

Sterling Commerce

An IBM Company

Selling and Fulfillment Foundation: Customization Basics Guide

Release 8.5

Last updated in HF16

June 2010



Copyright Notice

Copyright © 1999 - 2010

Sterling Commerce, Inc.

ALL RIGHTS RESERVED

STERLING COMMERCE SOFTWARE

TRADE SECRET NOTICE

THE STERLING COMMERCE SOFTWARE DESCRIBED BY THIS DOCUMENTATION ("STERLING COMMERCE SOFTWARE") IS THE CONFIDENTIAL AND TRADE SECRET PROPERTY OF STERLING COMMERCE, INC., ITS AFFILIATED COMPANIES OR ITS OR THEIR LICENSORS, AND IS PROVIDED UNDER THE TERMS OF A LICENSE AGREEMENT. NO DUPLICATION OR DISCLOSURE WITHOUT PRIOR WRITTEN PERMISSION. RESTRICTED RIGHTS.

This documentation, the Sterling Commerce Software it describes, and the information and know-how they contain constitute the proprietary, confidential and valuable trade secret information of Sterling Commerce, Inc., its affiliated companies or its or their licensors, and may not be used for any unauthorized purpose, or disclosed to others without the prior written permission of the applicable Sterling Commerce entity. This documentation and the Sterling Commerce Software that it describes have been provided pursuant to a license agreement that contains prohibitions against and/or restrictions on their copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright notice. Commerce, Inc. copyright notice.

U.S. GOVERNMENT RESTRICTED RIGHTS. This documentation and the Sterling Commerce Software it describes are "commercial items" as defined in 48 C.F.R. 2.101. As and when provided to any agency or instrumentality of the U.S. Government or to a U.S. Government prime contractor or a subcontractor at any tier ("Government Licensee"), the terms and conditions of the customary Sterling Commerce commercial license agreement are imposed on Government Licensees per 48 C.F.R. 12.212 or § 227.7202 through § 227.7202-4, as applicable, or through 48 C.F.R. § 52.244-6.

This Trade Secret Notice, including the terms of use herein is governed by the laws of the State of Ohio, USA, without regard to its conflict of laws provisions. If you are accessing the Sterling Commerce Software under an executed agreement, then nothing in these terms and conditions supersedes or modifies the executed agreement.

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

Copyright © 1999 - 2010

Third-Party Software

Portions of the Sterling Commerce Software may include products, or may be distributed on the same storage media with products, ("Third Party Software") offered by third parties ("Third Party Licensors"). Sterling Commerce Software may include Third Party Software covered by the following copyrights: Copyright © 2006-2008 Andres Almiray. Copyright © 1999-2005 The Apache Software Foundation. Copyright (c) 2008 Azer Koçulu <http://azer.kodfabrik.com>. Copyright © Einar Lielmanis, einars@gmail.com. Copyright (c) 2006 John Reilly (www.inconspicuous.org) and Copyright (c) 2002 Douglas Crockford (www.crockford.com). Copyright (c) 2009 John Resig, <http://jquery.com/>. Copyright © 2006-2008 Json-lib. Copyright © 2001 LOOX Software, Inc. Copyright © 2003-2008 Luck Consulting Pty. Ltd. Copyright 2002-2004 © MetaStuff, Ltd. Copyright © 2009 Michael Mathews micmath@gmail.com. Copyright © 1999-2005 Northwoods Software Corporation. Copyright (C) Microsoft Corp. 1981-1998. Purple Technology, Inc. Copyright (c) 2004-2008 QOS.ch. Copyright © 2005 Sabre Airline Solutions. Copyright © 2004 SoftComplex, Inc. Copyright © 2000-2007 Sun Microsystems, Inc. Copyright © 2001 VisualSoft Technologies Limited. Copyright © 2001 Zero G Software, Inc. All rights reserved by all listed parties.

The Sterling Commerce Software is distributed on the same storage media as certain Third Party Software covered by the following copyrights: Copyright © 1999-2006 The Apache Software Foundation. Copyright (c) 2001-2003 Ant-Contrib project. Copyright © 1998-2007 Bela Ban. Copyright © 2005 Eclipse Foundation. Copyright © 2002-2006 Julian Hyde and others. Copyright © 1997 ICE Engineering, Inc./Timothy Gerard Endres. Copyright 2000, 2006 IBM Corporation and others. Copyright © 1987-2006 ILOG, Inc. Copyright © 2000-2006 Infragistics. Copyright © 2002-2005 JBoss, Inc. Copyright LuMriX.net GmbH, Switzerland. Copyright © 1998-2009 Mozilla.org. Copyright © 2003-2009 Mozdev Group, Inc. Copyright © 1999-2002 JBoss.org. Copyright Raghu K, 2003. Copyright © 2004 David Schweinsberg. Copyright © 2005-2006 Darren L. Spurgeon. Copyright © S.E. Morris (FISH) 2003-04. Copyright © 2006 VisualSoft Technologies. Copyright © 2002-2009 Zipwise Software. All rights reserved by all listed parties.

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

Copyright 1999-2003 The Apache Software Foundation. All rights reserved.

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 IMPLIED WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS 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.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. The GetOpt.java, GetOptsException.java, IllegalArgumentException.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: `install/jar/commons_upload/1_0/ CommonsFileUpload_License.txt`, `install/jar/jetspeed/1_4/RegExp_License.txt`, `install/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

Copyright (c) 2001-2003 Ant-Contrib project. All rights reserved.

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 end-user documentation included with the redistribution, if any, must include the following acknowledgement:

"This product includes software developed by the Ant-Contrib project (<http://sourceforge.net/projects/ant-contrib>)."

Alternately, this acknowledgement may appear in the software itself, if and wherever such third-party acknowledgements normally appear.

4. The name Ant-Contrib must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact ant-contrib-developers@lists.sourceforge.net.

5. Products derived from this software may not be called "Ant-Contrib" nor may "Ant-Contrib" appear in their names without prior written permission of the Ant-Contrib project.

THIS SOFTWARE IS PROVIDED `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 ANT-CONTRIB PROJECT OR ITS CONTRIBUTORS 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 Ant-Contrib Software and does not apply to the Sterling Commerce Software or to any other Third-Party Software.

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

DOM4J Software

The Sterling Commerce Software is distributed with or on the same storage media as the Dom4h Software which is free software distributed under the terms of the following license:

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.
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 name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com.
4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.
5. Due credit should be given to the DOM4J Project - <http://www.dom4j.org>

THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS ``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 METASTUFF, LTD. OR ITS CONTRIBUTORS 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.

Copyright 2001-2004 (C) MetaStuff, Ltd. All Rights Reserved.

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 ehcache v.1.5 software (Copyright © 2003-2008 Luck Consulting Pty. Ltd.) ("Ehcache 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/smcfcs/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.

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

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

* 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.

* Neither the name of Azer Koçulu, nor the names of any other contributors may be used to endorse or promote products derived from this software without specific prior written permission of Parakey Inc.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS 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 COPYRIGHT OWNER OR CONTRIBUTORS 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.

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 `install_dir/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: `install_dir/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 not. 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 disclaimer 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\ear\data\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:

Copyright 2000 by Sun Microsystems, Inc. All Rights Reserved.

