Sterling Commerce
An IBM Company

Selling and Fulfillment Foundation: Installation Guide

Release 8.5

Last updated in HF40

March 2011

Copyright Notice

STERLING COMMERCE SOFTWARE

Sterling Commerce, Inc.
4600 Lakehurst Court
Dublin, Ohio 43016-2000

Third-Party Software

Certain components of the Sterling Commerce Software are distributed on the same storage media as Third Party Software which are not listed above. Additional information for such Third Party Software components of the Sterling Commerce Software is located at: installdir/mesa/studio/plugins/SCI_Studio_License.txt.

Third Party Software which is included, or are distributed on the same storage media with, the Sterling Commerce Software where use, duplication, or disclosure by the United States government or a government contractor or subcontractor, are provided with RESTRICTED RIGHTS under Title 48 CFR 2.101, 12.212, 52.227-19, 227.7201 through 227.7202-4, DFAR 252.227-7013(c) (1) (ii) and (2), DFAR 252.227-7015(b)(6/95), DFAR 227.7202-3(a), FAR 52.227-14(g)(2)(6/87), and FAR 52.227-19(c)(2) and (6/87) as applicable.

Additional information regarding certain Third Party Software is located at installdir/SCI_License.txt.

Some Third Party Licensors also provide license information and/or source code for their software via their respective links set forth below:

http://danadler.com/jacob/

http://www.dom4j.org

This product includes software developed by the Apache Software Foundation (http://www.apache.org). This product includes software developed by the Ant-Contrib project (http://sourceforge.net/projects/ant-contrib). This product includes software developed by the JDOM Project (http://www.jdom.org/). This product includes code licensed from RSA Data Security (via Sun Microsystems, Inc.). Sun, Sun Microsystems, the Sun Logo, Java, JDK, the Java Coffee Cup logo, JavaBeans , JDBC, JMX and all JMX based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. All other trademarks and logos are trademarks of their respective owners.

THE APACHE SOFTWARE FOUNDATION SOFTWARE

The Sterling Commerce Software is distributed with or on the same storage media as the following software products (or components thereof) and java source code files: Xalan version 2.5.2, Cookie.java, Header.java, HeaderElement.java, HttpException.java, HttpState.java, NameValuePair.java, CronTimeTrigger.java, DefaultTimeScheduler.java, PeriodicTimeTrigger.java, Target.java, TimeScheduledEntry.java, TimeScheduler.java, TimeTrigger.java, Trigger.java, BinaryHeap.java, PriorityQueue.java, SynchronizedPriorityQueue.java, GetOpt.java, GetOptsException.java, IllegalArgumentException.java, MissingOptArgException.java (collectively, "Apache 1.1 Software"). Apache 1.1 Software is free software which is distributed under the terms of the following license:

License Version 1.1

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgement: "This product includes software developed by the Apache Software Foundation (http://www.apache.org)." Alternatively, this acknowledgement may appear in the software itself, if and whenever such third-party acknowledgements normally appear.

4. The names "Commons", "Jakarta", "The Jakarta Project", "HttpClient", "log4j", "Xerces "Xalan", "Avalon", "Apache Avalon", "Avalon Cornerstone", "Avalon Framework", "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without specific prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without the prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMIPLIED WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTIBILITY, AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTIAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. The GetOpt.java, GetOptsException.java, IlligalArgumentException.java and MissingOptArgException.java software was originally based on software copyright (c) 2001, Sun Microsystems., http://www.sun.com. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

The preceding license only applies to the Apache 1.1 Software and does not apply to the Sterling Commerce Software or to any other Third-Party Software.

The Sterling Commerce Software is also distributed with or on the same storage media as the following software products (or components thereof): Ant, Antinstaller, Apache File Upload Package, Apache Commons Beans, Apache Commons BetWixt, Apache Commons Collection, Apache Commons Digester, Apache Commons IO, Apache Commons Lang., Apache Commons Logging, Apache Commons Net, Apache Jakarta Commons Pool, Apache Jakarta ORO, Lucene, Xerces version 2.7, Apache Log4J, Apache SOAP, Apache Struts and Apache Xalan 2.7.0, (collectively, "Apache 2.0 Software"). Apache 2.0 Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in the following directory files for the individual pieces of the Apache 2.0 Software: installdir/jar/commons_upload/1_0/ CommonsFileUpload_License.txt,
installdir/jar/jetspeed/1_4/RegExp_License.txt,
installdir/ant/Ant_License.txt
<install>/jar/antInstaller/0_8/antinstaller_License.txt,
<install>/jar/commons_beanutils/1_7_0/commons-beanutils.jar (/META-INF/LICENSE.txt),
<install>/jar/commons_betwixt/0_8/commons-betwixt-0.8.jar (/META-INF/LICENSE.txt),

<install>/jar/commons_collections/3_2/LICENSE.txt,
<install>/jar/commons_digester/1_8/commons-digester-1.8.jar (/META-INF/LICENSE.txt),
<install>/jar/commons_io/1_4/LICENSE.txt,
<install>/jar/commons_lang/2_1/Commons_Lang_License.txt,
<install>/jar/commons_logging/1_0_4/commons-logging-1.0.4.jar (/META-INF/LICENSE.txt),
<install>/jar/commons_net/1_4_1/commons-net-1.4.1.jar (/META-INF/LICENSE.txt),
<install>/jar/smcfs/8.5/lucene-core-2.4.0.jar (/META-INF/LICENSE.txt),
<install>/jar/struts/2_0_11/struts2-core-2.0.11.jar (./LICENSE.txt),
<install>/jar/mesa/gisdav/WEB-INF/lib/Slide_License.txt,
<install>/mesa/studio/plugins/xerces_2.7_license.txt,
<install>/jar/commons_pool/1_2/Commons_License.txt,
<install>/jar/jakarta_oro/2_0_8/JakartaOro_License.txt,
<install>/jar/log4j/1_2_15/LOG4J_License.txt,
<install>/jar/xalan/2_7/Xalan_License.txt,
<install>/jar/soap/2_3_1/Apache_SOAP_License.txt

Unless otherwise stated in a specific directory, the Apache 2.0 Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to Apache 2.0 Software, nor other Third Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the Apache 2.0 Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

NOTICE file corresponding to the section 4 d of the Apache License, Version 2.0, in this case for the Apache Ant distribution. Apache Ant Copyright 1999-2008 The Apache Software Foundation. This product includes software developed by The Apache Software Foundation (http://www.apache.org/). This product includes also software developed by :

- the W3C consortium (http://www.w3c.org) ,

- the SAX project (http://www.saxproject.org)

The <sync> task is based on code Copyright (c) 2002, Landmark Graphics Corp that has been kindly donated to the Apache Software Foundation.

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., http://www.ibm.com.

- software copyright (c) 1999, Sun Microsystems., http://www.sun.com.

- voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.

NOTICE file corresponding to the section 4 d of the Apache License, Version 2.0, in this case for the Apache Lucene distribution. Apache Lucene Copyright 2006 The Apache Software Foundation. This product includes software developed by The Apache Software Foundation (http://www.apache.org/). The snowball stemmers in contrib/snowball/src/java/net/sf/snowball were developed by Martin Porter and Richard Boulton. The full snowball package is available from http://snowball.tartarus.org/

Ant-Contrib Software

The Sterling Commerce Software is distributed with or on the same storage media as the Anti-Contrib software (Copyright (c) 2001-2003 Ant-Contrib project. All rights reserved.) (the "Ant-Contrib Software"). The Ant-Contrib Software is free software which is distributed under the terms of the following license:

The Apache Software License, Version 1.1

DOM4J Software

The preceding license only applies to the Dom4j Software and does not apply to the Sterling Commerce Software, or any other Third-Party Software.

THE ECLIPSE SOFTWARE FOUNDATION

The Sterling Commerce Software is also distributed with or on the same storage media as the following software:

com.ibm.icu.nl1_3.4.4.v200606220026.jar, org.eclipse.ant.core.nl1_3.1.100.v200606220026.jar,
org.eclipse.ant.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.compare.nl1_3.2.0.v200606220026.jar,
org.eclipse.core.boot.nl1_3.1.100.v200606220026.jar,
org.eclipse.core.commands.nl1_3.2.0.v200606220026.jar,
org.eclipse.core.contenttype.nl1_3.2.0.v200606220026.jar,
org.eclipse.core.expressions.nl1_3.2.0.v200606220026.jar,
org.eclipse.core.filebuffers.nl1_3.2.0.v200606220026.jar,
org.eclipse.core.filesystem.nl1_1.0.0.v200606220026.jar,
org.eclipse.core.jobs.nl1_3.2.0.v200606220026.jar,
org.eclipse.core.resources.nl1_3.2.0.v200606220026.jar,
org.eclipse.core.runtime.compatibility.auth.nl1_3.2.0.v200606220026.jar,
org.eclipse.core.runtime.compatibility.nl1_3.1.100.v200606220026.jar,
org.eclipse.core.runtime.nl1_3.2.0.v200606220026.jar,
org.eclipse.core.variables.nl1_3.1.100.v200606220026.jar,
org.eclipse.debug.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.debug.ui.nl1_3.2.0.v200606220026.jar,
org.eclipse.equinox.common.nl1_3.2.0.v200606220026.jar,
org.eclipse.equinox.preferences.nl1_3.2.0.v200606220026.jar,
org.eclipse.equinox.registry.nl1_3.2.0.v200606220026.jar,
org.eclipse.help.appserver.nl1_3.1.100.v200606220026.jar,
org.eclipse.help.base.nl1_3.2.0.v200606220026.jar, org.eclipse.help.nl1_3.2.0.v200606220026.jar,
org.eclipse.help.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.apt.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.jdt.apt.ui.nl1_3.2.0.v200606220026.jar,
org.eclipse.jdt.core.manipulation.nl1_1.0.0.v200606220026.jar,
org.eclipse.jdt.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.jdt.debug.ui.nl1_3.2.0.v200606220026.jar,
org.eclipse.jdt.doc.isv.nl1_3.2.0.v200606220026.jar,
org.eclipse.jdt.doc.user.nl1_3.2.0.v200606220026.jar,
org.eclipse.jdt.junit4.runtime.nl1_1.0.0.v200606220026.jar,
org.eclipse.jdt.launching.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.nl1_3.2.0.v200606220026.jar,
org.eclipse.jdt.ui.nl1_3.2.0.v200606220026.jar,
org.eclipse.jface.databinding.nl1_1.0.0.v200606220026.jar,
org.eclipse.jface.nl1_3.2.0.v200606220026.jar, org.eclipse.jface.text.nl1_3.2.0.v200606220026.jar,
org.eclipse.ltk.core.refactoring.nl1_3.2.0.v200606220026.jar,
org.eclipse.ltk.ui.refactoring.nl1_3.2.0.v200606220026.jar,
org.eclipse.osgi.nl1_3.2.0.v200606220026.jar, org.eclipse.osgi.services.nl1_3.1.100.v200606220026.jar,
org.eclipse.osgi.util.nl1_3.1.100.v200606220026.jar, org.eclipse.pde.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.pde.doc.user.nl1_3.2.0.v200606220026.jar,
org.eclipse.pde.junit.runtime.nl1_3.2.0.v200606220026.jar,
org.eclipse.pde.nl1_3.2.0.v200606220026.jar, org.eclipse.pde.runtime.nl1_3.2.0.v200606220026.jar,
org.eclipse.pde.ui.nl1_3.2.0.v200606220026.jar,
org.eclipse.platform.doc.isv.nl1_3.2.0.v200606220026.jar,
org.eclipse.platform.doc.user.nl1_3.2.0.v200606220026.jar,

org.eclipse.rcp.nl1_3.2.0.v200606220026.jar, org.eclipse.search.nl1_3.2.0.v200606220026.jar,
org.eclipse.swt.nl1_3.2.0.v200606220026.jar, org.eclipse.team.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.team.cvs.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.team.cvs.ssh.nl1_3.2.0.v200606220026.jar,
org.eclipse.team.cvs.ssh2.nl1_3.2.0.v200606220026.jar,
org.eclipse.team.cvs.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.team.ui.nl1_3.2.0.v200606220026.jar,
org.eclipse.text.nl1_3.2.0.v200606220026.jar, org.eclipse.ui.browser.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.cheatsheets.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.console.nl1_3.1.100.v200606220026.jar,
org.eclipse.ui.editors.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.externaltools.nl1_3.1.100.v200606220026.jar,
org.eclipse.ui.forms.nl1_3.2.0.v200606220026.jar, org.eclipse.ui.ide.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.intro.nl1_3.2.0.v200606220026.jar, org.eclipse.ui.navigator.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.navigator.resources.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.presentations.r21.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.views.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.views.properties.tabbed.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.workbench.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.workbench.texteditor.nl1_3.2.0.v200606220026.jar,
org.eclipse.update.configurator.nl1_3.2.0.v200606220026.jar,
org.eclipse.update.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.update.scheduler.nl1_3.2.0.v200606220026.jar,
org.eclipse.update.ui.nl1_3.2.0.v200606220026.jar,
com.ibm.icu_3.4.4.1.jar,
org.eclipse.core.commands_3.2.0.I20060605-1400.jar,
org.eclipse.core.contenttype_3.2.0.v20060603.jar,
org.eclipse.core.expressions_3.2.0.v20060605-1400.jar,
org.eclipse.core.filesystem.linux.x86_1.0.0.v20060603.jar,
org.eclipse.core.filesystem_1.0.0.v20060603.jar, org.eclipse.core.jobs_3.2.0.v20060603.jar,
org.eclipse.core.runtime.compatibility.auth_3.2.0.v20060601.jar,
org.eclipse.core.runtime_3.2.0.v20060603.jar, org.eclipse.equinox.common_3.2.0.v20060603.jar,
org.eclipse.equinox.preferences_3.2.0.v20060601.jar, org.eclipse.equinox.registry_3.2.0.v20060601.jar,
org.eclipse.help_3.2.0.v20060602.jar, org.eclipse.jface.text_3.2.0.v20060605-1400.jar,
org.eclipse.jface_3.2.0.I20060605-1400.jar, org.eclipse.osgi_3.2.0.v20060601.jar,
org.eclipse.swt.gtk.linux.x86_3.2.0.v3232m.jar, org.eclipse.swt_3.2.0.v3232o.jar,
org.eclipse.text_3.2.0.v20060605-1400.jar,
org.eclipse.ui.workbench.texteditor_3.2.0.v20060605-1400.jar,
org.eclipse.ui.workbench_3.2.0.I20060605-1400.jar, org.eclipse.ui_3.2.0.I20060605-1400.jar,
runtime_registry_compatibility.jar, eclipse.exe, eclipse.ini, and startup.jar
(collectively, "Eclipse Software").
All Eclipse Software is distributed under the terms and conditions of the Eclipse Foundation Software User Agreement (EFSUA) and/or terms and conditions of the Eclipse Public License Version 1.0 (EPL) or other license agreements, notices or terms and conditions referenced for the individual pieces of the Eclipse Software, including without limitation "Abouts", "Feature Licenses", and "Feature Update Licenses" as defined in the EFSUA .

A copy of the Eclipse Foundation Software User Agreement is found at
<install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/notice.html,
<install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/plugins/notice.html,
<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux_x86/eclipse/notice.html, and
<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux_x86/eclipse/plugins/notice.html.

A copy of the EPL is found at
<install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/plugins/epl-v10.htm,
<install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/epl-v10.htm,
<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux.x86/eclipse/plugins/epl-v10.html, and
<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux.x86/eclipse/epl-v10.html.

The reference to the license agreements, notices or terms and conditions governing each individual piece of the Eclipse Software is found in the directory files for the individual pieces of the Eclipse Software as described in the file identified as installdir/SCI_License.txt.

These licenses only apply to the Eclipse Software and do not apply to the Sterling Commerce Software, or any other Third Party Software.

The Language Pack (NL Pack) piece of the Eclipse Software, is distributed in object code form. Source code is available at http://archive.eclipse.org/eclipse/downloads/drops/L-3.2_Language_Packs-200607121700/index.php. In the event the source code is no longer available from the website referenced above, contact Sterling Commerce at 978-513-6000 and ask for the Release Manager. A copy of this license is located at <install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/plugins/epl-v10.htm and

<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux.x86/eclipse/plugins/epl-v10.html.

The org.eclipse.core.runtime_3.2.0.v20060603.jar piece of the Eclipse Software was modified slightly in order to remove classes containing encryption items. The org.eclipse.core.runtime_3.2.0.v20060603.jar was modified to remove the Cipher, CipherInputStream and CipherOutputStream classes and rebuild the org.eclipse.core.runtime_3.2.0.v20060603.jar.

Ehcache Software

The Sterling Commerce Software is also distributed with or on the same storage media as the ehache v.1.5 software (Copyright © 2003-2008 Luck Consulting Pty. Ltd.) ("Ehache Software"). Ehcache Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/jar/smcfs/8.5/ehcache-1.5.0.jar (./LICENSE.txt).

The Ehcache Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to the Ehcache Software, nor other Third Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the Ehcache Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

EZMorph Software

The Sterling Commerce Software is also distributed with or on the same storage media as the EZMorph v. 1.0.4 software (Copyright © 2006-2008 Andres Almiray) ("EZMorph Software"). EZMorph Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/jar/ezmorph/1_0_4/ezmorph-1.0.4.jar (./LICENSE.txt).

The EZMorph Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to the EZMorph Software, nor other Third Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the EZMorph Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

Firebug Lite Software

The Sterling Commerce Software is distributed with or on the same storage media as the Firebug Lite Software which is free software distributed under the terms of the following license:

Copyright (c) 2008 Azer Koçulu http://azer.kodfabrik.com. All rights reserved.

ICE SOFTWARE

The Sterling Commerce Software is distributed on the same storage media as the ICE Software (Copyright © 1997 ICE Engineering, Inc./Timothy Gerard Endres.) ("ICE Software"). The ICE Software is independent from and not linked or compiled with the Sterling Commerce Software. The ICE Software is a free software product which can be distributed and/or modified under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License or any later version.

A copy of the GNU General Public License is provided at installdir/jar/jniregistry/1_2/ICE_License.txt. This license only applies to the ICE Software and does not apply to the Sterling Commerce Software, or any other Third Party Software.

The ICE Software was modified slightly in order to fix a problem discovered by Sterling Commerce involving the RegistryKey class in the RegistryKey.java in the JNIRegistry.jar. The class was modified to comment out the finalize () method and rebuild of the JNIRegistry.jar file.

Source code for the bug fix completed by Sterling Commerce on January 8, 2003 is located at: installdir/jar/jniregistry/1_2/RegistryKey.java. Source code for all other components of the ICE Software is located at http://www.trustice.com/java/jnireg/index.shtml.

The ICE Software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

JBOSS SOFTWARE

The Sterling Commerce Software is distributed on the same storage media as the JBoss Software (Copyright © 1999-2002 JBoss.org) ("JBoss Software"). The JBoss Software is independent from and not linked or compiled with the Sterling Commerce Software. The JBoss Software is a free software product which can be distributed and/or modified under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License or any later version.

A copy of the GNU Lesser General Public License is provided at:
<install_dir>\jar\jboss\4_2_0\LICENSE.html

This license only applies to the JBoss Software and does not apply to the Sterling Commerce Software, or any other Third Party Software.

The JBoss Software is not distributed by Sterling Commerce in its entirety. Rather, the distribution is limited to the following jar files: el-api.jar, jasper-compiler-5.5.15.jar, jasper-el.jar, jasper.jar, jboss-common-client.jar, jboss-j2ee.jar, jboss-jmx.jar, jboss-jsr77-client.jar, jbossmq-client.jar, jnpserver.jar, jsp-api.jar, servlet-api.jar, tomcat-juli.jar.

The JBoss Software was modified slightly in order to allow the ClientSocketFactory to return a socket connected to a particular host in order to control the host interfaces, regardless of whether the ClientSocket Factory specified was custom or note. Changes were made to org.jnp..server.Main. Details concerning this change can be found at http://sourceforge.net/tracker/?func=detail&aid=1008902&group_id=22866&atid=376687.

Source code for the modifications completed by Sterling Commerce on August 13, 2004 is located at: http://sourceforge.net/tracker/?func=detail&aid=1008902&group_id=22866&atid=376687. Source code for all other components of the JBoss Software is located at http://www.jboss.org.

JGO SOFTWARE

The Sterling Commerce Software is distributed with, or on the same storage media, as certain redistributable portions of the JGo Software provided by Northwoods Software Corporation under a commercial license agreement (the "JGo Software"). The JGo Software is provided subject to the disclaimers set forth above and the following notice:

U.S. Government Restricted Rights

The JGo Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (C)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor / manufacturer of the JGo Software is Northwoods Software Corporation, 142 Main St., Nashua, NH 03060.

JSLib Software

The Sterling Commerce Software is distributed with or on the same storage media as the JSLib software product (Copyright (c) 2003-2009 Mozdev Group, Inc.) ("JSLib Software"). The JSLib Software is distributed under the terms of the MOZILLA PUBLIC LICENSE Version 1.1. A copy of this license is located at <install>\repository\eardata\platform_uifwk_ide\war\designer\MPL-1.1.txt. The JSLib Software code is distributed in source form and is located at http://jslib.mozdev.org/installation.html. Neither the Sterling Commerce Software nor any other Third-Party Code is a Modification or Contribution subject to the Mozilla Public License. Pursuant to the terms of the Mozilla Public License, the following notice applies only to the JSLib Software (and not to the Sterling Commerce Software or any other Third-Party Software):

"The contents of the file located at http://www.mozdev.org/source/browse/jslib/ are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/MPL-1.1.html.

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozdev Group, Inc. code. The Initial Developer of the Original Code is Mozdev Group, Inc. Portions created by_Mozdev Group, Inc. are Copyright © 2003 Mozdev Group, Inc. All Rights Reserved. Original Author: Pete Collins <pete@mozdev.org>one Contributor(s):_____none listed________.

Alternatively, the contents of this file may be used under the terms of the ____ license (the "[___] License"), in which case the provisions of [___] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [___] License and not allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

The preceding license only applies to the JSLib Software and does not apply to the Sterling Commerce Software, or any other Third-Party Software.

Json Software

The Sterling Commerce Software is also distributed with or on the same storage media as the Json 2.2.2 software (Copyright © 2006-2008 Json-lib) ("Json Software"). Json Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/jar/jsonlib/2_2_2/json-lib-2.2.2-jdk13.jar.

This product includes software developed by Douglas Crockford (http://www.crockford.com).

The Json Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to the Json Software, nor other Third Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the Json Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

Purple Technology

The Sterling Commerce Software is distributed with or on the same storage media as the Purple Technology Software (Copyright (c) 1995-1999 Purple Technology, Inc.) ("Purple Technology Software"), which is subject to the following license:

Copyright (c) 1995-1999 Purple Technology, Inc. All rights reserved.

PLAIN LANGUAGE LICENSE: Do whatever you like with this code, free of charge, just give credit where credit is due. If you improve it, please send your improvements to alex@purpletech.com. Check http://www.purpletech.com/code/ for the latest version and news.

LEGAL LANGUAGE LICENSE: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors and the names "Purple Technology," "Purple Server" and "Purple Chat" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact server@purpletech.com.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND PURPLE TECHNOLOGY "AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR PURPLE TECHNOLOGY BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The preceding license only applies to the Purple Technology Software and does not apply to the Sterling Commerce Software, or any other Third Party Software.

Rico Software

The Sterling Commerce Software is also distributed with or on the same storage media as the Rico.js software (Copyright © 2005 Sabre Airline Solutions) ("Rico Software"). Rico Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/repository/eardata/platform/war/ajax/scripts/Rico_License.txt.

The Rico Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to the Rico Software, nor other Third-Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the Rico Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third-Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

Rhino Software

The Sterling Commerce Software is distributed with or on the same storage media as the Rhino js.jar (Copyright (c) 1998-2009 Mozilla.org.) ("Rhino Software"). A majority of the source code for the Rhino Software is dual licensed under the terms of the MOZILLA PUBLIC LICENSE Version 1.1. or the GPL v. 2.0. Additionally, some files (at a minimum the contents of toolsrc/org/Mozilla/javascript/toolsdebugger/treetable) are available under another license as set forth in the directory file for the Rhino Software.

Sterling Commerce's use and distribution of the Rhino Software is under the Mozilla Public License. A copy of this license is located at <install>/3rdParty/rico license.doc. The Rhino Software code is distributed in source form and is located at http://mxr.mozilla.org/mozilla/source/js/rhino/src/. Neither the Sterling Commerce Software nor any other Third-Party Code is a Modification or Contribution subject to the Mozilla Public License. Pursuant to the terms of the Mozilla Public License, the following notice applies only to the Rhino Software (and not to the Sterling Commerce Software or any other Third-Party Software):

"The contents of the file located at <install>/jar/rhino/1_7R1/js.jar are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/.

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Rhino code, released May 6, 1999. The Initial Developer is Netscape Communications Corporation. Portions created by the Initial Developer are Copyright © 1997-1999. All Rights Reserved. Contributor(s):_____none listed.

The preceding license only applies to the Rico Software and does not apply to the Sterling Commerce Software, or any other Third-Party Software.

Sun Microsystems

The Sterling Commerce Software is distributed with or on the same storage media

as the following software products (or components thereof): Sun JMX, and Sun JavaMail (collectively, "Sun Software"). Sun Software is free software which is distributed under the terms of the licenses issued by Sun which are included in the directory files located at:

SUN COMM JAR - <install>/Applications/Foundation/lib

SUN ACTIVATION JAR - <install>/ Applications/Foundation/lib

SUN JavaMail - <install>/jar/javamail/1_4/LICENSE.txt

The Sterling Commerce Software is also distributed with or on the same storage media as the Web-app_2_3.dtd software (Copyright © 2007 Sun Microsystems, Inc.) ("Web-App Software"). Web-App Software is free software which is distributed under the terms of the Common Development and Distribution License ("CDDL"). A copy of the CDDL is found in http://kenai.com/projects/javamail/sources/mercurial/show.

The source code for the Web-App Software may be found at: <install>/3rdParty/sun/javamail-1.3.2/docs/JavaMail-1.2.pdf

Such licenses only apply to the Sun product which is the subject of such directory and does not apply to the Sterling Commerce Software or to any other Third Party Software.

The Sterling Commerce Software is also distributed with or on the same storage media as the Sun Microsystems, Inc. Java (TM) look and feel Graphics Repository ("Sun Graphics Artwork"), subject to the following terms and conditions:

Sun grants you ("Licensee") a non-exclusive, royalty free, license to use, and redistribute this software graphics artwork, as individual graphics or as a collection, as part of software code or programs that you develop, provided that i) this copyright notice and license accompany the software graphics artwork; and ii) you do not utilize the software graphics artwork in a manner which is disparaging to Sun. Unless enforcement is prohibited by applicable law, you may not modify the graphics, and must use them true to color and unmodified in every way.

