "/icons/rcp/$param1$.gif"
      CompressionEnabled = "N"
    </Config>
  </Location>
</Locations>
```

3. Define the Locations root element.
4. Define the Location element under the Locations root element with id such as DEFAULT, LOCAL, REMOTE, and so forth. You can configure the proxy server and application server URL settings for each location.

Note: You must have one Location element with the id attribute value set as "DEFAULT". This Location element must have a Config element whose Name attribute must have the value set as "DEFAULT".

Note: When you log in to a Rich Client Platform application using a particular location, the system checks whether or not the loaded location has a "DEFAULT" Config element defined for it. If the selected location has the "DEFAULT" Config element, the system loads the "DEFAULT" configuration. Otherwise, the system loads the "DEFAULT" configuration defined in the "DEFAULT" location.

For more information about location configuration settings, see “Deploying RCP - Configuring Location Settings” on page 183.

Deploying RCP - Modifying the locations.ycfg.sample XML File About this task

To modify the locations.ycfg.sample file:

Procedure

1. Copy the locations.ycfg.sample file from the <INSTALL_DIR>/platformrcp/<PLATFORM_VERSION>/rcpclient/com.yantra.yfc.rcp_<version> folder and store it in the <RCP_EXTN_FOLDER>/resource directory.
where <RCP_EXTN_FOLDER> refers to the folder that you created for storing Rich Client Platform-based PCA client application extensions. For more information about creating the <RCP_EXTN_FOLDER> folder, see “Deploying RCP - Creating the RCP_EXTN_FOLDER Folder” on page 176.
2. Rename the locations.ycfg.sample file to locations.ycfg file.
3. Modify the location configurations settings as needed. For information about location configuration settings, see “Deploying RCP - Configuring Location Settings” on page 183.

Deploying RCP - Localizing Bundle and Theme Files

You can localize the Rich Client Platform application’s locale-specific files based on the user’s locale. The Rich Client Platform supports the bundle and theme locale-specific files. All the Rich Client Platform application plug-ins contain the <Plug-in_id>_<name>.properties bundle file and <Plug-in_id>_<theme_name>.yhtm theme file. For more information about localizing bundle and theme files, see the *Sterling Selling and Fulfillment Foundation: Localization Guide*.

Deploying RCP - Enabling HTTPS

If you are using the HTTPS connection to communicate with the application server, copy all SSL (Secure Socket Layer) certificates in the truststore directory. For more information about the truststore directory, see “Deploying RCP - Creating the RCP_EXTN_FOLDER Folder” on page 176.

For more information about configuring connection settings for HTTPS connection, see “Deploying RCP - Configuring Connection Settings for HTTPS Connection” on page 187.

Deploying RCP - Applying Updates

About this task

The Rich Client Platform's update process is based on the timestamp of the files. In the <INSTALL_DIR>/properties/yfs.properties.in file, the yfs.rcp.pca.updates.dir property points to the directory where updates for the PCAs are located. The yfs.rcp.pca.updates.cache.dir property points to the local directory on the application server where updates for the PCAs can be cached.

To deploy updates for the Rich Client Platform application on a client:

Procedure

1. Modify the <INSTALL_DIR>/properties/customer_overrides.properties file to configure the following properties:
 - Configure the yfs.rcp.pca.updates.dir property by specifying the path of the directory where updates for the PCAs are located. The directory that you

specify can also be a shared directory on the network. For example, `yfs.yfs.rcp.pca.updates.dir = <INSTALL_DIR>/<PCA_UPDATES_DIR>` where `yfs.yfs.rcp.pca.updates.dir` is the property, `<INSTALL_DIR>` is the directory where you have installed Sterling Selling and Fulfillment Foundation, and `<PCA_UPDATES_DIR>` is the directory which contains individual updates folder for each Sterling Selling and Fulfillment Foundation PCA.

For example, if the root folder for PCA updates is maintained in the `<INSTALL_DIR>/<PCA_UPDATES_DIR>` directory, and for IBM Sterling Warehouse Management System, if the application identifier is `YFSSYS00011`, PCA code is `com20`, and operating system configuration is `win32.win32.x86`, the client searches for updates based on the application ID, PCA code, and operating system configuration. The IBM Sterling Warehouse Management System updates are maintained in the `<INSTALL_DIR>/<PCA_UPDATES_DIR>/YFSSYS00011/com20/win32.win32.x86` directory.

You can find the following resources in this directory:

- Rich Client Platform client plug-in
- IBM Sterling Warehouse Management System and related plug-ins
- Eclipse related plug-ins

Note: The JRE files are not updated.