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

1	Checklist for Customization Projects	
1.1	Customization Projects.....	1
2	Extensibility Overview	
2.1	Extending Your Application	5
2.2	Extending the Console User Interface.....	6
2.3	Extending the Applications Manager User Interface	7
2.4	Extending the Mobile User Interface.....	7
2.5	Extending the Database	9
2.6	ExtendingTransactions	9
2.7	Extending the Rich Client Platform User Interface	9
3	Setting Up the Development Environment	
3.1	Prerequisites for Extending Your Application.....	11
3.2	Understanding the Development Environment.....	11
3.3	Preparing the Development Environment on WebLogic	12
3.4	Preparing the Development Environment on WebSphere.....	15
3.5	Preparing the Development Environment on JBoss.....	17
3.6	Developing and Testing in the Development Environment	20
3.6.1	Testing UI Customizations	20
3.7	Configuring the UI Cache Refresh Actions	21
4	Customization Using Microsoft COM+	
4.1	Creating a COM+ Application on Windows.....	23
4.2	Adding Components to a COM+ Application	24
4.3	Configuring the COM+ Service	25
4.4	Creating a Client Proxy	26
4.5	Installing a Client Proxy	26

5 Building and Deploying Extensions

5.1	After You Create Your Extensions	27
5.2	Building Resources Extensions	27
5.3	Building Database Extensions.....	28
5.4	Building Other Extensions.....	29
5.5	Deploying Extensions.....	31
5.6	Building and Deploying Enterprise-Level Extensions	31
5.6.1	Building Enterprise-Level Extensions	32
5.6.2	Building Enterprise-Level Resources Extensions	33
5.6.3	Building Enterprise-Level Database Extensions	34
5.6.4	Building Enterprise-Level Template Extensions	34
5.6.5	Deploying Enterprise-Level Extensions	35
5.7	Customizing web.xml for Multiple Applications.....	35
5.8	Customizing web.xml for Session Timeout	36
5.9	Deploying Multiple EARs on One Application Server	37
5.9.1	Defining the JNDI Context Namespace	37
5.9.2	Defining Context Root Entries	37

6 File Names, Keywords, and Other Conventions

6.1	File Naming Conventions	39
6.2	Reserved Keywords	39
6.3	Multibyte Characters.....	40

Index

Preface

This manual explains how to develop enterprise-wide customizations to Selling and Fulfillment Foundation and provides information about how to set up the environment in which you create your customizations. This manual explains how to prepare your environment to perform customizations to the user interface and database.

Intended Audience

This manual is intended for use by those who are responsible for customizing Selling and Fulfillment Foundation.

Structure

This document contains the following chapters:

Chapter 1, "Checklist for Customization Projects"

This chapter describes a checklist of the tasks you need to perform to customize the different components of Selling and Fulfillment Foundation.

Chapter 2, "Extensibility Overview"

This chapter introduces the types of extensibility possible through Selling and Fulfillment Foundation.

Chapter 3, "Setting Up the Development Environment"

This chapter provide information directing you to instructions on how to set up the development environment.

Chapter 4, "Customization Using Microsoft COM+ "

This chapter describes how to customize Selling and Fulfillment Foundation using COM+ application.

Chapter 5, "Building and Deploying Extensions"

This chapter provides step-by-step instructions for building and deploying all the extensions you make to Selling and Fulfillment Foundation.

Chapter 6, "File Names, Keywords, and Other Conventions"

This chapter describes Selling and Fulfillment Foundation reserved keywords and special characters that are only used internally.

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.

Convention	Meaning
/ 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.
<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.

Convention	Meaning
<ANALYTICS_HOME>	User-supplied location of the Sterling Analytics installation directory. Note: This convention is used only in the <i>Selling and Fulfillment Foundation: Business Intelligence Guide</i> .
<COGNOS_HOME>	User-supplied location of the IBM Cognos 8 Business Intelligence installation directory. Note: This convention is used only in the <i>Selling and Fulfillment Foundation: Business Intelligence Guide</i> .
<MQ_JAVA_INSTALL_PATH>	User-supplied location of the IBM WebSphere® MQ Java components installation directory. Note: This convention is used only in the <i>Selling and Fulfillment Foundation: System Management and Administration Guide</i> .
<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.

Checklist for Customization Projects

This chapter provides a high-level checklist for the tasks involved in customizing or extending Selling and Fulfillment Foundation.

1.1 Customization Projects

Projects to customize or extend Selling and Fulfillment Foundation vary with the type of changes that are needed. However, most projects involve an interconnected series of changes that are best carried out in a particular order. The checklist identifies the most common order of customization tasks and indicates which guide in the documentation set provides details about each stage.

1. Prepare your development environment

Set up a development environment that mirrors your production environment, including whether you deploy Selling and Fulfillment Foundation on a WebLogic, WebSphere, or JBoss application server. Doing so ensure that you can test your extensions in a real-time environment.

You install and deploy Selling and Fulfillment Foundation in your development environment following the same steps that you used to install and deploy Selling and Fulfillment Foundation in your production environment. Refer to Selling and Fulfillment Foundation system requirements and installation documentation for details.

An option is to customize Selling and Fulfillment Foundation with Microsoft COM+. Using COM+ provides you with advantages such as increased security, better performance, increased manageability of server applications, and support for clients of mixed environments. If

this is your choice, see the *Selling and Fulfillment Foundation: Customization Basics Guide* about additional installation instructions.

2. Plan your customizations

Are you adding a new menu entry, customizing the Sign In screen and logo, creating new themes, customizing views and wizards, or adding new screens? Each type of customization varies in scope and complexity. For background, see the *Selling and Fulfillment Foundation: Customization Basics Guide*, which summarizes the types of changes that you can make.

Important guidelines about file names, keywords, and other conventions are found in the *Selling and Fulfillment Foundation: Customization Basics Guide*.

3. Extend the Database

For many customization projects, the first task is to extend the database so that it supports the other UI or API changes that you make later. For instructions, see the *Selling and Fulfillment Foundation: Extending the Database Guide* which include information about the following topics:

- Important guidelines about what you can and cannot change in the database.
- Information about modifying APIs. If you modify database tables so that any APIs are impacted, you must extend the templates of those APIs or you cannot store or retrieve data from the database. This step is required if table modifications impact an API.
- How to generate audit references so that you improve record management by tracking records at the entity level. This step is optional.

4. Make other changes to APIs

Selling and Fulfillment Foundation can call or invoke standard APIs or custom APIs. For background about APIs and the services architecture in Selling and Fulfillment Foundation, including service types, behavior, and security, see the *Selling and Fulfillment Foundation: Customizing APIs Guide*. This guide includes information about the following types of changes:

- How to invoke standard APIs for displaying data in the UI and also how to save the changes made to the UI in the database.
- Invoke customized APIs for executing your custom logic in the extended service definitions and pipeline configurations.
- APIs use input and output XML to store and retrieve data from the database. If you don't extend these API input and output XML files, you may not get the results you want in the UI when your business logic is executing.
- Every API input and output XML file has a DTD and XSD associated to it. Whenever you modify input and output XML, you must generate the corresponding DTD and XSD to ensure data integrity. If you don't generate the DTD and XSD for extended Application XMLs, you may get inconsistent data.

