Sterling Commerce
An IBM Company

Sterling Multi-Channel Fulfillment Solution

Localization Guide

Release 8.0

January 2008

Copyright Notice

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 <install_dir>/Readme.html.

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

http://wwws.sun.com/software/xml/developers/xsdlib2

http://www.dhtmlab.com/

http://java.sun.com/j2se/downloads.html

http://java.sun.com/products/jsse/index-103.html

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.

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

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

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>/rcpdependencies/windows/eclipse/notice.html,
<install_dir>/rcpdependencies/windows/eclipse/plugins/notice.html,
<install_dir>/rcpdependencies/gtk.linux_x86/eclipse/notice.html, and
<install_dir>/rcpdependencies/gtk.linux_x86/eclipse/plugins/notice.html.

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

ICE SOFTWARE and TEE UTILITY 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") and the Tee Utility
Software (Copyright © 2002 Karl M. Syring) ("Tee Utility Software"). The ICE Software and the Tee
Utility Software are independent from and not linked or compiled with the Sterling Commerce Software.

The ICE Software and Tee Utility Software are free software products 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 and at <install_dir>/Tee_Utility.txt. This license only applies to the ICE Software and the Tee Utility Software and does not apply to the Sterling Commerce Software, or any other Third Party Software.

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

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

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

IEEMU.JS

The Sterling Commerce Software is distributed with or on the same storage media as the IEEMU.js (Copyright (c) 1999-2007 Erik Arvidsson and Emil A. Eklund.) ("IEEMU Software"). Created by Erik Arvidsson and Emil A. Eklund http://webfx.eae.net/contact.html#erik) for WebFX (http://webfx.eae.net/). The IEEMU Software is distributed under the terms of the MOZILLA PUBLIC LICENSE Version 1.1. A copy of this license is located at <install_dir>/3rdParty/ieemu license.doc. The IEEMU.js code is distributed in source form and is located at <install_dir>/repository/eardata/platform/war/yfcscripts/ieemu.js. 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 IEEUM.js code (and not to the Sterling Commerce Software or any other Third Party Software):

"The contents of the file located at <install_dir>/3rdParty/ieemu license.doc 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 ieemu.js. The Initial Developer is Erik Arvidsson and Emil A. Eklund. Portions created by_____None Listed_____and the Initial Developer are Copyright © 1999-2007. All Rights Reserved. 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 IEEMU Software and does not apply to the Sterling Commerce Software, or any other Third Party Software.

JGO SOFTWARE

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

U.S. Government Restricted Rights

The JGo Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (C)(1)(ii) of the

Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (C)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor / manufacturer of the JGo Software is Northwoods Software Corporation, 142 Main St., Nashua, NH 03060.

MYSQL SOFTWARE

The Sterling Commerce Software is distributed on the same storage media as the MySQL Software (Copyright © 1995-2002 MySQL AB) ("MySQL Software"). Before installing the MySQL Software, the terms and conditions of the MySQL license must be accepted.

A copy of the MySQL license is provided at <install_dir>/mysql/MySQL_License.txt. This license only applies to the MySQL Software and does not apply to the Sterling Commerce Software, or any other Third Party Licensor Software.

THE APACHE SOFTWARE FOUNDATION 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, Axis, Apache Commons Lang, Apache Jakarta Commons Collections, Apache Jakarta Commons Pool, Apache Jakarta ORO, Xerces version 2.7, Apache Log4J, Apache SOAP, 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_dir>/ant/Ant_License.txt,
<install_dir>/jar/antInstaller/0_8/antinstaller_License.txt
<install_dir>/jar/commons_pool/1_2/Commons_License.txt
<install_dir>/jar/jakarta_oro/2_0_8/JakartaOro_License.txt
<install_dir>/jar/log4j/1_2_11/LOG4J_License.txt
<install_dir>/Xalan_License.txt
<install_dir>/jar/soap/2_3_1/Apache_SOAP_License.txt
<install_dir>/jar/commons_collections/2_1/Commons_License.txt
<install_dir>/jar/commons_lang/2_1/Commons_Lang_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.

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 <install_dir>/3rdParty/rico license.doc.

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."

SUN MICROSYSTEMS

The Sterling Commerce Software is distributed with or on the same storage media as the following software products (or components thereof): Sun Activation Framework, Sun JSSE, Sun JNet, and Sun JavaMail (collectively, "Sun Software").   Sun Software is free software which is distributed under the terms of the Sun Microsystems, Inc. Binary Code License Agreement ("BCLA"). A copy of the specific BCLA is found in the following directory files for the individual pieces of the Sun Software:

SUN JSSE and JNET JARS - <install_dir>/noapp/lib
SUN JavaMail - <install_dir>/jar/javamail/1_3_2

The licenses relating to the following Sun products are included in the directory files located at:

SUN COMM JAR - <install_dir>/ repository/eardata/platform/war/yfscommon
SUN ACTIVATION JAR - <install_dir>/jar/jaf/1_0_2

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 <install_dir>/repository/eardata/platform/war/WEB-INF/web_app_licence.txt

The source code for the Web-App Software may be found at: <install_dir>/repository/eardata/platform/war/WEB-INF/web-app+2_3.dtd.

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.

W3C Software

The Sterling Commerce Software is distributed on the same storage media as the W3C Software to which the following notice applies:

W3C XML Schema

Copyright © 1994-2007 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved. This work is distributed under the W3C® Software License [1] in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

[1] http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

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

Contents

2 Localizing the Sterling Multi-Channel Fulfillment Solution

3 Localizing the Sterling Rich Client Application

A   Localizable XML Attributes

Index

Preface

This manual provides an outline for localizing the Sterling Multi-Channel Fulfillment Solution suite.

Intended Audience

This manual is intended to provide a general localization procedure for users responsible for localizing the Sterling Multi-Channel Fulfillment Solution suite for new languages, as well as languages that are provided by the Sterling Multi-Channel Fulfillment Solution. Prior to reading this document, be sure that you have read the *Sterling Multi-Channel Fulfillment Solution Installation Guide* for knowledge and understanding of the Sterling Multi-Channel Fulfillment Solution database installation.

Structure

This manual contains the following sections:

Chapter 1, "Internationalization in the Sterling Multi-Channel Fulfillment Solution"

This chapter briefly lists and describes how the Sterling Multi-Channel Fulfillment Solution has been internationalized for localization, and lists the components that cannot be localized.

Chapter 2, "Localizing the Sterling Multi-Channel Fulfillment Solution"

This chapter describes the Sterling Multi-Channel Fulfillment Solution components that can be localized, and provides general guidelines on how to localize each component.