This software graphics artwork is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE GRAPHICS ARTWORK.

IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE GRAPHICS ARTWORK, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

If any of the above provisions are held to be in violation of applicable law, void, or unenforceable in any jurisdiction, then such provisions are waived to the extent necessary for this Disclaimer to be otherwise enforceable in such jurisdiction.

The preceding license only applies to the Sun Graphics Artwork and does not apply to the Sterling Commerce Software, or any other Third Party Software.

WARRANTY DISCLAIMER

This documentation and the Sterling Commerce Software which it describes are licensed either "AS IS" or with a limited warranty, as set forth in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

The Third Party Software is provided "AS IS" WITHOUT ANY WARRANTY AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. FURTHER, IF YOU ARE LOCATED OR ACCESSING THIS SOFTWARE IN THE UNITED STATES, ANY EXPRESS OR IMPLIED WARRANTY REGARDING TITLE OR NON-INFRINGEMENT ARE DISCLAIMED.

Without limiting the foregoing, the ICE Software and JBoss Software are distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Contents

5 Installing and Configuring Database Tier Software on UNIX/Linux

6 Installing and Configuring Database Tier Software on Windows

7 Installing Selling and Fulfillment Foundation in UNIX or Linux Environments

8 Installing Selling and Fulfillment Foundation in a Windows Environment

9 Installing Selling and Fulfillment Foundation Language Pack

14 Deploying Selling and Fulfillment Foundation

Preface

This manual explains how to install the various components of Selling and Fulfillment Foundation and contains information relevant to new installs and upgrades of Selling and Fulfillment Foundation. It also describes the major tools and components of Selling and Fulfillment Foundation and provides information about how to set them up in a typical implementation.

Intended Audience

This manual provides installation and administration information for individuals responsible for installing and maintaining Selling and Fulfillment Foundation.

Structure

This document contains the following sections:

Chapter 1, "Getting Started"

This chapter provides a high-level introduction to the tasks involved in installing Selling and Fulfillment Foundation.

Chapter 2, "System Requirements"

This chapter describes the hardware and software requirements for installing Selling and Fulfillment Foundation.

Chapter 3, "Creating a Security Plan"

This chapter provides security recommendations and guidelines.

Chapter 4, "Installing and Configuring Application Tier Software"

This chapter provides information directing you to installation instructions for specific application server software.

Chapter 5, "Installing and Configuring Database Tier Software on UNIX/Linux"

This chapter provides step-by-step instructions for installing and configuring your database server to run Selling and Fulfillment Foundation on UNIX or Linux operating systems.

Chapter 6, "Installing and Configuring Database Tier Software on Windows"

This chapter provides step-by-step instructions for installing and configuring your database server to run Selling and Fulfillment Foundation on Windows operating systems.

Chapter 7, "Installing Selling and Fulfillment Foundation in UNIX or Linux Environments"

This chapter provides step-by-step instructions for installing Selling and Fulfillment Foundation on UNIX or Linux operating systems.

Chapter 8, "Installing Selling and Fulfillment Foundation in a Windows Environment"

This chapter provides step-by-step instructions for installing Selling and Fulfillment Foundation on Windows operating systems.

Chapter 9, "Installing Selling and Fulfillment Foundation Language Pack"

This chapter provides step-by-step instructions for installing the Selling and Fulfillment Foundation language pack on each of the supported operating systems.

Chapter 10, "Installing the Print Server"

This chapter describes the Sterling WMS-specific settings for the installation and configuration of the Loftware Label Manager (LLM) and Loftware Print Server (LPS).

Chapter 11, "Installing the Weighing Scale"

This chapter describes how to install the weighing scale software used by the Sterling WMS.

Chapter 12, "Installing the Mobile Application"

This chapter describes how to install the Mobile Application on PocketPC, WinCE, and VT220 mobile devices.

Chapter 13, "Configuring Utilities"

This chapter describes how to configure the utilities provided with Selling and Fulfillment Foundation, such as the installation, runtime, and migration script files.

Chapter 14, "Deploying Selling and Fulfillment Foundation"

This chapter describes how to deploy Selling and Fulfillment Foundation on an application server.

Chapter 15, "Deploying and Updating the Rich Client Platform Applications"

This chapter explains how to deploy and update the Rich Client Platform applications such as Sterling Call Center and Sterling Store (SCCS) and Store Operations (SOP) Packaged Composite Applications (PCAs) in different geographical locations.

Selling and Fulfillment Foundation Documentation

For more information about the Selling and Fulfillment Foundation components, see the following manuals:

- *Selling and Fulfillment Foundation: Release Notes*
- *Selling and Fulfillment Foundation: Installation Guide*
- *Selling and Fulfillment Foundation: Upgrade Guide*
- *Selling and Fulfillment Foundation: Configuration Deployment Tool Guide*
- *Selling and Fulfillment Foundation: Performance Management Guide*
- *Selling and Fulfillment Foundation: High Availability Guide*

- *Selling and Fulfillment Foundation: System Management Guide*
- *Selling and Fulfillment Foundation: Localization Guide*
- *Selling and Fulfillment Foundation: Customization Basics Guide*
- *Selling and Fulfillment Foundation: Customizing APIs Guide*
- *Selling and Fulfillment Foundation: Customizing Console JSP Interface for End User Guide*
- *Selling and Fulfillment Foundation: Customizing the RCP Interface Guide*
- *Selling and Fulfillment Foundation: Customizing User Interfaces for Mobile Devices Guide*
- *Selling and Fulfillment Foundation: Customizing Web UI Framework Guide*
- *Selling and Fulfillment Foundation: Customizing Swing Interface Guide*
- *Selling and Fulfillment Foundation: Extending the Condition Builder Guide*
- *Selling and Fulfillment Foundation: Extending the Database Guide*
- *Selling and Fulfillment Foundation: Extending Transactions Guide*
- *Selling and Fulfillment Foundation: Using Sterling RCP Extensibility Tool Guide*
- *Selling and Fulfillment Foundation: Integration Guide*
- *Selling and Fulfillment Foundation: Product Concepts Guide*
- *Sterling Warehouse Management™ System: Concepts Guide*
- *Selling and Fulfillment Foundation: Application Platform Configuration Guide*
- *Sterling Distributed Order Management™: Configuration Guide*
- *Sterling Supply Collaboration: Configuration Guide*
- *Sterling Global Inventory Visibility™: Configuration Guide*
- *Catalog Management™: Configuration Guide*
- *Sterling Logistics Management: Configuration Guide*

- *Sterling Reverse Logistics™: Configuration Guide*
- *Sterling Warehouse Management System: Configuration Guide*
- *Selling and Fulfillment Foundation: Application Platform User Guide*
- *Sterling Distributed Order Management: User Guide*
- *Sterling Supply Collaboration: User Guide*
- *Sterling Global Inventory Visibility: User Guide*
- *Sterling Logistics Management: User Guide*
- *Sterling Reverse Logistics: User Guide*
- *Sterling Warehouse Management System: User Guide*
- *Selling and Fulfillment Foundation: Mobile Application User Guide*
- *Selling and Fulfillment Foundation: Business Intelligence Guide*
- *Selling and Fulfillment Foundation: Javadocs*
- *Sterling Selling and Fulfillment Suite™: Glossary*
- *Parcel Carrier: Adapter Guide*
- *Selling and Fulfillment Foundation: Multitenant Enterprise Guide*
- *Selling and Fulfillment Foundation: Password Policy Management Guide*
- *Selling and Fulfillment Foundation: Properties Guide*
- *Selling and Fulfillment Foundation: Catalog Management Concepts Guide*
- *Selling and Fulfillment Foundation: Pricing Concepts Guide*
- *Business Center: Item Administration Guide*
- *Business Center: Pricing Administration Guide*
- *Business Center: Customization Guide*
- *Business Center: Localization Guide*

Conventions

In this manual, Windows refers to all supported Windows operating systems.

The following conventions may be used in this manual:

Convention	Meaning
. . .	Ellipsis represents information that has been omitted.
< >	Angle brackets indicate user-supplied input.
`mono-spaced text`	Mono-spaced text indicates a file name, directory path, attribute name, or an inline code example or command.
/ or \	Slashes and backslashes are file separators for Windows, UNIX, and Linux operating systems. The file separator for the Windows operating system is "\" and the file separator for UNIX and Linux systems is "/". The UNIX convention is used unless otherwise mentioned.
<INSTALL_DIR>	User-supplied location of the Selling and Fulfillment Foundation installation directory. This is only applicable for Release 8.0 or later.
<INSTALL_DIR_OLD>	User-supplied location of the Selling and Fulfillment Foundation installation directory (for Release 8.0 or later). **Note:** This is applicable only for users upgrading from Release 8.0 or later.
<YANTRA_HOME>	User-supplied location of the Sterling Supply Chain Applications installation directory. This is only applicable for Releases 7.7, 7.9, and 7.11.
<YANTRA_HOME_OLD>	User-supplied location of the Sterling Supply Chain Applications installation directory (for Releases 7.7, 7.9, or 7.11). **Note:** This is applicable only for users upgrading from Releases 7.7, 7.9, or 7.11.

Convention	Meaning
<YFS_HOME>	For Releases 7.3, 7.5, and 7.5 SP1, this is the user-supplied location of the Sterling Supply Chain Applications installation directory. For Releases 7.7, 7.9, and 7.11, this is the user-supplied location of the <YANTRA_HOME>/`Runtime` directory. For Release 8.0 or above, the <YANTRA_HOME>/`Runtime` directory is no longer used and this is the same location as <INSTALL_DIR>.
<YFS_HOME_OLD>	This is the <YANTRA_HOME>/`Runtime` directory for Releases 7.7, 7.9, or 7.11. **Note:** This is only applicable for users upgrading from Releases 7.7, 7.9, or 7.11.
<ANALYTICS_HOME>	User-supplied location of the Sterling Analytics installation directory. **Note:** This convention is used only in the *Selling and Fulfillment Foundation: Business Intelligence Guide*.
<COGNOS_HOME>	User-supplied location of the IBM Cognos 8 Business Intelligence installation directory. **Note:** This convention is used only in the *Selling and Fulfillment Foundation: Business Intelligence Guide*.
<MQ_JAVA_INSTALL_PATH>	User-supplied location of the IBM WebSphere® MQ Java components installation directory. **Note:** This convention is used only in the *Selling and Fulfillment Foundation: System Manangement and Administration Guide*.
<DB>	Refers to Oracle®, IBM DB2®, or Microsoft SQL Server® depending on the database server.
<DB_TYPE>	Depending on the database used, considers the value oracle, db2, or sqlserver.

Note: The Selling and Fulfillment Foundation documentation set uses the following conventions in the context of the product name:

- Yantra is used for Release 7.7 and earlier.
- Sterling Supply Chain Applications is used for Releases 7.9 and 7.11.
- Sterling Multi-Channel Fulfillment Solution is used for Releases 8.0 and 8.2.
- Selling and Fulfillment Foundation is used for Release 8.5.

1

Getting Started

This chapter provides a high-level introduction and checklist for the tasks required to install Selling and Fulfillment Foundation.

1.1 Before You Begin

Before you begin installing 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 *Selling and Fulfillment Foundation: Performance Management Guide* which contains information that helps you optimize the performance of your Selling and Fulfillment Foundation.

1.2 Installation Checklist

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

During the installation and setup processes, you should also refer to the *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.

Table 1–1 Installation Checklist

Checklist
1. Ensure that you have the necessary system requirements to install and run Selling and Fulfillment Foundation.
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 (UNIX) or (Windows).
7. Size your database (UNIX) or (Windows).
8. Install the Selling and Fulfillment Foundation application (UNIX) or (Windows).
9. Install the Selling and Fulfillment Foundation language pack (optional)
10. Optionally install the print server.
11. Optionally install the weighing scale software.
12. Optionally Install the Mobile Application.
13. Configure the Selling and Fulfillment Foundation properties to use with the database, agent servers, LDAP servers, logging, and so forth. See the *Selling and Fulfillment Foundation: Properties Guide* for more information.
14. Configure the Selling and Fulfillment Foundation utilities for installation, runtime, migration, and production.
15. Set up the application server for use with Selling and Fulfillment Foundation on (WebLogic), (WebSphere), or (JBoss).
16. Build your Enterprise Archive (EAR) on (WebLogic), (WebSphere), or (JBoss).
17. Deploy the EAR to your application server as appropriate on (WebLogic), (WebSphere), or (JBoss).

Checklist
18. Optionally deploy and update the Sterling Rich Client applications.
19. Optionally run the configuration deployment tool to migrate your configuration data. For information about this tool, see the *Selling and Fulfillment Foundation: Configuration Deployment Tool Guide*.
20. Optionally log into the Business Center application, which is part of the Selling and Fulfillment Foundation installation. For more information about Business Center, see the *Business Center: Item Administration Guide* and the *Business Center: Pricing Administration Guide*.

1.3 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 Sterling Commerce. It provides online access to the 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 Sterling, the documentation set is updated dynamically with changes and hotfixes.

The Local Documentation Library is hosted locally by your enterprise. It provides access to the 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 hotfixes except through product upgrades. This documentation is always accessible, even if you choose to implement the Online Documentation Library.

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

Figure 1–1 Product Documentation Menu

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

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

For information about how to enable either Documentation Library, see the *Selling and Fulfillment Foundation: Properties Guide*.

Note: The context-sensitive online help that you can access by clicking the help button shown in Figure 1–1 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 Section 14.2.1, "Deploying the Context-Sensitive Help (WebLogic)", or Section 14.5.1, "Deploying the Context-Sensitive Help (WebSphere)", or Section 14.8.1, "Deploying the Context-Sensitive Help (JBoss)", as appropriate for your installation.

2

System Requirements

Selling and Fulfillment Foundation is an n-tier application, using a combination of application server, Web server, and database server software. This chapter lists all the supported operating systems and the required software used in the deployment of Selling and Fulfillment Foundation. Before installing Selling and Fulfillment Foundation, verify that you have already installed the applicable software listed in this chapter.

This chapter also provides the information required to complete Step as indicated on the Installation Checklist on page 1-1.

Minimum Requirements

This chapter describes the *minimum* supported options for which Selling and Fulfillment Foundation is already tuned for optimal performance. Your own results are derived from your specific hardware, data volumes, and user activities.

Obtaining Maximum Performance

For information about how to obtain the *maximum* performance from Selling and Fulfillment Foundation on this supported hardware and software, you must thoroughly read, evaluate, and apply any relevant recommendations described in the *Selling and Fulfillment Foundation: Performance Management Guide*. For example, Selling and Fulfillment Foundation predefines a set of indexes, but also expects your Database Administrator to monitor the system and add or remove indexes as necessary.

Note: To avoid data integrity issues, do not remove any *unique* indexes that are provided by Selling and Fulfillment Foundation.

2.1 Technical Stack Matrix

The Selling and Fulfillment Foundation's technical stack consists of the various tiered hardware and software required by Selling and Fulfillment Foundation. The technical stack consists of a specific application server, JDK, and database server. You can select various supported configurations from the matrix to create a supported technical stack by following these rules and instructions:

- Select one Database Server.
- Select one Application Server.
- Find a match for the operating system based on your selections of the database and application server.

  For example, you may use an Oracle 11G database server with WebSphere on one of the following operating system/processor combinations:

  - Red Hat Enterprise Linux 5.3 on an Intel 64 bit Xeon or AMD processor
  - SUSE Linux 10 SP1 on a 64 bit bit Xeon or AMD processor
- Although heterogeneous stacks are supported, when possible, one should consider a homogeneous stack to eliminate having to manage multiple vendors.
- Supported Oracle configurations are either single-node, or Real Application Clusters (RAC) 2-node.

2.2 IPv6 Certification

IPv6 certification is done using the Sterling Warehouse Management System components of Selling and Fulfillment Foundation. Figure 2–1 illustrates the IPv6 deployment for Selling and Fulfillment Foundation.

Figure 2–1 IPv6 Deployment for Selling and Fulfillment Foundation

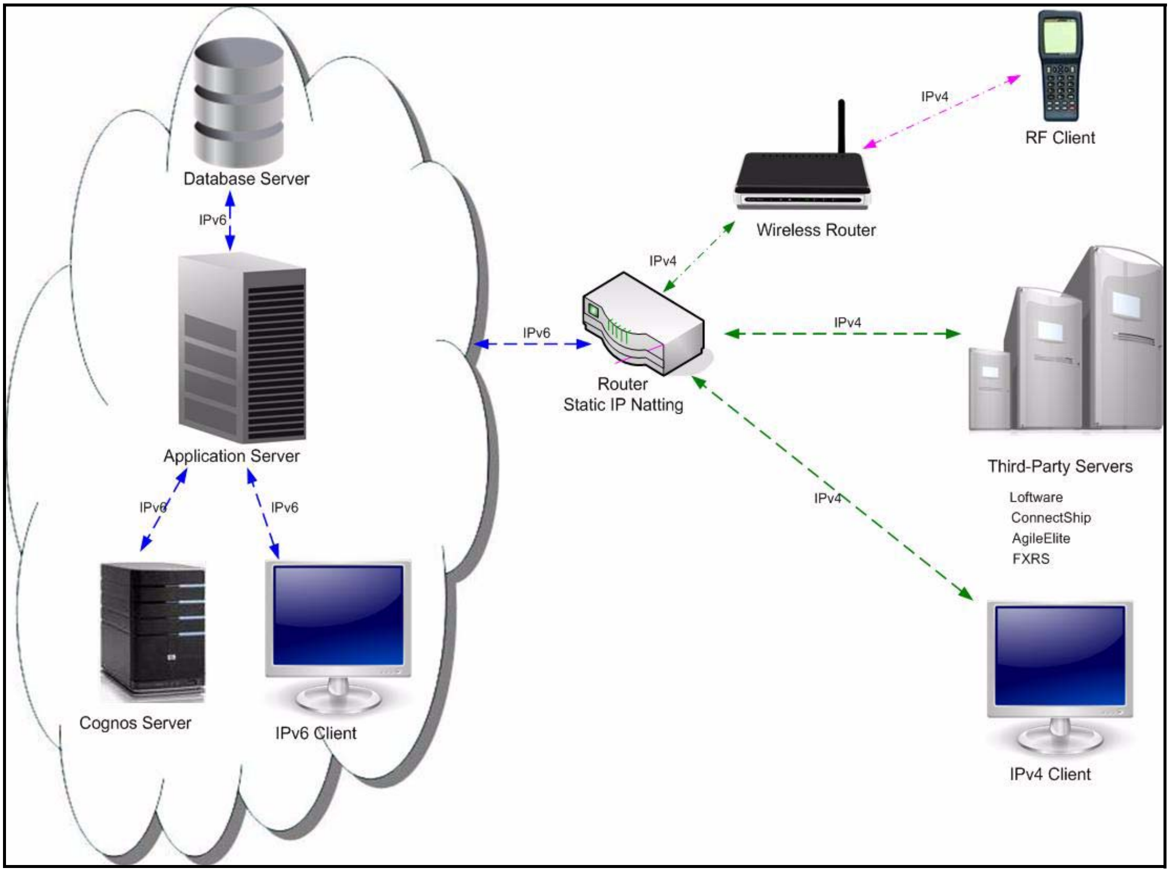


Table 2–1 and Table 2–2 describes the certified stacks of IPv6 deployment.

Table 2–1 IPv6 Deployment, Certified Stack 1

IPv6 Components	IPv4 Components
• Application Server - Weblogic 10.3 on Windows 2008 • Database - DB2 9.5 on AIX 6.1 • Client PC - Windows Vista - IE7	• Third-Party Software - Loftware 9.5, ConnectShip 6.2, AgileElite 6.5.34, FXRS 0776 • Printers - used by the Sterling Warehouse Management System • Client PC - Windows Vista - IE7 • Symbol MC9090

Table 2–2 IPv6 Deployment, Certified Stack 2

IPv6 Components	IPv4 Components
• Application Server - WebSphere 7.0 on Red Hat Enterprise Linux 5.3 • Database - DB2 9.5 on AIX 6.1 • Cognos 8.4 on Windows 2008 • Client PC - Windows Vista - IE7	• Third-Party Software - Loftware 9.5, ConnectShip 6.2, AgileElite 6.5.34, FXRS 0776 • Printers - used by the Sterling Warehouse Management System • Client PC - Windows Vista - IE7 • Symbol MC9090
Note: The Sterling Warehouse Management System reports were tested with the Client on IPv6 only.	

Note: For more information about IPv6 certification for all the third-party software you are using, refer to the respective vendor's documentation.

2.3 Database Tier Requirements

This section describes the minimum supported options for the database tier supported in Selling and Fulfillment Foundation. You can select one database server based on your choice of operating system as shown in Table 2–3, "Supported Database Tier".

Table 2–3 Supported Database Tier

Database Version	Operating System
Oracle 10g(10.2.0.3) **Oracle 11g(11.1.0.6)**	HP-UX 11i v3 on Itanium
	IBM AIX 6.1 IBM AIX 5.3 ML4
	Solaris 10 on SPARC and AMD Opteron
	Red Hat Enterprise Linux 5.3 / AP 64 bit Xeon or AMD processor
	SUSE Linux 10 SP1 on 64 bit Xeon or AMD processor
	Windows 2008 Enterprise / Standard Edition
DB2 9.1 FP3, DB2 9.5 FP4	IBM AIX 6.1 **only** on WebSphere and WebLogic 10 IBM AIX 5.3 ML4
*__Note:__ Oracle RAC is supported on 2-node configurations.	

Note: Support for Oracle Database does not include support for the Oracle Exadata platform.

2.4 Application Server Requirements

This section describes the minimum supported requirements for the agent and application server tier. You can select one application server based on your choice of operating system and Java Messaging Service (JMS) as shown in Table 2–4.

> **Note:** Install the JDK that is shipped with your application server, unless otherwise noted.

Table 2–4 Supported Application Server Tier

Application Server	Operating System	JMS	JDK
Oracle Weblogic 10.3	HP-UX 11i v3 on Itanium	Default JMS server that comes with the application server or TIBCO EMS 5.0, 5.1.2, 5.1.4	HP JDK 6.0.02 (64 bit)
	IBM AIX 6.1, 5.3 ML4	Default JMS server that comes with the application server or TIBCO EMS 5.0, 5.1.2, 5.1.4	IBM JDK 6.0 (64 bit)
	Solaris 10 on SPARC and AMD Opteron (Solaris with Fujitsu SPARC processor)	Default JMS server that comes with the application server or TIBCO EMS 5.0, 5.1.2, 5.1.4	Sun JDK 6.0 Update 14 (64 bit)
	SUSE Linux Enterprise 10 SP1 on 64 bit Xeon or AMD processor	Default JMS server that comes with the application server	JRockit 6.0 (R27.6.0-50) (64 bit)
	Red Hat Enterprise Linux 5.3/AP 64 bit Xeon or AMD processor	Default JMS server that comes with the application server or TIBCO EMS 5.0, 5.1.2, 5.1.4	JRockit 6.0 (R27.6.0-50) (64 bit)
	Windows 2008	Default JMS server that comes with the application server	Sun JDK 6.0 Update 14 or JRockit 6.0 (R27.6.0-50) (64 bit)

Table 2–4 Supported Application Server Tier

Application Server	Operating System	JMS	JDK
IBM Websphere 7.0.0.5 (plus APAR PK97427), IBM Websphere 6.1.0.11	IBM AIX 6.1, 5.3 ML4	IBM WebSphere MQ 6.0.2.5 or default messaging provider that comes with the application server or TIBCO EMS 5.0, 5.1.2, 5.1.4	IBM JDK 6.0 (Websphere 7.0.0.5) and IBM JDK 5 (Websphere 6.1.0.11)
	Solaris 10 on SPARC and AMD Opteron (Solaris with Fujitsu SPARC processor)	IBM WebSphere MQ 6.0.2.5 or default messaging provider that comes with the application server or TIBCO EMS 5.0, 5.1.2, 5.1.4	IBM JDK 6.0 (Websphere 7.0.0.5) and IBM JDK 5 (Websphere 6.1.0.11)
	Red Hat Enterprise Linux 5.3 /AP 64 bit on Xeon or AMD processor	IBM WebSphere MQ 6.0.2.5 or default messaging provider that comes with the application server or TIBCO EMS 5.0, 5.1.2, 5.1.4	IBM JDK 6.0 (Websphere 7.0.0.5) and IBM JDK 5 (Websphere 6.1.0.11)
	SUSE Linux Enterprise 10 SP1 on 64 bit Xeon or AMD-Opteron processor	IBM WebSphere MQ 6.0.2.5 or default messaging provider that comes with the application server	IBM JDK 6.0 (Websphere 7.0.0.5) and IBM JDK 5 (Websphere 6.1.0.11)
IBM WebSphere 7.0.0.5	Red Hat Enterprise Linux 5.4/AP 64 bit on Xeon or AMD processor	IBM WebSphere MQ 6.0.2.5 or IBM WebSphere MQ 7.0.1.2	IBM JDK 6.0 (WebSphere 7.0.0.5)
JBoss 4.3	Red Hat Enterprise Linux 5.3 /AP 64 bit on Xeon or AMD processor	JBoss 4.3, IBM WebSphere MQ 6.0.2.5, TIBCO EMS 5.0, 5.1.2, 5.1.4	Sun JDK 6.0 Update 14
	Solaris 10 on SPARC and AMD Opteron (Solaris with Fujitsu SPARC processor)	JBoss 4.3, TIBCO EMS 5.0, 5.1.2, 5.1.4	Sun JDK 6.0 Update 14

To achieve load balancing and failover, you can install the web server or proxy server based on the application server you choose.

2.5 Utility Requirements

Following are the utility requirements for installation.

> **Note:** For the utilities listed in this section, use the same version of JDK that you use for your application server.

> **Note:** For WebSphere, use the WebSphere JDK from the WebSphere Application Server with WebSphere Application Server jar files.

- **Runtime Utilities**
  - Integration Server
  - Agent Server
  - Agent Trigger
- **Installation Utilities**
  - Installer
  - loadFactoryDefaults
  - All steps after the Installer through ear precompilation (for example, merge, ear compilation)
- **Upgrade Utilities**
  - Migration Validator
  - migrator
- **Development Utilities**
  - Configuration Deployment Tool
  - Transaction Data Truncation Tool

2.6 Internet Browser and Plugin Requirements

The minimum requirements for the Internet browser and Java plugin are shown in Table 2–5, "Supported Browsers and the Selling and Fulfillment Foundation Client".