5. Customize the UI

Sterling Commerce applications support several UI frameworks. Depending on your application and the customizations you want to make, you may work in only one or in several of these frameworks. Each framework has its own process for customizing components like menu items, logos, themes, and etc. Depending on the framework you want, consult one of the following guides:

- *Selling and Fulfillment Foundation: Customizing Console JSP Interface for End User Guide*
- *Selling and Fulfillment Foundation: Customizing the Swing Interface Guide*
- *Selling and Fulfillment Foundation: Customizing User Interfaces for Mobile Devices Guide*
- *Selling and Fulfillment Foundation: Customizing the RCP Interface Guide* and *Selling and Fulfillment Foundation: Using the Sterling RCP Extensibility Tool Guide*
- *Customizing the Web UI Framework Guide*

6. Extend Transactions

You can extend the standard Selling and Fulfillment Foundation to enhance the functionality of your implementation of Selling and Fulfillment Foundation and to integrate with external systems. For background about transaction types, security, dynamic variables, and extending the

Condition Builder, see the *Selling and Fulfillment Foundation: Extending Transactions Guide* *Selling and Fulfillment Foundation: Extending the Condition Builder Guide* . These guides includes information about the following types of changes:

- How to extend Selling and Fulfillment Foundation Condition Builder to define complex and dynamic conditions for executing your custom business logic and using a static set of attributes.
 - How to define variables to dynamically configure properties belonging to actions, agents, and services configurations.
 - How to set up transactional data security for controlling who has access to what data, how much they can see, and what they can do with it.
 - How to create custom time-triggered transactions. You can invoke and schedule these custom time-triggered transactions in much the same manner as you invoke and schedule Selling and Fulfillment Foundation standard time-triggered transactions. Finally, you can coordinate your custom, time-triggered transactions with external transactions and run them either by raising an event, calling a user exit, or invoking a custom API or service.
7. Build and deploy your customizations or extensions

After performing the customizations that you want, you must build and deploy your customizations or extensions. First, build and deploy these customizations or extensions in the test environment for verification. When you are ready, repeat the same process to build and deploy your customizations and extensions in the production environment. For instructions, see the *Selling and Fulfillment Foundation: Customization Basics Guide* which includes information about the following topics:

- How to build and deploy standard resources, database, and other extensions (such as templates, user exits, java interfaces).
- How to build and deploy Enterprise-Level extensions.

Extensibility Overview

2.1 Extending Your Application

This chapter introduces the types of extensibility possible through Selling and Fulfillment Foundation. It provides an overview, describes high-level concepts, and provides technical architectural diagrams of Selling and Fulfillment Foundation.

Look and Feel Extensibility

Selling and Fulfillment Foundation provides a Presentation Framework toolkit that enables you to change the way information is rendered or displayed without changing the way it functions.

Transactional Extensibility

Selling and Fulfillment Foundation provides a Service Definition Framework, which is an infrastructure that automates the conversion and transportation of data between Selling and Fulfillment Foundation and third-party applications, and then converts that data into formats readable by each system. The Service Definition Framework also handles logging and exceptions. It enables you to create custom transactions that extend the functionality of Selling and Fulfillment Foundation.

Database Extensibility

In addition to customizing the user interface and transactions, you can extend the database to store additional attributes specific for your business.

Printed Documents Extensibility

You can customize printed documents. For example, you can extend the default length of bar codes.

You can use JasperReports for generating printed reports in Selling and Fulfillment Foundation. Additionally, you can also generate reports as a PDF object in Selling and Fulfillment Foundation. The JasperReports installation guidelines are provided in the `<INSTALL_DIR>/xapidocs/code_examples/jasperreports/readme.html` file. A sample JasperReport called `sampleOrderReport.pdf` is also available in the same directory.

A Jasper print component is defined in the Service Definition Framework which can be used to automatically print a document based on an event. It is a standard XML-based component that accepts XML as input and provides an output XML. For more information on this component and the print transaction, refer to the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

Browser Portal Extensibility

You can export views into the Selling and Fulfillment Foundation database that lists a user's frequently used search criteria.

2.2 Extending the Console User Interface

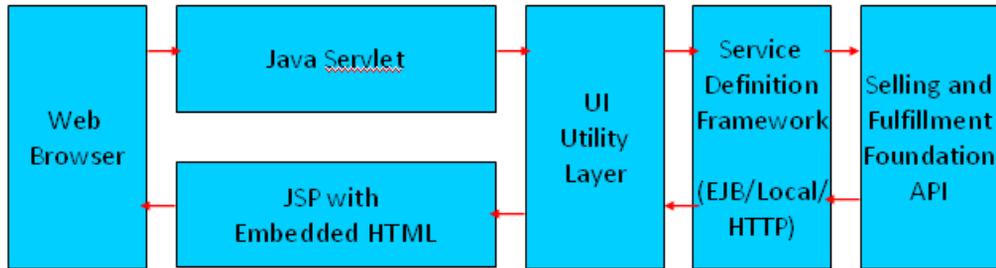
The Application Console is the user interface for conducting and tracking day-to-day transactional business, such as orders and inventory. The Application Console is also known by the terms Application Consoles or Consoles.

The Application Consoles UI uses HTML within Java Server Pages (JSPs). The user interface layer accesses the exposed APIs through services defined in the Service Definition Framework, which ensures that only exposed APIs are used.

The UI layer of the Service Definition Framework uses very minimal XML manipulation. Wherever significant manipulation of XML output becomes necessary, changes to the APIs provide a more UI friendly output.

Figure 2–1 shows the technical architecture of the Application Consoles User Interface.

Figure 2–1 Application Consoles User Interface Architecture



For detailed information about extending the Application Consoles user interface, see the *Selling and Fulfillment Foundation: Customizing Console JSP Interface for End User Guide*.

2.3 Extending the Applications Manager User Interface

The Applications Manager is the user interface for configuring the setup of an organization's business rules and transactions. The Applications Manager is composed of Java Swing pages.

For detailed information about extending the Applications Manager user interface, see *Selling and Fulfillment Foundation: Customizing the Swing Interface Guide*.

2.4 Extending the Mobile User Interface

Selling and Fulfillment Foundation enables you to develop and display a custom user interface for mobile devices used in warehouse operations.

The mobile architecture consists of two components: A client and a server.

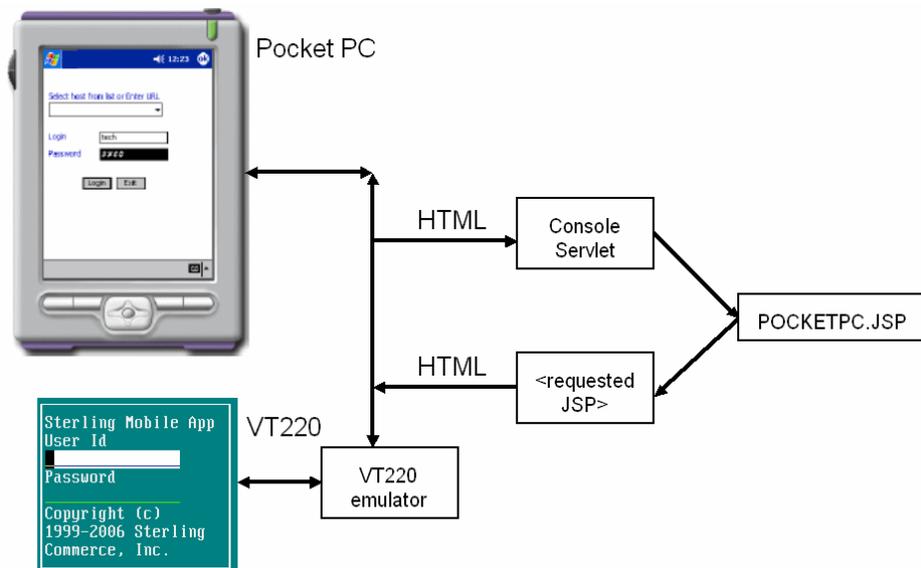
- Selling and Fulfillment Foundation mobile client—Typically a Pocket PC-based handheld device, but it can also be a VT220 emulation terminal. In this documentation, the mobile client is referred to as a *mobile device*.
- Application server—Selling and Fulfillment Foundation running on an application server.

The client and server communicate using the HTTP protocol to transfer HTML according to the request and response model. Each client request is of the type `"/console/*.ppc"`.

The application server directs requests for any `.ppc` to the controlling servlet (called the Console Servlet) which in turn redirects the request to the `pocketpc.jsp` file. The `pocketpc.jsp` file in turn redirects the request to the JSP as specified in the `uidentity`. The JSP in turn renders an HTML response, which is displayed by the Selling and Fulfillment Foundation Mobile Client.

Figure 2–2, "Mobile Architecture" illustrates this architecture.