Chapter 3, "Localizing the Sterling Rich Client Application"

This chapter describes and provides general guidelines on how to localize the Sterling Multi-Channel Fulfillment Solution-based Sterling Rich Client applications.

Appendix A, "Localizable XML Attributes"

This chapter lists the localizable factory setup XML attributes.

Sterling Multi-Channel Fulfillment Solution Documentation

For more information about the Sterling Multi-Channel Fulfillment Solution$^{®}$ components, see the following manuals:

- *Sterling Multi-Channel Fulfillment Solution$^{®}$ Release Notes*
- *Sterling Selling and Fulfillment Suite$^{®}$ Release Notes*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ Installation Guide*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ Upgrade Guide*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ Configuration Deployment Tool Guide*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ Performance Management Guide*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ High Availability Guide*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ System Management Guide*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ Localization Guide*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ Customization Guide*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ Integration Guide*
- *Sterling Selling and Fulfillment Suite$^{®}$ Integration Guide*
- *Sterling Multi-Channel Fulfillment Solution$^{®}$ Product Concepts*
- *Sterling Warehouse Management System$^{®}$ Concepts Guide*
- *Sterling Multi-Channel Fulfillment Solution Platform$^{®}$ Configuration Guide*

- *Sterling Distributed Order Management® Configuration Guide*
- *Sterling Supply Collaboration® Configuration Guide*
- *Sterling Global Inventory Visibility® Configuration Guide*
- *Sterling Product Management® Configuration Guide*
- *Sterling Logistics Management® Configuration Guide*
- *Sterling Reverse Logistics® Configuration Guide*
- *Sterling Warehouse Management System® Configuration Guide*
- *Sterling Multi-Channel Fulfillment Solution 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*
- *Sterling Multi-Channel Fulfillment Solution Mobile Application® User Guide*
- *Sterling Multi-Channel Fulfillment Solution Analytics® Guide*
- *Sterling Multi-Channel Fulfillment Solution® Javadocs*
- *Sterling Multi-Channel Fulfillment Solution® Glossary*
- *Sterling Parcel Carrier Adapter® Guide*

Conventions

The following conventions may be used in this manual:

Convention	Meaning
. . .	Ellipsis represents information that has been omitted.
< >	Angle brackets indicate user-supplied input.

Convention	Meaning
`mono-spaced text`	Mono-spaced text indicates a file name, directory path, attribute name, or an inline code example or command.
/ or \	Slashes and backslashes are file separators for Windows, UNIX, and Linux operating systems. The file separator for the Windows operating system is "\" and the file separator for UNIX and Linux systems is "/". The UNIX convention is used unless otherwise mentioned.
<INSTALL_DIR>	User-supplied location of the Sterling Multi-Channel Fulfillment Solution installation directory. This is only applicable for Release 8.0 or above.
<INSTALL_DIR_OLD>	User-supplied location of the Sterling Multi-Channel Fulfillment Solution installation directory for previously installed releases. This is only applicable for Release 8.0 or above.
<YANTRA_HOME>	User-supplied location of the Sterling Supply Chain Applications installation directory. This is only applicable for Release 7.7, 7.9, and 7.11.
<YANTRA_HOME_OLD>	User-supplied location of the Sterling Supply Chain Applications installation directory for previously installed releases. This is only applicable for Releases 7.7, 7.9, and 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, 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 of previously installed releases. This is only applicable for Releases 7.7, 7.9, and 7.11. |

Convention	Meaning
<ANALYTICS_HOME>	User-supplied location of the Sterling Multi-Channel Fulfillment Solution Analytics installation directory. **Note:** This convention is used only in the *Sterling Multi-Channel Fulfillment Solution Analytics Guide*.
<COGNOS_HOME>	User-supplied location of the Cognos installation directory. **Note:** This convention is used only in the *Sterling Multi-Channel Fulfillment Solution Analytics Guide*.
<MQ_JAVA_INSTALL_PATH>	User-supplied location of the IBM WebSphere MQ Java components installation directory. **Note:** This convention is used only in the *Sterling Multi-Channel Fulfillment Solution System Management Guide*.
<DB>	Refers to the Oracle, DB2, or MSSQL depending on the database server.
<DB_TYPE>	Depending on the database used, considers the value oracle, db2, or sqlserver.

1

Internationalization in the Sterling Multi-Channel Fulfillment Solution

This chapter provides an overview of concepts regarding localization and internationalization. Also, it explains how they apply to the Sterling Multi-Channel Fulfillment Solution.

1.1 Localizable Components and Data

This section describes the internationalized components and data that can be localized in your system.

Every user created within the Sterling Multi-Channel Fulfillment Solution can have an associated preference for number formatting, date layout, and language. These preferences are called a locale and are identified as a pairing of a language code and a country code. Examples include `en_US` (English, US), `fr_CA` (French, Canada), `fr_FR` (French, France). A specific locale definition includes the following information:

- Country

- Language

- Date and time format

- Time zones

- Numeric Formats

- Currency

- Unit of measure for dimensions, volume, and weight

To associate data such as dates, times, and strings with locale-specific formats, the Sterling Multi-Channel Fulfillment Solution associates a

locale with each user profile. This allows each user to interact with a locale-specific version of the product.

The locale definition associated with any organization defined in the Sterling Multi-Channel Fulfillment Solution is used to determine currency and unit of measure only. Date, time, and time zone information is strictly related to each user.

1.1.1 Literals and Data

All user interface and exception message literals are retrieved from a set of external files or database tables.

The Sterling Multi-Channel Fulfillment Solution retrieves images and literals from locale-specific files. In order to provide a single-installation multi-lingual solution, the Sterling Multi-Channel Fulfillment Solution stores multiple instances of the literals for a screen. Each instance is identified by a specific country and language pairing.

1.1.2 Multi-Byte Character Sets

Multi-byte character sets are appropriately and thoroughly taken into consideration in the database, application server, and browser tiers of the Sterling Multi-Channel Fulfillment Solution. To represent all of the characters in a language it is sometimes necessary to use 2 (double-byte) or 3 (multi-byte) bytes for each character. The longer character representations can represent a space and transmission challenge for application development.

- *Double-byte Character Set (DBCS)***:** One of a number of character sets defined for representing Chinese, Japanese, or Korean text (for example, JIS X 0208-1990). These character sets are often encoded in such a way as to allow double-byte character encoding to be mixed with single-byte character encoding.