Table 2–5 Supported Browsers and the Selling and Fulfillment Foundation Client

Internet Browser	Operating System	JRE
(For Legacy HTML Framework) MS Internet Explorer 8, MS Internet Explorer 7, MS Internet Explorer 6 SP1	Windows XP, VISTA	Java Plugin 1.5.0_11
(For Business Center UI screens and Sterling Web) MS Internet Explorer 8, MS Internet Explorer 7, MS Internet Explorer 6 SP1, Mozilla Firefox 3.0, Mozilla Firefox 2.0	Not applicable	Java Plugin 1.5.0_11

Note: If you are running the Application Console and Applications Manager from the same browser window in MS Internet Explorer 7.0, the Console becomes disabled when you open a pop-up window in the Applications Manager. This is due to the way tabs and popups are handled in MS Internet Explorer 7.0. To enable the Console, close the pop-up window in the Applications Manager.

Note: By default, the Java plugin memory should be set to 128M. When using the Fulfillment Network Model, set the java plugin memory to 256M. For more information about the Fulfillment Network Model, see the *Sterling Distributed Order Management: Configuration Guide*.

Note: To avoid heap space errors when using the Applications Manager, it is recommended that you set the XmxNNN in the Java Plug-in Control Panel. Find the \Documents and Settings\<USER_NAME>\ApplicationData\Sun\Java\Deployment\deployment.properties file and add your parameter to the deployment.javapi.jre.<JRE_VERSION>.args line.

Note: For better visibility of the menu options in the Application Console, ensure that the dpi setting is 96.

Note: When you perform the following steps in a MS Internet Explorer 7.0 browser, Internet Explorer automatically redirects you to the tab pertaining to the most recent log in:

1. Log in as a user of a business application in a new Internet Explorer window.
2. Open a new tab in the same Internet Explorer window, and try to log in to the same business application, but as a different user.

If you want to log in as two different users of the same business application, use a new Internet Explorer window for the second log in. This rule applies even if you attempt to log in as two different users of two different business applications in the scenario described here.

2.7 Web Server Requirements

The following web servers are supported for Sterling Web:

- Apache HTTP Server 2.2
- Microsoft Internet Information Services (IIS) 7.0

2.8 Online Documentation Library

The following web browsers are supported for access to the Sterling-hosted Online Documentation Library:

- MS Internet Explorer 8, MS Internet Explorer 7, MS Internet Explorer 6
- Mozilla Firefox 3.0, Mozilla Firefox 2.0

2.9 Third-Party Software Requirements

The requirements for third-party systems such as Cognos Reports, Loftware Print Server, weighing scale, and so forth are provided in Table 2–6. The software mentioned in the table is supported for all of the operating systems unless otherwise noted.

Table 2–6 Supported Third-Party Software

Name	Version
Build tools	Apache ANT 1.7, ANT-CONTRIB (bundled with Selling and Fulfillment Foundation)
Analytics Reports	Cognos 8 Business Intelligence Version 8.4
ConnectShip	ConnectShip Version 6.2
Ext JS	Ext JS 2.2
FedEx	FXRS 0776
FedEx Printer	Eltron LP2844
FusionCharts	Fusion Charts 3.0
JasperReports	Jasperreports-1.2.0.jar
Loftware Print Server	Loftware Version 9.7, Loftware Version 9.5
Lucene	Lucene 2.4
Pierbridge	AgileElite 6.5.34
RFID	Gen 96 Bit Alien Squiggle
Voxware	VoiceLogisticsPro 2.1.3.5
Weighing Scale	Mettler-Toledo PS30, PS60, or equivalent

2.10 Mobile Application Requirements

The minimum system requirements supported for installing the Mobile Application are shown in Table 2–7, "Supported Devices to Run the Mobile Application".

Table 2–7 Supported Devices to Run the Mobile Application

Terminals	Supported
Mobile Terminals	Pocket PC 2003 OS with 1.0 SP3 .NET CFT
	Windows CE 4.1 OS with 1.0 SP3 .NET CFT
	Windows CE 5.0 OS with 2.0 SP1 .NET CFT
	VT220 emulation software
ncurses (VT220 emulation software)	Version 5.3
ncurses build utilities	`gcc-3.3.2`, `bison-1.875`, `make-3.80`, and `flex-2.5.4a`. These utilities are available in binary format at http://hpux.cs.utah.edu.

Note: Selling and Fulfillment Foundation is specifically tested with the Symbol PPT8846, Symbol PDT8146, Symbol PPT 8800, and Symbol VRC7900 series; the Denso BHT-400B, Denso BHT-260Q, and Denso BHT-420BW series; and the LXE_VXC001 and LXE_MX7001 series.

The device requirements such as memory, screen resolution, keys, and network connectivity are described in Table 2–8, "Device Requirements".

Table 2–8 Device Requirements

Options	Description
Keys	Space, Backspace, Tab, Enter, arrow keys, 0-9, A-Z, a-z, function keys (F1-F12), and special symbols such as!@#$%^&*()-_+=[]\<>?/.,. The special symbols are required only if the data (for fields such as Item ID, Location ID, and Shipment Number) contains special characters.
Barcode Scanner	This device should be equipped with an integrated barcode scanner and should have the ability to send a TAB character after the scanned data.
Screen Resolution	Width:240, Height:320. Screens are designed for this resolution. Screen performance on devices with other resolutions may be sub-optimal. For VT220 screens, Selling and Fulfillment Foundation assumes 8 rows by 20 columns.
Network Connectivity	802.11x (802.11b or higher).
Memory	Minimum of 32MB RAM.
Note: Microsoft ActiveSync 3.7 or above is required to synchronize the PC with the mobile terminal to run the Mobile Application. Microsoft ActiveSync 3.7 can be installed on any PC. For more information about Microsoft ActiveSync, and related system requirements, go to `http://www.microsoft.com`.	

2.11 Rich Client Platform Application Requirements

This section lists all the supported operating systems and the required software used in the deployment of the Rich Client Platform application. Before installing the Rich Client Platform application, verify that you have already installed the applicable software listed in this chapter.

2.11.1 Operating System Requirements

The minimum operating system requirements supported for installing the Rich Client Platform application are shown in Table 2–9, "Supported Operating Systems".

Table 2–9 Supported Operating Systems

Name	Configuration
Windows	Windows XP, VISTA
Linux	Red Hat Enterprise Linux WS 5.3 SUSE Linux Enterprise Desktop 10 SP1

Important: For optimal resolution quality of the menu options and other user interface components of the Rich Client Platform applications, set your system resolution to 96 dpi.

2.11.2 Minimum Hardware Requirements

The minimum hardware requirements for installing the Rich Client Platform applications are listed in Table 2–10, "Minimum and Recommended Hardware Requirements".

Table 2–10 Minimum and Recommended Hardware Requirements

Component	Minimum	Recommended
Processor	350 MHz	1 GHz
Memory	512 MB	1 GB

2.11.3 Selling and Fulfillment Foundation Plugin Requirements

The Selling and Fulfillment Foundation plugins supported for installing the Rich Client Platform application are shown in Table 2–11, "Supported Selling and Fulfillment Foundation Plugins".

Table 2–11 Supported Selling and Fulfillment Foundation Plugins

Name	Version
Rich Client Platform plugin	RCP Plugin 1.0.0
Rich Client Platform Tools plugin	RCP Tools Plugin 1.1.0

To verify that you have a supported version of the plugins listed in Table 2–11, please follow the steps listed below:

1. Open Eclipse
2. Navigate to Help > About Eclipse SDK
3. Click Plug-in Details
4. Verify that the Plug-ins listed in table Table 2–12 match those that are listed in Eclipse.

Table 2–12 Plug-ins Installed in Eclipse

Name	Plug-in Name
RCP plugin	Rich Client Platform Foundation Plugin 1.0.0
	Rich Client Platform Libs Plugin
	Rich Client Platform Plugin 1.0.0
RCP Tools plugin	RCP Extensibility Tools 1.1.0
	RCP Tools Core Plug-in 1.1.0
	RCP Tools UI Editor Plug-in 1.1.0

Table 2–13 RCP Client and Tool Requirements

Name	Requirement	JRE
Client	SUSE Linux Enterprise Desktop 10 SP1, SUSE Linux Enterprise Desktop 11	
	Windows XP, VISTA	
	Red Hat Linux WS 5.3	
Tool - Eclipse 3.2	Windows XP, VISTA	Java Plugin 6

2.11.4 Third-Party Software Requirements

The third-party software such as Eclipse SDK and its related plugins, JDK, and so forth, are provided in Table 2–14, "Supported Third-Party Software". The software mentioned in the table are supported for all the operating systems except where noted.

Table 2–14 Supported Third-Party Software

<table>
<tr><th>Name</th><th>Version</th></tr>
<tr><td>Eclipse SDK</td><td>Eclipse SDK 3.2

The following table lists the Eclipse-related plugins and their versions Selling and Fulfillment Foundation supports:

<table>
<tr><th>Name</th><th>Version</th></tr>
<tr><td>GEF plugin</td><td>GEF Plugin 3.2, SDK</td></tr>
<tr><td>EMF plugin</td><td>EMF Plugin 2.2.0, SDK (includes EMF, SDO, XSD)</td></tr>
<tr><td>VE plugin</td><td>Visual Editor Plugin 1.2, SDK</td></tr>
</table>
</td></tr>
<tr><td>JDK</td><td>JDK 6.0.

The JDK is used for both building and extending the client application.</td></tr>
<tr><td>JRE</td><td>JRE 6.0.

The JRE is used to run the Rich Clients.</td></tr>
</table>

To install the Eclipse SDK and its related plugins, go to the following link: http://www.eclipse.org/downloads/ and download the appropriate version of Eclipse SDK and its related plugins for the appropriate operating system.

To install the JDK/JRE, go to the following link: http://java.sun.com/downloads/ and download the appropriate version of the JDK.

2.12 Installation Memory Requirements

The minimum memory requirements for Selling and Fulfillment Foundation are based on the products you are installing. Table 2–15, "Minimum Memory Requirements" shows the installation command parameters to use for supported operating systems.

Table 2–15 Minimum Memory Requirements

Operating System	Installation Command Parameters
Solaris 10 on SPARC and AMD Opteron (Solaris with Fujitsu SPARC processor	-J-Xms256m -J-Xmx1408m -XX:MaxPermSize=512m
Windows 2008, Windows XP	-J-Xms1024m -J-Xmx2048m -XX:MaxPermSize=512m
IBM AIX 6.1, IBM AIX 5.3 ML4	-J-Xms1024m -J-Xmx1536m
HP-UX 11i v3 on Itanium	-J-Xms256m -J-Xmx1408m -XX:MaxPermSize=512m
Red Hat Enterprise Linux 5.3/AP 64 bit Xeon or AMD processor	-J-Xms1024m -J-Xmx1664m -XX:MaxPermSize=512m
SUSE Linux Enterprise 10 SP1 on 64 bit Xeon or AMD processor	-J-Xms1024m -J-Xmx1664m -XX:MaxPermSize=512m

Note: Table 2–15 provides the suggested installation command parameters for the corresponding operating system. However, if you encounter an out-of-memory error message, modify the value of the installation command parameters appropriately.

Note: For Red Hat Enterprise Linux and SUSE Linux Enterprise operating systems, the `-XX:MaxPermSize=512m` command parameter need not be set for JRockit JDK.

For information about these memory parameters as they relate to the properties ADDITIONAL_ANT_JAVA_TASK_ARGS and ADDITIONAL_ANT_COMPILER_TASK_ARGS, see the *Selling and Fulfillment Foundation: Properties Guide*.

2.13 Disk Space Requirements for Installation

During the Selling and Fulfillment Foundation installation, the installer checks for a minimum of 5GB of free disk space on the specified drive. If the system does not meet this requirement, the installation fails.

3

Creating a Security Plan

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

This chapter also provides the information required to complete Step indicated on the Installation Checklist on page 1-1.

Because we recognize that you may have unique business or operational requirements, 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-llnw.oracle.com/docs/cd/E12840_01/wls/docs103/secmanage/ for tips on how to secure Oracle WebLogic 10
- http://msdn.microsoft.com/en-us/library/bb283235.aspx for tips on how to secure Microsoft SQL Server 2005
- The Rhino9 Team, *The Modern Hackers Desk Reference*; available from 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.

3.1 Planning Your Deployment Architecture

Prior to procuring and implementing the hardware and software that make up 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 Selling and Fulfillment Foundation, see Section 3.1.1, "Current Security Infrastructure Analysis" on page 3-24.
- Conduct an analysis of authentication and authorization mechanisms in your organization to identify the steps needed to incorporate them into Selling and Fulfillment Foundation. For more information on the mechanism used for authentication in Selling and Fulfillment Foundation, see Section 3.1.2, "Authentication and Authorization" on page 3-25.
- Conduct an analysis of your data encryption mechanisms for deploying Selling and Fulfillment Foundation over the internet. For more information on the different variations of the data encryption mechanisms, refer to Section 3.1.3, "Data Encryption" on page 3-25.
- Conduct an analysis of your organization's network topology required to deploy Selling and Fulfillment Foundation. For more information on the various methods to deploy the application, refer to Section 3.1.4, "Network Topology" on page 3-26.

Completing these tasks enables you to:

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

3.1.1 Current Security Infrastructure Analysis

In order to ensure that your 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 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.

3.1.2 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, 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 Selling and Fulfillment Foundation and integrating with LDAP, see the *Selling and Fulfillment Foundation: Properties Guide*. The default authentication mechanism is implemented against the Selling and Fulfillment Foundation database.

3.1.3 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 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 Selling and Fulfillment Foundation are:

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

3.1.4 Network Topology

Where is Selling and Fulfillment Foundation being accessed from?

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

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 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 off-load your SSL processing to specialized devices such as an F5 load-balancer with built-in hardware SSL engines.

3.1.4.1 Accessing Over the Public Internet

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

3.1.4.2 Deployment Over a Virtual Private Network

If you are deploying 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.

3.1.4.3 Deployment Over a Local Area Network

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

3.2 Java Protocol Security Measures

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

3.2.1 Disabling Java Protocols

Each of the following sections provide instructions to disable the respective protocols not used.

EJB

To disable Enterprise JavaBeans (EJB) from 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 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.

HTTP

To disable Hypertext Transfer Protocol (HTTP) as the means to enter API information in 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 Selling and Fulfillment Foundation are stored in the <INSTALL_DIR>/repository/eardata/smcfs/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.

JMS

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

To ensure that JMS is not used without authorization there should be appropriate permissions on the JMS server and in 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

Selling and Fulfillment Foundation: Application Platform Configuration Guide.

3.2.2 Securing Java Protocols

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 *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.WLInitialContext
Factory
```

```
yif.java.naming.provider.url=t3://<ipaddress>:<port>
```

For WebSphere:

```
yif.java.naming.factory.initial=com.ibm.websphere.naming.WsnIn
itialContextFactory
```

```
yif.java.naming.provider.url=iiop://<ipAddress>:<port>
```

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

EJB

When the 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 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 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 Selling and Fulfillment Foundation. This ensures that all the requests are for HTML, and not tunneled RMI or JNDI over HTTP.
- If 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 Selling and Fulfillment Foundation or any screens that extend 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 Selling and Fulfillment Foundation or the custom screens you create.

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

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

HTTP API Tester

The HTTP API tester is provided *only* to test APIs in development mode. Authentication and authorization are not used in this utility. 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 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 Selling and Fulfillment Foundation are stored in the <INSTALL_DIR>/repository/eardata/smcfs/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.

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.

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

3.3 Web Security

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

Sterling Commerce also recommends that you write log files to several servers. There are several applications that do this with no specific need for 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.

3.3.1 Post Installation Recommendations

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

1. Change the password of the default user (admin).
2. Secure the <INSTALL_DIR>/database, <INSTALL_DIR>/repository, and <INSTALL_DIR>/installed_data/smcfs/components/complete_installation/entity directories because they expose components of the data model.
3. Selling and Fulfillment Foundation supplies Web pages that help you test your implementation while running Selling and Fulfillment Foundation in development mode. Remove the <INSTALL_DIR>/installed_data/platform_afc/files/repository/eardata/platform/war/yfshttpapi/yantrahttpapitester.jsp and <INSTALL_DIR>/repository/eardata/platform/war/yfshttpapi/yantrahttpapitester.jsp files from production systems. If you plan to provide access to this jsps in production, secure access to them as described in "HTTP API Tester" on page 3-30.
4. Change permissions on <INSTALL_DIR>/bin/migrator.* files to non-executable.
5. 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 *Selling and Fulfillment Foundation: Properties Guide*.

3.3.2 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.

3.3.3 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`

3.3.4 Documentation

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

3.3.5 Routing

Routing should not be enabled on a production web server.

3.3.6 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.

3.4 Database Security

Set up separate accounts on the database server for installing the 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.

3.4.1 Credit Card Encryption

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 *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

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 *Selling and Fulfillment Foundation: Extending Transactions Guide* and the *Selling and Fulfillment Foundation: Javadocs*.

3.5 Internet Explorer Security Settings

When using 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.

3.5.1 Configuring Browser Security Settings

To configure the Internet Explorer security and privacy settings:

1. From the Internet Explorer menu, select Tools > Internet Options > Security.
2. Select the Web content zone from which 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:
   - Table 3–1, "Internet Explorer Version 6.0 Security Settings for the Selling and Fulfillment Foundation"
   - Table 3–2, "Internet Explorer Version 6.0 Privacy Settings for the Selling and Fulfillment Foundation" on page 3-37

Table 3–1 Internet Explorer Version 6.0 Security Settings for the Selling and Fulfillment Foundation

Internet Explorer Security Setting	Selling and Fulfillment Foundation
ActiveX Controls and Plugins	
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 plugins	Prompt/Enable
Script ActiveX controls marked as safe for scripting	Enable
Cookies	
Allow cookies that are stored on your computer	Disable (Enable only if you are using Sterling Analytics)
Allow per-session cookies	Enable
Downloads	
File download	Enable
Font download	Prompt
Microsoft VM	
Java permissions	High Safety
Miscellaneous	

Table 3–1 Internet Explorer Version 6.0 Security Settings for the Selling and Fulfillment Foundation

Internet Explorer Security Setting	Selling and Fulfillment Foundation
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 3–2 Internet Explorer Version 6.0 Privacy Settings for the Selling and Fulfillment Foundation

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

3.5.2 Adding Selling and Fulfillment Foundation as a Trusted Web site

You should set 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 Selling and Fulfillment Foundation to the list of trusted Web sites:

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. Uncheck 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.

4

Installing and Configuring Application Tier Software

Before installing an application server, ensure that you have installed the required software mentioned in Chapter 2, "System Requirements", noting any recommendations supplied by the software provider and by Sterling Commerce, Inc. This chapter supplies information to help you install software on the application server and web server tier.

> **Note:** Before proceeding with the steps in this chapter ensure that you know the precise installation location for Selling and Fulfillment Foundation (referred to as <INSTALL_DIR>).

This chapter also provides the information required to complete Step and Step indicated on the Installation Checklist on page 1-1.

4.1 Installing Your Application Server

Before installing your application server, check the requirements in Chapter 2, "System Requirements" to make sure you have the applicable hardware and software versions installed.

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

- If you purchased Oracle WebLogic directly from Oracle, see the *Installing Oracle WebLogic Platform* on the product CD-ROM disk for installation instructions.

- If you purchased WebSphere directly from IBM, see the *WebSphere Installation Guide* on the product CD-ROM disk for installation instructions.
- If you purchased JBoss directly from Red Hat, see the *JBoss Installation Guide* on the product CD-ROM disk for installation instructions.

4.2 Installing and Configuring Application Server Utilities

You need to configure certain application server utilities before installing Selling and Fulfillment Foundation.

Install the Java Development Kit (JDK) that is shipped with your application server unless otherwise stated in Chapter 2, "System Requirements". When upgrading the JDK, be sure to set the correct JAVA_HOME environment variable and update the PATH.

4.2.1 Installing JDK Upgrades

You should install the Java Development Kit (JDK) that is shipped with your application server (unless otherwise specified in Chapter 2, "System Requirements"). When upgrading the JDK, be sure to set the correct JAVA_HOME environment variable and update the PATH.

4.3 Installing and Configuring Your Proxy Server

Installing a proxy web server on a 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 Selling and Fulfillment Foundation installed on your application server. Sterling Commerce recommends that you install and configure the Web server version as specified by your application server provider.

4.3.1 Configuring Proxy Server for SSL or HTTPS

This section explains how to set up a web server as an SSL proxy and a load balancer. This procedure applies to the Apache HTTP server and the Oracle WebLogic application server.

For information about configuring the proxy server for SSL or HTTPS using the Apache HTTP Server and WebSphere, see the IBM documentation.

Using any standard Web browser, 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://www.jboss.org/community/wiki/UsingMod_proxyWithJBoss.

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 allows users to use one secure URL to access any number of application servers that run the Sterling applications.

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

1. Install and run Selling and Fulfillment Foundation on the application servers.
2. Copy the appropriate plug-in to the `/etc/apache2/modules` directory.

   For WebLogic 10.3, this is:

   `<WL_HOME>/wlserver_100/server/plugins/<OS>/<processor type>/plugin`

   where `<WL_HOME>` refers to the WebLogic installation directory.

   - 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:

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

Note: To enable an SSL, ensure to add "include conf.d/ssl.conf" as instructed by Apache. By default, RHAS3 has "Include conf.d/*", which includes ssl.conf.

For an HTTP proxy, outside any VirtualHost, add the following section:

Note: 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:

Note: 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_hostna
me/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 Section 15.5, "Security Certificates".
6. Restart Apache, and verify access with any browser.

> **Note:** Continue and complete steps 7-10 if using an RCP application only.

7. Copy the security certificate to the `<RCP_EXTENSIONS_FOLDER>/truststoredirectory`.
8. Build the RCP client.
9. 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.
10. Start the client.

4.4 Setting Up the Image Server

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:

1. 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.

2. Use the default port # 80 (or any available port #) while installing the Image Server & exclude this port from the OS firewall, if required.

3. 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

   where <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 Chapter 15.4.1, "Configuring Connection Settings for Fetching Images from the Server".

For example, if you install Apache as the web server, then to configure it as the image server, do the following:

- 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

  where <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 ""

<Directory ""

AllowOverride None

Order allow, deny

Allow from all

</Directory>
```

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

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

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

> **Note:** 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.

5

Installing and Configuring Database Tier Software on UNIX/Linux

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

This chapter also provides the information required to complete Step and Step indicated on the Installation Checklist on page 1-1.

Before installing your database server, verify that you have the applicable software versions. For more information see Chapter 2, "System Requirements".

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

5.1 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 Section 5.1.1, "Capacity Planning" on page 5-48, Section 5.1.2, "Disk Estimation for the Distributed Order Management Module" on page 5-48 and Section 5.1.3, "Disk Estimation for the Networked Warehouse Management System Module" on page 5-51.

5.1.1 Capacity Planning

There are many factors to consider when estimating the amount of disk space that is required for 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

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 the Distributed Order Management (DOM) module, use Table 5–1, "Steps for Disk Space Estimation for the Order Management Module" on page 5-51.
- If you use or are planning to use both the Distributed Order Management (DOM) and the networked Warehouse Management System (WMS) modules, use Table 5–1, "Steps for Disk Space Estimation for the Order Management Module" on page 5-51 and Table 5–2, "Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS" on page 5-53.
- If you are planning to use only the WMS module use Table 5–3, "Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS" on page 5-55.

5.1.2 Disk Estimation for the Distributed Order Management Module

The disk estimation provided here pertains to the Order Management module of 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 Step 1 mentioned above, by a storage usage factor.
3. Finally add a minimum base amount.

However, the following information is essential to keep in mind before calculating the estimated disk space:

Note 1

You need to gather some information about the amount of time required to maintain the database, such as:

1. How long do you plan to keep data in the main transactional database before orders are purged to the history database?
2. How long are orders kept in the history database before they are purged?
3. Are you purchasing the storage for the first few years into the implementation?

Consider the following examples to achieve answers for the above mentioned 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 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.

Note 2

The order discussed in Table 5–1, "Steps for Disk Space Estimation for the Order Management Module" on page 5-51 includes sales, transfer, return, and work orders.

Note 3

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***.

Note 4

When procuring your storage, ensure that the storage device has at least the amount of usable space specified in Step 8 of Table 5–1, "Steps for Disk Space Estimation for the Order Management Module". This table provides an idea of the usable space for the storage device in your company. However, the actual amount you may need to order, 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.

Now that you have noted the above points you can proceed to the estimation of required disk space as outlined in Table 5–1.

Table 5–1 Steps for Disk Space Estimation for the Order Management Module

1.	Enter the number of years worth of information to be kept in the system (retention time). For a more detailed example, refer to "Note 1".	___________
2.	Enter the number of orders you expect to be in the system during the time period specified in Step 1. For the different types of orders refer to "Note 2".	___________
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 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 Selling and Fulfillment Foundation. For more information on the storage estimate, refer to "Note 3".	500 MB
9.	Enter the total estimated storage obtained by adding the values from Step 6, Step 7, and Step 8. For more information on the amount of usable space, refer to "Note 4".	___________

5.1.3 Disk Estimation for the Networked Warehouse Management System Module

The disk estimation discussed in this section pertains to the networked WMS module of Selling and Fulfillment Foundation.

This estimation methodology consists of three parts:

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.

If you are planning to use both the Selling and Fulfillment Foundation DOM and WMS modules use Table 5–2, "Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS" on page 5-53 or else if you are planning to use only the WMS module use Table 5–3, "Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS" on page 5-55.

However, the following information is essential to keep in mind before calculating the estimated disk space:

Note 1

You need to gather some information about the amount of time required to maintain the database, such as:

1. How long do you plan to keep data in the main transactional database before shipment data is purged to the history database?

2. How long is the shipment data kept in the history database before it is purged?

3. Are you purchasing the storage for the first few years into the implementation?

Consider the following example to achieve answers for the above mentioned questions.

Case 1 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.

Note 2

The shipment lines discussed in Table 5–2 on page 5-53 and Table 5–3 on page 5-55 include space requirements for demand-based replenishment.

Note 3

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***.

Note 4

When procuring your storage, ensure that the storage device has at least the amount of usable space specified in the last step of Table 5–2 on page 5-53 or Table 5–3 on page 5-55. 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.