- Configure the `yfs.yfs.rcp.pca.updates.cache.dir` property by specifying the path of the local directory on the application server where updates for PCAs need to be cached. For example, `yfs.yfs.rcp.pca.updates.cache.dir = <INSTALL_DIR>/<PCA_UPDATES_DIR>/<UPDATES_CACHE_DIR>`

Note: Make sure that the directory specified in the `yfs.rcp.pca.updates.dir` property is different from the directory specified in the `yfs.rcp.pca.updates.cache.dir` property.

For additional information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

2. Modify the `locations.ycfg` file to define the type of update you want to deploy on the client in the `updateType` attribute of the `Location` element. The Rich Client Platform supports two methods of deploying updates on the client: Client Pull and Push Updates. For more information about the different types of updates that the Rich Client Platform supports, see “Deploying RCP - Types of Updates.”

Deploying RCP - Types of Updates

The Rich Client Platform's update process is based on the timestamp of files. The Rich Client Platform supports two methods of deploying updates for a Rich Client application on the client:

- Client Pull or Automatic Update—Client Pull is the automatic way of deploying updates on the client. In this type of update, when a user logs in to a Rich Client Platform application, based on the location configuration settings, the client application automatically starts searching for updates in a background thread and installs them. Once all updates are downloaded successfully and installed, the user is authorized to restart the application.
- Push Updates or Manual Update—Push Updates is the manual way of deploying updates on the client. If you want to use push updates option, copy the contents of the update directory based on the client application which you want to update to the client machine. For example, if you have specified the

update directory as: <INSTALL_DIR>/<PCA_UPDATES_DIR>, copy the contents from the following directory to the client machine.

```
<INSTALL_DIR>/<PCA_UPDATES_DIR>/<PCA_APPLICATION_ID>/  
<PCA_APPLICATION_VERSION>/<OS_CONFIG>
```

where <INSTALL_DIR> refers to the directory where you have installed the Sterling Selling and Fulfillment Foundation.

<PCA_UPDATES_DIR> is the directory where updates are located,

<PCA_APPLICATION_ID> is the identifier of the client application for which you want to deploy updates using the Push Update method,

<PCA_APPLICATION_VERSION> is the version number of the client application, and <OS_CONFIG> refers to the <Windowing_System>.<OS>.<OS_ARCH> operating system.

Deploying RCP - Running the Ant Script

Run the application-specific ant script with the appropriate ant target as needed. The ant script is provided by the appropriate Rich Client Platform application. For example, if you want to deploy the Sterling Call Center, run the `buildcomapplication.xml` file.

The ant file contains multiple ant targets to generate the deployable folder for all unique combinations of the Operating System and Application such as `buildCOMForWindows`, `buildCOMForLinuxGTK`, `buildSOPForWindows`, and so forth.

For example, if you want to deploy the Sterling Call Center on Windows, run the following ant script from the <INSTALL_DIR> directory with an ant target:

For Windows:

```
sci_ant.cmd -f bin/buildcomapplication.xml buildCOMForWindows
```

For UNIX:

```
sci_ant.sh -f bin/buildcomapplication.xml buildCOMForWindows
```

Note: For this ant script to run, the following arguments or variables need to be exported:

- <INSTALL_DIR>—name of the folder where Sterling Selling and Fulfillment Foundation is installed.
- <RCP_EXTN_FOLDER>—specify the name of the <RCP_EXTN_FOLDER> folder that you created for storing Rich Client Platform-based PCA client application extensions. For more information about creating <RCP_EXTN_FOLDER> folder, see “Deploying RCP - Creating the RCP_EXTN_FOLDER Folder” on page 176.

After you run this ant script runtime the following resources or directory structure is created or generated for the call center application:

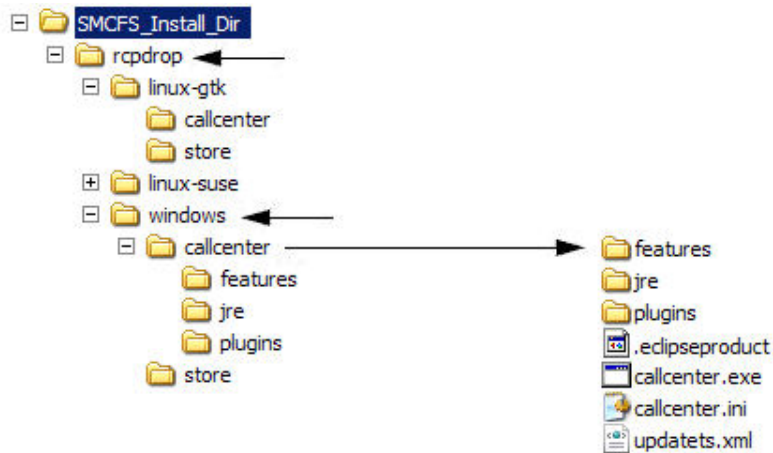


Figure 2. Sample Directory Structure for a Call Center Application

- The rcpdrop folder is created within the <INSTALL_DIR> directory. where <INSTALL_DIR> refers to the directory where you have installed Sterling Selling and Fulfillment Foundation.