- *Multi-byte Character Set (MBCS)***:** A character set encoded with a variable number of bytes for each character. Many large character sets have been defined as multi-byte character set in order to keep strict compatibility with the standards of the ASCII subset, the ISO and IEC 2022.

The Sterling Multi-Channel Fulfillment Solution architecture ensures that:

- All data is stored in the database using a standard compression algorithm known as UTF-8.

- The application is coded in Java which can handle multi-byte character sets without any special changes.

- All communication between the database and the application server is through Java Database Connectivity (JDBC) which transforms the UTF-8 database representation of data to and from the multi-byte character set.

- All communication between the application server and the client is through UTF-8, which minimizes data transmission volume.

- All clients are expected to receive and send data using the UTF-8 algorithm.

1.1.3 Date and Time Formats

The Sterling Multi-Channel Fulfillment Solution can present stored dates and times in any valid date or time format. Date and time fields in the Sterling Multi-Channel Fulfillment Solution are expected to be entered relative to the locale where the Sterling Multi-Channel Fulfillment Solution database resides. Some typical date formats are as follows:

- MM/dd/yyyy

- dd/MM/yyyy

- yyyy/MM/dd

> **Note:** In each date format, the month can be expressed as a word instead of a numeral as long as the entire date does not exceed ten characters in length.

Some typical time formats are as follows:

- HH:mm:ss

- HH:mm

1.1.4 Time Zones

Besides being sensitive to local time zone considerations, the Sterling Multi-Channel Fulfillment Solution is aware of worldwide time zones. For

example, if an order is entered in Germany for fulfillment in the United States and the order is not filled in time, the software considers the differences in time zones in order to raise an exception at the appropriate hour in the United States.

The following characteristics guarantee sensitivity to time zones:

- When a date and time value is stored, it is converted to the locale of the database. For example, assume that a database is in New York and a customer service representative is in London. The customer service representative enters an order. When the date and time of the order is stored in the database, which resides in New York, the values are converted to Eastern Standard Time.

- When a date and time field is displayed through the user interface, the Sterling Multi-Channel Fulfillment Solution performs time zone calculations based on the current locale of the user and presents the time accordingly. For example, when a customer service representative in London displays an order, which resides in a New York database, the date and time are converted from Eastern Standard Time to Greenwich Mean Time.

> **Note:** When a date field does not contain a time component, the time is assumed to be 12 AM. Such fields are not adjusted for time zone when viewed from various locales with different time zones.

- The Sterling Multi-Channel Fulfillment Solution APIs output the date and time as stored in the database. For time-sensitive fields the time zone difference from the Universal Time Coordinate (UTC) is appended to the date and time in the output.

1.1.5 Numeric Formats

Numeric formats are dependent on the country and language set up in the locale definition.

1.1.6 Currency

The Sterling Multi-Channel Fulfillment Solution allows each order to be processed in the preferred currency of the customer. This currency is known as the "transactional currency".

A currency is also associated with the locale that is associated with each enterprise. This currency is known as the "enterprise currency".

Order management capabilities provide order handling in multiple currencies between buyer and seller. These capabilities include a multi-currency view of an order's value as well as currency conversion procedures and rate tables.

Data structures hold flexible charge and tax taxonomies for order and invoice entities. The Sterling Multi-Channel Fulfillment Solution has deliberately *not* taken the approach of building complex taxation rules into the application. Integration with sophisticated tax calculation programs, such as Taxware or Vertex, complement our solution and provide you with a complete taxation system. The Sterling Multi-Channel Fulfillment Solution optionally provides a standard integration to the Taxware product line.

> **Note:** All charges and taxes display in the user interface in the order (transactional) currency. Users can switch to view payment information of an order in either the transactional or enterprise currency.

1.1.6.1 Currency Precision

The Sterling Multi-Channel Fulfillment Solution has the following currency precision characteristics:

- Unit price and unit cost are entered and stored to a maximum of six decimal places.

- Totals are entered and stored to a maximum of two decimal places.

- Totals are rounded with traditional rounding (round up any digit of 5 or more).

1.1.6.2 Currency Conversion

The Sterling Multi-Channel Fulfillment Solution has the following currency conversion characteristics:

- Currency conversion rates are defined between two currencies and are bound by effective dates. For example:

Table 1–1   Currency Conversion Rates

From	To	From	To	Rate
British Pound	US Dollar	1/1/2004	1/31/2004	1.51
British Pound	US Dollar	2/1/2004	2/28/2004	1.56
British Pound	US Dollar	3/1/2004	3/31/2004	1.62

A conversion rate definition is understood to imply a 1:x relationship. Thus, if one British Pound is equal to 1.51 US dollars during January 2004, two British pounds are equal to 3.02 US dollars during January 2004.

- Rate definitions are not reciprocal. A conversion from US dollars to British pounds cannot use the inverse of the British pound to US dollar exchange rate. A rate for converting US dollars to British pounds must be available. Otherwise,the Sterling Multi-Channel Fulfillment Solution reports an error. The only exception to this restriction is that a reciprocal relation does exist between the euro and its member currencies.

- The Sterling Multi-Channel Fulfillment Solution provides the `updateConversionRates` API that allows users to import exchange rates and maintain the exchange rates in their database. Using this API, new exchange rates can be loaded every few hours if desired. You can also set up conversions using the Sterling Multi-Channel Fulfillment Solution Configurator.

- The Sterling Multi-Channel Fulfillment Solution provides a user exit at the point of performing a currency conversion. This user exit allows users to perform currency conversions outside of the Sterling Multi-Channel Fulfillment Solution. This allows real-time application of the current exchange rate instead of using the last updated exchange rate.

1.1.6.3 Currency Conversion Scenario

The following scenario illustrates how currency is displayed for an order. Suppose that a customer enters an order in France with an English company. The customer pays in francs (the transactional currency). The

Sterling Multi-Channel Fulfillment Solution converts the amount from francs to pounds using triangulation through the euro.

Because the order is placed in francs, if a company employee in England displays the order, the order is displayed in francs.

To allow auditing or reviewing of currency conversions, the Sterling Multi-Channel Fulfillment Solution stores the exchange rates for each order as an order attribute. When an order is entered, the software stores the exchange rate used between the transactional currency and the enterprise currency. If a conversion involves triangulation, the Sterling Multi-Channel Fulfillment Solution stores the composite exchange rate used between the euro, the transactional currency, and the enterprise currency.

1.1.7 Units of Measure

A Sterling Multi-Channel Fulfillment Solution user can select different units of measure (UOM) from various places in the Sterling Multi-Channel Fulfillment Solution User Interface.