Now that you have noted the above points you can proceed to the estimation of required disk space as outlined in Table 5–2 on page 5-53 or Table 5–3 on page 5-55.

Table 5–2 Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS

1.	Enter the number of years worth of information to be kept in the system (retention time). For a more detailed example, refer to "Note 1".	___________
2.	Enter the number of shipment lines you expect to be in the system during the time period specified in Step 1. For the different types of shipments, refer to "Note 2".	___________

Table 5–2 Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS

3.	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 Selling and Fulfillment Foundation system: (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. (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. (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. (d) 20 KB - for warehouses using no tag-controlled items, no serial tracking, largely CASE LPNs, and more than 80% PARCEL shipping. (e) 25 KB - for warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and PARCEL as well as TL - LTL shipping. (f) 30 KB - for warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and more than 80% TL - LTL shipping.	___________
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 one of the following storage factors that most closely approximates a description of your 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.	___________

Table 5–2 Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS

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 Selling and Fulfillment Foundation. For more information on the storage estimates refer to the "Note 3".	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. For more information on the amount of usable space, refer to "Note 4".	___________
12.	Enter the value of Step 9 from Table 5–1 on page 5-51.	___________
13.	Enter the total estimated storage obtained by adding the values from Step 11 and Step 12. For more information on the amount of usable space, refer to "Note 4"	___________

Table 5–3 Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS

1.	Enter the number of years worth of information to be kept in the system (retention time). For a more detailed example, refer to "Note 1".	___________
2.	Enter the number of shipment lines you expect to be in the system during the time period specified in Step 1. For the different types of shipments, refer to "Note 2".	___________

Table 5–3 Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS

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 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.	___________
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 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.	___________

Table 5–3 Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS

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 Selling and Fulfillment Foundation. For more information on the storage estimates refer to "Note 3".	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. For more information on the amount of usable space, refer to "Note 4".	___________

5.1.4 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.

5.2 Installing Oracle (UNIX/Linux)

You can use an Oracle database for maintaining information on Selling and Fulfillment Foundation. The following sections provide the necessary steps to install and configure an Oracle database for production.

To install Oracle:

Follow the steps below to install Oracle with single or multiple byte characters:

1. If you do not have Oracle installed, follow the installation procedures in your Oracle Installation manuals.

2. Run the create instance procedure. Use a character set appropriate for your desired language.

```
CHARACTER SET "UTF8"
```

3. 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=<10.2.0.3>
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:

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

prior to running any create table scripts.

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.

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

> **Note:** 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.

4. Download the Oracle JDBC driver `ojdbc6.jar` from the Oracle Web site and copy it to a well known location for reference during installation.

   The Oracle JDBC driver can be found at:

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

   You can use the `ojdbc6.jar` file which comes along with the Oracle installation, or download the driver version that matches the installed Oracle version.

   For example, if you have installed Oracle 11.1.0.6, then the JDBC driver version should also be 11.1.0.6.

5.2.1 Oracle Database User Privileges

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

Following are some of the basic privileges that should be granted to the Selling and Fulfillment Foundation administrative user who is involved in creating and modifying the Oracle database:

- ALTER ANY SEQUENCE
- ALTER SESSION
- ALTER USER DEFAULT ROLE
- 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
- GRANT "CONNECT"
- INSERT ANY TABLE
- SELECT ANY DICTIONARY
- SELECT ANY SEQUENCE
- SELECT ANY TABLE
- SELECT_CATALOG_ROLE
- UPDATE ANY TABLE

The following are some of the basic privileges that should be granted to the application user who will only be running the application:

- ALTER SESSION
- CREATE SESSION
- DELETE ANY TABLE
- EXECUTE ANY PROCEDURE
- INSERT ANY TABLE
- SELECT ANY SEQUENCE
- SELECT ANY TABLE
- UPDATE ANY TABLE

Note: Ensure that the user who is responsible for creating and modifying the Oracle database has a specified quota (extent) assigned to him in the tablespace even if the user was assigned a unlimited tablespace when the user was created. Otherwise, the installer will throw the "ORA-01950: no privileges on tablespace name" error.

Note: 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.

5.2.2 Configuring an Oracle Database for Production

You need to configure your Oracle database for running in a production environment with 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 Sterling Warehouse Management System installation.
- Set the database connection properties.

To create the Oracle database to handle multiple byte characters:

1. Do not modify the Selling and Fulfillment Foundation DDL.
2. Choose the correct data encoding format for your language. See "To install Oracle:" on page 5-57 for more information.
3. Choose the character set suitable for your language. See "To install Oracle:" on page 5-57 for specific settings to ensure the database field sizes.

To set up scripts (if you are using locally managed tablespaces or another utility to size your database):

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.

To set up the scripts:

1. Create tablespaces where the 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.

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

> **Note:** Index creations should be run only *after* everything else in the tablechanges script.

To run the scripts:

1. Log into the Oracle server manager as sysdba.
2. Create the user that is the designated schema owner.
3. Grant the privileges listed in Section 5.2.1, "Oracle Database User Privileges" on page 5-59 to the newly created user.
4. Log out of the Oracle Server Manager and log back in as the newly created user.
5. Verify the database as described in Section 13.1.1.2, "Verifying the Database Schema" on page 13-194.
6. Load the Selling and Fulfillment Foundation database factory defaults as described in Section 13.1.1.3, "Loading the Selling and Fulfillment

Foundation Database Factory Defaults After Installation" on page 13-202.

7. Check for the degree of parallelism, using information from the *Selling and Fulfillment Foundation: Performance Management Guide*.

5.2.2.1 Enabling the Text Search Feature

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. Sterling Commerce recommends that you use the CTXCAT index.

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

This section explains the following:

- Enabling the Text Search Feature for CTXCAT Index
- Enabling the Text Search Feature for CONTEXT Index

5.2.2.1.1 Enabling the Text Search Feature for CTXCAT Index

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:

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 *Selling and Fulfillment Foundation: Properties Guide*.

5.2.2.1.2 Enabling the Text Search Feature for CONTEXT Index

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:

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 *Selling and Fulfillment Foundation: Properties Guide*.

5.2.3 Using an Oracle Database Server

You can use both Oracle 10g and 11g database with Selling and Fulfillment Foundation. See Chapter 2, "System Requirements" for supported version information. Selling and Fulfillment Foundation supports either single node or 2-node RAC environment.

To use an Oracle 11g database:

- Create the database. Refer to the Oracle documentation for information about creating the database, including creating a schema repository, login, and tablespace. Be sure to install the correct version and patches.
- Configure the database by completing the following tasks:
  - Setting Database Parameters in Oracle
  - Rolling Back or Undoing Changes in Oracle
  - Granting Permissions in Oracle
  - Installing the JDBC Driver in Oracle

5.2.3.1 Setting Database Parameters in Oracle

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

5.2.3.2 Rolling Back or Undoing Changes in Oracle

You can roll back or undo changes in Oracle using the following method:

(Oracle versions 10g or later) These versions support AUTO UNDO management. It is recommended that you use this option. This avoids manual monitoring of UNDO segments.

If a server is upgraded from Oracle 9i, 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.

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

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

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 *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 *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.

> **Note:** For information about the `sandbox.cfg` and `customer_overrides.properties` files, refer to the *Selling and Fulfillment Foundation: Properties Guide.*

5.2.4 Installing the JDBC Driver in Oracle

Selling and Fulfillment Foundation requires the appropriate JDBC driver for Oracle 11g databases. These drivers are thin client, 100% Pure Java™ JDBC drivers. See Chapter 2, "System Requirements" for supported version information.

The supported versions of the JDBC driver build the correct Selling and Fulfillment Foundation directory structure.

After you obtain the correct JDBC driver file, record the absolute path to its location on your system. You must supply this absolute path when you install Selling and Fulfillment Foundation.

5.2.5 Configuring the NLS_LANG Parameter for Oracle Client

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.

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)

To set the value of the NLS_LANG parameter in Windows, verify the HKEY_LOCAL_MACHINE/SOFTWARE/ORACLE/NLS_LANG entry in the registry.

To set the value of the NLS_LANG parameter in UNIX, NLS_LANG is set as a local environment variable.

5.3 Installing DB2 (UNIX/Linux)

You can use a DB2 database for maintaining information on Selling and Fulfillment Foundation. The following sections provide the necessary steps to install and configure a DB2 database for production.

Note: For the Selling and Fulfillment Foundation, set the DB2LOCK_TO_RB registry variable as follows:

```
DB2LOCK_TO_RB=STATEMENT
```

This registry variable specifies whether lock timeouts cause an entire transaction or only the current statement to be rolled back. If DB2LOCK_TO_RB is set to STATEMENT, locked timeouts cause only the current statement to be rolled back. Any other setting results in transaction rollback.

In addition, set the following registry variables as shown. (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_BUFPD=
```

To install DB2:

1. If you do not have DB2 installed, follow the installation procedures in your DB2 Installation manual.

Note: When creating the DB2 database, the appropriate codepage needs to be selected for international language characters (for example, UTF-8).

2. You need to set the following parameter to avoid memory leaks and DB2 crashes:

```
db2set DB2_NUM_CKPW_DAEMONS=0
```

5.3.1 DB2 Database User Privileges

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

5.3.2 Configuring a DB2 Database for Production

You need to configure your DB2 database for running in a production environment with Selling and Fulfillment Foundation. To configure a DB2 database for a production environment, you must:

- Size the database by estimating the required disk space.
- 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.

Manually creating objects on DB2

To set up the scripts:

1. Create tablespaces where the 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.

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

To run the scripts:

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 Section 5.3.1, "DB2 Database User Privileges" on page 5-69 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 Section 13.1.1.2, "Verifying the Database Schema" on page 13-194.

6. Load the Selling and Fulfillment Foundation database factory defaults as described in Section 13.1.1.3, "Loading the Selling and Fulfillment Foundation Database Factory Defaults After Installation" on page 13-202.

7. Check for the degree of parallelism, using information from the *Selling and Fulfillment Foundation: Performance Management Guide*.

Manually creating views on DB2

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_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

5.3.2.1 Enabling the Text Search Feature

To enable the text search feature on DB2 database:

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.

> **Note:** 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.

3. Verify that the text search index creation was successful.
4. Use the `customer_overrides.properties` file that is located in the <INSTALL_DIR>/properties directory to set the `yfs.db.textsearch` property to `Y`. For additional information about overriding properties using the `customer_overrides.properties` file, see the *Selling and Fulfillment Foundation: Properties Guide*.

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

5.3.3 Using a DB2 Database Server

You can use a DB2 database with Selling and Fulfillment Foundation. See Chapter 2, "System Requirements" for supported version information.

To use a DB2 server:

- 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 patch. Be sure to install the client components and compilers before you install the fixpack.
- Configure the database by completing the following tasks:
  - Installing Client Components, Compilers, and Fix Pack
  - Setting Parameters for DB2
  - Granting Permissions for DB2
  - Installing JDBC Drivers for DB2

5.3.3.1 Installing Client Components, Compilers, and Fix Pack

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

1. Install the Administration client.
2. Install the 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.
4. Set the path for the compiler by using the **db2set** command.

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

5.3.3.2 Setting Parameters for DB2

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

5.3.3.3 Installing JDBC Drivers for DB2

For DB2, install the appropriate DB2 JDBC Type 4 driver and any correlating patches. See Chapter 2, "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:

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 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 the .jar file you created on the Preinstallation Checklist.

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

6

Installing and Configuring Database Tier Software on Windows

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

This chapter also provides the information required to complete Step and Step indicated on the Installation Checklist on page 1-1.

Before installing your database server, verify that you have the applicable software versions. For more information see Chapter 2, "System Requirements".

6.1 Creating and Configuring the Database Server (Windows)

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

> **Caution:** If you are reinstalling 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 Section 6.2, "Data Backup and Restore Scripts".
>
> After creating and configuring your database, recycle the database. Then stop and restart Selling and Fulfillment Foundation to apply the changes.

In a Windows environment, Selling and Fulfillment Foundation supports the following databases:

- MS SQL 2005 SP2
- MS SQL 2008
- Oracle® 10g, 11g
- DB2

See Chapter 2, "System Requirements" for supported version information.

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

6.2 Data Backup and Restore Scripts

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

```
sci_ant.cmd -f backupScriptGen.xml -DdbType=<database_type>
```

This script generates sample backup and restore scripts in the <INSTALL_DIR>/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 6–1 backupScripGen.xml Arguments

Argument	Purpose	Accepted Values
`-Dos=`	Determines what kind of script is generated. If "windows" is selected, a .cmd file is created. If "linux" or "unix" is selected, a .sh file is created. If "all" is selected, both .cmd and .sh files are created.	• `windows` • `unix` • `linux` • `all`
`-DdbType=`	Determines which database(s) the scripts will be generated for.	• `oracle` • `db2` • `sqlserver` • `iseries` • `all`
`-DtableType=`	Determines which entities to generate scripts for. Valid values are any TableType attribute defined for an entity.	• `ALL` • `CONFIGURATION` • `MASTER` • `METADATA` • `STATISTICS` • `TRANSACTION` **Note:** The value ALL is supported only for single-schema mode.

Note: 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`

- Microsoft SQL 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`
- DB2 scripts depend on `export` or `load` utilities. You can modify and use the following scripts:
  - `backup_config_db2.cmd`
  - `restore_config_db2.cmd`
  - `delete_configuration_oracle.sql`

6.3 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 Section 6.3.1, "Capacity Planning" on page 6-78, Section 6.3.2, "Disk Estimation for the Distributed Order Management Module" on page 6-79 and Section 6.3.3, "Disk Estimation for the Networked Warehouse Management System Module" on page 6-82.

6.3.1 Capacity Planning

There are many factors to consider when estimating the amount of disk space that is required for 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.

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 the Distributed Order Management (DOM) module, use Table 6–2, "Steps for Disk Space Estimation for the Order Management Module" on page 6-81.
- If you use or are planning to use both Distributed Order Management (DOM) and networked Warehouse Management System (WMS) modules, please use Table 6–2, "Steps for Disk Space Estimation for the Order Management Module" on page 6-81 and Table 6–3, "Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS" on page 6-84.
- If you are planning to use only the WMS module use Table 6–4, "Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS" on page 6-86.

6.3.2 Disk Estimation for the Distributed Order Management Module

The disk estimation provided here pertains to the Order Management module of 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 Step 1 mentioned above, by a storage usage factor.
3. Finally add a minimum base amount.

However, the following information is essential to keep in mind before calculating the estimated disk space:

Note 1

You need to gather some information about the amount of time required to maintain the database, such as:

1. How long do you plan to keep data in the main transactional database before orders are purged to the history database?

2. How long are orders kept in the history database before they are purged?
3. Are you purchasing the storage for the first few years into the implementation?

Consider the following examples to achieve answers for the above mentioned 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 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.

Note 2

The order discussed in Table 6–2, "Steps for Disk Space Estimation for the Order Management Module" on page 6-81 includes sales, transfer, return, and work orders.

Note 3

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***.

Note 4

When procuring your storage, ensure that the storage device has at least the amount of usable space specified in Step 8 of Table 6–2, "Steps for Disk Space Estimation for the Order Management Module". This table provides an idea of the usable space for the storage device in your company. However, the actual amount you might need to order, 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.

Now that you have noted the above points you can proceed to the estimation of required disk space as outlined in Table 6–2.

Table 6–2 Steps for Disk Space Estimation for the Order Management Module

1.	Enter the number of years worth of information to be kept in the system (retention time). For a more detailed example, refer to "Note 1".	___________
2.	Enter the number of orders you expect to be in the system during the time period specified in Step 1. For the different types of orders refer to "Note 2".	___________
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).	___________

Table 6–2 Steps for Disk Space Estimation for the Order Management Module

5.	Enter the order line multiplier: Choose one of the following storage factors that most closely approximates a description of your 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 Selling and Fulfillment Foundation. For more information on the storage estimate, refer to "Note 3".	500 MB
9.	Enter the total estimated storage obtained by adding the values from Step 6, Step 7, and Step 8. For more information on the amount of usable space, refer to "Note 4".	___________

6.3.3 Disk Estimation for the Networked Warehouse Management System Module

The disk estimation discussed in this section pertains to the networked WMS module of Selling and Fulfillment Foundation.

This estimation methodology consists of three parts:

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.

If you are planning to use both the Selling and Fulfillment Foundation DOM and WMS modules, use Table 6–3, "Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS" on page 6-84 or else if you are planning to use only the WMS module use Table 6–4, "Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS" on page 6-86.

However, the following information is essential to keep in mind before calculating the estimated disk space:

Note 1

You need to gather some information about the amount of time required to maintain the database, such as:

1. How long do you plan to keep data in the main transactional database before shipment data is purged to the history database?
2. How long is the shipment data kept in the history database before it is purged?
3. Are you purchasing the storage for the first few years into the implementation?

Consider the following example to achieve answers for the above mentioned questions.

Case 1 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.

Note 2

The shipment lines discussed in Table 6–3 on page 6-84 and Table 6–4 on page 6-86 include space requirements for demand-based replenishment.

Note 3

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***.

Note 4

When procuring your storage, ensure that the storage device has at least the amount of usable space specified in the last step of Table 6–3 on page 6-84 and Table 6–4 on page 6-86. These tables provide an idea of the usable space for the storage device in your company. However, the actual amount you might 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.

Now that you have noted the above points you can proceed to the estimation of required disk space as outlined in Table 6–3 on page 6-84 and Table 6–4 on page 6-86.

Table 6–3 Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS

1.	Enter the number of years worth of information to be kept in the system (retention time). For a more detailed example, refer to "Note 1".	___________
2.	Enter the number of shipment lines you expect to be in the system during the time period specified in Step 1. For the different types of shipments, refer to "Note 2".	___________

Table 6–3 Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS

3.	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 Selling and Fulfillment Foundation system: (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. (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. (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. (d) 20 KB - for warehouses using no tag-controlled items, no serial tracking, largely CASE LPNs, and more than 80% PARCEL shipping. (e) 25 KB - for warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and PARCEL as well as TL - LTL shipping. (f) 30 KB - for warehouses using tag-controlled items, serial tracking, largely CASE LPNs, and more than 80% TL - LTL shipping.	____________
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 one of the following storage factors that most closely approximates a description of your 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.	____________

Table 6–3 Steps for Disk Space Estimation for the Networked WMS Module - If you have both DOM and WMS

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 Selling and Fulfillment Foundation. For more information on the storage estimates refer to the "Note 3".	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. For more information on the amount of usable space, refer to "Note 4".	___________
12.	Enter the value of Step 9 from Table 6–2 on page 6-81.	___________
13.	Enter the total estimated storage obtained by adding the values from Step 11 and Step 12. For more information on the amount of usable space, refer to "Note 4"	___________

Table 6–4 Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS

1.	Enter the number of years worth of information to be kept in the system (retention time). For a more detailed example, refer to "Note 1".	___________
2.	Enter the number of shipment lines you expect to be in the system during the time period specified in Step 1. For the different types of shipments, refer to "Note 2".	___________

Table 6–4 Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS

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 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.	____________
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 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.	____________

Table 6–4 Steps for Disk Space Estimation for Networked WMS Module - If you have only WMS

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 Selling and Fulfillment Foundation. For more information on the storage estimates refer to "Note 3".	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. For more information on the amount of usable space, refer to "Note 4".	___________

6.3.4 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.

6.4 Installing Microsoft SQL Server 2005/2008 (Windows)

You can use a Microsoft SQL Server 2005/2008 database for maintaining information on Selling and Fulfillment Foundation. When using a Microsoft SQL Server database with Selling and Fulfillment Foundation, see Chapter 2, "System Requirements" for supported version information.

If you do not have Microsoft SQL Server installed, follow the installation procedures in your Microsoft SQL Server installation manual. Refer to the Microsoft SQL Server documentation for information about creating the database, including creating a schema repository, login, and tablespace. Be sure to install the correct version and patch.

Note: Ensure that Named Pipes & TCP/IP protocols are enabled in the network utility of the Microsoft SQL Server.

Note: For Microsoft SQL Server 2005, do not use case-sensitive column names in the database. Case-sensitive names will prevent the Microsoft SQL Server 2005 System Management Console from loading.

Note: Set the Microsoft SQL Server 2005 to disallow page locks by using the following command:

```
sp_msforeachtable 'ALTER INDEX ALL ON ? SET (ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = OFF)'
```

6.4.1 Setting Database Parameters in Microsoft SQL Server

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

6.4.2 Microsoft SQL Server Database User Privileges

In Microsoft SQL Server, you must grant DBO (Database Owner) permission to the Selling and Fulfillment Foundation user. The DB_DDLADMIN role is required for creating objects in the Microsoft SQL Server database.

6.4.3 Configuring a Microsoft SQL Server Database for a Production Environment

You need to configure your Microsoft SQL Server database for running in a production environment with Selling and Fulfillment Foundation. To configure a Microsoft SQL Server database for a production environment, you must:

- Size the database by estimating the required disk space.
- Set the database connection properties.

6.4.3.1 Running Scripts for a Microsoft SQL Server Database

To run the scripts:

1. Make sure you have a Microsoft SQL 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 a Microsoft SQL Server command-line utility.
3. Examine the log files for errors.

> **Note:** 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.

When running DBVerify-generated SQL scripts for Microsoft SQL Server, you may see warnings about maximum row size. You can ignore these warnings.

However, when adding indexes on Microsoft SQL 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.
```

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.

6.4.3.2 Enabling the Text Search Feature

To enable the text search feature on the Microsoft SQL Server database:

1. Make sure that the Microsoft Search service is running on the machine on which the Microsoft SQL Server is installed.
2. Log in to the Microsoft SQL Server manager with a user ID having DBA privileges.

> **Note:** The text search indexes that are created on the Microsoft SQL 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
   ```

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

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

6.4.4 Installing the JDBC Driver in Microsoft SQL Server

Selling and Fulfillment Foundation requires the correct Microsoft SQL Server driver. See Chapter 2, "System Requirements" for supported version information. The supported version of the JDBC driver builds the correct directory structure.

Go to:

http://www.microsoft.com/downloads

For Microsoft SQL Server 2005/2008, download Microsoft SQL Server JDBC version 2.0.

6.5 Installing Oracle (Windows)

You can use an Oracle database for maintaining information on Selling and Fulfillment Foundation. The following sections provide the necessary steps to install and configure an Oracle database for production.

To install Oracle:

Follow the steps below to install Oracle with single or multiple byte characters:

1. If you do not have Oracle installed, follow the installation procedures in your Oracle Installation manuals.

2. Run the create instance procedure. Use a character set appropriate for your desired language.