- Based on the ant target that you specified, when you run the ant script, a folder for the operating system is created. For example, the windows folder is created if you specify buildCOMForWindows as the ant target to deploy the IBM Sterling Warehouse Management System on the Windows operating system.
- Under the windows folder, the <application> folder is created. For example, com. The com folder contains the required files and resources for the application that are to be built. These resources are accumulated from the following folders:
 - <INSTALL_DIR>/rcpclient/
 - <INSTALL_DIR>/rcpdependencies/windows
 - <INSTALL_DIR>/rcp/COM/rcpclient
 - <RCP_EXTN_FOLDER>
 where <INSTALL_DIR> refers to the directory where you have installed Sterling Selling and Fulfillment Foundation.
 <RCP_EXTN_FOLDER> refers to the folder that you created for the storing Rich Client Platform-based PCA client application extensions. For more information about creating the <RCP_EXTN_FOLDER> folder, see “Deploying RCP - Creating the RCP_EXTN_FOLDER Folder” on page 176.
- Also the updatets.xml file is created which is used by the Rich Client Platform to check for auto updates. The updatets.xml file contains a list of files that are present in the application. It also includes the timestamp for these files.

Note: The updatets.xml file is automatically generated by the ant script provided with Sterling Selling and Fulfillment Foundation for building a PCA Application.

Deploying RCP - Deploying Clients through a Remote Terminal

RCP clients can be deployed and accessed on a terminal server through a remote login from a client machine, by using Windows Terminal Server. Terminal Server is the server component of Terminal services. It authenticates clients, provides access to remote clients and also controls the level of access for each client. This service

uses the Remote Desktop Protocol (RDP), which enables a user to connect to the remote server (running Microsoft Terminal Services). Any client that supports RDP can be used as terminal client to connect to the server.

To run RCP clients through a remote login, add the Terminal Server and route your LAN through it.

Note: Only one user per client can log in to the terminal server.

Deploying RCP - Configuring Location Settings

About this task

Location configurations are defined in the `locations.ycfg` file. You can set different preferences for each location.

To define a new location configuration:

Procedure

1. Set the attributes of the Location element. For Location element attributes and their descriptions, see Table 15.

Table 15. Location Element Attribute List

Attribute	Description
id	Specify a unique identifier for the geographical location. For example, DEFAULT, REMOTE, LOCAL, and so forth.
proxyServer	Specify the unique proxy server used to connect to the internet, if applicable.
proxyPort	Specify the port number of the proxy server.
updateType	Set this attribute only when you are updating a Rich Client Platform application. Specify the mode of update you want to perform, if applicable. Valid update modes are: pull and push. For more information about update modes, see "Deploying RCP - Applying Updates" on page 179.

2. Define a Config element under the Location element to configure the connection settings. Each location has multiple Config elements. For example, DEFAULT, IMAGE_SMALL, IMAGE_BIG, and so forth. Using the Config element, define the various configuration settings. Set all attributes of the Config element to specify the application server URL you want to use. For more information about configuring connection settings, see "Deploying RCP - Connection Settings" on page 184.

Note: You must have one Location element with id attribute value as "DEFAULT" and this Location element must have a Config element whose Name attribute should have the value as "DEFAULT".

When you log into a Rich Client Platform application using a particular location, the system checks whether or not loaded location has a "DEFAULT" Config element defined for it. If the selected location has "DEFAULT" Config element, the system loads the "DEFAULT" configuration. Otherwise the system loads the "DEFAULT" configuration defined in the "DEFAULT" location.

A sample configuration data used to define a location configuration is as follows:


```

<Location id = "DEFAULT"
  proxyServer="proxy.sterling.com"
  proxyPort="8080">
  <Config Name = "DEFAULT"
    Protocol = "http"
    BaseUrl = "localhost"
    PortNumber = "7001"
    ApiUrl = "/<WEB_APP_NAME>smcfs/RcpServlet"
    CompressionEnabled = "N">
  </Config>
</Location>

```

When you start the Rich Client Platform application, the system reads the `locations.ycfg` file and loads the location information available in this file.

When you start the application for the first time, the Location Preferences window displays.