Figure 2–2 Mobile Architecture



For detailed information about extending the mobile user interface, see the *Selling and Fulfillment Foundation: Customizing User Interfaces for Mobile Devices Guide*.

2.5 Extending the Database

Database extensibility enables you to add columns and tables to capture additional data.

For more information, see the *Selling and Fulfillment Foundation: Extending the Database Guide*.

2.6 Extending Transactions

Selling and Fulfillment Foundation provides a mechanism for processing and resolving errors during data transformation and transportation. This mechanism, called the Service Definition Framework, enables access to the following transactional processes:

- System APIs exposed by Selling and Fulfillment Foundation
- Event handlers to route the data published by Selling and Fulfillment Foundation to the transport services layer
- Time-triggered transactions that monitor and run tasks as needed

The Service Definition Framework provides error checking through the log4j utility, which writes both trace and debug information to a log file.

In addition, you can extend Selling and Fulfillment Foundation by creating the following types of custom code:

- Extended (custom) APIs
- User exits that override the default business algorithm
- User exits that extend the business algorithm used by APIs and time-triggered transactions
- Custom time-triggered transactions

For more information about extending transactions, see the *Selling and Fulfillment Foundation: Extending Transactions Guide*.

2.7 Extending the Rich Client Platform User Interface

The Rich Client Platform provides a highly interactive Rich Client, which can be remotely deployed, updated, and easily managed. A Rich Client is a client that processes the bulk of data operations without depending on

the server to which it is connected. However, it is dependent on the server, primarily for data storage. The Rich client is rich in features and functionality and has complete access to the programming functions of the operating system.

For more information, see the *Selling and Fulfillment Foundation: Customizing the RCP Interface Guide*.

Setting Up the Development Environment

3.1 Prerequisites for Extending Your Application

This guide assumes that you:

- Have already installed Selling and Fulfillment Foundation.
- Are familiar with creating and running the Selling and Fulfillment FoundationEnterprise Archive (EAR) file in deployment mode, as described in the *Selling and Fulfillment Foundation: Installation Guide*.
- Are familiar with the standard installation, otherwise known as the factory default installation.

Throughout this guide, <INSTALL_DIR> refers to the directory where you have installed Selling and Fulfillment Foundation.

3.2 Understanding the Development Environment

The development environment you need depends on the type of work you are doing.

If you intend to customize Selling and Fulfillment Foundation, you need a test environment that enables you to verify that your changes work as you intend. To save development time, you can customize your test environment to run Selling and Fulfillment Foundation in *development mode*.

If you extend the database, include the `yfsdbextn.jar` file before the `yantrashared.jar` in the CLASSPATH in all scripts.

If you have installed any PCA, for example, Sterling Call Center and Sterling Store, Sterling Store Operations, and so on, the `yantrautil.jar`

should be removed from the application server CLASSPATH before starting the application server for running the application in development mode.

Development mode saves time by enabling your application server to automatically load the latest version of edited JSP files directly from specific directories rather than reading them from the Selling and Fulfillment Foundation file. This enables you to customize and test iteratively, without having to repeatedly create the Selling and Fulfillment Foundation file.

Development mode also enables you to immediately test UI customizations.

The way you set up the development environment depends on the application server you use.

3.3 Preparing the Development Environment on WebLogic

To enable WebLogic to run Selling and Fulfillment Foundation without creating an EAR, you must define an application in WebLogic with the appropriate settings and then configure your startup script to set up the CLASSPATH required by Selling and Fulfillment Foundation.

Setting up the Selling and Fulfillment Foundation directory structure enables WebLogic to read from files directly rather than from the Selling and Fulfillment Foundation file. The following steps provide the necessary details on setting up the scripts to run WebLogic in exploded mode.

To configure WebLogic to run Selling and Fulfillment Foundation in exploded mode:

1. Edit the <WEBLOGIC_DOMAIN>/bin/startWebLogic.cmd script for windows (startWebLogic.sh for UNIX), and set the following argument in Options as java parameters:

```
-Dsci.opsproxy.disable=Y -Dvendor=shell -DvendorFile=/servers.properties  
-Dsci.naming.provider.url=t3://<hostname or ip-address of your WebLogic  
Server>:<port number of your WebLogic Server>.
```

2. Copy the web.xml file and the weblogic.xml file from the <INSTALL_DIR>/repository/eardata/smcfs/descriptors/weblogic/WAR/WEB-

INF directory to the <INSTALL_
DIR>/repository/eardata/smcfs/war directory.

3. Copy the weblogic.xml file from the <INSTALL_
DIR>/repository/eardata/smcfs/descriptors/weblogic/WAR/WEB-
INF directory to the <INSTALL_DIR>/extensions/smcfs directory.
4. Copy the ycpapibundle.properties file and ycpapibundle_<lang>_
<country>.properties (if applicable) from the <INSTALL_
DIR>/resources directory to the <INSTALL_
DIR>/repository/eardata/smcfs/war/yfscommon directory.
5. Copy the yscpapibundle.properties file and yscpapibundle_
<lang>_ <country>.properties (if applicable) from the <INSTALL_
DIR>/resources directory to the <INSTALL_
DIR>/repository/eardata/smcfs/war/yfscommon directory.
6. Copy the extnbundle.properties file and extnbundle_<lang>_
<country>.properties (if applicable) from the <INSTALL_
DIR>/resources/extn directory to the <INSTALL_
DIR>/repository/eardata/smcfs/war/yfscommon directory.
7. (Optional) If a PCA is installed, you must copy the following files to
the <INSTALL_DIR>/repository/eardata/smcfs/war/yfscommon
directory:
 - a. <INSTALL_DIR>/resources/com.yantra.yfc.rcp.common_
bundle.properties
 - b. <INSTALL_DIR>/resources/com.yantra.yfc.rcp_
bundle.properties
 - c. <INSTALL_DIR>/resources/<PCA_Code>bundle.properties. For
example, for Sterling COM PCA, copy the ycdbundle.properties
file.
8. Copy all the extensions you have made to the <INSTALL_
DIR>/extensions/global/webpages directory.
9. Rebuild the EAR file as you did during installation process.
10. Unzip the following war files from the smcfs.ear file:
 - smcfs.war
 - sbc.war
 - sma.war

Then extract each of these war files into a directory of your choice.

11. Extract the remaining jar files from the smcfs.ear and include them in the classpath.
12. Deploy each directory on WebLogic as a Web Application.
13. Test your customizations using the following standards:

Table 3–1 WebLogic Hot Deployment Test Mode

If you modify...	In these files...	Then...
Startup parameters	properties	Restart WebLogic
UI extensibility	JSP, JavaScript, CSS, theme XML	Load dynamically
Localization literals	alertmessages and localization bundle files	Restart WebLogic
Database extensions	entity XMLs	You need to rebuild the <code>entities.jar</code> file and include the jar in the classpath directory; then restart WebSphere.
APIs and other template files	template XMLs	You need to rebuild the <code>resources.jar</code> file and include the jar in the classpath directory; then restart WebLogic.

Note: Selling and Fulfillment Foundation does not support documentation extensions for online help in exploded mode. If you want to use online help in exploded mode, build a separate `smcfsdocs.ear` doc EAR file and deploy it.

Note: Selling and Fulfillment Foundation deployed in exploded mode works in the same way as the Solution deployed in EAR mode. There are no performance implications specific to exploded mode.

Sterling Commerce recommends the EAR mode of deployment in production. In case an application server hosts multiple applications, there is no interference for jars or classes across applications. This is because each application is packaged or deployed as a single EAR file isolated from other application packages (EARs). However, in exploded mode, the class that is first added to the classpath is always considered.