For supported carriers, pack dimensions entered by a shipper can be converted to different dimensions needed by the carrier. For example, if a shipper enters pack dimensions in kilos, kilos can be converted to pounds for a carrier. To accomplish this conversion for other carriers, custom coding is needed.

1.2 Non-Localizable Components and Data

The Sterling Multi-Channel Fulfillment Solution does not support the localization of components that are explicitly technical in nature. These components are typically used by professionals who are working with a Sterling Commerce employee to perform installation, tuning tasks, and so forth. The components include documentation and the following tools:

- Configuration Deployment Tool
- VT220 Mobile Terminal

1.3 Localizing the Sterling Multi-Channel Fulfillment Solution Using Language Packs

You can localize the Sterling Multi-Channel Fulfillment Solution in a single base language with one or multiple language packs. The base language refers to the display language of the factory setup data in the Sterling Multi-Channel Fulfillment Solution Configurator. For more information about switching the base language, see Section 2.4.1.2, "Switching the Sterling Multi-Channel Fulfillment Solution Base Language".

> **Note:** For language pack installations on Windows XP, where the base language is switched to a non-English language on an English operating system, the following operating system settings must be applied:
>
> **1.** Go to Start>Settings>Control Panel>Regional and Language Options.
>
> **2.** Under Regional Options, select the applicable language.
>
> **3.** Depending on the language that you are localizing, under the Languages tab, select the Install files for East Asian languages check box.
>
> **4.** Select the Advanced tab, select the applicable language.

> **Note:** For language pack installations on Linux, you need to install the language fonts. These can be obtained from the installation disks. For example, to install Chinese fonts, install the zh_CN-2.14-6.noarch.rpm file.

> **Note:** The languages listed here are for informational purposes only. For actual available languages, please contact Sterling Commerce's sales division at 1-978-513-6000.

The following list provides a description of the terms used in the tables.

- Base Language - Refers to the language for displaying factory setup data in the Sterling Multi-Channel Fulfillment Solution Configurator.

- Config Literals - Refers to the labels on the Sterling Multi-Channel Fulfillment Solution Configurator screen.

- Console Literals - Refers to the labels on the Sterling Multi-Channel Fulfillment Solution Consoles screen.

- Config Data - Refers to the factory setup data. A specific list of fields can be translated. For example, the status description can be localized.

- Config data on Console - Refers to the display of the config data on the console user interface. For example, Carrier is a config data that appears on the console screen.

A schematic diagram with the above discussed terms is given below:
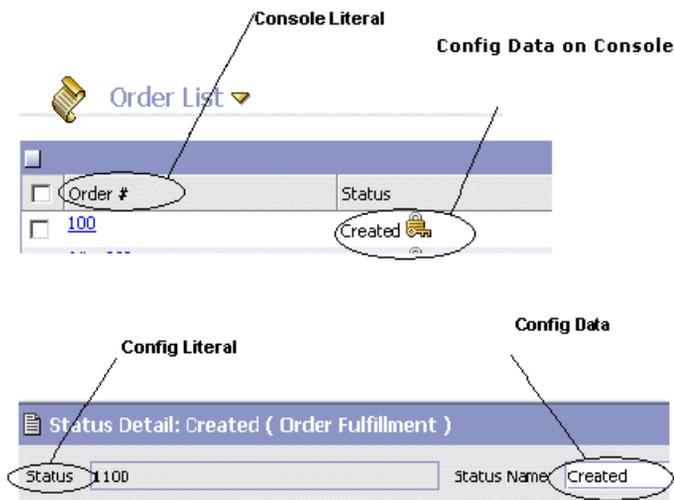
The following sections provide more information on the localized literals and data if you have purchased single language CD or multiple language CDs.

Single Language CD

The Table 1–2, "Localization Options for Single Language CD" provides two different options for the localization information if you get a single language pack along with a standard base language.

Table 1–2   Localization Options for Single Language CD

Options	User Locale	Config Literals	Config Data	Console Literals	Config Data on Console
Option 1: Base Language English with Japanese Language Pack	**Japanese**	Japanese	English	Japanese	Japanese
	English	English	English	English	English
Option 1: Base Language Japanese with English Language Pack	**Japanese**	Japanese	Japanese	Japanese	Japanese
	English	English	Japanese	English	English

Multiple Language CDs

Each language pack is shipped as a separate CD, if you have purchased a Japanese and Chinese language packs you would have 2 language CDs.

The Table 1–3, "Localization Options for Multiple Language CD" provides three different options for the localization information if you get multiple language packs along with a standard base language.

> **Note:**   Each of the below specified option is supported as part of your implementation design. However, you need to decide the best possible option for your business needs.

Table 1–3   Localization Options for Multiple Language CD

Options	User Locale	Config Literals	Config Data	Console Literals	Config Data on Console
Option 1: Base Language Japanese with Chinese and English Language Pack	**Japanese**	Japanese	Japanese	Japanese	Japanese
	English	English	Japanese	English	English
	Chinese	Chinese	Japanese	Chinese	Chinese

Table 1–3   Localization Options for Multiple Language CD

Options	User Locale	Config Literals	Config Data	Console Literals	Config Data on Console
Option 2: Base Language Chinese with Japanese and English Language Pack	**Japanese**	Japanese	Chinese	Japanese	Japanese
	English	English	Chinese	English	English
	Chinese	Chinese	Chinese	Chinese	Chinese
Option 3: Base Language English with Japanese and Chinese Language Pack	**Japanese**	Japanese	English	Japanese	Japanese
	English	English	English	English	English
	Chinese	Chinese	English	Chinese	Chinese

2

Localizing the Sterling Multi-Channel Fulfillment Solution

This chapter provides steps for localizing the Sterling Multi-Channel Fulfillment Solution for both languages that are not yet provided and languages that are supported by the Sterling Multi-Channel Fulfillment Solution out of the box.

2.1 Before You Begin

Before you begin localizing the Sterling Multi-Channel Fulfillment Solution, keep the information in this section in mind.

Rules Governing Localization

Some of the localization rules that the Sterling Multi-Channel Fulfillment Solution follows are governed by standards and rules external to the Sterling Multi-Channel Fulfillment Solution, such as Java JVM, JRE, operating systems, and external applications. The Java programming language specifies the implementation of the locale logic. For example, see http://java.sun.com/j2se/1.5.0/docs/api/java/util/ResourceBundle.html for more information about the logic around resource bundles. For more information about specifying a Java JRE, see Chapter 2.2.1, "Specifying the JRE Settings".