3. Select a location from the drop-down list.
4. Configure the proxy server settings, if applicable.

Results

Based on the location preferences, you are logged into the application.

Deploying RCP - Connection Settings

To connect to the application server, you must configure the Rich Client Platform application. In the `locations.ycfg` file, set the protocol, base URL, port number, API URL, and other attributes of the Config element. For Config element attributes, see Table 16. You can configure the connection settings for fetching images from the server or connecting to HTTPS.

Table 16. Config Element Attribute List

Attribute	Description
Name	Specify a unique name for the server configuration. For example, LOCAL, DEFAULT, and so forth.
Protocol	Specify the name of the protocol to use to communicate with the application server. For example, http or https. For more information about configuring connection settings for HTTPS protocol, see “Deploying RCP - Connection Settings for Fetching Images from the Server” on page 185.
BaseUrl	Specify the base URL path of the application server. For example localhost or 10.11.25.80 or www.myserver.com.
PortNumber	Specify the port number based on the protocol you specified. For example, 7001 or 7002.
ApiUrl	Specify the URL path of the application server where all APIs are stored. For example, /<WEB_APP_NAME>smcfs/RCPServlet. If you want to display images from the server, the URL path must contain \$param1\$ parameter. For more information about configuring connection settings for fetching images from the server, see “Deploying RCP - Connection Settings for Fetching Images from the Server” on page 185.
CompressionEnabled	If the data received from the server is in the compressed format, set the CompressionEnabled attribute to "Y". The Rich Client Platform supports only Gzip compression format. For more information about the supported compression format, see “Deploying RCP - Compression in the Rich Client Platform” on page 188.

A sample configuration data used to configure a server is as follows:

```
<Config Name = "DEFAULT"
    Protocol = "http"
    BaseUrl = "localhost"
    PortNumber = "7001"
    ApiUrl = "/<WEB_APP_NAME>smcfs/RcpServlet"
    CompressionEnabled = "Y">
</Config>
<Config Name = "LOCAL"
    Protocol = "http"
    BaseUrl = "localhost"
    PortNumber = "7001"
    HttpsPortNumber = "7002"
    ApiUrl = "/<WEB_APP_NAME>smcfs/RcpServlet"
    CompressionEnabled = "N">
</Config>
```

Note: You must have one location element with id attribute value as "DEFAULT". This location element must have a Config element with Name attribute value as "DEFAULT", which defines the DEFAULT URL for the connecting to the application server.

The Rich Client Platform application is initially launched by connecting to the server specified in the "DEFAULT" URL. You can define the URL at each command level, if applicable. If the command element in the <Plug-in_id>_commands.ycml file is not associated with the URL, the system considers the "DEFAULT" URL for that command.

Deploying RCP - Connection Settings for Fetching Images from the Server

About this task

You can configure the connection settings to fetch images from the server by setting the protocol, base URL, port number, API URL, and other attributes of the Config element in the locations.ycfg file. For Config element attributes, see Table 17. You can create more than one configurations to display different types of images.

Table 17. Config Element Attribute List

Attribute	Description
Name	Specify a unique name for the server configuration.
Protocol	Specify the name of the protocol to use to communicate with the application server. For example, http or https.
BaseUrl	Specify the base URL path of the server. For example localhost, 10.11.25.80, or www.myserver.com.
PortNumber	Specify the port number based on the protocol that you have specified. For example, 80.
ApiUrl	Specify the URL path of the server where all the images are stored. The URL path must contain \$param1\$ parameter. For example, /icons/rcp/\$param1\$.gif.
DefaultApiUrl	Specify the URL path of the image that displays if the image specified in the ApiUrl is not found, if applicable. For example, /icons/rcp/404.jpeg.

Note: You can create the following server configurations to fetch images of different types such as GIF, JPEG, PNG, and so forth:

- IMAGE
- IMAGE_SMALL
- IMAGE_MEDIUM
- IMAGE_BIG

Each location must have a server configuration named "IMAGE" which defines the URL for fetching images from the server. You can configure the "IMAGE" URL to get images of type GIF, JPEG, PNG, and so forth. All other server configurations are optional.