```
CHARACTER SET "UTF8"
```

3. 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=<10.2.0.3>
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:

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

prior to running any create table scripts.

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.

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

> **Note:** 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.

4. Download the Oracle JDBC driver `ojdbc16.jar` from the Oracle website and copy to a well known location for reference during installation.

   The Oracle JDBC driver can be found at:

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

   You can use the `ojdbc6.jar` file which comes along with the Oracle installation, or download the driver version that matches the installed Oracle version.

   For example, if you have installed Oracle 11.1.0.6, then the JDBC driver version should also be 11.1.0.6.

6.5.1 Oracle Database User Privileges

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

Following are some of the basic privileges that should be granted to the Selling and Fulfillment Foundation administrative user who is involved in creating and modifying the Oracle database:

- ALTER ANY SEQUENCE
- ALTER SESSION
- ALTER USER DEFAULT ROLE
- 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
- GRANT "CONNECT"
- INSERT ANY TABLE
- SELECT ANY DICTIONARY
- SELECT ANY SEQUENCE
- SELECT ANY TABLE
- SELECT_CATALOG_ROLE
- UPDATE ANY TABLE

The following are some of the basic privileges that should be granted to the application user whose involvement is restricted just to running the application:

- ALTER SESSION
- CREATE SESSION
- DELETE ANY TABLE
- EXECUTE ANY PROCEDURE
- INSERT ANY TABLE
- SELECT ANY SEQUENCE
- SELECT ANY TABLE

- UPDATE ANY TABLE

> **Note:** 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.

6.5.2 Configuring an Oracle Database for Production (Windows)

You need to configure your Oracle database for running in a production environment with 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 Sterling Warehouse Management System installation.
- Set the database connection properties.

To create the Oracle database to handle multiple byte characters:

1. Do not modify the Selling and Fulfillment Foundation DDL.
2. Choose the correct data encoding format for your language. See "To install Oracle:" on page 6-92 for more information.
3. Choose the character set suitable for your language. See "To install Oracle:" on page 6-92 for specific settings to ensure the database field sizes.

To set up scripts (if you are using locally managed tablespaces or another utility to size your database):

1. Create tablespaces where the Selling and Fulfillment Foundation tables and indexes reside.
2. Modify the `<INSTALL_DIR>\repository\scripts\EFrame_TableChanges.sql` file to reference your newly created tablespaces.

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

To run the scripts:

1. Log into the Oracle server manager as sysdba.
2. Create the user that is the designated schema owner.
3. Grant the privileges listed in Section 6.5.1, "Oracle Database User Privileges" on page 6-94 to the newly created user.
4. Log out of the Oracle Server Manager and log back in as the newly created user.
5. Verify the database as described in Section 13.1.1.2, "Verifying the Database Schema" on page 13-194.
6. Load the Selling and Fulfillment Foundation database factory defaults as described in Section 13.1.1.3, "Loading the Selling and Fulfillment Foundation Database Factory Defaults After Installation" on page 13-202.
7. Check for the degree of parallelism, using information from the *Selling and Fulfillment Foundation: Performance Management Guide*.

6.5.2.1 Enabling the Text Search Feature

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. Sterling Commerce recommends that you use the CTXCAT index.

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

This section explains the following:

- Enabling the Text Search Feature for CTXCAT Index
- Enabling the Text Search Feature for CONTEXT Index:

6.5.2.1.1 Enabling the Text Search Feature for CTXCAT Index

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:

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 *Selling and Fulfillment Foundation: Properties Guide*.

6.5.2.1.2 Enabling the Text Search Feature for CONTEXT Index:

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:

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 <INSTALL_DIR>\repository\scripts\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 *Selling and Fulfillment Foundation: Properties Guide*

6.5.3 Using an Oracle Database Server (Windows)

You can use both Oracle 10g and 11g database with Selling and Fulfillment Foundation. See Chapter 2, "System Requirements" for supported version information. Selling and Fulfillment Foundation supports either single node or 2-node RAC environment.

To use an Oracle 11g database:

- Create the database. Refer to the Oracle documentation for information about creating the database, including creating a schema repository, login, and tablespace. Be sure to install the correct version and patches.
- Configure the database by completing the following tasks:
  - Setting Database Parameters in Oracle
  - Rolling Back or Undoing Changes in Oracle
  - Granting Permissions in Oracle
  - Installing the JDBC Driver in Oracle

6.5.3.1 Setting Database Parameters in Oracle

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

6.5.3.2 Rolling Back or Undoing Changes in Oracle

You can roll back or undo changes in Oracle using the following method:

(Oracle versions 10g or later) These versions support AUTO UNDO management. It is recommended that you use this option. This avoids manual monitoring of UNDO segments.

6.5.4 Installing the JDBC Driver in Oracle (Windows)

Selling and Fulfillment Foundation requires the appropriate JDBC driver for Oracle 11g databases. These drivers are thin client, 100% Pure Java™ JDBC drivers. See Chapter 2, "System Requirements" for supported version information.

The supported versions of the JDBC driver build the correct Selling and Fulfillment Foundation directory structure.

After you obtain the correct JDBC driver file, record the absolute path to its location on your system. You must supply this absolute path when you install Selling and Fulfillment Foundation.

6.5.5 Enabling Failover in a Multiple Node Oracle RAC Database Cluster (Windows)

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

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 *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 *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.

   Note: For information about the `sandbox.cfg` and `customer_overrides.properties` files, refer to the *Selling and Fulfillment Foundation: Properties Guide.*

7

Installing Selling and Fulfillment Foundation in UNIX or Linux Environments

This chapter explains how to install Selling and Fulfillment Foundation in UNIX or Linux environments. This chapter also provides the information required to complete Step of the “Installation Checklist” on page 1-1 for these environments.

You can install Selling and Fulfillment Foundation on a UNIX or Linux system locally in an X Windows environment, or remotely, in a text-based console environment.

7.1 Before You Begin

If you are upgrading from a prior release, see the *Selling and Fulfillment Foundation: Upgrade Guide* ***before*** continuing with the installation of Selling and Fulfillment Foundation.

Before installing Selling and Fulfillment Foundation, ensure that you have installed the appropriate software listed in Chapter 2, "System Requirements".

Throughout this document, <INSTALL_DIR> refers to the directory where you install Selling and Fulfillment Foundation. For example, <INSTALL_DIR>=/Supply_Chain_Apps.

7.2 Preinstallation Setup for a UNIX or Linux Environment

The following topics will assist you with the preinstallation tasks when planning to install Selling and Fulfillment Foundation in a UNIX or Linux environment:

- "Key Terms (UNIX/Linux)" on page 7-104
- "Checklist for UNIX or Linux Preinstallation" on page 7-104
- "Checking System Requirements (UNIX/Linux)" on page 7-109
- "Installing the Java Software Development Kit (UNIX/Linux)" on page 7-109
- "Creating a UNIX Account" on page 7-109
- "Applying Database Definition Language (DDL) Statements (UNIX/Linux)" on page 7-109
- "About Silent Installations (UNIX/Linux)" on page 7-110

7.2.1 Key Terms (UNIX/Linux)

The following terms and definitions will assist you in understanding the concepts discussed in this document:

- 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.

7.2.2 Checklist for UNIX or Linux Preinstallation

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

Note: 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.

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 included detailed explanations of some of the items on the checklist.

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, refer to “Checking System Requirements (UNIX/Linux)” on page 7-109.	
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. • Version of the JDK • Absolute path to the JDK For more information, refer to “Installing the Java Software Development Kit (UNIX/Linux)” on page 7-109 and Chapter 2, "System Requirements".	
4	Set Umask to 002.	

Preinstallation Checklist for UNIX or Linux

Step	Description	Your Notes
5	Determine and record information about your Oracle or MS SQL 2005/2008, DB2 database server; determine and record information about your database server. Be aware that this information may be case sensitive. • Database user name and associated password • Database catalog name (For more information, see “Key Terms (UNIX/Linux)” on page 7-104.) • Database host name (or IP address) • Database host port number • (Oracle or Microsoft SQL Server 2005/2008 only) 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. For more information, refer to “Applying Database Definition Language (DDL) Statements (UNIX/Linux)” on page 7-109.	

Preinstallation Checklist for UNIX or Linux

Step	Description	Your Notes
7	Determine and record the directory in which you plan to install Selling and Fulfillment Foundation. • 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 "About Silent Installations (UNIX/Linux)" on page 7-110 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.	

Preinstallation Checklist for UNIX or Linux

<table>
<tr><th>Step</th><th>Description</th><th>Your Notes</th></tr>
<tr><td>8</td><td>If you are running a silent installation, create your <code>install.silent</code> property file. For more information, refer to “About Silent Installations (UNIX/Linux)” on page 7-110.</td><td></td></tr>
<tr><td>9</td><td> Note: This step is only for application systems that use the Linux operating system.
Make the following system change:

If the base locale for the system is English, set the LANG environment variable to <code>en_US</code>.
Reboot the system.
</td><td></td></tr>
<tr><td>10</td><td> Note: This step is only for application systems that use the RedHat Enterprise Linux operating system.
Make the following system changes:

 If the base locale for the system is English, edit the /etc/sysconfig/i18n file by changing the SUPPORTED variable from <code>en_US.utf8</code> to <code>en_US</code>. You can also allow multiple support using the following format:
<code>en_US.utf8:en_US</code>
Save and close the /etc/sysconfig/i18n file.
 Edit the /etc/security/limits.conf file by adding the following lines:
<pre>
*   hard   nofile   8196
*   soft   nofile   2048
*   hard   memlock   3000000
*   soft   memlock   4000000
*   hard   nproc   16000
*   soft   nproc   16000
*   hard   stack   512000
*   soft   stack   512000
</pre>
This updates the system ulimits.
Save and close the /etc/security/limits.conf file.
Reboot the system.
</td><td></td></tr>
</table>

7.2.3 Checking System Requirements (UNIX/Linux)

Before you begin, verify that your system meets the hardware and software requirements specified for Release 8.5 of Selling and Fulfillment Foundation. The hardware requirements listed are the minimum required. For current information, see Chapter 2, "System Requirements".

7.2.4 Installing the Java Software Development Kit (UNIX/Linux)

You must install the Java Software Development Kit (JDK) and the patches specific to your system. You must supply the absolute path when installing the Java Software Development Kit (JDK). To determine which JDK version and patches you need, see Chapter 2, "System Requirements". After you install the JDK, record the absolute path to its location on your system. You will use this path information during the installation.

See the *Selling and Fulfillment Foundation: Properties Guide* for more information about memory parameter values in sandbox.cfg.

7.2.5 Creating a UNIX Account

In a UNIX or Linux environment, you must create a UNIX administrative account on the host server for each installation of 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.

7.2.6 Applying Database Definition Language (DDL) Statements (UNIX/Linux)

When you install 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 Selling and Fulfillment Foundation database user. The rights to create tables, indexes, and so forth 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.

7.2.7 About Silent Installations (UNIX/Linux)

You can use a silent installation process, which automates part of the installation process and limits manual interaction with the installation program. This type of installation is detailed in "Running the Installation Program in UNIX or Linux (from Manually Edited Silent Install File)" on page 7-136.

7.2.7.1 Creating the Silent Installation File

To use the silent install process, you first create a silent installation file using a text editor. The file must contain the entries marked as required in the following table. When you are finished editing the file, record its name and location. You will use this information during the installation.

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 running on MSSQL 2000, this attribute must be set to FALSE. If you are not running on MSSQL, do not include this attribute. **NOTE:** This attribute is case-sensitive.

Example Entry	Description
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. (For more information, see "Key Terms (UNIX/Linux)" on page 7-104.)
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, MSSQL2005/2008, or DB2) Full path to the JDBC driver file(s): • Oracle, Microsoft SQL Server 2005 and 2008 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: • <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	Indicates whether you want to load factory setup defaults during installation (true) or manually after installation (false). For information about manually loading the factory defaults, see Section 13.1.1.3, "Loading the Selling and Fulfillment Foundation Database Factory Defaults After Installation".
COPY_FCXML_TO_REPOSITORY=true	(Required) Must be set to true (default), which specifies that all factory setup files are copied from the default <INSTALL_DIR>/installed_data/<package-name>/factorysetup directory to the <INSTALL_DIR>/repository/factorysetup directory. This enables you to delete the <INSTALL_DIR>/installed_data/<package-name>/factorysetup directory after installation, as it contains other files that take up space. The GUI installer sets this property to true.
NO_DBVERIFY=false	Valid values are true/false. When set to true during installation and installservice, DBVerify will not be run. This means that Selling and Fulfillment Foundation will not generate DDL to make the database like the XML entity repository.
REINIT_DB=true	Valid values are true/false. By default, the value is set to true. If the value is set to false, the Selling and Fulfillment Foundation installation will complete successfully, but no database operation will be performed as part of the installation process.

Example Entry	Description
multischema.applyddl=true	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.
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 Section 7.2.7.1.2, "Sample Multischema.xml File" for more information.
multischema.version=<version_number>	(Required) This attribute indicates which version is being installed. For the Selling and Fulfillment Foundation Release 8.5, you must enter 8.5. **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=8.5	(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>	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 *Selling and Fulfillment Foundation: Properties Guide* for more information about these properties.
ONLINE_DOC_URL=<url>	The value of this property is the URL for the Sterling web-based Online Documentation Library that is in HTML and PDF format: `http://www.sterlingcommerce.com/Documentation/MCSF85/HomePage.htm`
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 Suite 8.5	(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 Table 2–15, "Minimum Memory Requirements" 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. See the *Selling and Fulfillment Foundation: Properties Guide* for more information about these parameters. **Note:** The ADDITIONAL_ANT_JAVA_TASK_ARGS property must not be set for IBM and JRockit JDK.
SUPPORT_MULTIBYTE	Valid values are Y or N. If you are installing on a DB2 or MSSQL server and need to localize your database using a multi-byte character set, set this flag to Y. This ensures that the database column sizes are large enough to handle the multibyte characters correctly

7.2.7.1.1 Sample Silent Installation File

Following is a sample silent installation file:

```
INSTALL_DIR=/<INSTALL_DIR>
DB_VENDOR=Oracle
DB_USER=joe_smith
DB_DATA=l10n
DB_PASS=joes_password
DB_HOST=10.10.23.90
DB_PORT=1221
DB_DRIVERS=/Oracle_Drivers/ojdbc6.jar
DB_DRIVERS_VERSION=11
DB_SCHEMA_OWNER=DB0
multischema.applyddl=false
multischema.enabled=true
multischema.version=8.5
multischema.file=multischema.xml
STERLING_FOUNDATION_PRODUCT_LABEL=smcfs
STERLING_FOUNDATION_PRODUCT_VERSION=8.5
ACCEPT_LICENSE=Y
ACTIVE_DOC_URL=ONLINE
ONLINE_DOC_URL=http://www.sterlingcommerce.com/Documentation/M
CSF85/HomePage.htm
LOCAL_DOC_URL=URL=/smcfsdocs/yfscommon/online_help/en_US/wwhel
p/wwhimpl/js/html/wwhelp.htm
JAVADOC_PRODUCT_LABEL=Sterling Selling and Fulfillment Suite
8.5
JDK64BIT=true
ADDITIONAL_ANT_JAVA_TASK_ARGS=-XX:MaxPermSize=512m
```

```
ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms256mm -J-Xmx1408m
```

> **Note:** See Table 2–15, "Minimum Memory Requirements" for memory parameter values based on your operating system.

7.2.7.1.2 Sample Multischema.xml File

The following sample file is invoked by the installation process if multischema is enabled in the silent installation file. It is customer-created and specifies database account information for four multischema data tables: Metadata, Statistics, System Configuration, and Transaction/Master.

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

You can change only the prarameters that are shown in the following table:

Parameter	Definition
<jdbc_url>	Specify the URL to connect to the database. • If using Oracle, set to: `jdbc:oracle:thin:@<DatabaseServerHostname/IPaddress>:<TNSListenerPortNumber>:<DatabaseSID>` • If using Microsoft SQL Server 2005/2008, set to: `jdbc:sqlserver://<Database ServerHostname>:<PortNumber>;DatabaseName=<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.

Parameter	Definition
<db_driver_class>	Specify the class name of your database driver as follows. • If using Oracle, set to: `oracle.jdbc.driver.OracleDriver` • If using Microsoft SQL Server 2005/2008, 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.

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. The *Selling and Fulfillment Foundation: Multitenant Enterprise Guide* explains the schemas shown in this sample file as well as colonies and other multischema information.

```
<?xml version="1.0" encoding="UTF-8"?>
<colonyconfig>
   <colonies>
      <colony name="DEFAULT" pkprefix="20" version="8.5">
        <schema poolid="DEFAULT_METADATA" tabletype="METADATA"/>
        <schema poolid="DEFAULT_STATISTICS" tabletype="STATISTICS"/>
        <schema poolid="DEFAULT_CONFIGURATION" tabletype="CONFIGURATION"/>
        <schema poolid="DEFAULT_TRANSACTION" tabletype="TRANSACTION"/>
        <schema poolid="DEFAULT_TRANSACTION" tabletype="MASTER"/>
     </colony>
   </colonies>
   <pools>
      <pool id="DEFAULT_METADATA">
             <jdbc>
                 <param name="url" value="jdbc:db2://10.10.20.82:50000/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"/>
            </jdbc>
    </pool>
    <pool id="DEFAULT_STATISTICS">
            <jdbc>
                <param name="url" value="jdbc:db2://10.10.20.82:50000/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"/>
            </jdbc>
    </pool>
    <pool id="DEFAULT_CONFIGURATION">
            <jdbc>
                <param name="url" value="jdbc:db2://10.10.20.82:50000/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"/>
               </jdbc>
     </pool>
     <pool id="DEFAULT_TRANSACTION">
               <jdbc>
                    <param name="url" value="jdbc:db2://10.10.20.82:50000/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"/>
               </jdbc>
           </pool>
     </pools>
</colonyconfig>
```

7.3 Installing Selling and Fulfillment Foundation in a UNIX or Linux Environment

Installing Selling and Fulfillment Foundation in a UNIX or Linux environment includes the following sections:

- "Running the Installation Program in UNIX or Linux (GUI-Based)" on page 7-121
- "Running the Installation Program in UNIX or Linux (Text-Based)" on page 7-129
- "Running the Installation Program in UNIX or Linux (from Manually Edited Silent Install File)" on page 7-136

7.3.1 Before You Install: For AIX Installations Only

During the installation process, you specify the name of the directory to be created for Selling and Fulfillment Foundation. The installation process creates the directory and uses it as the Home folder for the 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`*

where *`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`.

7.3.2 Running the Installation Program in UNIX or Linux (GUI-Based)

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.

> **Note:** The following instructions assume that you received an installation CD. If you downloaded Selling and Fulfillment Foundation or a Service Pack (SP) from the Electronic Software Distribution (ESD) Portal, unzip the downloaded file to an empty directory. The directory containing the unzipped 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.

> **Note:** 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.

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

1. Place the Selling and Fulfillment Foundation installation CD in the appropriate drive.
2. From the installation CD, copy the `SCIInstallWizard.jar` and `SMCFS_8.5.jar` files to your home directory or base directory and change to that directory.

   If you are using FTP to copy the files, verify that your session is set to binary mode.

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

   a. Use a connectivity client to connect to your UNIX/Linux account.

   b. 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)

   > **Note:** In the above command, `:0.0` can be a different value, for example; `:8.0`.

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

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

   > **Note:** 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.

   > **Note:** You **must** be in the directory where SCIInstallWizard.jar resides when issuing this command.

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

   > **Note:** The `-Djavataskargs` property must not be set for IBM and JRockit JDK.

   Example based on Red Hat Enterprise Linux 5.3:

```
<JAVA_HOME>/bin/java -Djavataskargs="-XX:MaxPermSize=512m"
-Dcomptaskargs="-J-Xms256m -J-Xmx1408m" -jar SCIInstallWizard.jar
```

See Table 2–15, "Minimum Memory Requirements" for memory parameter values based on your operating system.

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

The installation dialog box appears in a GUI.

6. Click Next to start the installation program.
7. Review the license agreement, and click Accept to accept the terms.
8. 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.

   > **Note:** If you click No, ensure that you download Version 2.7.1 of xerces and xalan from Apache. Copy the following files into the `jre/lib/endorsed` directory of your JDK:
   >
   > - serializer.jar
   > - xalan.jar
   > - xercesImpl.jar
   > - xml-apis.jar

9. 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, check 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 check 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/Linux" for information about running view scripts after the initial installation and Chapter 13, "Configuring Utilities" for information about manually installing the database DDLs and factory setup.

- 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, leave the option unchecked and click Next.

10. Choose an installation directory for Selling and Fulfillment Foundation. Click Select Folder and navigate to the folder 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.

> **Note:** The GUI installation creates the following installation directory structure:
>
> <INSTALL_DIR>/Foundation
>
> The installation process lays down the Selling and Fulfillment Foundation files and subdirectories under <INSTALL_DIR>/Foundation.

11. Choose the installation jar. 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.
12. Select the database vendor that you want to use (Oracle, Microsoft SQL Server 2005/2008, or DB2) and click Next.
13. If you do not want to enable Multiple Schema Support, leave the box unchecked and skip to Step 13 in these instructions. If you would like

to enable Multiple Schema Support, click the check box, click Next, and proceed as follows:

a. Enter your jdbc driver file. If your database is DB2, also enter your second jdbc driver 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 (For more information, see "Key Terms (UNIX/Linux)" on page 7-104.)
- Database host name (or IP address)
- Database port

> **Note:** If you wish 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 Section 7.2.7.1, "Creating the Silent Installation File".

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. 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. An sample of this file is shown in Section 7.2.7.1.2, "Sample Multischema.xml File".

e. Skip to Step 15.

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

- (Oracle only) Select whether to use the BLOB (binary large object) or the Long Raw data type for caching. You can significantly improve performance by using the BLOB data type. For more information, refer to the the application documentation for slow performance in Oracle.

  To use the BLOB data type, check the check box. To use the Long Raw data type, clear the check box.

- (Microsoft SQL Server 2005/2008 or DB2 only) Select whether to support multibyte character sets.
- Database user name
- Database password
- Confirm database password
- Database catalog name (For more information, see "Key Terms (UNIX/Linux)" on page 7-104.)
- Database host name (or IP address)
- Database port
- (Oracle or Microsoft SQL Server 2005/2008 only) Absolute path and file name for one JDBC driver.
- (DB2 only) Absolute paths and file names for two JDBC drivers.

  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 Selling and Fulfillment Foundation to the DB2 server.

> **Note:** It is not possible to pass multiple database drivers when running the installation program in a UNIX or Linux (GUI-Based) mode. Since DB2 and Microsoft SQL Server 2000 require you to provide multiple database drivers, you should unzip the two available jars and place them under a single jar. The new jar can be provided as input for the database drivers using the GUI mode.

15. 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.

16. The Documentation Access screen is displayed. Select Online (default) or Local for the type of Documentation Library you want to enable.

    The *Selling and Fulfillment Foundation: Properties Guide* contains information about these Documentation Library properties and how to switch them after installation, if you wish.

    Click Next.

17. The installation asks whether this is a 32 or 64-bit JDK. Click the appropriate button. The default is 64-bit JDK.

    Click Next.

18. The Pending Installation Tasks screen is displayed. This screen is read-only. The following tasks are shown:

    - *Verify Sufficient Disk Space*
    - *Install Foundation Components*
    - *Save 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 GUI installer installs Selling and Fulfillment

Foundation in the Foundation subdirectory, unlike the text-based and silent installations, which install the product directly into the folder you specify. Click Next.

19. 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.

20. When the installation is finished, the message *Installation Wizard completed. Please see the installation guide for next steps* 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.

21. 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=8.5
    ```

    – Run the dbverify script on multischema colonies, as described in Section 13.1.1.2, "Verifying the Database Schema".

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

> **Note:** 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 (Oracle or DB2) for production in Chapter 5, "Installing and Configuring Database Tier Software on UNIX/Linux" and Chapter 13, "Configuring Utilities".

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

tmpSterlingInstall.12345678/

You can delete these directories after the installation has completed.

7.3.3 Running the Installation Program in UNIX or Linux (Text-Based)

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

> **Note:** The following instructions assume that you received an installation CD. If you downloaded Selling and Fulfillment Foundation or a Service Pack (SP) from the Electronic Software Distribution (ESD) Portal, unzip the downloaded file to an empty directory. The directory containing the unzipped 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.

> **Note:** 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.

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

> **Note:** A root user cannot install Selling and Fulfillment Foundation.

1. Place the Selling and Fulfillment Foundation installation CD in the appropriate drive.
2. From the installation CD, copy the `SCIInstallWizard.jar` and `SMCFS_8.5.jar` files to a UNIX/Linux directory.

If you are using FTP to copy the files, verify that your session is set to binary mode.

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

4. To begin the installation, type the absolute path to the JDK directory followed by the command:

> **Note:** 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.

> **Note:** You **must** be in the directory where SCIInstallWizard.jar resides when issuing this command.

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

For example:

```
<JAVA_HOME>/bin/java -Djavataskargs="-XX:MaxPermSize=512m"
-Dcomptaskargs="-J-Xms256m -J-Xmx1408m" -jar SCIInstallWizard.jar
```

See Table 2–15, "Minimum Memory Requirements" for memory parameter values based on your operating system.

5. At the "Welcome to the Selling and Fulfillment Foundation Installer" prompt, press Enter.

6. At the "Press Enter to view the license agreement" prompt, press Enter to begin. The first page of the license agreement is displayed. Type N for the next page of the agreement, or scroll directly to the end of the agreement by pressing any other key. After the last page of the license agreement is displayed, the prompt "Do you accept the license? Y or N" is displayed. Press Y for yes, N for no.

7. At the "Select JDK directory" prompt, type the path of the JDK directory. 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.

8. At "Do you want a local copy of the JDK to be created in your installation?" enter the number 1 for yes (or press Enter) or 2 for no.
9. 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 Factory setup?" Use the following bullets as a guide for your answer:
   - If you are upgrading from a previous release, type the number 1 and press Enter. See the *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, type false and press Enter.
   - 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, type the number 1 and press Enter. 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.
10. 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.
11. 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.

Note: The Text-Based installation creates the following installation directory structure:

<INSTALL_DIR>/Foundation

The installation process lays down the Selling and Fulfillment Foundation files and subdirectories under <INSTALL_DIR>/Foundation.

12. 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.

13. At the prompt, "Would you like to enable Multiple Schema support", enter Yes or No. If you enter No, skip to Step 14 in these instructions. If you would like to enable Multiple Schema Support, enter Yes, click Next, and proceed as follows:

    a. When prompted, enter your jdbc driver file. (Oracle or Microsoft SQL Server 2005/2008) Absolute path and file name for one JDBC driver.

       (DB2 only) Absolute paths and file names for two JDBC drivers.

       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.

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

       Starting with Metadata, 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 (For more information, see "Key Terms (UNIX/Linux)" on page 7-104.)
- Database host name (or IP address)
- Database port

> **Note:** If you wish 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 Section 7.2.7.1, "Creating the Silent Installation File".

c. After you enter the Metadata Schema information and press Enter, the Confirm Database Information for Metadata Schema screen (read-only) displays the information you entered on the previous screen. 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. The installation guides you through entering database information for the remaining three database schemas: Statistics, System Configuration, and Transaction/Master. When finished, press Enter.

   The database information you entered for Multischema Support tables is saved in `<INSTALL_DIR>\multischema.xml`. A sample of this file is shown in Section 7.2.7.1.2, "Sample Multischema.xml File".

e. If all database connections are successful, the installation process begins.

Skip to Step 16.

14. 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
- Database catalog name (For more information, see "Key Terms (UNIX/Linux)" on page 7-104.)
- Database host name (or IP address)
- Database host port number (For DB2, use port 50000)
- (Oracle or Microsoft SQL Server 2005/2008 only) Absolute path and file name for one JDBC driver.
- (DB2 only) Absolute paths and file names for two JDBC drivers.

  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.

15. After you have entered the database information item and pressed Enter, the Confirm Database Information screen (read-only) displays the information you entered on the previous screen. 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).

16. When prompted, enter the type of Documentation Library you want to enable (Online or Local).

The *Selling and Fulfillment Foundation: Properties Guide* contains information about these Documentation Library properties and how to switch them after installation, if you wish.

Press Enter.

17. When prompted, enter whether you are using a 32-bit or 64-bit JDK.

    Press Enter.

    The installation process begins.

18. 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.

19. 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=8.5
      ```

    - Run the dbverify script on multischema colonies, as described in Section 13.1.1.2, "Verifying the Database Schema".

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

> **Note:** 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 (Oracle or DB2) for production in Chapter 5, "Installing and Configuring Database Tier Software on UNIX/Linux" and Chapter 13, "Configuring Utilities".

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

tmpSterlingInstall.12345678/

You can delete these directories after the installation has completed.

7.3.4 Running the Installation Program in UNIX or Linux (from Manually Edited Silent Install File)

Use the following instructions to install in a UNIX or Linux environment from a command line, using a manually edited silent installation file.

> **Note:** The following instructions assume that you received an installation CD. If you downloaded the Selling and Fulfillment Foundation or a Service Pack (SP) from the Electronic Software Distribution (ESD) Portal, unzip the downloaded file to an empty directory. The directory containing the unzipped 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.

> **Note:** 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.

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

> **Note:** A root user cannot install Selling and Fulfillment Foundation.

1. Place the Selling and Fulfillment Foundation installation CD in the appropriate drive.
2. From the installation CD, copy the `SMCFS_8.5.jar` to a UNIX/Linux directory.

If you are using FTP to copy the file, verify that your session is set to binary mode.

3. Set up your silent installation file, using the guidelines in “About Silent Installations (UNIX/Linux)” on page 7-110. Record the path to your silent installation file.

4. Type one of the following commands, which include paths to the JDK, the application jar file, and the silent installation file:

> **Note:** On Linux, do not use any soft or symbolic links in the path to the jar file. Make sure that you specify the full path to the jar file.

> **Note:** The directory path to `SMCFS_8.5.jar` cannot include any spaces.

- If you are installing a new Selling and Fulfillment Foundation system, enter:

  ```
  <JAVA_HOME>/bin/java -jar SMCFS_8.5.jar -f install.silent
  ```

  The `install.silent` file is the one you created during preinstallation setup, as explained in “About Silent Installations (UNIX/Linux)” on page 7-110.

- If you are doing an upgrade, set LOAD_FACTORY_SETUP=false in your silent install file and enter:

  ```
  <JAVA_HOME>/bin/java -jar SMCFS_8.5.jar -f install.silent
  -cluster
  ```

  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_8.5.jar -f install.silent
  -cluster -nocopyjvm
  ```

  If the application is running, stop the previous installation before proceeding.