> **Note:** If possible, Sterling Commerce recommends that an exact match is used for all localization files (`_<lang>_<country>` in the file name), to ensure that all files are correctly returned.

International Standards Organization (ISO) Codes

The Sterling Multi-Channel Fulfillment Solution uses locale-related codes as specified by the ISO. Throughout this guide, locale is represented by <language>_<country>. Locale is comprised of <language>, a 2-character lower-case ISO-639 code, and <country>, a 2-character upper-case ISO-3166 code.

Character Encoding

When planning the localization of the Sterling Multi-Channel Fulfillment Solution, ensure that your operating system default character encoding is set to the one specified by the `yfs.ui.defaultEncoding` property in the `yfs.properties` file during product installation.

To modify this property, add an entry for it in the `<INSTALL_DIR>/properties/customer_overrides.properties` file. For additional information about modifying properties and the `customer_overrides.properties` file, see the *Sterling Multi-Channel Fulfillment Solution Installation Guide*.

When deploying on WebSphere, in order to enable entering any characters other than the ASCII set (for example Asian characters and vowels with accents), ensure that the WebSphere JVM `client.encoding.override` property is set to UTF-8. For more information about required WebSphere JVM settings, see the *Sterling Multi-Channel Fulfillment Solution Installation Guide*.

Literals External to the Sterling Multi-Channel Fulfillment Solution

External applications used in conjunction with the Sterling Multi-Channel Fulfillment Solution may display on-screen information. For example, WebLogic displays a status message during startup. Non-Sterling Multi-Channel Fulfillment Solution literals of this type may need to be localized. For instructions on how to localize these types of messages, see the documentation supplied by that software provider.

Windows Clients

When localizing the Sterling Multi-Channel Fulfillment Solution Configurator to run on Windows, ensure that client computers have regional options set appropriately within the Windows Control Panel.

For example, if the client computer is running Windows and you want to display the title "Sterling Multi-Channel Fulfillment Solution Configurator" in Japanese, the client Windows regional options must be set to Japanese.

When localizing the Sterling Multi-Channel Fulfillment Solution Consoles to run on Windows, ensure that client computers can correctly display Unicode characters by configuring the display appearance within Control Panel > Display > Appearance.

Mobile Device Clients

The `ycpapibundle.properties` and `yscpapibundle.properties` files contains literals that apply specifically to mobile devices. The key for these mobile device literals are preceded with "`Mobile_`". When localizing the Sterling Multi-Channel Fulfillment Solution to run on a mobile device, localize the literals associated with these keys.

2.2 Setting Up Character Encoding

For your implementation of the Sterling Multi-Channel Fulfillment Solution, use the guidelines specified within this section when using Latin-1 and UTF-8 encoding.

2.2.1 Specifying the JRE Settings

In a multi-language deployment, install the international version of the JRE onto each client computer.

To install the plugin, download the JRE from http://java.sun.com/javase/downloads/index_jdk5.jsp and install it on each client computer. For clients that must display non-English literals, choose the "Windows (all languages, including English)" JRE.

2.2.2 Latin-1 Character Encoding

ISO 8859-1 is the ISO standard Latin-1 character set and encoding format. CP1252 is what Microsoft defined as the superset of ISO 8859-1,

so there are approximately 27 extra characters that are not included in the standard ISO 8859-1.

When using Latin-1 character encoding, Sterling Commerce strongly recommends that you also refer to the following documents:

- For more information about "Basic Latin" and the "Latin-1 Supplement" that together comprise the "Latin-1" (or ISO 8859-1) character set and encoding, see http://www.unicode.org/charts/.

- For an official chart download, see

  http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=28245&ICS1=35&ICS2=40&ICS3= ().

- For a Microsoft site that lists both ISO 8859-1 and CP1252, see

  http://www.microsoft.com/globaldev/reference/WinCP.mspx.

2.2.3 UTF-8 Character Encoding

Encoding must be specified in order for international characters to display correctly on UI components. For example:

- Inventory Graph

- Delivery Map

- Shipnode in Inventory Console

- Item ID in Inventory Console

On Unix, the UTF-8 character encoding must be specified on your Unix application server.

To set up a UTF-8 environment:

5. Modify all process startup scripts (for the application server, time-triggered transactions, and so forth) to include the -Dfile.encoding=UTF-8 parameter for the Java command.

   -OR-

6. On the Unix application server, determine whether or not it has UTF-8 capability by running the following command:

   ```
   locale -a
   ```

— If the command returns *any* line that indicates UTF-8, go to Step 5.

For example:

```
POSIX
common
en_US.UTF-8
C
iso_8859_1
```

— If the command returns no lines that indicate UTF-8, go to Step 7.

7. From the international language option pack appropriate to your Unix operating system, install at least one language that has the UTF-8 character set, then return to Step 6 to test its correct installation.

2.3 Database Overview

Data Encoding Format

The Sterling Multi-Channel Fulfillment Solution is tested and shipped using the UTF-8 transformation format. If you use a different transformation encoding format, the number of characters that you can store in standard database sizes diminishes. In this case, you must ensure that you review and modify the Sterling Multi-Channel Fulfillment Solution database creation process to size the database fields accordingly.

Character Set

Use a character set appropriate for your desired localization language. For example, single-byte language character sets typically require UTF-8, while a multi-byte language may require a UTF-16 character set.

The character set you choose may impact field sizes. For example, a Varchar(40) field can only store 40/3 Japanese characters using the UTF-8 character set. This has implications on the table field sizes at the time of creation. Table creation scripts must be changed to ensure that the field lengths are correct.

String Length Checker

To run the string length checker to ensure that translated strings do not exceed the field lengths of the tables, follow the following steps:

1. Create a folder named `/Length`.

2. Copy the contents of `<INSTALL_DIR>/repository/entity`, `including all subfolders, into /Length/entities`.

3. Copy the contents of `<INSTALL_DIR>/installed_` `data/sscap/components/complete_` `installation/factorysetup/XMLS into /Length/XMLS`.