The sample configuration data that is used to configure server for displaying images is given below:

```
<Config-List>
  <Config Name = "IMAGE"
    Protocol = "http"
    BaseUrl = "localhost"
    PortNumber = "80"
    ApiUrl = "/icons/imgservlet/?file=$param1$"
  </Config>
  <Config Name = "IMAGE_SMALL"
    Protocol = "http"
    BaseUrl = "localhost"
    PortNumber = "80"
    ApiUrl = "/icons/rcp/$param1$.gif"
    DefaultApiUrl = "/icons/rcp/404.gif"
  </Config>
  <Config Name = "IMAGE_BIG"
    Protocol = "http"
    BaseUrl = "localhost"
    PortNumber = "80"
    ApiUrl = "/icons/rcp/$param1$.jpeg"
    DefaultApiUrl = "/icons/rcp/404.gif"
  </Config>
</Config-List>
```

For example, to get an image from the server using the `http://localhost:80/icons/imgservlet/?file=Y001` URL, define a Config element named IMAGE as shown in the sample code (above). To fetch an image from the server using the `http://localhost:80/icons/rcp/Y001.gif` URL, define a Config element named IMAGE_SMALL as shown in the sample code (above). In both the cases, the `$param1$` variable is replaced by the image's name.

Note: You can modify the Config element for the IMAGE URL. But ensure that you do not delete it.

For example, if you want to get an image for an OrderNo label:

Procedure

1. Set the source binding for the label as:

```
1b1OrderNo.setSourceBinding("ServerImageList:Images/Icons/RCP/Image1/@OrderNo");
```

where `1b1OrderNo` is the label name and `ServerImageList` is the namespace for the model.
2. Set the server image configuration for the label to display the image from the server as shown:

```
lblBindingData.setServerImageConfiguration(YRCConstants.IMAGE_SMALL);
```

where `lblBindingData` is the binding object and `IMAGE_SMALL` is the value of the `Name` attribute of the `Config` element, which is defined in the configuration file.

Results

When getting the image for the `lblOrderNo` label, the `$param1$` parameter is replaced by the value of the `OrderNo` attribute. If the value of the `OrderNo` attribute is "Y001", the image `Y001.gif` displays for the `lblOrderNo` label.

Deploying RCP - Configuring Connection Settings for HTTPS Connection

About this task

To configure the connection settings to communicate with application servers:

Procedure

1. In the `locations.ycfg` file when defining the connection settings, set the value of `Protocol` attribute of the `Config` element as "https". Also, specify the port number for the HTTPS protocol in the `PortNumber` attribute of the `Config` element. For more information about configuring the connection settings, see "Deploying RCP - Connection Settings" on page 184.
2. By default, during handshaking, if there is a mismatch between the URL's hostname and the server's identification hostname, the Rich Client Platform allows the HTTPS connection.
3. Copy all SSL or public key certificates required for configuring an HTTPS connection in the `truststore` directory under the `extensions` folder that you created as shown:

```
<RCP_EXTN_FOLDER>/truststore.
```

where `<RCP_EXTN_FOLDER>` refers to the folder that you created for storing the Rich Client Platform-based PCA client application extensions. For more information about creating `<RCP_EXTN_FOLDER>` folder, see "Deploying RCP - Creating the `RCP_EXTN_FOLDER` Folder" on page 176.

A trusted Certificate Authority (CA) like VeriSign issues these security certificates. For more information about SSL or security certificates, see "Deploying RCP - Security Certificates" on page 188.

Deploying RCP - Configuring Connection Settings for Context-Sensitive Help

To access the Context-Sensitive Help, configure the connection settings by setting the protocol, base URL, port number, and API URL attributes of the `Config` element in the `locations.ycfg` file. For `Config` element attributes, see Table 17 on page 185.

The following sample configuration data can be used to configure the connection settings for accessing the Context-Sensitive Help:

```
<Config Name = "HELP"
  Protocol = "http"
  BaseUrl = "localhost"
  PortNumber = "80"
  ApiUrl =
```

```
"/smcfsdocs/yfscommon/online_help/$param1/wwhelp/wwhimpl/c  
ommon/html/wwhelp.htm?context=$param2_userguide&topic=$  
param3"  
</Config>
```

Here \$param1 refers to locale, \$param2 refers to module id, and \$param3 refers to help topic id.

Deploying RCP - Security Certificates

An SSL certificate or public key certificate is a certificate that uses a digital signature to bind a public key with an identity information such as the name of the person or an organization, address, and so forth. An SSL certificate has information about the owner of the certificate, the usage of the certificate, validity details, resource location or web site address, e-mail address and the certificate ID of the person who certifies (signs) this information. SSL certificates are used for secure communication over the HTTPS protocol.

Whenever a client needs to verify the authenticity of an SSL server, the SSL certificate used by the server needs to be signed by the Certificate Authority that is already trusted by the client. The well-known certificate authorities such as Thawte and VeriSign serve as an authoritative, trusted third party for authentication. They sign the SSL certificates that are used when dealing with sensitive information or services. If these SSL certificates are signed by a trusted authority, it is possible to verify the identity of a server by supplying the SSL certificate.