3.4 Preparing the Development Environment on WebSphere

When using WebSphere, you can test the modifications that you have made to Selling and Fulfillment Foundation as described in this section.

To configure WebSphere to run Selling and Fulfillment Foundation in exploded mode:

1. Set the following JVM arguments for this deployment:

```
-Dsci.opsproxy.disable=Y -Dvendor=shell -DvendorFile=/servers.properties
-Dsci.naming.provider.url=t3://<hostname or ip-address of your WebSphere
Server>:<port number of your WebSphere Server>.
```

2. Deploy the Selling and Fulfillment Foundation EAR, using the documentation provided by IBM. During deployment, WebSphere copies all the contents of WAR and EAR files to the <WAS_HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME>/ directory.

After deployment, any files copied to the <WAS_HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME>/ directory can be modified as needed. For example, if you are extending a custom code written as part of the database extensibility. The custom code files can be directly moved to the appropriate directory under <WAS_

HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME>/ directory for testing. IBM calls this ability to modify and move files as needed "hot deployment."

The custom JSPs written as part of UI extensibility can be directly incorporated into the Selling and Fulfillment Foundation WAR file.

Note: Selling and Fulfillment Foundation does not support documentation extensions for online help in exploded mode. If you want to use online help in exploded mode, build a separate `smcfsdocs.ear` doc EAR file and deploy it.

3. Build your extensions.
4. Stop the application server.
5. Copy the jars created as part of building and deploying extensions and overwrite the jars in <WAS_HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME>.

For example:

- If you are extending your database, build and deploy the `entities.jar` and copy the jar to the <WAS_HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME> directory.
 - If you are extending UI resources, build and deploy the `resources.jar`, and copy the jar to the <WAS_HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME> directory.
6. Copy your customized files (for example, localization literal files, JSPs), to the appropriate <WAS_HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME>/smcfs.war/<Module_Name> directory.

For example, if you have some customizations in the Catalog module, add the files in the <WAS_HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME>/smcfs.war/ycm directory.

7. Restart the application server.
8. Test your customizations using the following standards:

Table 3–2 WebSphere Hot Deployment Test Mode

If you modify...	In these files...	Then...
Startup parameters	properties	Restart WebSphere
UI extensibility	JSP, JavaScript, CSS, theme XML	Load dynamically
Localization literals	alertmessages and localization bundle files	Restart WebSphere
Database extensions	entity XMLs	You need to rebuild the <code>entities.jar</code> file and include the jar in the <code><WAS_HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME></code> directory; then restart WebSphere.
APIs and other template files	template XMLs	You need to rebuild the <code>resources.jar</code> file and include the jar in the <code><WAS_HOME>/AppServer/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/<APP_NAME></code> directory; then restart WebSphere.

3.5 Preparing the Development Environment on JBoss

This section explains how to test the modifications made to Selling and Fulfillment Foundation when using JBoss.

To configure JBoss to run Selling and Fulfillment Foundation in exploded mode:

1. Edit the `<JBOSS_DOMAIN>/bin/run.cmd` script for windows (`run.sh` for UNIX), and set the following argument in Options as java parameters:

```
-Dsci.opsproxy.disable=Y -Dvendor=shell -DvendorFile=/servers.properties  
-Dsci.naming.provider.url=t3://<hostname or ip-address of your JBoss  
Server>:<port number of your JBoss Server>.
```

2. Rebuild the EAR file as you did during the installation process.

Note: Selling and Fulfillment Foundation does not support documentation extensions for online help in exploded mode. If you want to use online help in exploded mode, build a separate `smcfsdocs.ear` doc EAR file and deploy it.

3. Stop the application server and execute the following steps.

For example, if you have some customizations in the Catalog module, add the files in the `<JBOSS_HOME>/server/<SERVER_NAME>/deploy/smcfs.ear/smcfs.war/ycm` directory.

Create a new directory and name it `smcfs.ear`.

4. Unzip the EAR into the `smcfs.ear` directory you created.
5. Within the `smcfs.ear` directory, a `smcfs.war` file exists. Rename this `.war` file or copy it into another directory.
6. Within the `smcfs.ear` directory, create a new subdirectory, and name it `smcfs.war`.
7. Unzip and extract all the files from the `smcfs.war` file into the `smcfs.ear/smcfs.war` subdirectory.
8. Delete the `smcfs.war` file that you renamed or copied Step 5.
9. Copy the jars created as part of building and deploying extensions and overwrite the jars in `<JBOSS_HOME>/server/<SERVER_NAME>/deploy/smcfs.ear` directory.

For example:

- If you are extending your database, build and deploy the `entities.jar` and copy the jar to the `<JBOSS_HOME>/server/<SERVER_NAME>/deploy/smcfs.ear` directory.

- If you are extending UI resources, build and deploy the `resources.jar`, and copy the jar to the `<JBOSS_HOME>/server/<SERVER_NAME>/deploy/smcfs.ear` directory.
10. Copy your customized files (for example, localization literal files, JSPs), to the appropriate `<JBOSS_HOME>/server/<SERVER_NAME>/deploy/smcfs.ear/smcfs.war` directory.
 11. Restart the application server.
 12. After deploying, you can modify the files copied to the `<JBOSS_HOME>/server/<SERVER_NAME>/deploy` directory. For example, if you extend a custom code written as part of database extensibility, you can directly move the extended custom code to the appropriate directory under the `<JBOSS_HOME>/server/<SERVER_NAME>/deploy` directory for testing. JBoss identifies the changes and redeploys the application (Hot Deployment).
 13. Test your customizations using the standards described in [Table 3–3](#).

Table 3–3 JBoss Hot Deployment Test Mode

If you modify...	In these files...	Then...
Startup parameters	properties	Restart JBoss
UI extensibility	JSP, JavaScript, CSS, theme XML	Load dynamically
Localization literals	alertmessages and localization bundle files	Restart JBoss
Database extensions	entity XMLs	You need to rebuild the <code>entities.jar</code> file and include the jar in the <code><JBOSS_HOME>/server/<SERVER_NAME>/deploy/smcfs.ear</code> directory; then restart JBoss.

Table 3–3 JBoss Hot Deployment Test Mode

If you modify...	In these files...	Then...
APIs and other template files	template XMLs	You need to rebuild the <code>resources.jar</code> file and include the jar in the <code><JBOSS_HOME>/server/<SERVER_NAME>/deploy/smcfs.ear</code> directory; then restart JBoss.
Application configuration files	<code>web.xml</code> , <code>application.xml</code> , and property files	Restart JBoss

3.6 Developing and Testing in the Development Environment

Now that you have set up your development environment, you are ready to begin customizing Selling and Fulfillment Foundation, using the directions provided throughout this guide. This section explains how to use the icons that enable you immediately test UI customizations.

Note: After any modifications are made to the following files, the application server must be restarted:

- `datatypes.xml` file
- `yfsdatatypeemap.xml` file

3.6.1 Testing UI Customizations

After making changes to UI Resources within the Resource Hierarchy tree, you can test your changes immediately by using the Cache Refresh icons.

To use the cache refresh icons:

1. Make changes as needed.
2. Select the refresh cache icon that fits your needs as follows:

- If you want to update one entity and its child resources - Select the specific entity and select the  *Refresh Entity Cache* icon
 - If you want to update all resources - Select the  *Refresh Cache* icon
3. Log into Selling and Fulfillment Foundation again to test your changes.

3.7 Configuring the UI Cache Refresh Actions

In a standard deployment of Selling and Fulfillment Foundation, any configuration changes made within the Resource Hierarchy tree do not take effect until the application server has been restarted. This means that testing your UI extensions can be a time-consuming activity. Therefore, Selling and Fulfillment Foundation provides actions within the Resource Hierarchy tree that enable you to refresh the resources so that your modifications can be tested immediately.