4. Copy the `<INSTALL_DIR>/repository/datatypes/datatypes.xml` into `/Length`.

5. Copy the following JAR files to `/Length/lib` directory:

   - `<INSTALL_DIR>/jar/platform_afc/3000/platform_afc.jar`

   - `<INSTALL_DIR>/jar/install_foundation.jar`

   - `<INSTALL_DIR>/jar/log4j/1_2_12/log4j-1.2.12.jar`

   - `<INSTALL_DIR>/jdk/jre/lib/endorsed/xalan.jar`

   - `<INSTALL_DIR>/jdk/jre/lib/endorsed/xercesImpl.jar`

   - `<INSTALL_DIR>/jdk/jre/lib/endorsed/xmlParserAPIs.jar`

6. Copy `<INSTALL_DIR>/installed_` `data/sscap/components/complete_` `installation/factorysetup/XMLS/<language>_` `><country>/<baselanguage>_<basecountry>_` `ycplocalizedstrings_<language>_<country>.properties` to `/Length`

7. Set `CLASSPATH=` `/Length/lib/xmlParserAPIs.jar;/Length/lib/xalan.jar;/Length` `/lib/xercesImpl.jar;/Length/lib/platform_` `afc.jar;/Length/lib/install_` `foundation.jar;/Length/lib/log4j-1.2.12.jar`

8. Run the following Java command. This command runs the string length checker in GENERATE mode. In this mode, the output file contains a list of translatable literals and their maximum string lengths.

```
call <JAVA_HOME>/bin/java
com.yantra.ycp.tools.localization.YCPLocalizedStringLengthT
ool -OUTPUT_FILE /Length/LengthsFile.txt -MODE GENERATE
```

```
-ENTITY_DIR /Length/entities -DTYPES_FILE
/Length/datatypes.xml -FC_DIR /Length/XMLS
```

9. Run the following java command. This command runs the string length checker in CHECK mode. In this mode, the localizedstrings file (for instance en_US_ycplocalizedstrings_ja_JP.properties) is compared with the LengthsFile.txt file that is generated from running the string length checker in GENERATE mode. Running this tool in CHECK mode also creates MissingLength.txt, which contains the literals that are missing from the LengthsFile.txt, and MissingTranslations.txt, which contains the literals missing from the localizedstrings file that was passed in the input.

```
call <JAVA_HOME>/bin/java
com.yantra.ycp.tools.localization.YCPLocalizedStringLengthT
ool -OUTPUT_FILE inconsistencies.txt -MODE CHECK -LENGTHS_
FILE /Length/LengthsFile.txt -TRANSLATIONS_FILE
/Length/<baselanguage>_<basecountry>_
<prefix>localizedstrings_<language>_<country>.properties.
```

Factory Defaults

Factory defaults are limited to one language. This means that in order to switch from one language to another, you must load new factory defaults to your database.

The UI literals are not part of the factory defaults. They can be switched from one language to another as needed, as long as they have been translated appropriately.

When using an Oracle database, it is possible to use Japanese characters in English factory defaults. In fact, it could be any character in valid Unicode and UTF-8 range. When using a SQL Server database, it is not possible to use more than one encoding and code page, based on the database collation chosen at time of the database creation. Therefore, if the database is created with a collation that is not Japanese, the factory defaults cannot contain Japanese character collation, such as Latin1 (SQL_Latin1_General_CP1_CI_AS).

2.3.1 Oracle Database Setup

In this release of the Sterling Multi-Channel Fulfillment Solution, multi-lingual (including a multi-byte character set) environments with Oracle databases have been tested and certified using Oracle 10g created

with a UTF-8 character set. See the *Sterling Multi-Channel Fulfillment Solution Installation Guide* for more information about installing Oracle 10g and the settings for the UTF-8 character set.

2.3.2 SQL Server Database Setup

SQL Server is limited to the collation and code page with which the database is created to store characters. Because of this, during database creation, you must carefully consider collation issues regarding storage of data with non-English international characters, including supported Asian code pages.

Sterling Commerce recommends that SQL Server requires case-insensitive collation (specified using the CI argument). For example, to choose a case-insensitive collation for the Japanese language, one of the valid collation choices is Japanese_CI_AI.

The localizable factory setup is limited to the selected collation and code page as well, since they are stored in database.

SQL Server permits one locale for each database.

To set up SQL Server:

1. From the SQL Server collation name drop-down menu, select the collation name that supports the language you want to specify. Be sure to select a collation that is case insensitive. This creates your database.

2. Configure the `charset` value for the JDBC URL property to match the code page of the collation name selected in SQL Server in the `<INSTALL_DIR>/properties/customer_overrides.properties` as shown below:

   Jdbc url format is
   "jdbc:inetdae7:<hostName>?charset=<yourCharset>"

   For additional information about modifying properties and the `customer_overrides.properties` file, see the *Sterling Multi-Channel Fulfillment Solution Installation Guide*.

3. Install the application tables and other components, and then load the database factory defaults as described in the *Sterling Multi-Channel Fulfillment Solution Installation Guide*.

2.3.3 DB2 Database Setup

DB2 database can be localized by passing the codeset as UTF-8. For more information about DB2 database setup, see the *Sterling Multi-Channel Fulfillment Solution Installation Guide*.

2.4 Data

Besides storing your transactional data, the database also stores configuration data, such as error codes and item descriptions of various attributes. This means that the database may need to store values in a language-specific format. If these database literals are not localized, screen literals display inconsistently, with some displaying in the localized language and others displaying in English.

You can store item descriptions in your database in multiple languages. For more information about item descriptions, see the *Sterling Product Management Configuration Guide*.

> **Note:** The localized item description is only available for Order Details and Order Line Details screens.

2.4.1 Localization for a Multi-Language Installation

The database factory default values can be localized for a multi-language installation, to use one or more locales in addition to the installed locale.

To localize factory default XML and user-configured attributes for multiple locales:

1. Create the necessary locale using the Sterling Multi-Channel Fulfillment Solution Configurator. For more information about creating a locale, see the *Sterling Multi-Channel Fulfillment Solution Platform Configuration Guide*.

2. Run the LocalizedStringReconciler tool from `<INSTALL_DIR>/bin` in EXPORT mode as follows:

   For UNIX:

   ```
   ant.sh -f localizedstringreconciler.xml export
   -Ddestdir=<INSTALL_DIR>/installed_data/sscap/components/complete_
   installation/factorysetup/XMLS
   ```

For Windows:

```
ant.cmd -f localizedstringreconciler.xml
export -Ddestdir=<INSTALL_DIR>/installed_data/sscap/components/complete_
installation/factorysetup/XMLS
```

This processes the entity XMLs and identifies the missing literals for the localizable columns in the YFS_LOCALIZED_STRINGS table.

The data is then exported to one or more properties files depending on the number of locales present in the YFS_LOCALE table.