Deploying RCP - Compression in the Rich Client Platform

The Rich Client Platform enables you to send and receive compressed data to and from the application server. When you enable compression, the Rich Client Platform enables bidirectional compression.

Benefits

- The bidirectional compression helps in reducing the traffic in both directions as only the XML data is passed to an API or service. For example, input XMLs and output templates passed to an API or service.
- The compression is most useful for applications that rely more on multiple API calls because it avoids multiple trips to and from the application server.
- There is minimal overhead in performing compression. For example, when an XML file size is large, we can reduce the size of the data by about 90%.

Note: The Rich Client Platform supports the Gzip compression format.

Note: The Rich Client Platform does not support compression of images or compressed files when fetching images or extracting updates from the server.

To enable compression, in the `locations.ycfg` file, you must set the value of the `CompressionEnabled` attribute of the `Config` element to "Y". These settings are done when you are configuring the connection settings for a Rich Client Platform application. For more information about configuring connection settings, see "Deploying RCP - Connection Settings" on page 184.

Deploying RCP - Creating Server-Side Commands Without Running the Application

If the RCP PCA application is run with the `yfs.rcp.devmode` property set to "FALSE" in the `yfs.properties` file, ensure that the following command file is present in the server:

```
( <runtime>\templates\com.yantra.yfc.rcp\commands\<APP_ID>\ )
```

The name of the command file can be either `commands.ycml` or `commands_<COMMANDS_VERSION>.ycml`. The `<COMMANDS_VERSION>` value is read from the `client.properties` file. If this value is not available, then the `commands.ycml` file will be used.

To create the merged command file, the following utility class is provided:

```
com.yantra.yfc.rcp.internal.YRCCommandsMergeUtils in  
Platform/rcpclient/com.yantra.yfc.rcp_1.0.0/yrui.jar
```

Invoke this utility class using the following arguments:

- `rcpClientDir=<RCP_CLIENT_DIR>`
- `rcpCommonDir=<RCP_COMMON_DIR>`
- `destDir=<DEST_DIR>`
- `commandsDirs=<PCA_COMMANDS_DIR >`
- `extnCommandsDir=<EXTN_COMMANDS_DIR>`
- `applicationId=<APP_ID>`

Here,

- `<RCP_CLIENT_DIR>` refers to the `rcpClient` directory.
- `<RCP_COMMON_DIR>` refers to the `rcpCommon` directory.
- `<DEST_DIR>` refers to the directory for the merged files.
- `<PCA_COMMANDS_DIR>` refers to the directory containing the comma-separated values of all the PCA commands.
- `<EXTN_COMMANDS_DIR>` refers to the directory containing the directories for each of the extension plugins that contain all the command (.ycml) files. The directory name of each plugin should be the plugin ID of that plugin.
- `<APP_ID>` refers to the application ID.

The following files must be present in the class path when running the Java class:

- `yrui.jar`
- `eclipse Plugins jars`
- `resources.jar`

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2011. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2011.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center[®], Connect:Direct[®], Connect:Enterprise[®], Gentran[®], Gentran[®]:Basic[®], Gentran:Control[®], Gentran:Director[®], Gentran:Plus[®], Gentran:Realtime[®], Gentran:Server[®], Gentran:Viewpoint[®], Sterling Commerce[™], Sterling Information Broker[®], and Sterling Integrator[®] are trademarks or registered trademarks of Sterling Commerce[™], Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Index

A

- access control 129, 144
- Activesync 103
- agent criteria
 - override 122
- authentication
 - using JAAS 6
 - using LDAP 6

B

- backing up
 - databases 37
- backupScriptGen.xml 37
- barcode printing
 - WebLogic 129
 - WebSphere 143
- browser
 - clearing cache 172

C

- capacity planning 19
- catalog name for database 49
- clearing browser 172
- client character 172
- client characters
 - displaying special characters 172
 - displaying Unicode character 172
- cold boot
 - reinstallation 105
- COM+ (extended component object model) 10
- component object model. 10
- copying WMS label formats 100
- custom classes deployment 138, 152, 166

D

- Database catalog name 49
- database name 49
- database schema creation (single schema) 114
- database security 12