See the Upgrade Guide for more detailed information about performing a Selling and Fulfillment Foundation upgrade.

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

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

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=8.5
```

- Run the dbverify script on multischema colonies, as described in Section 13.1.1.2, "Verifying the Database Schema".

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

See Section 3.3.1, "Post Installation Recommendations" for security measures that Sterling recommends you should consider.

> **Note:** 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 (Oracle or DB2) for production in Chapter 5, "Installing and Configuring Database Tier Software on UNIX/Linux" and Chapter 13, "Configuring Utilities".

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

```
tmpSterlingInstall.12345678/
```

You can delete these directories after the installation has completed.

8

Installing Selling and Fulfillment Foundation in a Windows Environment

This chapter explains how to install Selling and Fulfillment Foundation in a Windows environment. This chapter also provides the information required to complete Step of the See "Installation Checklist" on page 1-1. for this environment.

8.1 Before You Begin

If you are upgrading from a prior release, see the *Selling and Fulfillment Foundation: Upgrade Guide* ***before*** continuing with the installation of Selling and Fulfillment Foundation.

Before installing Selling and Fulfillment Foundation, ensure that you already have installed the appropriate software listed in Chapter 2, "System Requirements".

Throughout this document, <INSTALL_DIR> refers to the directory where you install Selling and Fulfillment Foundation. For example, <INSTALL_DIR>=C:/Supply_Chain_Apps.

8.2 Preinstallation Setup for a Windows Environment

The following topics will assist you with the preinstallation tasks when planning to install Selling and Fulfillment Foundation in a Windows environment:

- "Key Terms (Windows)" on page 8-142
- "Checklist for Windows Preinstallation" on page 8-142

- "Checking System Requirements (Windows)" on page 8-144
- "Installing the Java Software Development Kit (Windows)" on page 8-144
- "About Silent Installations (Windows)" on page 8-146

8.2.1 Key Terms (Windows)

The following terms and definitions will assist you in understanding the concepts of installing Selling and Fulfillment Foundation in a Windows environment discussed in this document:

- 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.

8.2.2 Checklist for Windows Preinstallation

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

Note: 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.

You may want to make a copy of the following checklist and use it to record the information you collect for installing Selling and Fulfillment Foundation:

Step	Description	Your Notes
1	Verify that your system meets the hardware and software requirements specified for Release 8.5. For more information, refer to "Checking System Requirements (Windows)" on page 8-144.	

Step	Description	Your Notes
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. • Version of the JDK • Absolute path to the JDK. The path name can not include spaces. For more information, refer to "Installing the Java Software Development Kit (Windows)" on page 8-144.	
4	Determine and record information about your Oracle, Microsoft SQL Server 2005/2008 or DB2 database server. Be aware that this information may be case sensitive. • Database vendor • Database user name and associated password • Database catalog name (For more information, see "Key Terms (Windows)" on page 8-142.) • Database host name (or IP address) • Database host port number • (Oracle or Microsoft SQL Server 2005 only) Absolute path and file name for one JDBC driver. • (DB2 only) Absolute paths and file names for two JDBC drivers. 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	Decide if you are going to manually or automatically apply database definition language (DDL) statements (schema) to the database. For more information, refer to "Applying Database Definition Language (DDL) Statements (Windows)" on page 8-145.	

Step	Description	Your Notes
6	Determine and record the directory in which you plan to install Selling and Fulfillment Foundation. • 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 "About Silent Installations (Windows)" on page 8-146 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	If you are running a silent installation, manually create your silent installation file. For more information, refer to "About Silent Installations (Windows)" on page 8-146.	

8.2.3 Checking System Requirements (Windows)

Before you begin, verify that your system meets the hardware and software requirements specified for Release 8.5. The hardware requirements listed are the minimum required to run Selling and Fulfillment Foundation. For current information, see Chapter 2, "System Requirements".

8.2.4 Installing the Java Software Development Kit (Windows)

You must install the Java Software Development Kit (JDK) and the patches specific to your system. To determine which JDK version and patches you need, see Chapter 2, "System Requirements". After you install the JDK, record the absolute path to its location on your system.

You must supply the absolute path when you install Selling and Fulfillment Foundation.

Caution: In Windows, the directory name where the JDK resides cannot include a space.

The JAVA_HOME environment variable needs to be changed to equal the directory where you installed JDK.

See the *Selling and Fulfillment Foundation: Properties Guide* for more information about memory parameter values in sandbox.cfg.

8.2.5 Applying Database Definition Language (DDL) Statements (Windows)

When you install 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 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.

8.2.6 About Silent Installations (Windows)

You can create and use a silent installation process, which automates part of the installation process and limits manual interaction with the installation program. This type of installation is detailed in "Running the Installation Program in Windows (Manually Edited Silent Install File)" on page 8-163.

8.2.6.1 Creating the Silent Installation File

You create the silent installation file using a text editor. The file must contain the entries marked as required in the following table. When you are finished editing the file, record its name and location. You will use this information during the installation.

> **Note:** 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:/Sterling
>
> or
>
> D:/Applications/Myinstall

Example Entry	Description
INSTALL_DIR=<INSTALL_DIR>	(Required) Directory in which to install. **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.

Example Entry	Description
MSSQL2005=<TRUE/FALSE>	If you are running on MSSQL 2005 or later, this attribute must be set to TRUE. If you are running on MSSQL 2000, this attribute must be set to FALSE. 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 (Windows)" on page 8-142.)
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, MSSQL2005/2008, or DB2) Full path to the JDBC driver file(s): • Oracle or Microsoft SQL Server 2005/2008 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: • `<JDBC_driver_dir>/jdbc.jar` • `<JDBC_driver_dir>/db2_1_jdbc.jar:<JDBC_driver_dir>/db2_2_jdbc.jar`

Example Entry	Description
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). For information about manually loading the factory defaults, see Section 13.1.1.3, "Loading the Selling and Fulfillment Foundation Database Factory Defaults After Installation".
COPY_FCXML_TO_REPOSITORY=true	(Required) Must be set to true (default), which specifies that all factory setup files are copied from the default <INSTALL_DIR>/installed_data/<package-name>/factorysetup directory to the <INSTALL_DIR>/repository/factorysetup directory. This enables you to delete the <INSTALL_DIR>/installed_data/<package-name>/factorysetup directory after installation, as it contains other files that take up space. The GUI installer sets this property to true.
NO_DBVERIFY	Valid values are true/false. When set to true during installation and installservice, dbverify will not be run. This means that Selling and Fulfillment Foundation will not generate DDL to make the database like the XML entity repository.

Example Entry	Description
REINIT_DB=true	Valid values are true/false. By default, the value is set to true. If the value is set to false, the Selling and Fulfillment Foundation installation will complete successfully, but no database operation will be performed as part of the installation process.
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.
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 Section 8.2.6.1.2, "Sample Multischema.xml File" for more information.
multischema.version=<version_number>	(Required)This attribute indicates which version is being installed. For the Selling and Fulfillment Foundation Release 8.5, you must enter 8.5. **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=8.5	(Required) Specifies the product version you are installing.

Example Entry	Description
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.
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 *Selling and Fulfillment Foundation: Properties Guide* for more information about these properties.
ONLINE_DOC_URL=<url>	The value of this property is the URL for the Sterling web-based Online Documentation Library that is in HTML and PDF format: `http://www.sterlingcommerce.com/Documentation/MCSF85/HomePage.htm`
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 Suite 8.5	(Required) Specifies the Javadocs to be installed.

Example Entry	Description
JDK64BIT=<true/false>	Specifies whether you are using a 32-bit JDK or a 64-bit JDK. Default=true
ADDITIONAL_ANT_JAVA_TASK_ARGS=-XX:MaxPermSize=512m ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms256m -J-Xmx1408m	See Table 2–15, "Minimum Memory Requirements" 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. See the *Selling and Fulfillment Foundation: Properties Guide* for more information about these parameters.
SUPPORT_MULTIBYTE	Valid values are Y or N. If you are installing on a DB2 or MSSQL server and need to localize your database using a multi-byte character set, set this flag to Y. This ensures that the database column sizes are large enough to handle the multibyte characters correctly

8.2.6.1.1 Sample Silent Installation File

Following is a sample silent installation file:

```
INSTALL_DIR=C:/<INSTALL_DIR>

DB_VENDOR=Oracle

DB_USER=joe_smith

DB_DATA=l10n

DB_PASS=joes_password

DB_HOST=10.10.42.93

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=8.5

multischema.file=multischema.xml

STERLING_FOUNDATION_PRODUCT_LABEL=smcfs

STERLING_FOUNDATION_PRODUCT_VERSION=8.5

ACCEPT_LICENSE=Y

ACTIVE_DOC_URL=ONLINE

ONLINE_DOC_URL=http://www.sterlingcommerce.com/Documentation/M
CSF85/HomePage.htm

LOCAL_DOC_URL=URL=/smcfsdocs/yfscommon/online_help/en_US/wwhel
p/wwhimpl/js/html/wwhelp.htm

JAVADOC_PRODUCT_LABEL=Sterling Selling and Fulfillment Suite
8.5

JDK64BIT=true

ADDITIONAL_ANT_JAVA_TASK_ARGS=-XX:MaxPermSize=512m
```

```
ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms256m -J-Xmx1408m
```

Note: See Table 2–15, "Minimum Memory Requirements" for memory parameters based on your operating system.

8.2.6.1.2 Sample Multischema.xml File

The following sample file is invoked by the installation process if multischema is enabled in the silent installation file. It is customer-created and specifies database account information for four multischema data tables: Metadata, Statistics, System Configuration, and Transaction/Master.

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

You can change only the prarameters that are shown in the following table:

Parameter	Definition
<jdbc_url>	Specify the URL to connect to the database. • If using Oracle, set to: `jdbc:oracle:thin:@<DatabaseServerHostname/IPaddress>:<TNSListenerPortNumber>:<DatabaseSID>` • If using Microsoft SQL Server 2005/2008, set to: `jdbc:sqlserver://<Database ServerHostname>:<PortNumber>;DatabaseName=<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.

Parameter	Definition
<db_driver_class>	Specify the class name of your database driver as follows. • If using Oracle, set to: `oracle.jdbc.driver.OracleDriver` • If using Microsoft SQL Server 2005/2008, 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.

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. The *Selling and Fulfillment Foundation: Multitenant Enterprise Guide* explains the schemas shown in this sample file as well as colonies and other multischema information.

```
<?xml version="1.0" encoding="UTF-8"?>
<colonyconfig>
   <colonies>
      <colony name="DEFAULT" pkprefix="20" version="8.5">
        <schema poolid="DEFAULT_METADATA" tabletype="METADATA"/>
        <schema poolid="DEFAULT_STATISTICS" tabletype="STATISTICS"/>
        <schema poolid="DEFAULT_CONFIGURATION" tabletype="CONFIGURATION"/>
        <schema poolid="DEFAULT_TRANSACTION" tabletype="TRANSACTION"/>
        <schema poolid="DEFAULT_TRANSACTION" tabletype="MASTER"/>
     </colony>
   </colonies>
   <pools>
      <pool id="DEFAULT_METADATA">
             <jdbc>
                 <param name="url" value="jdbc:db2://10.10.20.82:50000/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"/>
            </jdbc>
     </pool>
     <pool id="DEFAULT_STATISTICS">
            <jdbc>
                <param name="url" value="jdbc:db2://10.10.20.82:50000/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"/>
            </jdbc>
     </pool>
     <pool id="DEFAULT_CONFIGURATION">
            <jdbc>
                <param name="url" value="jdbc:db2://10.10.20.82:50000/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"/>
               </jdbc>
      </pool>
      <pool id="DEFAULT_TRANSACTION">
               <jdbc>
                    <param name="url" value="jdbc:db2://10.10.20.82:50000/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"/>
               </jdbc>
           </pool>
      </pools>
</colonyconfig>
```

8.3 Installing Selling and Fulfillment Foundation in a Windows Environment

Installing Selling and Fulfillment Foundation in a Windows environment includes the following sections:

- "Running the Installation Program in Windows (GUI-Based)" on page 8-156
- "Running the Installation Program in Windows (Manually Edited Silent Install File)" on page 8-163

8.3.1 Running the Installation Program in Windows (GUI-Based)

Note: The following instructions assume that you received an installation CD for Selling and Fulfillment Foundation. If you downloaded Selling and Fulfillment Foundation or a Service Pack (SP) from the Electronic Software Distribution (ESD) Portal, unzip the downloaded file to an empty directory. Do not change the directory structure of the newly unzipped files. The directory containing the unzipped 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.

Note: To install more than one instance of 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.

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

> **Note:** A root user cannot install Selling and Fulfillment Foundation.

1. Close all open Windows programs and any command prompt windows.
2. Copy the `SCIInstallWizard.jar` and `SMCFS_8.5.jar` files from your installation CD to a Windows directory.
3. Start the installation process. From a command prompt, enter the following command:

   > **Note:** You **must** be in the directory where SCIInstallWizard.jar resides when issuing this command.

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

   Example based on Red Hat Enterprise Linux 5.3:

   ```
   <JAVA_HOME>/bin/java -Djavataskargs="-XX:MaxPermSize=512m"
   -Dcomptaskargs="-J-Xms256m -J-Xmx1408m" -jar SCIInstallWizard.jar
   ```

   See Table 2–15, "Minimum Memory Requirements" for memory parameter values based on your operating system.

   The installation dialog box appears.
4. Click Next to start the installation program.
5. Review the license agreement, and click Accept to accept the terms.
6. Type the path of your JDK directory, or search for it using the Select Folder button.

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

7. 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, check 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 check 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" for information about running view scripts after the initial installation and Chapter 13, "Configuring Utilities" for information about manually installing the database DDLs and factory setup.

   - 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, leave the option unchecked and click Next.

8. Choose an installation directory for Selling and Fulfillment Foundation. Click Select Folder and navigate to the folder you want to use as the installation 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 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.

Note: The GUI installation creates the following installation directory structure:

<INSTALL_DIR>\Foundation

The installation process lays down the Selling and Fulfillment Foundation files and subdirectories under <INSTALL_DIR>\Foundation.

9. Select the database vendor that you want to use (Oracle, Microsoft SQL Server 2005/2008, or DB2) and click Next.
10. If you do not want to enable Multiple Schema Support, leave the box unchecked and skip to Step 11 in these instructions. If you would like to enable Multiple Schema Support, click the check box, click Next, and proceed as follows:
    a. Enter your jdbc driver file. If your database is DB2, also enter your second jdbc driver 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 (For more information, see “Key Terms (Windows)” on page 8-142.)
       - Database host name (or IP address)
       - Database port

Note: If you wish 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 Section 8.2.6.1, "Creating the Silent Installation File".

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. 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. An sample of this file is shown in Section 8.2.6.1.2, "Sample Multischema.xml File".

e. Skip to Step 13.

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

- Database user name
- Database password
- Confirm database password
- Database catalog name (For more information, see “Key Terms (Windows)” on page 8-142.)
- Database host name (or IP address)
- Database port
- (Oracle or Microsoft SQL Server 2005/2008 only) Absolute path and file name for one JDBC driver.

- (DB2 only) Absolute paths and file names for two JDBC drivers.

  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 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.

13. The Documentation Access screen is displayed. Select Online (default) or Local for the type of Documentation Library you want to enable.

    The *Selling and Fulfillment Foundation: Properties Guide* contains information about these Documentation Library properties and how to switch them after installation, if you wish.

    Click Next.

14. The installation asks whether this is a 32 or 64-bit JDK. Click the appropriate button. The default is 64-bit JDK.

    Click Next.

15. The Pending Installation Tasks screen is displayed. This screen is read-only. The following tasks are shown:

    - *Verify Sufficient Disk Space*
    - *Install Foundation Components*
    - *Save 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 Selling and Fulfillment Foundation in the Foundation subdirectory, unlike the silent installation, which installs the product directly into the folder you specify. Click Next.

16. 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.

17. When the installation is finished, the message *Installation Wizard completed. Please see the installation guide for next steps* 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.

18. 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=8.5
    ```

    – Run the dbverify script on multischema colonies, as described in Section 13.1.1.2, "Verifying the Database Schema".

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

Note: 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 2005/2008, Oracle 11g, or DB2) for production in Chapter 6, "Installing and Configuring Database Tier Software on Windows" and Chapter 13, "Configuring Utilities".

8.3.2 Running the Installation Program in Windows (Manually Edited Silent Install File)

Use the following instructions to install in a Windows environment from a command line, using a manually edited silent installation file.

> **Note:** The following instructions assume that you received an installation CD. If you downloaded Selling and Fulfillment Foundation or a Service Pack (SP) from the Electronic Software Distribution (ESD) Portal, unzip the downloaded file to an empty directory. The directory containing the unzipped 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.

> **Note:** 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.

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

> **Note:** A root user cannot install Selling and Fulfillment Foundation.

1. Place the Selling and Fulfillment Foundation installation CD in the appropriate drive.
2. From the installation CD, copy the `SMCFS_8.5.jar` to a Windows directory.

   If you are using FTP to copy the file, verify that your session is set to binary mode.

3. Set up your silent installation file, using the guidelines in “About Silent Installations (Windows)” on page 8-146. Record the path to your silent installation file.

4. At a command prompt, type one of the following commands, which include paths to the JDK, the SMCFS_8.5.jar file, and the silent installation file:

> **Note:** The directory path to SMCFS_8.5.jar cannot include any spaces.

- If you are installing a new Selling and Fulfillment Foundation system, enter the following:

```
<JAVA_HOME>\bin\java -jar SMCFS_8.5.jar -f install.silent
```

  The Silent Installation File is the one you created during preinstallation setup, as explained in “Creating the Silent Installation File” on page 8-146.

- If you are doing an upgrade, set LOAD_FACTORY_SETUP=false in your silent install file and enter:

```
<JAVA_HOME>/bin/java -jar SMCFS_8.5.jar -f install.silent -cluster
```

  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_8.5.jar -f install.silent -cluster
-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 *Installation has completed successfully* when done.

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=8.5
```

- Run the dbverify script on multischema colonies, as described in Section 13.1.1.2, "Verifying the Database Schema".

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

See Section 3.3.1, "Post Installation Recommendations" for security measures that Sterling recommends you should consider.

> **Note:** 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 2005/2008, Oracle 11g, or DB2) for production in Chapter 6, "Installing and Configuring Database Tier Software on Windows" and Chapter 13, "Configuring Utilities".

9

Installing Selling and Fulfillment Foundation Language Pack

This chapter explains how to install, load the factory defaults, and check the import mode of the Selling and Fulfillment Foundation language packs. This chapter also provides the information required to complete Step indicated on the Installation Checklist on page 1-1.

9.1 Installing the Language Pack

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

> **Note:** Before installing the language pack, ensure that you have successfully installed Selling and Fulfillment Foundation, Release 8.5.

9.2 Setting up Properties

Use the <INSTALL_DIR>/properties/customer.overrides.properties file to set the yfs.install.localecode property to <your locale code>. For more information about overriding properties using the `customer.overrides.properties` file, see the *Selling and Fulfillment Foundation: Properties Guide*.

9.3 Loading the Selling and Fulfillment Foundation Language Pack Factory Defaults

Prior to loading the 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/Linux" or Chapter 6, "Installing and Configuring Database Tier Software on Windows", as the case may be.

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

To load the language-specific factory defaults, run the `loadDefaults.sh` script for UNIX and Linux or 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>_<cou
ntry>_locale_installer.xml
<INSTALL_DIR>\repository\factorysetup\complete\XMLS
```

For more information about the configuration steps involved in loading the factory defaults, see Section 13.1.1.3, "Loading the Selling and Fulfillment Foundation Database Factory Defaults After Installation".

9.3.1 Loading the Selling and Fulfillment Foundation Language Pack Translations

Prior to loading the Selling and Fulfillment Foundation Language Pack factory defaults, ensure that you have successfully completed all the instructions provided in Chapter 8, "Installing Selling and Fulfillment Foundation in a Windows Environment".

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:

For Windows:

```
sci_ant.cmd -f localizedstringreconciler.xml import
-Dsrc=<INSTALL_DIR>/repository/factorysetup/complete_installat
ion/XMLS -Dbasefilename=ycplocalizedstrings
```

where <INSTALL_DIR> refers to the installation directory.

For UNIX:

```
sci_ant.sh -f localizedstringreconciler.xml import
-Dsrc=<INSTALL_DIR>/repository/factorysetup/complete_installat
ion/XMLS -Dbasefilename=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 Sterling Call Center and 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.

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

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

9.3.2 Switching the 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 *Selling and Fulfillment Foundation: Localization Guide*.

9.4 Creating and Deploying the Enterprise Archive

If you are installing both 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 14, "Deploying Selling and Fulfillment Foundation".

9.5 Configuring Locales

Selling and Fulfillment Foundation runs on any locale that Java supports. If you want to run 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:

1. Enter locale -a. A list of locales is displayed.
2. Set your locate 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:

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.

10

Installing the Print Server

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

This chapter also provides the information required to complete Step indicated on the Installation Checklist on page 1-1.

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*.

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

10.1 Installation of 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*.

Selling and Fulfillment Foundation supports printing in the following modes:

- File Copy Mode

- TCP/IP Sockets Mode

  The `yfs.loftware.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.loftware.tcpip.sockets` property to Y. For more information about overriding properties using the `customer.overrides.properties` file, see the *Selling and Fulfillment Foundation: Properties Guide*.

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

10.2 Define 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*.

10.3 Define Printers for the Sterling WMS Installation

For more information about configuring printers for the Sterling WMS, see the *Sterling Warehouse Management System: Configuration Guide*.

10.4 Copying the Sterling WMS Standard Label Formats

The Sterling WMS 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.

10.5 Installation of JasperReports

JasperReports is an 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 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 JasperReports, Selling and Fulfillment Foundation uses the jasperreports-1.2.0.jar file. For more information about JasperReports and supporting jars and components, see the `readme.html` file located either in the <INSTALL_DIR>/xapidocs/code_examples/jasperreports directory.

11

Installing the Weighing Scale

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

This chapter also provides the information required to complete Step indicated on the "Installation Checklist" on page 1-1.

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

11.1 Installation of the Weighing Scale

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

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

1. Launch your Internet Explorer browser.
2. In the Address bar, type
   `http://<hostname>:<portnumber>/smcfs/yfscommon/win32com.dll`

   and press Enter. The File Download window appears.
3. Choose Save. The Save As window appears.
4. Save the file in any directory present in the System Class path. For example, on Windows NT, go to `C:/WINNT/system32`.

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

For more information about system requirements, see Chapter 2, "System Requirements".

12

Installing the Mobile Application

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

This chapter also provides the information required to complete Step indicated on the "Installation Checklist" on page 1-1.

For information on PocketPC and WinCE mobile terminal system requirements, see Chapter 2, "System Requirements".

> **Important:** Install the Microsoft .NET Compact Framework on your local PC, BEFORE installing the Mobile Application. For more information about the supported versions for the Microsoft .NET Compact Framework, see Section 2.10, "Mobile Application Requirements". This may be downloaded from http://www.microsoft.com.
>
> Installing the Microsoft .NET Compact Framework creates multiple .CAB files on the system, for multiple operating systems and processors of the device.

12.1 Installation on Mobile Terminals

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

1. Connect the Mobile Terminal to the PC that has ActiveSync installed. For more information, refer to the Communications section of the *Symbol Installation Guide*.

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 webserver and use the Internet Explorer browser on the device to download it.

If the Sterling WMS 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, or WCE4.ARMV4T.

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

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.

> **Note:** 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.

> **Note:** Servers include application servers used for production, test, and other environments, if applicable.

> **Note:** The Mobile Application does not support http:// and https:// at the same time, that is, the `YantraHostList.xml` file must not contain entries for both securemode and non-securemode.

10. Locate the properties file for your mobile device as specified in Table 12–1 and rename that file to `YMAProperties.xml`.

Table 12–1 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 *Symbol Device Installation* documentation.

The *Symbol Device Installation* 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:** Sterling Commerce recommends you to configure the barcode scanner such that the TAB character is suffixed with the scanned barcode data.

12.1.1 Ensuring Re-Installation 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.

12.1.1.1 On a PocketPC Mobile Terminal

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.

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 Section 12.1, "Installation on Mobile Terminals".
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.

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.

12.1.1.2 On a Symbol VRC7900 WinCE Mobile Terminal

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:

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 Section 12.1, "Installation on Mobile Terminals" on page 12-179.
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.

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

12.1.1.3 On a Denso BHT400B Win CE 5.0 Mobile Terminal

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:

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 Section 12.1, "Installation on Mobile Terminals".

> **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.

12.2 Installing on VT220 Mobile Terminals

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.

12.2.1 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, Chapter 2, "System Requirements".

12.2.2 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 run-time dependencies on libgcc and gettext. Therefore, you must install these packages while installing libiconv.

12.2.3 Installing the Mobile Application on VT220 Mobile Terminals

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:

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 12–2 Operating System to OS-folder-name

Operating System	OS-folder-name
AIX	aix
HP-UX on Intel Itanium (ia64) Processor	itanuim
Red Hat Enterprise Linux WS release 3	linux
Solaris on SPARC Processor	solaris
Solaris 10 on AMD Opteron Processor	solaris10

2. Grant execute permission to <VT220_HOME>/yantramobileapp and <VT220_HOME>/keyseq.

3. Set up the VT220 emulation terminal as described in "Setting Up the Terminal" on page 12-188. Using the <VT220_HOME>/keyseq binary, verify that the keys F1 through F12 display the respective keys on the keyseq program output.