To export for a specific locale, use optional parameters language, and country. This usage is as follows:

For UNIX:

```
ant.sh -f localizedstringreconciler.xml export -Ddestdir=<INSTALL_
DIR>/installed_data/sscap/components/complete_
installation/factorysetup/XMLS -Dlanguage=fr -Dcountry=FR
```

For Windows:

```
ant.cmd -f localizedstringreconciler.xml export -Ddestdir=<INSTALL_
DIR>/installed_data/sscap/components/complete_
installation/factorysetup/XMLS -Dlanguage=fr -Dcountry=FR
```

> **Note:** In an instance where the YFS_LOCALE table contains two entries other than the base locale, two files are created. For example, if fr_FR (representing French, and France) is the locale, a file named en_US_ ycpmissinglocalizedstrings_fr_FR.properties is created in the destination folder specified (destdir).

3. Edit the aa_BB_ycpmissinglocalizedstrings_xx_YY.properties file for the relevant locale, in the <INSTALL_DIR>/installed_ data/sscap/components/complete_ installation/factorysetup/XMLS/<xx_YY> directory, where aa is the language code for the 'from' locale', BB is the country code for the 'from' locale, xx is the language code for the 'to' locale, and YY is the country code for the 'to' locale.

The `aa_BB_ycpmissinglocalizedstrings_xx_YY.properties` file contains entries in the following format:

```
Acceptance_Process=
Add_Service=
Line=
```

Add relevant transaction values to the entries. An example (for French literals) is:

```
Acceptance_Process=Processus d'acceptation
Add_Service=Ajouter service
Line=Ligne
```

4. Save the modified `aa_BB_ycpmissinglocalizedstrings_xx_YY.properties` file in the UTF-8 format.

5. Save the modified file as `aa_BB_ycplocalizedstrings_xx_YY.properties`, where aa is the language code for the 'from' locale', BB is the country code for the 'from' locale, xx is the language code for the 'to' locale, and YY is the country code for the 'to' locale.

6. Run the LocalizedStringReconciler tool from `<INSTALL_DIR>/bin` in IMPORT mode as follows:

   For UNIX:

   ```
   ant.sh -f localizedstringreconciler.xml import -Dsrc=<INSTALL_
   DIR>/installed_data/sscap/components/complete_
   installation/factorysetup/XMLS
   ```
   For windows:

   ```
   ant.cmd -f localizedstringreconciler.xml import -Dsrc=<INSTALL_
   DIR>/installed_data/sscap/components/complete_
   installation/factorysetup/XMLS
   ```

   This inserts the values specified in the properties file into the database.

2.4.1.1 Extending Default Factory-Shipped Translations

The factory default translations can be extended to create custom translations for the localization literals.

To modify the default factory-shipped translations with custom localization literals:

1. Create a new `extn` directory in the `<INSTALL_DIR>/installed_ data/sscap/components/complete_ installation/factorysetup/XMLS/<language>_<country>` directory.

   > **Note:** For example, if `fr_FR` is the factory-shipped translated locale, the `extn` directory should be created in the `<INSTALL_DIR>/installed_ data/sscap/components/complete_ installation/factorysetup/XMLS/fr_FR` directory.

2. Copy the `<baselanguage>_<basecountry>_ycplocalizedstrings_ <language>_<country>.properties` file from the `<INSTALL_ DIR>/installed_data/sscap/components/complete_ installation/factorysetup/XMLS/<language>_<country>` directory to the newly created `extn` folder.

3. Edit the translations in the `properties` file in the `extn` directory:

   - Modify the translations for existing localization literals with the new translations.

   - Add new localization literals and their translations, if needed.

   - Remove any obsolete or unwanted translations that are not overridden for localization literals. This enables you to retain the Sterling Multi-Channel Fulfillment Solution-provided translations during an upgrade.

   > **Note:** It is recommended that you manage the extended translations during an upgrade.

4. Run the LocalizedStringReconciler tool from `<INSTALL_DIR>/bin` in IMPORT mode as follows:

   For UNIX:

   ```
   ant.sh -f localizedstringreconciler.xml import -Dsrc=<INSTALL_
   DIR>/installed_data/sscap/components/complete_
   installation/factorysetup/XMLS
   ```

   For Windows:

```
ant.cmd -f localizedstringreconciler.xml import -Dsrc=<INSTALL_
DIR>/installed_data/sscap/components/complete_
installation/factorysetup/XMLS
```

This tool first inserts the values specified in the properties file present in the `<INSTALL_DIR>/installed_data/sscap/components/complete_installation/factorysetup/XMLS/<language>_<country>` directory into the database.

This entry is then replaced with the values specified in the properties file in the `<INSTALL_DIR>/installed_data/sscap/components/complete_installation/factorysetup/XMLS/<language>_<country>/extn` directory.

2.4.1.2 Switching the Sterling Multi-Channel Fulfillment Solution Base Language

The base language is the language that all translations are in relation to. For example, when a new common code is added, the description is in the base language, and there are translations from this language to other languages.

> **Note:** The base language can only be switched **once**. Switching a base language more than once may result in a loss of data or other potential errors.

By performing this switch the following processes occur:

- Every translation to the desired base language from your current base language is changed to a translation from your desired base language to your current base language.

- Every translation from your current base language to another language is changed to a translation from your desired base language to another language.

- Every localizable column is converted from your current base language to your desired base language.

For example, your Sterling Multi-Channel Fulfillment Solution install has English as a base language, and you have setup translations to French and German. The translations are interpreted as follows:

- Translations from English to French

- Translations from English to German

By switching your base language to French, your translations are interpreted as follows:

- Translations from French to English

- Translations from French to German

> **Note:** In order to successfully perform a base language switch, all localizable descriptions from the current base language to the desired base language must exist. Furthermore, there must be at least one entry in the YFS_ LOCALE table corresponding to the desired base language.

Considerations When Switching the Base Language

The following points need to be taken into consideration prior to performing a base language switch.

- Multiple 'from' strings in a language may translate to the same 'to' string in another language. During the switch, only one of these records is retained, and ambiguous records are removed.

    For example, ABC and AAC in English may both translate to FABC in French. When the base language is switched from English to French, only one record with a `from' string of FABC is retained.

- If a locale does not provide a translation, the Sterling Multi-Channel Fulfillment Solution falls back to the base language.

    For example, there are three locales, en_US, en_GB, and fr_FR. Currently en_US is the base language. fr_FR has full translations, but en_GB only has a few translations, such as *color* to colour. Using the en_GB locale, the en_GB translation for *apple* does not exist, therefore the Sterling Multi-Channel Fulfillment Solution falls back to the en_US translation of *apple*.