 - credit card encryption 12
- database sizing 19
 - capacity planning 19
 - disk estimation
 - DOM module 19
 - WMS module 21
 - future disk estimation 25
- database user privileges 28, 42
 - DB2 34
 - oracle
 - administrative user 28, 42
 - application user 29, 43
 - Oracle 28, 42
- database verification (single schema) 114

databases

- backup and restore 37
- configuring 37
- DB2 8.1 Fixpack 5 33
- DB2 Fixpack 32
- Oracle 30, 41
- Oracle (Windows) 49
- DatabaseSecurity 12
- DB2 8.1 Fixpack 5 databases 33
- DB2 database
 - sizing. See Also database sizing 27, 32, 41
- DB2 Fixpack databases 32
- deploying
 - custom classes (WebLogic) 138, 152, 166
- deploying Selling and Fulfillment Foundation 127
- deployment architecture
 - analyzing
 - authentication mechanism 6
 - current security infrastructure 6
 - data encryption 6
 - network topology 6
- disk estimation
 - DOM module 19
 - estimation methodology 19
 - steps involved 21
 - future requirements 25
 - WMS and DOM module
 - steps involved 22
 - WMS module 21
 - estimation methodology 21
 - WMS module only
 - steps involved 24
- DOM module
 - disk space estimation 21
- DOM module disk estimation 19

E

- EFrame_indexadds.sql 116
- EFrame_indexdrops.sql 116
- EFrame_sequence.sql 116
- EFrame_tablechanges.sql 116
- EFrame_TableChanges.sql 115
- EFrame_tabledrops.sql 116
- EFrame_TextIndexAdds.sql 116
- EFrame_TextIndexDrops.sql 116
- EFrame_TextIndexModify.sql 116
- EFrame_TextIndexUpdates.sql 116
- EFrame_UpdateQueries.sql 117
- EJB (Enterprise JavaBeans) 8
- ejbdeploy.sh 142
- enterprise JavaBeans. See EJB 8
- estimates of database 19

F

- factory defaults 119
- installer restart file 119

G

- generating EJB stubs 142
- graph display on WebLogic 129
- graph display on WebSphere 143

H

- history database 20, 22
- HTTP
 - in-memory session replication 130
- HTTP (hypertext transfer protocol) 8
- HTTPS (hypertext transfer protocol secure) 9
- hypertext transfer protocol secure. See HTTPS 9
- hypertext transfer protocol. See HTTP 8

I

- installation checklist 1
- installation parameters
 - database catalog name 49
- installing
 - DB2 32
 - Microsoft SQL Server 38
 - Oracle 41
 - Sterling Sensitive Data Capture Server 91
 - Windows 51
- installing (from command line)
 - Linux 74, 77
 - UNIX 74, 77
- installing (from silent install file)
 - Linux 85, 89
- installing (from Windows) 59
- installing application server 15
- installing database schema (single schema) 114
- installing database software
 - DB2. See DB2 32
 - Microsoft SQL Server. See Microsoft SQL Server 38
 - Oracle. See Oracle 41
- installing database software (Windows) 27, 37
- installing DB2 32
- installing jasperreports 100
- installing Microsoft SQL Server 38
- installing mobile application 103
- installing Oracle 41
- installing print server 99
- installing Sterling Supply Chain Applications language pack 93
- installing weighing scale 101

- Installing MSSQL2005 38
- Internet Explorer security
 - adding trusted web site 14
 - browser settings 12
- invoking
 - custom classes 138, 152, 166

J

- JAAS (Java Authentication and Authorization Service) 6
- jasperreports
 - installation 100
- Java Authentication and Authorization Service. See JAAS 6
- Java messaging service. See JMS 8
- Java naming and directory interface. See JNDI 9
- java plugin
 - clearing cache 173
- Java protocol
 - disabling 8
 - EJB 8
 - HTTP 8
 - JMS 8
- Java protocols
 - securing 9
 - COM+ 10
 - EJB 9
 - HTTP API tester 10
- JavaProtocolSecurityMeasures 7
- JavaServer pages. See JSP 9
- JBoss
 - using Web services 159
- jcrt.jar 128
- JMS (Java messaging service) 8
- JNDI (Java naming and directory interface) 9
- jnet.jar 128
- JSP (JavaServer Pages) 9
- jsse.jar 128
- JVM settings
 - for WebSphere 143

L

- LAN (local area network) 7
- language pack
 - context-sensitive help 93
 - creating ear 96
 - creating EAR 96
 - deploying EAR 96
 - installing 93
 - loading factory defaults 93, 94
 - loading language pack translations 94, 95
 - switching base language 96
- languages, supported 96, 97
- LDAP (lightweight directory access protocol) 6
- lightweight directory access protocol. See LDAP 6
- Linux
 - installing (from command line) 74, 77

- Linux (*continued*)
 - installing (from silent install file) 85, 89
 - silent installation file 80
 - silent installation file parameters 81
 - UNIX accounts 68
- LLM (loftware label manager) 99
- loading factory defaults, database factory defaults 119
- local area network. See LAN 7