Setting Up the Terminal

When setting up a terminal for use with the Mobile Application for VT220 terminal emulation, you must perform the following preliminary actions:

1. Set your environment variable TERM to vt220.
2. Under the <VT220_HOME> directory, create a directory called terminfo.
3. 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 de-compiles the terminal information, and the resulting file can be edited to map the keystrokes observed by running keyseq.
4. 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>
   ```
5. 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.

6. 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 12–3, "Terminal Information - Common Keys and Codes" on page 12-190 to decide which values to edit.

   In the example above, the F1 key maps to kf1 (from Table 12–3). Therefore, you must change the value of kf1 in the `vt220.ti` file (generated in Step 3) to:

   `\E[11~`

7. Repeat Step 6 for all the keys you want to map.

8. Re-compile the edited `vt220.ti` file as directed in Step 5.

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.

The most common keys and their codes in the terminal information file are in Table 12–3:

Table 12–3 Terminal Information - Common Keys and Codes

Code	Key
kcub1	Left arrow
kcuf1	Right arrow
kcuu1	Up arrow
kcud1	Down arrow
kf1 – kf12	F1 – F12 keys

To launch the Mobile Application using the VT220 emulation terminal and access context-sensitive help:

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".

13

Configuring Utilities

Selling and Fulfillment Foundation provides script files (.sh for UNIX and .cmd for Windows) that you must customize using the directions provided in this chapter.

This chapter describes all the utilities supplied by 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.

This chapter also provides instructions required to complete Step indicated on the Installation Checklist.

13.1 Installation Utilities

Installation utilities enable you to install 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 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 Section 13.1.1.3, "Loading the Selling and Fulfillment Foundation Database Factory Defaults After Installation".

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 Section 13.1.1, "Creating Database Schemas and Loading Factory Defaults After Installation". For detailed information about setting up and running dbverify, see Section 13.1.1.2, "Verifying the Database Schema".

install3rdparty

This utility copies supplied resources into the 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 patch 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.

13.1.1 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 Section 7.3.2, "Running the Installation Program in UNIX or Linux (GUI-Based)" or Section 8.3.1, "Running the Installation Program in Windows (GUI-Based)".
- 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 Section 7.3.3, "Running the Installation Program in UNIX or Linux (Text-Based)".
- 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 and use the -cluster option as part of the installation command. For more information about silent installation, see Section 7.2.7, "About Silent Installations (UNIX/Linux)" or Section 8.2.6, "About Silent Installations (Windows)".

13.1.1.1 Updating Properties Files After Installation

After installing 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 *Selling and Fulfillment Foundation: Properties Guide*.

13.1.1.2 Verifying the Database Schema

You can run the dbverify utility to verify the database schema as follows:

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, edit the <INSTALL_DIR>/bin/dbverify.sh (or .cmd on Windows) script. For MS SQL and DB2 databases only, add the -DBNAME <database_name> parameter as mentioned below. For example,

```
%JAVA_HOME%\bin\java
com.yantra.tools.dbverify.DbVerifyCommandLine -b %<INSTALL_DIR>%
-DBNAME %<database_name>% -u %USERNAME% -p %PASSWD% -d %DRIVER%
-url %URL% -g Y -DT
%<INSTALL_DIR>%/repository/datatypes/datatypes.xml
```

   Where <database_name> refers to the name of the database for which text search feature is enabled. The -DBNAME parameter is required only for MS SQL and DB2 databases.

   Note: If you 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_<Colony_Id>_<TableType>_TextIndexUpdates.sql file. This file name syntax used in multischema deployments contains:

   - The name of the Colony Id, which can be up to 40 characters in length
   - The schema's table type, which could be CONFIGURATION, MASTER, TRANSACTION, STATISTICS, or METADATA.

   For example, the script for the DEFAULT colony's CONFIGURATION create and drop scripts would be named:

   EFrame_DEFAULT_CONFIGURATION_TextIndexUpdates.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 following scripts are generated:

> **Note:** All scripts listed below can be found in the `<INSTALL_DIR>/bin` directory.

Single Schema Script Name	Multischema Script Name	Description of the script
`EFrame_Sequence.sql`	`EFrame_<Colony_Id>_<TableType>_Sequence.sql`	Contains all the additional sequences that need to be created. **Note:** If you are using an Microsoft SQL Server 2005/2008 database, the `EFrame_Sequence.sql` script is not created when you run the dbverify command.
`EFrame_TableChanges.sql`	`EFrame_<Colony_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.

Single Schema Script Name	Multischema Script Name	Description of the script
EFrame_Drops.lst	EFrame_<Colony_Id>_<TableType>_Drops.lst	This list is provided as an example of tables that can be removed. **Note:** Selling and Fulfillment Foundation does not provide a .sql file for removing tables from the database.
EFrame_IndexAdds.sql	EFrame_<Colony_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_<Colony_Id>_<TableType>_IndexDrops.sql	Removes any extra indexes in the database.
EFrame_TextIndexAdds.sql	EFrame_<Colony_Id>_<TableType>_TextIndexAdds.sql	Adds new text search indexes that need to be created in the database.
EFrame_TextIndexDrops.sql	EFrame_<Colony_Id>_<TableType>_TextIndexDrops.sql	Removes text search indexes from the database.
EFrame_TextIndexModify.sql	EFrame_<Colony_Id>_<TableType>_TextIndexModify.sql	Updates the text search indexes in the database.
EFrame_TextIndexUpdates.sql	EFrame_<Colony_Id>_<TableType>_TextIndexUpdates.sql	When executed, updates the content of the text indexes.
EFrame_UpdateQueries.sql	EFrame_<Colony_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_<Colony_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 lists. 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.

Oracle

Multischema deployment:

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`.

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.

Additionally, for a multischema deployment, run `yfs_addnl_index.sql` in `<INSTALL_DIR>/database/oracle/scripts/CustomDBIndexes`

...where <tabletype> is `configuration` and `transaction`.

Single-Schema Deployment:

Run the following script:
`<INSTALL_DIR>/database/oracle/scripts/yfs_master_db_script.sql`

Microsoft SQL Server 2005/2008

Multischema Deployment:

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`.

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.

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`.

Single-Schema Deployment:

Run the following script:
`<INSTALL_DIR>/database/sqlserver/scripts/yfssqlserver_master_db_script.cmd`

DB2

Multischema Deployment:

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`.

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.

Additionally, for a multischema deployment, run `yfs_addnl_index.sql` in

`<INSTALL_DIR>/database/db2/scripts/CustomDBIndexes`

...where `<tabletype>` is `configuration` and `transaction`.

Single-Schema Deployment:

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

13.1.1.3 Loading the Selling and Fulfillment Foundation Database Factory Defaults After Installation

To load the Selling and Fulfillment Foundation database factory defaults after the product installation, load the defaults using the script applicable to your operating system. 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.

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 overide 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 *Selling and Fulfillment Foundation: Properties Guide*.

13.1.1.4 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.

13.1.2 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/fairlopmaint/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/zip files
-j for jar/zip 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/welogic/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: 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 `Dynamicclasspath.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.

13.2 Development Utilities

Development utilities enable you to customize Selling and Fulfillment Foundation to suit your business needs. They are for use while running Selling and Fulfillment Foundation in development mode.

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 *Selling and Fulfillment Foundation: Configuration Deployment Tool Guide*.

Transaction Data Truncation Tool

When deploying Selling and Fulfillment Foundation to a production environment, you may not want to include all of your transaction data. 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:

1. From the <INSTALL_DIR>/bin directory use the following command appropriate for your database:

   For Oracle and SQL Sever:

   ```
   ./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.

13.3 Runtime Utilities

These utilities start processes that run in the background. The setup of these utilities is described in detail in the *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 Selling and Fulfillment Foundation Integration Server allows Selling and Fulfillment Foundation to collaborate with different systems, organizations, and businesses all through a standard, uniform interface to all systems. The 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.

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<flowNam
e/systemApiName> <is firstParameter Flow (Y/N)> <xmlFileName>
```

where:

- 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.

13.3.1 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 *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 14, "Deploying Selling and Fulfillment Foundation".

13.3.1.1 WebLogic

Include the following jar files for WebLogic JMS:

<INSTALL_DIR>/bin/install3rdparty script to install the `wlfullclient.jar` file and include it in the `AGENTDynamicclasspath.cfg` dynamic classpath file.

For more information about developing a WebLogic Full Client, see http://download.oracle.com/docs/cd/E13222_01/wls/docs100/client/t3.html

Note: For more information about using the install3rdparty script, see Section 13.1.2, "Installing Third-Party JAR Files".

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`

Note: To create wlfullclient.jar:

- Change directories to `WL_HOME/server/lib`
- Run `java -jar wljarbuilder.jar`

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`

13.3.1.2 WebSphere

Both WebSphere and MQ jars are required for running the agent and integration servers. You can obtain these jars from the WebSphere or MQ server.

13.3.1.2.1 WebSphere 6.1.

- `com.ibm.ws.runtime_6.1.0.jar`
- `com.ibm.ws.emf_2.1.0.jar`
- `org.eclipse.emf.ecore_2.2.1.v200609210005.jar`
- `org.eclipse.emf.common_2.2.1.v200609210005.jar`
- `com.ibm.ws.webservices.thinclient_6.1.0.jar`
- `ibmorb.jar`
- `com.ibm.ws.wccm_6.1.0.jar`
- `com.ibm.ws.sib.server_2.0.0.jar`
- `com.ibm.ws.sib.utils_2.0.0.jar`
- `iwsorbutil.jar`

Note: If you are using WebSphere 6.1, set the following property to true in the WebSphere console under `Application Servers > <your server> > web container > customer properties`:

`com.ibm.ws.webcontainer.invokefilterscompatibility=true`

This is required for Business Center and the System Management Administrator.

13.3.1.2.2 WebSphere 7.0

- `<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 7.0.0.5, set the following property to true in the WebSphere console under `Application Servers > <your server> > web container > customer properties`:
>
> `com.ibm.ws.webcontainer.dispatcherrethrowser=true`
>
> Specify this property in all lowercase.

13.3.1.2.3 WebSphere MQ (6.0.2.5) Using "fscontext" jndi

For using WebSphere MQ, you can obtain these jars from MQ_HOME/java/lib.

- `com.ibm.mq.jar`
- `com.ibm.mqjms.jar`
- `fscontext.jar`
- `dhbcore.jar`
- `jms.jar`
- `jta.jar`
- `providerutil.jar`

13.3.1.2.4 WebSphere MQ (7.0.1.2) Using "fscontext" jndi

For using WebSphere MQ, you can obtain these jars from MQ_HOME/java/lib.

- `com.ibm.mq.jar`
- `com.ibm.mqjms.jar`
- `fscontext.jar`
- `dhbcore.jar`
- `jms.jar`
- `jta.jar`
- `providerutil.jar`

- com.ibm.mq.jmqi.jar

13.3.1.3 JBoss

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 *Selling and Fulfillment Foundation: Performance Management Guide*.

13.3.1.4 TIBCO

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.

14

Deploying Selling and Fulfillment Foundation

After configuring Selling and Fulfillment Foundation according to your business needs, deploy it into production based on your application server. This chapter describes how to deploy Selling and Fulfillment Foundation on Oracle WebLogic, IBM WebSphere, and JBoss. Deployment is part of the general path that you follow when installing and deploying Selling and Fulfillment Foundation:

1. Installing the application server (JBoss, WebLogic, or WebSphere). Refer to the documentation for the application server.
2. Installing 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 Selling and Fulfillment Foundation.

This chapter also provides the information required to complete Step , Step , and Step indicated in the Installation Checklist.

If you need to deploy Selling and Fulfillment Foundation in a development environment using exploded (non-ear) mode, see the *Selling and Fulfillment Foundation: Customizing Console JSP Interface for End User Guide*.

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 *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 *Selling and Fulfillment Foundation: Performance Management Guide*.

Note: If you are planning on installing any of the 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.

14.1 Setting Up the WebLogic Application Server

The following sections include information for setting up the WebLogic application server, including:

- *"Setting Up the WebLogic Script File"* on page 14-214
- *"Configuring WebLogic for Selling and Fulfillment Foundation"* on page 14-216
- *"Disabling Instrumented Stack Traces in WebLogic"* on page 14-217
- "Setting Up WebLogic to Display Barcodes and Graphs" on page 14-217
- "Setting Up WebLogic to Use HTTP In-Memory Session Replication" on page 14-218

14.1.1 Setting Up the WebLogic Script File

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:

1. Add the following properties to the startWebLogic.sh (or .cmd) file supplied by Oracle. Each property and its proper syntax are described in the following table.

Property	Description
JAVA_OPTIONS	Java command line options for running the server. Depending on your JVM vendor, specify as follows: • For IBM, set this value to `-Xms512m -Xmx512m` • For JRockit, set this value to `-Xms512m -Xmx512m` • For HP, set this value to `-XX:MaxPermSize=256m -Xms512m -Xmx512m` • For Sun, set this value to `-XX:MaxPermSize=256m -Xms512m -Xmx512m`
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: CLASSPATH="${CLASSPATH}${CLASPATHSEP}${MEDREC_WEBLOGIC_CLASSPATH}" Change this so that the path to the drivers .jars is first. For example: DBDRIVERS=/<directory_path_to_oracle_drivers>/ojdbc16.jar CLASSPATH="${DBDRIVERS}${CLASSPATHSEP}${CLASSPATH}${CLASSPATHSEP}${MEDREC_WEBLOGIC_CLASSPATH}"
JITC_COMPILEOPTS	For AIX, specify as "NQCLSINIT"
-Dfile.encoding	To ensure that all the Selling and Fulfillment Foundation UI screens display UTF-8 characters, specify as follows for java commands: `-Dfile.encoding=UTF-8` This is applicable to all the 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. `-Dvendor=shell`
-DvendorFile	System property. Specify as an argument to the java command. `-DvendorFile=/servers.properties`

Property	Description
-Dsci.naming.provider.url	Sets the local jndi that the server will register with. Specify as an argument to the java command using the name or ip of the server and the port it is listening on. -Dsci.naming.provider.url=t3://<hostname>:<port>

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
   - jcert.jar
   - jsse.jar

14.1.2 Configuring WebLogic for Selling and Fulfillment Foundation

You must configure WebLogic to run properly with Selling and Fulfillment Foundation.

To configure WebLogic:

1. From the WebLogic Console menu, choose **Services** > **XML Registries**.
2. Click **New**.

> **Note:** You do *not* need to set an XML registry parameter for UTF-8. This is predefined.

3. Click **Next**. Select the WebLogic application server or cluster to which you would like to deploy this XML Registry.
4. Click **Finish**.

14.1.3 Disabling Instrumented Stack Traces in WebLogic

You can eliminate additional stack traces resulting from an error on an API call in EJB mode.

To eliminate stack traces:

1. From the WebLogic System Administration Console, select each server on which Selling and Fulfillment Foundation is deployed.
2. Select Logging.
3. Uncheck the checkbox for Instrument Stack Traces and choose Apply.

14.1.4 Setting Up WebLogic to Display Barcodes and Graphs

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:

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 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.

14.1.5 Setting Up WebLogic to Use HTTP In-Memory Session Replication

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:

1. Copy the `<INSTALL_DIR>/repository/eardata/smcfs/descriptors/weblogic/WAR/WEB-INF/weblogic.xml` file to the `<INSTALL_DIR>/extensions/<package-name>` directory, where `<package-name>` is the application name, for example, `smcfs`.
2. 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>
```

3. Rebuild the EAR file.

14.2 Building the Enterprise Archive (EAR) Package (WebLogic)

> **Note:** Selling and Fulfillment Foundation supports overriding the context root during EAR deployment.

When deploying Selling and Fulfillment Foundation on WebLogic, use the `smcfs.ear` file, which contains:

- `smcfs.war` - Web module that contains all of the 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 Business Center Web application components.
- `yantrawebservices.war` - Web module that contains all of the Selling and Fulfillment Foundation Web services interface classes.
- `smcfsejb.jar` - The EJB module that contains all the 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 webservices backend jar file. For example, if you specify an EAR file as `xyz.ear`, the webservice 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.

14.2.1 Deploying the Context-Sensitive Help (WebLogic)

The 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
-Dappserver=weblogic -Dwarfiles=smcfs,sma,sbc
-Dearfile=smcfs.ear
```

To make use of the context-sensitive help files associated with 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 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 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=/`*`sterling`*`docs/yfscommon/online_help`. If your installation locale code is not en_US, set `yfs.onlinehelp.path.overrideforlocale.<your_local_code>=/`*`sterling`*`docs/yfscommon/online_help/<your_local_code>`.

For information about the full Product Documentation Library, see Section 1.3, "Documentation Library".

14.2.2 Preparing to Build Web Services (WebLogic)

If you are planning to run the Selling and Fulfillment Foundation components as Web services, additional setup is required. The setup takes place in the `namedwebservices.xml` file.

You can expose either all or selected APIs as Web services. Consider the following:

- To expose selected APIs as Web services, set the ExposeAllAPIs attribute value to N and specify each API you want to expose in an Api/Name attribute.

> **Note:** If API security is enabled, ensure that you expose the Login API. For more information about enabling APIs for Web services, refer to the *Platform Configuration Guide*.

- To expose all the Selling and Fulfillment Foundation APIs as Web services, set the ExposeAllAPIs attribute value to Y. If the ExposeAllAPIs attribute is set to Y, all <Api> node attributes are ignored.

If you are exposing individual services, edit the attributes of the `namedwebservices.xml` file, as described in the following table, before you create your `smcfs.ear` file.

Property	Description
ServiceName	The name of the service that you configured using the Selling and Fulfillment Foundation 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 webservice programmatically. When specifying a service name for ExposedName, choose a literal that does not match any of the standard Selling and Fulfillment Foundation API names.

To configure Selling and Fulfillment Foundation as Web services:

1. Edit the `<INSTALL_DIR>/properties/sandbox.cfg` file to set the value of the `BEA_DIR` property to specify the `<WL_HOME>` directory.
2. Run the `<INSTALL_DIR>/bin/setupfiles.sh` (or `setupfiles.cmd`) script.
3. Rename the `<INSTALL_DIR>/repository/eardata/platform/webservices/namedwebservices.xml.sample` file to `namedwebservices.xml`.
4. Edit `namedwebservices.xml` to remove any Selling and Fulfillment Foundation APIs and services that you do not want exposed as named Web services. Include the services you want to expose as named Web services.
5. Create the EAR as described in Section 14.2.4, "Creating the EAR (WebLogic)".
6. Later, if you want to add more APIs and services as webservices, repeat these steps again.

14.2.3 Including Custom Classes (WebLogic)

When deploying 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:

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 *Selling and Fulfillment Foundation: Customization Basics Guide*.
3. Rebuild the EAR as described in Section 14.2.4, "Creating the EAR (WebLogic)".

The custom classes are automatically included in the `smcfs.ear` file.

14.2.4 Creating the EAR (WebLogic)

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 Selling and Fulfillment Foundation application EAR file • `smcfsdocs.ear` - the Selling and Fulfillment Foundation documentation EAR file
create-doc-ear	Creates `smcfsdocs.ear -` the Selling and Fulfillment Foundation documentation EAR

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,sbc -Dearfile=smcfs.ear create-ear
```

> **Note:** Selling and Fulfillment Foundation supports the RPC encoded or document literal style and usage of invocation for WebServices. 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 Selling and Fulfillment Foundation application war file
- `sma.war` - the System Management Administrator application war file
- `sbc.war` - the Business Center application war file

> **Note:** You can add the following options to the end of the above command:
>
> - `-Dnowebservice=true` if you do not want to use Web services. If you do want to use Web services, see the Section 14.2.2, "Preparing to Build Web Services (WebLogic)".
> - `-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.

For more information about the System Management Administrator (SMA) see the *Selling and Fulfillment Foundation: System Manangement and Administration Guide*. For more information about 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).

14.2.4.1 Installing and Deploying Selling and Fulfillment Foundation on Different Servers

This section applies only to users who are installing and deploying 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 Selling and Fulfillment Foundation.

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.

For more information about sandbox.cfg and changing properties, see the *Selling and Fulfillment Foundation: Properties Guide*.

14.2.5 Precompiling the WAR File (WebLogic)

To improve the performance when initially loading UI resources, Sterling Commerce 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 *Selling and Fulfillment Foundation: Performance Management Guide*.

Deploy the new ear file as described in Section 14.3, "Deploying the Enterprise Archive (EAR) (WebLogic)".

14.3 Deploying the Enterprise Archive (EAR) (WebLogic)

Note: 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 Sterlng 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 *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 *Selling and Fulfillment Foundation: Performance Management Guide*.

For more information about how to implement multiple EAR files on the same application server, see the *Selling and Fulfillment Foundation: Customization Basics Guide*.

For instructions on deploying the EAR to your WebLogic application server, see your WebLogic documentation.

To verify the 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. If the Selling and Fulfillment Foundation Administrator's home page is not displayed, contact the Selling and Fulfillment Foundation Technical Support Services at:
   `http://www.sterlingcommerce.com/scm_support/`

4. To access Business Center:

   a. Access the Business Center login page by setting the enterprise appropriately. For more information about setting the enterprise while logging in to Business Center, see Section 14.14, "Setting an Enterprise for Logging In to Business Center".

   b. When prompted, enter your Login ID and Password. If the Business Center home page is not displayed, contact the Business Center Technical Support Services at:
   `http://www.sterlingcommerce.com/scm_support/`

14.4 Setting Up the WebSphere Application Server

Before configuring WebSphere, Sterling Commerce recommends that you start the WebSphere administrative server with the following memory parameters:

- -Xms512 MB or higher
- -Xmx512 MB or higher

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:

1. From the WebSphere Administrative Console, expand **Troubleshooting** in the left panel and click on **Logs** and **Trace**.
2. Select each server that hosts 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.

Also ensure that the WebSphere Classloader is set correctly for Classloader policy and Class loading modes as follows:

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.

The following sections include additional information for setting up the WebSphere application server, including:

- *"Application Clients Invoking the Selling and Fulfillment Foundation EJBs"* on page 14-227
- *"Configuring WebSphere JVM Settings"* on page 14-228
- *"Configuring WebSphere for Running the Selling and Fulfillment Foundation Agents"* on page 14-229
- "Setting Up WebSphere to Display Barcodes and Graphs" on page 14-229
- "Configuring DataSource Connection Pooling on WebLogic, WebSphere, and JBoss" on page 14-249

14.4.1 Application Clients Invoking the Selling and Fulfillment Foundation EJBs

In order to make EJB calls in 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:

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.

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
```

14.4.2 Configuring WebSphere JVM Settings

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:

1. From the WebSphere Administrative Console, select the application server specified for Selling and Fulfillment Foundation.
2. For IBM servers with IBM JDK 6.0:

   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.
-Dsci.naming.provider.url	-Dsci.naming.provider.url=corbaloc::*host*:*port*	Sets the local jndi that the server will register with. Specify as an argument to the java command using the name or IP of the server and the port it is listening on.

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.

14.4.3 Configuring WebSphere for Running the Selling and Fulfillment Foundation Agents

Selling and Fulfillment Foundation requires both WebSphere and MQ jars for running the agent and integration servers.

14.4.4 Setting Up WebSphere to Display Barcodes and Graphs

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:

1. From the WebSphere Administrative Console, go to Servers > Application Server and select the application server specified for 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 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.

14.5 Building the Enterprise Archive (EAR) Package (WebSphere)

When deploying Selling and Fulfillment Foundation on WebSphere, use the `smcfs.ear` file, which contains:

- `smcfs.war` - Web module that contains all of the 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 Business Center Web application components.
- yantrawebservices.war - Web module that contains all of the Selling and Fulfillment Foundation Web services interface classes.
- smcfsejb.jar - The EJB module that contains all the 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 webservices backend jar file. For example, if you specify an EAR file as xyz.ear, the webservice 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.

14.5.1 Deploying the Context-Sensitive Help (WebSphere)

The 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  -Dappserver=websphere
-Dwarfiles=smcfs,sma,sbc -Dearfile=smcfs.ear create-doc-ear
```

To make use of the documentation and help files associated with 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 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 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=/`*`sterling`*`docs/yfscommon/online_help`. If your installation locale code is not en_US, set `yfs.onlinehelp.path.overrideforlocale.<your_local_code>=/`*`sterling`*`docs/yfscommon/online_help/<your_local_code>`.

For information about the full Product Documentation Library, see Section 1.3, "Documentation Library".

14.5.2 Preparing to Build Web Services (WebSphere)

If you are planning to run the Selling and Fulfillment Foundation components as Web services, additional setup is required. The setup takes place in the `namedwebservices.xml` file.

You can expose either all or selected APIs as Web services. Consider the following:

- To expose selected APIs as Web services, set the ExposeAllAPIs attribute value to N and specify each API you want to expose in an Api/Name attribute.

Note: If API security is enabled, ensure that you expose the Login API. For more information about enabling APIs for Web services, refer to the *Platform Configuration Guide*.

- To expose all the Selling and Fulfillment Foundation APIs as Web services, you can set the ExposeAllAPIs attribute value to Y. If the ExposeAllAPIs attribute is set to Y, all <Api> node attributes are ignored.

If you are exposing individual services, edit the attributes of the `namedwebservices.xml` file, as described in the following table, before you create your `smcfs.ear` file.

Property	Description
ServiceName	The name of the service that you configured using the Selling and Fulfillment Foundation 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 Selling and Fulfillment Foundation API names.

To configure Selling and Fulfillment Foundation as Web services:

1. Edit the `<INSTALL_DIR>/properties/sandbox.cfg` file to set the value of the `WAS_DIR` property to specify the `<WAS_HOME>` directory.
2. Run the `<INSTALL_DIR>/bin/setupfiles.sh` (or `setupfiles.cmd`) script.
3. Rename the `<INSTALL_DIR>/repository/eardata/platform/webservices/namedwebservices.xml.sample` file to `namedwebservices.xml`.
4. Edit the `namedwebservices.xml` file to update the exposed names to start with a lowercase letter. Remove any Selling and Fulfillment Foundation APIs and services that you do not want exposed as named Web services and include the services you want to expose as named Web services.
5. Create the EAR as described in Section 14.5.4, "Creating the EAR (WebSphere)".
6. Later, if you want to add more APIs and services as webservices, repeat these steps again.

14.5.3 Including Custom Classes (WebSphere)

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:

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 *Selling and Fulfillment Foundation: Customization Basics Guide*.
3. Rebuild the EAR as described in Section 14.5.4, "Creating the EAR (WebSphere)".

The custom classes are automatically included in the `smcfs.ear` file.

14.5.4 Creating the EAR (WebSphere)

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 Selling and Fulfillment Foundation application EAR file • `smcfsdocs.ear` - the Selling and Fulfillment Foundation documentation EAR file
create-doc-ear	Creates `smcfsdocs.ear` - the Selling and Fulfillment Foundation documentation EAR

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: Selling and Fulfillment Foundation supports the RPC encoded or document literal style and usage of invocation for WebServices. 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 Selling and Fulfillment Foundation application war file
- `sma.war` - the System Management Administrator application war file
- `sbc.war` - the Business Center application war file

Note: You can add the following options to the end of the above command:

- `-Dnowebservice=true` if you do not want to use Web services. If you do want to use Web services, see the Section 14.5.2, "Preparing to Build Web Services (WebSphere)".
- `-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.