These actions can only be enabled in a development environment. They do not work in a deployment environment. This section explains how to enable these actions. These actions should be disabled in a deployment environment.

To configure the resource cache refresh actions:

1. Use the `<INSTALL_DIR>/properties/customer_overrides.properties` file to set the `yfs.uidev.refreshResources` property to `Y`. For additional information about overriding properties using the `customer_overrides.properties` file, see the *Selling and Fulfillment Foundation: Properties Guide*.
2. This enables the following actions on the Resource Hierarchy tree in the Applications Manager:
 -  *Refresh Cache* icon - Refreshes all resources.
 -  *Refresh Entity* icon - Refreshes the selected entity resource and its child resources.

For instructions on using the Refresh Cache icons, see [“Testing UI Customizations”](#) on page 3-20.

Customization Using Microsoft COM+

When using Selling and Fulfillment Foundation with COM+, you need to create and configure Selling and Fulfillment Foundation on a Windows server. You also need to create and install a client proxy.

4.1 Creating a COM+ Application on Windows

This section explains how to create a the Selling and Fulfillment Foundation COM+ application on a Windows server.

To create a the Selling and Fulfillment Foundation COM+ application on a server that has Windows operating system:

1. From the Windows Start menu, navigate to Administrative Tools > Component Services.
2. From the Component Services tree, navigate to Component Services > Computers > My Computer > COM+ Applications and then right-click COM+ Application. Select New > Application.
3. After the Welcome to COM Application Install Wizard screen appears, Click Next.
4. Select Create an Empty Application.
5. In the Create Empty window enter the following and then click next:
 - For Application Name enter Selling and Fulfillment Foundation.
 - For Activation Type select Server Application. This ensures that the components are started as dedicated processes.
6. In the Set Application Identity window select This User and enter the appropriate Windows user name and password. This user is the

identity under which the application is run. Make sure that the user belongs to the Administrators group. Click Next and then Click Finish.

The newly created the Selling and Fulfillment Foundation COM+ application appears under the Component Services tree.

Now you can add components to your Selling and Fulfillment Foundation COM+ application.

4.2 Adding Components to a COM+ Application

This section explains how to add components to a Selling and Fulfillment Foundation COM+ application.

To add components to a Selling and Fulfillment Foundation COM+ application:

1. From the Component Services tree, navigate to Component Services > Computers > My Computer > COM+ Applications > Selling and Fulfillment Foundation > Components and then right-click Components. Select New > Components.
2. After the Welcome to COM Component Install Wizard screen appears, click Next.
3. Select Install New Component.
4. Browse to <INSTALL_DIR>/bin/YIFComApi.dll. Select it and select Open. Then click Next and then click Finish.
5. Make sure that your system path contains the directories that store the following DLLs:
 - <INSTALL_DIR>/bin/YIFJNIApi.dll
 - <INSTALL_DIR>/bin/release/msvcrt.dll
 - <INSTALL_DIR>/bin/release/msvcp60.dll
 - jvm.dll (usually found under <JAVA_HOME>/jre/bin/hotspot on the client machine)
6. Verify that the <INSTALL_DIR>/properties directory has the yfs.properties file and the <INSTALL_DIR>/resources directory has the yifclient.properties file.

Note: If you have added any entries to the <INSTALL_DIR>/properties/customer_overrides.properties file, ensure that this file is included in the <INSTALL_DIR>/properties directory. For more information about overriding properties using customer_overrides.properties file, see the *Selling and Fulfillment Foundation: Properties Guide*.

4.3 Configuring the COM+ Service

This section explains how to configure the Selling and Fulfillment Foundation COM+ service.

To configure the Selling and Fulfillment Foundation COM+ service:

1. From the Component Services tree, right-click the newly created Selling and Fulfillment Foundation COM+ Application.
2. Select Properties.
The Selling and Fulfillment Foundation Properties dialog box is displayed.
3. Select the Advanced tab.
4. Under Server Process Shutdown panel, select Minutes Until Idle Shutdown, enter the time in minutes after which you want the process to shut down, and then Click OK.
5. Double-click the Selling and Fulfillment Foundation COM+ application.
6. Double-click Components.
7. Right-click YIFComApi.YIFComApi.1 and select Properties.
The YIFComApi.YIFComApi.1 Properties dialog box appears.
8. Select the Activation tab.
9. Select Enable object pooling.
10. In the Object pooling section, enter the Minimum and Maximum pool sizes based on the Component usage. Configure pooling to make optimal use of your hardware resources. The pool configuration can change as available hardware resources change.
11. Select Enable Just In Time Activation.

JIT activation activates an instance of an object just before the first call is made to it and then immediately deactivates the instance after it finishes processing its work.

4.4 Creating a Client Proxy

This section explains how to create a client proxy.

To create a client proxy:

1. Right-click your Selling and Fulfillment Foundation COM+ application and select Export.
2. After the Welcome to COM Application Export Wizard screen appears, click Next.
3. In the Application Export window, enter the path and file name where the export .MSI file is to be created.
4. Select Export as Application Proxy. Click Next and then Click Finish.

4.5 Installing a Client Proxy

To install a client proxy, make sure the following DLL files are in your system path:

- <INSTALL_DIR>/bin/YIFJNIApi.dll
- <INSTALL_DIR>/bin/release/msvcrt.dll
- <INSTALL_DIR>/bin/release/msvcp60.dll
- jvm.dll (located under the <JAVA>/jre/bin/hotspot directory on the client machine. Double-click the *.MSI file to install the component on the client.)

Note: This method of including the DLL files is deprecated in Release 7.7. The recommended method to call APIs and Services from a Microsoft Windows COM+ environment is through webservices or over http.

Building and Deploying Extensions

5.1 After You Create Your Extensions

After you are satisfied with all of the extensions you have made to Selling and Fulfillment Foundation (such as customizing the UI and database, creating custom code and files, and so forth), make sure that you build and deploy the new extensions. You should re-build and deploy the [Application] Enterprise Archive (EAR) with all the Java files, resource files, JSP files, custom classes, and so forth, that you created or modified.

5.2 Building Resources Extensions

To build extensions to your resources such as:

- Theme, CSS, Config resources, Data types files, and so forth
- Extended APIs, Events, and XSL templates
- Modifications made in the database, resources, and template directories.

you must re-build the `resources.jar` file by running the `deployer.sh` (or `deployer.cmd` on Windows) utility from the `<INSTALL_DIR>/bin` directory. For example:

```
./deployer.sh -t resourcejar
```

To incorporate your JSP or JS file modifications, re-build the Selling and Fulfillment Foundation EAR.

Note: If you are extending any event.xml files, create an extension directory in the `<INSTALL_DIR>/repository/xapi/template/merged/event` directory and place your extended *.xml files in the `<INSTALL_DIR>/extensions/global/template/event` folder.

Note: If you are extending an *.xsl file, place your extended files in the `<INSTALL_DIR>/extensions/global/template/xsl` directory (create this directory structure if necessary).

But when providing the name of the .xsl file during service definition, the path should be `/global/template/xsl/<CUSTOM-TEMPLATE-XSL>`.

Note: Ensure that all the extended JSP and JS files are stored in the `<INSTALL_DIR>/extensions/global/webpages` directory, if they are not already there.

5.3 Building Database Extensions

To build the extensions to your database, re-build the `entities.jar` by running the `deployer.sh` (or `deployer.cmd` on Windows) utility from the `<INSTALL_DIR>/bin` directory. For example:

```
./deployer.sh -t entitydeployer
```

Note: Before building the database extensions, make sure that all the extension files are stored in the `<INSTALL_DIR>/repository/entity/extensions` directory, if they are not already there.