- locales 96, 97
- localization 96, 97
- loftware label manager 99
 - bar code label 99
 - defining printers 100
 - designing label 99
 - for UNIX systems 100
 - supported modes 99
- log files
 - best practices, security 11
- LPM (loftware print server) 99

M

- Microsoft SQL Server database
 - enabling named pipes 39
 - enabling TCP/IP 39
 - production environment
 - running scripts 39
- mobile application
 - Activesync for copying files 103
 - installation 103
- Mobile Application
 - installation
 - PocketPC mobile terminal 103
- mobile installation
 - cold boot reinstallation 105
 - installing libiconv
 - HP-UX Itanium 108
 - steps to install 103

N

- named pipes 39
- network topology
 - deployment over internet 7
 - deployment over LAN 7
 - deployment over VPN 7
- networked WMS module disk estimation 21
- non-English language 96, 97

O

- operating system permissions 11
- Oracle
 - copying JDBC driver 28
 - installation 27, 41
 - running create instance 27, 41
- Oracle database
 - production environment
 - creating views 27, 41
- Oracle databases 30, 41
- Windows 49

P

- Pocket PC mobile terminal
 - reinstalling on cold boot 105
- PocketPC mobile terminal 103
- post installation recommendations 11
- pre-compiling JSP files 127
- Preparing the Database Server 27, 37
- print server
 - configuration utility 99
 - installing loftware components 99
- print server installation 99
- procuring storage 20, 22

R

- recommended data encryption
 - SSL 128-bit 6
 - VPN 3DES 6
- remote method invocation. See RMI 9
- restoring
 - databases 37
- retention policy 20
- RMI (remote method invocation) 9
- running
 - Microsoft SQL Server scripts 39
 - Oracle scripts 34

S

- security
 - database security 12
 - deployment architecture 5
 - Internet Explorer settings 12
 - Java protocol See Java protocol 7
 - web security See web security 10
- security plan
 - creating 5
- Selling and Fulfillment Foundation
 - deploying 127
- session replication 130
 - including session descriptors 130
- setting up
 - dbverify utility (single schema) 114
 - Oracle scripts 30
 - Web services 131, 145, 159
 - WebLogic scripts 127
- SID 49
- silent installation file
 - Linux 80
 - parameters for Linux 81
 - parameters for UNIX 81
 - parameters for Windows 55
 - UNIX 80
 - Windows 54
- sma.war 130, 144, 158
- smcfs.war 130, 144, 158
- smcfsejb.jar 130, 144
- smcfswsbe.jar 130, 144, 158
- stack traces
 - eliminating 129
- Sterling Sensitive Data Capture Server 91
- System Console 144

T

- TCP/IP 39
- time-triggered transactions
 - override agent criteria 122
- transaction data
 - truncating 121
- transactional database 20, 22
- truncating transaction data 121
- types of order 20

U

- UNIX
 - accounts 68
 - installing (from command line) 74, 77
 - installing (from silent install file) 85, 89
 - silent installation file 80
 - silent installation file parameters 81
- UNIX accounts
 - Linux 68
 - UNIX 68
- Utilities
 - development 121
 - transaction data truncation tool 121
 - runtime 122
 - agent server 122
 - agent trigger 122
 - Integration server 122
 - setting up 123

V

- verifying database
 - extra objects 117
- verifying database (single schema) 114
- virtual private network. See VPN 7
- VPN (virtual private network) 7
- VT220 Mobile Terminal 107
 - installing 107
 - steps to install 108

W

- web security 10
 - routing 11
 - session security 11
 - web server executables 11
 - writing log files 11
- Web services 131, 145, 159
- WebLogic
 - creating EAR
 - file descriptors 139
 - steps involved 139, 153, 167
 - eliminating stack traces 129
 - setting client character display 172
 - setting HTTP in-memory session replication 130
 - using Web services 131
- WebLogic application server
 - setting up 127, 141
- weblogic.xml 130

- WebSphere
 - avoiding warning messages 141
 - configuring JVM settings 143
 - invoking EJBs 142
 - making EJB calls 142
 - specifying memory parameters 141
 - using Web services 145
- weighing scale
 - installation 101
 - steps to install 101
- Win CE 5.0 mobile terminal
 - reinstalling on cold boot 107
- WinCE mobile terminal 103
- WinCE mobile terminals
 - reinstalling on cold boot 106
- Windows
 - installing 51
 - silent installation file 54
 - silent installation file parameters 55
- WMS and DOM module
 - disk space estimation 22
- WMS installation
 - defining printers 100
- WMS module only
 - disk space estimation 24

Y

- yantraejb.jar 142
- yantrawebservices.war 130, 144



Printed in USA