For more information about the System Management Administrator (SMA) see the *Selling and Fulfillment Foundation: System Manangement and Administration Guide*. For more information about 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

14.5.4.1 Installing and Deploying Selling and Fulfillment Foundation on Different Servers

This section applies only to users who are installing and deploying 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 Selling and Fulfillment Foundation.

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` script.
3. Rebuild the EAR file.
4. Edit sandbox.cfg to set the LOG_DIR value back to its original value.

For more information about sandbox.cfg and changing properties, see the *Selling and Fulfillment Foundation: Properties Guide*.

14.5.5 Precompiling the WAR File (WebSphere)

To improve the performance when initially loading UI resources, Sterling Commerce 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 *Selling and Fulfillment Foundation: Performance Management Guide*.

14.6 Deploying the Enterprise Archive (EAR) (using the WebSphere Admin Console)

Note: 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 Sterlng 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 *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 *Selling and Fulfillment Foundation: Performance Management Guide*.

For more information about how to implement multiple EAR files on the same application server, see the *Selling and Fulfillment Foundation: Customization Basics Guide*.

To deploy the EAR on WebSphere:

1. From the WebSphere Administrative Console menu in the left pane, select Applications > Install New Application.
2. The right pane is populated with the specifics for the EAR location.

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 if desired, change the application name. 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.

To verify the 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. If the Selling and Fulfillment Foundation Administrator’s home page is not displayed, contact the Selling and Fulfillment Foundation Technical

Support Services at:
`http://www.sterlingcommerce.com/scm_support/`

4. To access Business Center :

   a. Access the Business Center login page by setting the enterprise appropriately. For more information about setting the enterprise while logging in to Business Center, see Section 14.14, "Setting an Enterprise for Logging In to Business Center".

   b. When prompted, enter your Login ID and Password. If the Business Center home page is not displayed, contact the Business Center Technical Support Services at:
   `http://www.sterlingcommerce.com/scm_support/`

14.7 Setting Up the JBoss Application Server

> **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:

1. Add the following properties to the `<JBOSS_HOME>/bin/run.sh` (or `.cmd`) file supplied by JBoss. Each property and its proper syntax are described in the following table:

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=256m -Xms512m -Xmx512m` For information about supported JDK tiers, see Table 2–4, "Supported Application Server Tier". For information about memory requirements for specific operating systems, see Table 2–15, "Minimum Memory Requirements".

Property	Required Edits
-Dfile.encoding	To ensure that all the Selling and Fulfillment Foundation screens display UTF-8 characters for java commands, specify: `-Dfile.encoding=UTF-8` This is applicable to all the 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`
-Dsci.naming.provider.url	Sets the local jndi that the server will register with. Specify as an argument to the java command using the name or IP of the server and the port it is listening on. `-Dsci.naming.provider.url=jnp://host:portnum`

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`
- `jcert.jar`
- `jsse.jar`

In addition to this setup, see "Setting Up JBoss to Display Barcodes and Graphs" on page 14-240 for information about setting up the JBoss application server to display barcodes and graphs.

14.7.1 Setting Up JBoss to Display Barcodes and Graphs

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:

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 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.

14.8 Building the Enterprise Archive (EAR) Package (JBoss)

When deploying Selling and Fulfillment Foundation on JBoss, use the `smcfs.ear` file, which contains:

- `smcfs.war` - Web module that contains all of the 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 Business Center Web application components.
- `smcfsejb.jar` - The EJB module that contains all the 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 webservices backend jar file. For example, if you specify an EAR file as `xyz.ear`, the webservice 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.

14.8.1 Deploying the Context-Sensitive Help (JBoss)

The 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 -Dappserver=jboss
-Dwarfiles=smcfs,sma,sbc -Dearfile=smcfs.ear
```

To make use of the documentation and context-sensitive help files associated with 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 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 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=/*sterling*docs/yfscommon/online_help. If your installation locale code is not en_US, set yfs.onlinehelp.path.overrideforlocale.<your_local_code>=/*sterling*docs/yfscommon/online_help/<your_local_code>.

For information about the full Product Documentation Library, see Section 1.3, "Documentation Library".

14.8.2 Preparing to Build Web Services (JBoss)

If you are planning to run the Selling and Fulfillment Foundation components as Web services, additional setup is required. The setup takes place in the `namedwebservices.xml` file.

You can expose either all or selected APIs as Web services. Consider the following:

- To expose selected APIs as Web services, set the ExposeAllAPIs attribute value to N and specify each API you want to expose in an Api/Name attribute.

  > **Note:** If API security is enabled, ensure that you expose the Login API. For more information about enabling APIs for Web services, refer to the *Platform Configuration Guide*.

- To expose all the Selling and Fulfillment Foundation APIs as Web services, you can set the ExposeAllAPIs attribute value to Y. If the ExposeAllAPIs attribute is set to Y, all <Api> node attributes are ignored.

If you are exposing individual services, edit the attributes of the `namedwebservices.xml` file, as described in the following table, before you create your `smcfs.ear` file.

Property	Description
ServiceName	The name of the service that you configured using the Selling and Fulfillment Foundation 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 Selling and Fulfillment Foundation API names.

1. If you want to configure Selling and Fulfillment Foundation as Web services, add the following three variables to the `<INSTALL_DIR>/properties/sandbox.cfg` file:

   - JBOSS_DIR - set to the absolute path of the JBoss installation directory.

- JBOSS_PRECOMPILE_JSP - 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.
- EJB_3_ENABLED - set to true or false. The flag determines if the ejbs are generated according to the spec version 2 or 3. Jboss supports both.

Edit the `<INSTALL_DIR>/properties/sandbox.cfg` file to set the value of the `JBOSS_DIR` property to specify the `<JBOSS_HOME>` directory.

See the *Selling and Fulfillment Foundation: Properties Guide* for more information about editing the sandbox.cfg file.

2. Run the `<INSTALL_DIR>/bin/setupfiles.sh` (or `setupfiles.cmd`) script.
3. Rename the `<INSTALL_DIR>/repository/eardata/platform/webservices/namedwebservices.xml.sample` file to `namedwebservices.xml`.
4. Edit `namedwebservices.xml` to remove any Selling and Fulfillment Foundation APIs and services that you do not want exposed as named Web services. Include the services you want to expose as named Web services.
5. Create the EAR as described in Section 14.8.4, "Creating the EAR (JBoss)".
6. Later, if you want to add more APIs and services as webservices, repeat these steps again.

14.8.3 Including Custom Classes (JBoss)

When deploying 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:

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 *Selling and Fulfillment Foundation: Customization Basics Guide*.
3. Rebuild the EAR as described in Section 14.8.4, "Creating the EAR (JBoss)".

These classes are automatically included in the `smcfs.ear` file.

14.8.4 Creating the EAR (JBoss)

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 Selling and Fulfillment Foundation application EAR file • `smcfsdocs.ear` - the Selling and Fulfillment Foundation documentation EAR file
create-doc-ear	Creates `smcfsdocs.ear` - the Selling and Fulfillment Foundation documentation EAR

Note: To successfully build the application EAR file in 64-bit JBoss with the Business Center, System Management Administrator, Sterling Call Center and Sterling Store, and Selling and Fulfillment Foundation, use the following memory parameters:

`XX:MaxPermSize=768m -Xmx2048m -Xms2048m`

To create an application EAR file, run the following command from the `<INSTALL_DIR>/bin` directory:

```
.\buildear.sh (.cmd for Windows) -Dappserver=jboss
-Dwarfiles=smcfs,sma,sbc -Dearfile=smcfs.ear create-ear
```

> **Note:** Selling and Fulfillment Foundation supports the RPC encoded or document literal style and usage of invocation for WebServices. When choosing style and usage for JBoss, pass the following in the ear command line:
>
> `-D jboss-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 Selling and Fulfillment Foundation application war file
- `sma.war` - the System Management Administrator application war file
- `sbc.war` - the Business Center application war file

> **Note:** You can add the following options to the end of the above command:
>
> - `-Dnowebservice=true` if you do not want to use Web services. If you do want to use Web services, see the Section 14.8.2, "Preparing to Build Web Services (JBoss)".
> - `-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.

For more information about the System Management Administrator (SMA) see the *Selling and Fulfillment Foundation: System Manangement and Administration Guide*. For more information about 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

14.8.4.1 Installing and Deploying Selling and Fulfillment Foundation on Different Servers

This section applies only to users who are installing and deploying 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 Selling and Fulfillment Foundation.

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 script.
3. Rebuild the EAR file.
4. Edit sandbox.cfg to set the LOG_DIR value back to its original value.

For more information about sandbox.cfg and changing properties, see the *Selling and Fulfillment Foundation: Properties Guide*.

14.8.5 Precompiling the WAR File (JBoss)

See the *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.

14.9 Deploying the Enterprise Archive (EAR) (JBoss)

Note: 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 Sterlng 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 *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 *Selling and Fulfillment Foundation: Performance Management Guide*.

For more information about how to implement multiple EAR files on the same application server, see the *Selling and Fulfillment Foundation: Customization Basics Guide*.

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 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. If the Selling and Fulfillment Foundation Administrator's home page is not displayed, contact the Selling and Fulfillment Foundation Technical Support Services at: `http://www.sterlingcommerce.com/scm_support/`
4. To access Business Center:
   a. Access the Business Center login page by setting the enterprise appropriately. For more information about setting the enterprise while logging in to Business Center, see Section 14.14, "Setting an Enterprise for Logging In to Business Center".
   b. When prompted, enter your Login ID and Password. If the Business Center home page is not displayed, contact the Business Center Technical Support Services at: `http://www.sterlingcommerce.com/scm_support/`

14.10 Configuring DataSource Connection Pooling on WebLogic, WebSphere, and JBoss

How you perform this configuration depends on whether you chose to use the multischema feature during installation.

If using the multischema feature, you can use the manageDBPool API to set the datasource parameter for each pool. If not set, the datasource name defaults to pool id. Multischema database parameters are explained in Section 7.2.7.1.2, "Sample Multischema.xml File".

If you are using the datasource connection pooling and are not using the multischema feature, perform the following steps:

1. In the `customer_overrides.properties` file, add the following lines:

- For DB2, add `jdbcService.db2Pool.datasource=<datasourceName>`
- For Oracle, add `jdbcService.oraclePool.datasource=<datasourceName>`
- For Microsoft SQL Server, add `jdbcService.mssqlPool.datasource=<datasourceName>`

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

2. Configure the DataSource on the application server. The JNDI Name must be datasourcename. Refer to the documentation provided by your application server vendor for directions on completing this step.

   For JBoss, the format of the datasource name is:
   `jdbcService.oraclePool.datasource=java:/datasource`

3. Set the -Dvendor property in your startup script as follows:
   `-Dvendor=<application_server_type>`

   Example:
   `-Dvendor=weblogic`

4. Test the connection.

5. For WebSphere only: to avoid the warning messages regarding direct datasource lookups that occur at runtime:

   - From the WebSphere Administrative Console, expand Troubleshooting in the left panel and click on **Logs** and **Trace**.
   - Select each server that hosts the Platform Demo application and choose **Change Log Detail Levels** in the General Properties.
   - In the Components panel, select the class **com.ibm.ejs.j2c.ConnectionFactoryBuilderImpl**, and specify the log level as **severe**.

6. Save the changes to the master configuration.

Depending on your application server, you can find more information about -Dvendor and other properties in the application server startup scripts in Section 14.1, "Setting Up the WebLogic Application Server",

Section 14.4, "Setting Up the WebSphere Application Server", or Section 14.7, "Setting Up the JBoss Application Server".

14.11 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. To configure a client machine, select Control Panel > Regional and Language Options.

You can use apostrophes and international characters in database queries, the Condition Builder, the Configuration Deployment Tool, and other user interface fields.

14.12 Clearing Browser and Java Plugin Caches

Once Selling and Fulfillment Foundation is ready for deployment, each user must clear the browser and Java Plugin caches on their client machines before launching Selling and Fulfillment Foundation.

To clear the browser cache:

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.

To clear the Java plugin cache:

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.

14.13 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 Sterling Commerce 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).

14.14 Setting an Enterprise for Logging In to Business Center

Business Center supports enterprise-specific login. Therefore, when you log in to 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 Business Center, you should use the following URL:

```
http://<server>:<port>/sbc/sbc/launch.jsp?EnterpriseCode=<Ente
rprise_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 Business Center:

```
http://<server>:<port>/sbc/sbc/launch.jsp?EnterpriseCode=XYZ-1
23
```

15

Deploying and Updating the Rich Client Platform Applications

This chapter explains how to deploy and update the Rich Client Platform applications such as Sterling COM PCA (Packaged Composite Application) in different geographical locations.

This chapter also provides the information required to complete Step indicated on the Installation Checklist on page 1-1.

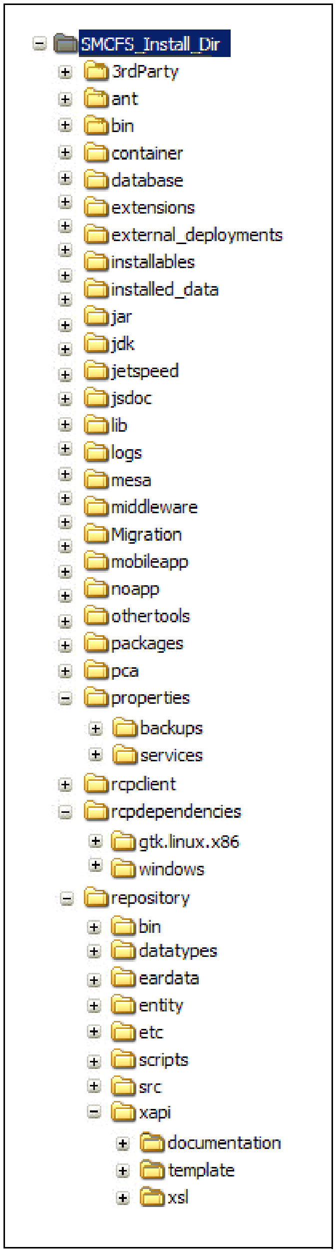
15.1 Before You Begin

Before you start deploying a Rich Client Platform application you must have installed Selling and Fulfillment Foundation. For more information about installing Selling and Fulfillment Foundation, see Chapter 7, "Installing Selling and Fulfillment Foundation in UNIX or Linux Environments" or Chapter 8, "Installing Selling and Fulfillment Foundation in a Windows Environment".

The components that are shipped as part of Selling and Fulfillment Foundation or Application Platform (that is, what is available when Selling and Fulfillment Foundation is installed) are:

- RCP Infrastructure plug-in's zip file
- RCP Foundation plug-in's zip file
- JREs for each of the supported operating systems. For more information about supported JREs and operating systems, see Chapter 2, "System Requirements".
- Eclipse dependencies for each of the supported operating systems

After you install Selling and Fulfillment Foundation, you can view the directory structure as shown:


SMCFS_Install_Dir
3rdParty
ant
bin
container
database
extensions
external_deployments
installables
installed_data
jar
jdk
jetspeed
jsdoc
lib
logs
mesa
middleware
Migration
mobileapp
noapp
othertools
packages
pca
properties
backups
services
rcpclient
rcpdependencies
gtk.linux.x86
windows
repository
bin
datatypes
eardata
entity
etc
scripts
src
xapi
documentation
template
xsl

The directory structure contains:

- The <INSTALL_DIR> folder—This contains the Rich Client Platform (Rich Client Platform) files, plug-ins, or JREs.
- The <INSTALL_DIR>/rcpclient folder—This contains the Rich Client Platform plug-in and tools plug-in zip files.
- The <INSTALL_DIR>/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 contains the customer_overrides.properties properties file. This file is used when enabling auto updates for the individual PCA. For more information about enabling auto updates, see Section 15.2.5, "Applying Updates".

What is available when a Rich Client Platform-based PCA is installed?

When you install a Rich Client Platform-based PCA client, a zip file that contains the Rich Client Platform application plug-ins or features is provided. For example, when you install the Sterling COM PCA application, the <INSTALL_DIR>/rcp/COM/rcpclient directory is automatically created. The com.zip file is stored in this directory, which contains the Sterling COM PCA-specific plug-ins or features.

15.2 Deploying the Rich Client Platform Application

Deploying the Rich Client Platform application involves:

- Creating the RCP_EXTN_FOLDER Folder
- Configuring Locations

- Localizing Bundle and Theme Files
- Enabling HTTPS
- Applying Updates
- Running the Ant Script
- Deploying RCP Clients through a Remote Terminal

15.2.1 Creating the RCP_EXTN_FOLDER Folder

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.

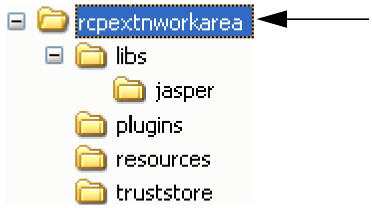
The <RCP_EXTN_FOLDER> folder structure is better explained with an example as follows:

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 as illustrated in Figure 15–1, "Sample <RCP_EXTN_FOLDER> Folder Structure":
   - `libs`
   - `plugins`
   - `resources`
   - `truststore`

Figure 15–1 Sample <RCP_EXTN_FOLDER> Folder Structure



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, `rcpsecureapis.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-1.2.0.jar

To download these jasper libs, see the `<INSTALL_DIR>/xapidocs/code_examples/jasperreports/readme.html` file.

15.2.1.1 Caching Data Types Locally in the Rich Client Platform Based-PCA Client

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:

1. Copy the `datatypes.xml` file from the `<INSTALL_DIR>/repository/datatypes` folder, and `yfsdatatypemap.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 `yfsdatatypemap.xml` from the extensions directory to the `<RCP_EXTN_FOLDER>/resources/extn` directory.

15.2.1.2 Supported Browser Version on Red Hat Linux Workstation 4

The default Firefox browser that is installed with the Red Hat Linux Workstation 4 is the certified version.

> **Note:** You need to set the environment variable, MOZILLA_FIVE_HOME to the folder containing your Firefox installation. For example, set the env MOZILLA_FIVE_HOME as `/usr/lib/firefox-3.x`

For more information about the supported browser version, see the following links:

- http://www.eclipse.org/swt/faq.php#browserlinux
- http://www.eclipse.org/swt/faq.php#browserlinuxrcp

15.2.1.3 Supported Browser Version on Suse Linux Desktop 10

Rich Client Platform supports Mozilla browser on the Suse Linux Desktop 10. The default Firefox browser that is installed with the Suse Linux Desktop 10 is not supported.

> **Note:** You need to set the environment variable, MOZILLA_FIVE_HOME to the folder containing your Mozilla installation. For example, set the env MOZILLA_FIVE_HOME as `/usr/lib/mozilla-1.7.12`.

For more information about the supported browser version, see the following links:

- http://www.eclipse.org/swt/faq.php#browserlinux
- http://www.eclipse.org/swt/faq.php#browserlinuxrcp

15.2.1.4 Installing Browser Plugins on Linux

In the Rich Client Platform-based applications, FusionCharts, and JasperReports can be viewed in the client application. This requires the following plugins to be installed on the Firefox browser:

- Flash Player Plugin to view FusionCharts
- Mozplugger to view JasperReports

15.2.1.5 Installing the Flash Player Plugin

To install the flash player plugin, open the following link in your Linux Firefox browser and follow the instructions provided:

http://www.adobe.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash

Note: During installation, the browser installation directory should point to the Firefox install directory. For example, `/usr/lib/firefox-3.x`

15.2.1.6 Installing Mozplugger

To install the Mozplugger:

1. Navigate to: http://mozplugger.mozdev.org/
2. Download Version 1.7.3 SOURCE of the MozPlugger to a local directory. The mozplugger-1.7.3.tar.gz file is downloaded.
3. Untar the mozplugger-1.7.3.tar.gz file. The mozplugger-1.7.3 directory is created.
4. Change the directory to mozplugger-1.7.3 and run the following commands:
    - make linux
    - make install

Note: During installation, the browser installation directory should point to the Firefox install directory. For example, `/usr/lib/firefox-3.x`

15.2.2 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. By default, the `locations.ycfg.sample` file is shipped by the Rich Client Platform. You can locate this file in the Rich Client Platform plug-in directory. The file in which the `locations.ycfg.sample` file is stored is shown below:

`<INSTALL_DIR>/rcpclient/com.yantra.yfc.rcp_<version>.zip`

You must extract the contents of this zip file.

To configure locations, you can either create a new `locations.ycfg` XML file or modify the existing `locations.ycfg.sample` XML file.

15.2.2.1 Creating and Configuring a New locations.ycfg XML File

To configure a new `locations.ycfg` file:

1. Create the `locations.ycfg` XML file 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 `<RCP_EXTN_FOLDER>` folder, see Section 15.2.1, "Creating the RCP_EXTN_FOLDER Folder".

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 ="/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".
>
> 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 Section 15.3, "Location Configuration Settings".

15.2.2.2 Modifying the locations.ycfg.sample XML File

To modify the `locations.ycfg.sample` file:

1. Extract and copy the `locations.ycfg.sample` from <INSTALL_DIR>/rcpclient/com.yantra.yfc.rcp_<version>.zip file 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 <RCP_EXTN_FOLDER> folder, see Section 15.2.1, "Creating the RCP_EXTN_FOLDER Folder".

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 Section 15.3, "Location Configuration Settings".

15.2.3 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>.ythm theme file. For more information about localizing bundle and theme files, see the *Selling and Fulfillment Foundation: Localization Guide*.

15.2.4 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 Section 15.2.1, "Creating the RCP_EXTN_FOLDER Folder".

For more information about configuring connection settings for HTTPS connection, see Section 15.4.2, "Configuring Connection Settings for HTTPS Connection".

15.2.5 Applying Updates

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:

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.rcp.pca.updates.dir` is the property, `<INSTALL_DIR>` is the directory where you have installed Selling and Fulfillment Foundation, and `<PCA_UPDATES_DIR>` is the directory which contains individual updates folder for each 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 Sterling COM PCA, 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 Sterling COM PCA 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

– Sterling COM PCA 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 *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 Section 15.2.5.1, "Type of Updates".

15.2.5.1 Type 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 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.

15.2.6 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 COM PCA, 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 COM PCA 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
```

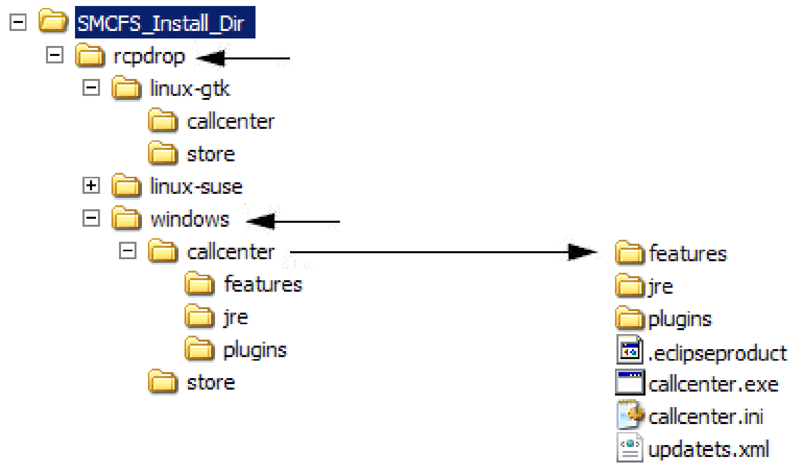
For UNIX:

```
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 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 Section 15.2.1, "Creating the RCP_EXTN_FOLDER Folder" on page 15-257.

After you run this ant script runtime the following resources or directory structure is created or generated for the call center application:

Figure 15–2 Sample Directory Structure for Call Center Application



- The `rcpdrop` folder is created within the `<INSTALL_DIR>` directory.

  where `<INSTALL_DIR>` refers to the directory where you have installed 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 Sterling COM PCA 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 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 Section 15.2.1, "Creating the RCP_EXTN_FOLDER Folder".

- 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 Selling and Fulfillment Foundation for building a PCA Application.

15.2.7 Deploying RCP 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.

15.3 Location Configuration Settings

Location configurations are defined in the `locations.ycfg` file. You can set different preferences for each location.

To define a new location configuration:

1. Set the attributes of the Location element. For Location element attributes and their descriptions, see Section 15–1, "Location Element Attribute List".

Table 15–1 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 Section 15.2.5, "Applying Updates" on page 15-265.

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 Section 15.4, "Configuring Connection Settings".

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.

1. Select a location from the drop-down list.
2. Configure the proxy server settings, if applicable.

Based on the location preferences, you are logged into the application.

15.4 Configuring 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 15–2. You can configure the connection settings for fetching images from the server or connecting to HTTPS.

Table 15–2 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 Section 15.4.2, "Configuring Connection Settings for HTTPS Connection".
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, `/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 Section 15.4.1, "Configuring Connection Settings for Fetching Images from the Server".
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 Section 15.6, "Compression in the Rich Client Platform".

A sample configuration data used to configure a server is as follows:

```
<Config Name = "DEFAULT"
             Protocol = "http"
             BaseUrl = "localhost"
             PortNumber = "7001"
             ApiUrl = "/smcfs/RcpServlet"
             CompressionEnabled = "Y">
</Config>
<Config Name = "LOCAL"
        Protocol = "http"
        BaseUrl = "localhost"
        PortNumber = "7001"
           HttpsPortNumber = "7002
        ApiUrl = "/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.

15.4.1 Configuring Connection Settings for Fetching Images from the Server

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 15–3. You can create more than one configurations to display different types of images.

Table 15–3 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:

1. Set the source binding for the label as:

   ```
   lblOrderNo.setSourceBinding("ServerImageList:Images/Icons/RCP/Image1/@OrderN
   o");
   ```

where `lblOrderNo` 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.

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.

15.4.2 Configuring Connection Settings for HTTPS Connection

To configure the connection settings to communicate with application servers:

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 Section 15.4, "Configuring Connection Settings".

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 Section 15.2.1, "Creating the RCP_EXTN_FOLDER Folder".

A trusted Certificate Authority (CA) like VeriSign issues these security certificates. For more information about SSL or security certificates, see Section 15.5, "Security Certificates".

15.4.3 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 15–3.

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&amp;topic=$
                param3"
</Config>
```

Here $param1 refers to locale, $param2 refers to module id, and $param3 refers to help topic id.

15.5 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.

15.6 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.
>
> The Rich Client Platform does not support compression of images or zip 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 Section 15.4, "Configuring Connection Settings".

15.7 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/yrcui.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:

- `yrcui.jar`
- `eclipse Plugins jars`
- `resources.jar`

Index

E

F

G

H

I

J

L

M

N

O

U

V

W

Y