To update the ERD documentation, re-build the `entities.jar` by running the `deployer.sh` (or `deployer.cmd` on Windows) utility from the `<INSTALL_DIR>/bin` directory. For example:

```
./deployer.sh -t updateERD
```

Note: By default, when you run the `entitydeployer` target or `InstallService` script, the `dbverify` tool is also run. But if you want to suppress this call to the `dbverify` tool again when you run `InstallService` script, override and set the `NO_DBVERIFY` property to `true` in the `<INSTALL_DIR>/install/properties/sandbox.cfg` file. For more information about overriding properties, refer to the *Selling and Fulfillment Foundation: Properties Guide*.

5.4 Building Other Extensions

To build your custom code extensions (user exits, extended APIs, custom implementations of supplied Java interfaces, and so forth) modifications, generate a JAR file containing these Java files and custom classes. After creating the JAR file, include the new JAR file in the `CLASSPATH` environment variable by running the `install3rdParty.sh` (or `install3rdParty.cmd` on Windows) utility from the `<INSTALL_DIR>/bin` directory. For example:

```
./install3rdParty.sh <vendorName> <vendorVersion> <-j | -l  
| -p | -r | -d > <filelist> [-targetJVM DCL | EVERY |  
NOWHERE | APP | AGENT]
```

Here, `<vendorName>` refers to the name of the vendor such as WebLogic, WebSphere, and JBoss. `<vendorVersion>` refers to the version of the vendor. Pass the appropriate argument based on the file type. You can pass the following arguments:

- `-j` for JAR or ZIP files
- `-l` for shared libraries
- `-p` for properties files
- `-r` for resource properties files
- `-d` for database JAR or ZIP files

`<filelist>` refers to the path to your custom file.

Note: If you want to make this custom JAR available to the Application Server and Agents when running the `install3rdParty` utility, based on your requirement pass the following `targetJVM` arguments:

- **DCL**—If you want to add the custom JAR to the main `dynamicclasspath.cfg` file only.
- **EVERY**—If you want to add the custom JAR to all the dynamic classpath files (for example, `APPDynamicclasspath.cfg`, `AGENTDynamicclasspath.cfg`, `OPSDynamicclasspath.cfg`, and `ACTIVEMQDynamicclasspath.cfg` files).
- **NOWHERE**—If you just want to add the custom JAR to the `<INSTALL_DIR>/jar` directory and do not want to update any of the dynamic classpath files.
- **AGENT**—If you want to add the custom JAR to the `AGENTDynamicclasspath.cfg` file.
- **APP**—If you want to add the custom JAR to the EAR file.

For example, if you want to add the custom JAR to the `AGENTDynamicclasspath.cfg` file, run the `install3rdparty` command with following arguments:

```
./install3rdParty.sh weblogic 10 -j <Path_to_your_custom_JAR> -targetJVM AGENT
```

Note: At times, mechanisms supplied by Selling and Fulfillment Foundation, such as time-triggered transactions, APIs, and user exits, are replaced by improved mechanisms. When these mechanism are replaced, they are designated as “deprecated.” Whenever possible, use the new mechanisms rather than the deprecated ones. If you do need to use a deprecated mechanism, it must be run in backward compatibility mode. In addition, note that deprecated mechanisms are supported for two major software versions, and then they are removed from the product.

5.5 Deploying Extensions

After you build the required Selling and Fulfillment Foundation extensions, you must deploy these extensions.

To deploy the extensions, re-build the EAR file as you did during installation.

Sterling Commerce recommends that you re-build and deploy the Selling and Fulfillment Foundation EAR file on your development system and test there first. Then, deploy your extensions to your production system and test them again.

Also, before deploying your extensions on a production system, verify that the `<INSTALL_DIR>/properties/customer_overrides.properties` file has the correct settings. For example, ensure that the cache refresh icons specified in the `yfs.uidev.refreshResources` property is set to N. For additional information about overriding properties using the `customer_overrides.properties` file, see the *Selling and Fulfillment Foundation: Properties Guide*

5.6 Building and Deploying Enterprise-Level Extensions

You can define resources used by Selling and Fulfillment Foundation such as templates, database extensions, UI resources, and so forth, at the Enterprise level. The Enterprise-Level resources are bundled into a services package for deployment.

Enterprise-Level resources can be developed and packaged as an Enterprise service package. This service package contains all of the components required to on-board an Enterprise. The Enterprise-Level resources are identified using a unique resource identifier. The unique resource identifier is used to locate the resources belonging to an Enterprise. Using the unique resource identifier, you can easily deploy or move the Enterprise-Level resources. You define these resource identifiers when you create an organization. For more information about creating an organization, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

5.6.1 Building Enterprise-Level Extensions

To build your modified Enterprise-Level extensions such as templates, resource files, customized webpages, entities, and so forth:

1. Create a new XML file named as `descriptor.xml` and add the following entry:

```
<ExtensionsDescriptor>
  <Package Name="<RESOURCE_IDENTIFIER>" />
</ExtensionsDescriptor>
```

where `<RESOURCE_IDENTIFIER>` refers to the unique resource identifier defined for identifying the resources belonging to an Enterprise.

2. Generate a custom JAR file containing the Enterprise-Level extension files. The custom JAR file should have the following directory structure depending on the extensions you have made:

Note: Before creating the custom JAR, make sure to copy the `descriptor.xml` file to the root of the JAR file.

- `/template/<TEMPLATE_SPECIFIC_FOLDER>`—for storing the extended template files

Where, `<TEMPLATE_SPECIFIC_FOLDER>` refers to the directory that contains the specific templates. For example:

- `api`—For storing the API-specific templates
- `email`—For storing E-mail-specific templates

- event—For storing Event-specific templates
- userexit—For storing UE-specific templates

In addition to templates, you can put JAR files, resource files, entity extensions, and customized webpages in the JAR file by creating the required folder in the root of the JAR file. For example,

- /jars—For storing the required JAR files
- /uijars—For storing the UI-specific JAR files
- /entities—For storing the extended entity XMLs
- /webpages—For storing the customized webpages
- /resources—For storing the modified resource files

3. After creating the JAR file, deploy the new JAR file by running the `InstallExtensions.sh` (or `InstallExtensions.cmd` on Windows) utility from the `<INSTALL_DIR>/bin` directory. For example:

```
./InstallExtensions.sh <filename>
```

Here, `<filename>` refers to the path to the JAR file you created in [Step 2](#).

Note: After building the template extensions, make sure that you re-build the `resources.jar`. For more information about building the `resources.jar`, see [Building Enterprise-Level Resources Extensions](#).

5.6.2 Building Enterprise-Level Resources Extensions

To build your modified Enterprise level UI resource files (such as theme, css, config resources, or datatype maps files), re-build the `resources.jar` file by running the `deployer.sh` (or `deployer.cmd` on Windows) utility from the `<INSTALL_DIR>/bin` directory. For example:

```
./deployer.sh -t resourcejar
```

To incorporate your JSP or JS file modifications, re-build the Selling and Fulfillment Foundation EAR.

Note: Make sure that all of the extended JSP and JS files are stored in the `<INSTALL_DIR>/repository/eardata/smcfs/war` directory, if they are not already there.

5.6.3 Building Enterprise-Level Database Extensions

To build your modified Enterprise-Level database, re-build the `entities.jar` file by running the `deployer.sh` (or `deployer.cmd` on Windows) utility from the `<INSTALL_DIR>/bin` directory. For example:

```
./deployer.sh -t entitydeployer
```

Note: Before building the database extensions, make sure that all of the extension files are stored in the `<INSTALL_DIR>/repository/entity/extensions` directory, if they are not already there.

Note: By default, when you run the `entitydeployer` target or `InstallService` script, the `dbverify` tool is also run. But if you want to suppress this call to the `dbverify` tool again when you run `InstallService` script, override and set the `NO_DBVERIFY` property to `true` in the `<INSTALL_DIR>/install/properties/sandbox.cfg` file. For more information about overriding properties, refer to the *Selling and Fulfillment Foundation: Properties Guide*.

5.6.4 Building Enterprise-Level Template Extensions

The system automatically reads the extended templates from the template folder that gets created in the `<INSTALL_DIR>/extensions` directory after you run the `InstallExtensions.sh` script.

5.6.5 Deploying Enterprise-Level Extensions

After you build the required Enterprise-Level extensions, you must deploy these extensions.

To deploy the Enterprise-Level extensions, re-build and deploy the Selling and Fulfillment Foundation Enterprise Archive (EAR). For more information on how to build an EAR, see the *Selling and Fulfillment Foundation: Installation Guide*.

Note: Sterling Commerce recommends that you re-build and deploy the Selling and Fulfillment Foundation EAR file on your development system and test there first. Then, deploy your extensions to your production system and test them again. For information about deploying Selling and Fulfillment Foundation, see the *Selling and Fulfillment Foundation: Installation Guide*.

Also, before deploying your extensions on a production system, ensure that the `<INSTALL_DIR>/properties/customer_overrides.properties` file has the correct settings. For example, ensure that the cache refresh icons specified in the `yfs.uiddev.refreshResources` property is set to N. For additional information about overriding properties using the `customer_overrides.properties` file, see the *Selling and Fulfillment Foundation: Properties Guide*.

5.7 Customizing web.xml for Multiple Applications

Selling and Fulfillment Foundation provides the following extensions for web.xml.

The extension package should have a ".extn" suffix and the "package_name" attribute should be used to specify the package. For example:

```
<WebComponents Package = "package_name.extn">
```

Applications have the capability to suppress some common configurations. Applications can suppress these configurations by using the Suppress element. The suppression will be done by removing any element that matches the suppression criteria. For example:

```
<WebComponents Package = "package_name.extn">
<Suppress>
<servlet><servlet-name>JasperPDFReport</servlet-name></servlet>
servlet-mapping><servlet-name>JasperPDFReport</servlet-name></
servlet-mapping>
</Suppress>
<web-app>
<!-- All the web.xml pieces needed, in standard web.xml
format. -->
</web-app>
</WebComponents>
```

In this example, all configurations are suppressed in which the servlet element contains a child `servlet-name` with the given name .

5.8 Customizing web.xml for Session Timeout

All the applications supported in Selling and Fulfillment Foundation picks the session timeout value from the `YFS_USER` table, by default the value for the session timeout is 100 minutes. This value overrides the session timeout value in the `web.xml`.

If the application do not want the session timeout value mentioned in the `web.xml` to be overridden by the `session_timeout` column value in the `YFS_USER` table, then the application must extend the `web.xml` as follows:

1. Add the context param;

```
<context-param>
<param-name>scui-suppress-user-level-sessiontimeout-override</
param-name>
<param-value>y</param-value>
</context-param>
```

2. Change the value of the session timeout with the desired value in minutes.

```
<session-config>
<session-timeout>60</session-timeout>
</session-config>
```

5.9 Deploying Multiple EARs on One Application Server

On the same application server, you can do either of the following:

- Deploy different customizations of the same or different versions of the application.
- Deploy different versions of the same application.

The number of different EARs that can be deployed on a single application server depends on the available resources on the application server. To support this deployment of multiple EARs, the different versions or customizations of the same application should be deployed as an EAR and not in exploded mode.

Note: Selling and Fulfillment Foundation assumes that each EAR file is generated from a different `<INSTALL_DIR>` directory.

5.9.1 Defining the JNDI Context Namespace

You must define the JNDI (Java Naming and Directory Interface) context namespace property to avoid JNDI clashes. To define the JNDI entries, do the following:

1. Edit the `<INSTALL_DIR>/properties/sandbox.cfg` file.
2. Add the `YFS_CONTEXT_NAMESPACE` property and assign it a name.
3. Run the following script from the `<INSTALL_DIR>/bin` directory to munge the value of this property to the `yifclient.properties` file:
 - `setupfiles.sh` (for UNIX/Linux)
 - `setupfiles.cmd` (for Windows)

5.9.2 Defining Context Root Entries

You must add the WAR file mappings for defining the context root in the `build.properties.in` file. This ensures that the web applications that are installed on single application server have unique context roots.

To add these entries, do the following:

1. Edit the `<INSTALL_DIR>/install/bin/build.properties.in` file.
2. Add the WAR file mappings for the context root paths for any web application. The key should be the name of the application's WAR file. For example,

```
platformdemo.war=/myplatformdemo  
yantrawebsservices.war=/yantrawebsservices  
platformdemodocs.war=/myplatformdemodocs
```

3. Run the following script from the `<INSTALL_DIR>/bin` directory to munge the context root WAR file mappings to the `build.properties` file:
 - `setupfiles.sh` (for UNIX/Linux)
 - `setupfiles.cmd` (for Windows)

Note: If you provide `yfs.context.namespace` property and don't provide the mapping of the WAR files in the `build.properties.in` file, the `buildEAR.cmd` (`buildear.sh` for UNIX/Linux) script will throw an error forcing the user to fill in the WAR file mappings. And if you provide the mapping of the WAR files in the `build.properties.in` file and don't provide `yfs.context.namespace` property, the WAR file mappings are ignored.

Also, the `yfs.context.namespace` property value and the WAR file mappings name should be different for each EAR that you want to deploy on the same application server.

6

File Names, Keywords, and Other Conventions

Selling and Fulfillment Foundation reserves keywords and special characters that are only used internally. For more details about using special characters, see the *Selling and Fulfillment Foundation: Localization Guide*.

6.1 File Naming Conventions

When naming files, Sterling Commerce recommends that you choose characters from the standard English-based character set, such as A through Z and 0 (zero) through 9. That way, if you need to localize the application in languages other than English, you do not need to rename files.

6.2 Reserved Keywords

Some keywords are reserved for use by Selling and Fulfillment Foundation and are important to keep in mind when programming with APIs and creating error codes. Do not create file names or error codes that start with the following:

- DCS
- INV
- OMD
- OMP
- OMR

- OMS
- SYS
- WMS
- YCM
- YCP
- YDM
- YFC
- YFE
- YFS
- YFX
- YIF
- YRET

6.3 Multibyte Characters

If you want to use multibyte characters, your database must be configured to support multibyte characters. For more information about multibyte characters and localization, see the *Selling and Fulfillment Foundation: Localization Guide*.

B

best practices

- API input, characters allowed, 39
- file names, 39
- naming files, 39
- reserved keywords, 39

C

configuring

- resource cache refresh actions, 21

customization checklist, 1

D

development environment

- developing, 20
- preparing on JBoss, 17
- preparing on Weblogic, 12
- preparing on WebSphere, 15
- testing, 20
- understanding, 11

development mode, 11

E

environment variable

- INSTALL_DIR, xxi
- INSTALL_DIR_OLD, xxi

I

INSTALL_DIR, xxi

INSTALL_DIR_OLD, xxi

internationalization

- multibyte characters, 40

L

localization

- file naming conventions, 39

M

Microsoft COM+

- adding components, 24
 - configuring service, 25
 - creating client proxy, 26
 - installing client proxy, 26
- ### multibyte characters, 40

N

non-ear mode, 11

S

special characters

- API input, 39
- in file names, 39
- keywords, 39
- multibyte support, 40
- reserved keywords, 39

U

UI cache refresh actions

 configuring, 21

UI customizations

 testing, 20

Y

yccapibundle.properties file

 deployment mode, 13

yfs.properties.in file

 resource cache refresh actions, 21

yscapibundle.properties file

 deployment mode, 13