When the base language is switched to French, the fall back language becomes French. Now when the en_GB translation of *apple* does not exist, the Sterling Multi-Channel Fulfillment Solution falls back to the fr_FR translation, *pomme.*

It is recommended that *all* translations be provided for *all* locales.

Switching the Base Language

1. To validate that there are no issues in making a base language switch, run the LocalizedStringReconciler tool from `<INSTALL_DIR>/bin` in SWITCHTEST mode as follows:

   For UNIX:

   ```
   ant.sh -f localizedstringreconciler.xml switchtest -Dlanguage=xx -Dcountry=YY
   ```

   For Windows:

   ```
   ant.cmd -f localizedstringreconciler.xml switchtest -Dlanguage=xx -Dcountry=YY
   ```

   where xx and YY are the desired base language and country. For example, fr and FR for French and France, respectively. This simulates the switch and report any errors it finds, without making any changes to the database.

2. To perform the base language switch, run the LocalizedStringReconciler tool from `<INSTALL_DIR>/bin` in SWITCH mode as follows:

   For UNIX:

   ```
   ant.sh -f localizedstringreconciler.xml switch -Dlanguage=xx -Dcountry=YY
   ```

   For Windows:

   ```
   ant.cmd -f localizedstringreconciler.xml switch -Dlanguage=xx -Dcountry=YY
   ```

   where xx and YY are the desired base language and country. For example, fr and FR for French and France, respectively.

2.4.1.3 Full Export to Back-up Existing Localization Literals

To create a full export to back-up existing localization literals:

1. Run the LocalizedStringReconciler tool from `<INSTALL_DIR>/bin` in
   EXTRACT mode as follows:

   For UNIX:

   ```
   ant.sh -f localizedstringreconciler.xml extract -Ddestdir=<INSTALL_
   DIR>/installed_data/sscap/components/complete_
   installation/factorysetup/XMLS
   ```

   For Windows:

   ```
   ant.cmd -f localizedstringreconciler.xml extract -Ddestdir=<INSTALL_
   DIR>/installed_data/sscap/components/complete_
   installation/factorysetup/XMLS
   ```

   This exports the literals that are currently defined in the `YFS_
   LOCALIZED_STRINGS` table.

   To extract existing localized strings for a specific locale, use the
   optional parameters `locale`, and `country`. The usage is as follows:

   For UNIX:

   ```
   ant.sh -f localizedstringreconciler.xml extract -Ddestdir=<INSTALL_
   DIR>/installed_data/sscap/components/complete_
   installation/factorysetup/XMLS -Dlanguage=fr -Dcountry=FR
   ```

   For Windows:

   ```
   ant.cmd -f localizedstringreconciler.xml extract -Ddestdir=<INSTALL_
   DIR>/installed_data/sscap/components/complete_
   installation/factorysetup/XMLS -Dlanguage=fr -Dcountry=FR
   ```

For example, if there are three locales that have been localized for
factory default data in the database, one file for each locale is created in
the following format:

```
<baselanguage>_<basecountry>_ycpdblocalizedstrings_
<language>_<country>_db.properties
```

Each file is saved within the `extn` directory within the respective locale
directory. For instance, for the `fr_FR` locale, the file `en_US_
ycpdblocalizedstrings_fr_FR_db.properties` is created and saved to
the `<INSTALL_DIR>/installed_data/sscap/components/complete_
installation/factorysetup/XMLS/fr_FR` directory.

> **Note:** To re-import the backed-up literals:
>
> Run the LocalizedStringReconciler tool from `<INSTALL_DIR>/bin` in IMPORT mode with the file for which you have exported the literals as follows:
>
> For UNIX: `ant.sh -f localizedstringreconciler.xml import -Dsrc=<INSTALL_DIR>/installed_data/sscap/components/complete_installation/factorysetup/XMLS`
>
> For Windows: `ant.cmd -f localizedstringreconciler.xml import -Dsrc=<INSTALL_DIR>/installed_data/sscap/components/complete_installation/factorysetup/XMLS`

2.4.1.4 Deleting Unused Localization Literals

To delete unused localization literals, run the LocalizedStringReconciler tool from `<INSTALL_DIR>/bin` in EXPORT mode as follows:

For UNIX:

```
ant.sh -f localizedstringreconciler.xml export -Ddestdir=<INSTALL_
DIR>/installed_data/sscap/components/complete_
installation/factorysetup/XMLS -Ddelete=true
```

For Windows:

```
ant.cmd -f localizedstringreconciler.xml export -Ddestdir=<INSTALL_
DIR>/installed_data/sscap/components/complete_
installation/factorysetup/XMLS -Ddelete=true
```

This deletes the unused literals, while exporting the literals that are currently defined in the `YFS_LOCALIZED_STRINGS` table.

2.5 Directory Structure and Filenames

The Sterling Multi-Channel Fulfillment Solution directory structure must be maintained based on the language and country localization. File names are limited to one locale and can only contain characters `a` through `z`, `0` (zero) through `9`, and the underscore (_).

Likewise, since many files are created using a transaction ID, this rule also applies to transaction names and transaction IDs. Similarly, you should adhere to these limitations when localizing the template filename for transaction events (`TransactionID.On_SUCCESS.xml`).

The Sterling Multi-Channel Fulfillment Solution follows the hierarchy of lookups given below:

1. When a certain locale is selected in the console it searches for some files with that particular suffix specified in the locale. For example, if the locale ja_JP is chosen in the console, the system searches for files such as `ycpapibundle_ja_JP.properties`, `validation_ja_JP.properties`, `sapphire_ja_JP.css`, and so forth.

2. If these files are not found, the systems looks at the default locale of the operating system that the application server is running. If it is en_US, the system tries to look for the above files with en_US suffix. For example, `ycpapibundle_en_US.properties`.

3. Only if the above two steps fails, the default files (for example, `ycpapibundle.properties`) are used.

The method of defaulting to the standard files occurs for the files that can have the locale suffix such as `alertmessages.js`, `sscapiconsbe.jar`, and so forth.

2.6 User Interface Themes

The user interface theme files specify screen colors and display fonts to use. Display fonts are dependent on what languages need to be supported. Some fonts may not support all languages. For example, Tahoma (the Sterling Multi-Channel Fulfillment Solution default font) does not support Japanese; for best results when localizing the Sterling Multi-Channel Fulfillment Solution to Japanese, use the MS Gothic font. When setting up a theme, choose a font that displays the specific language you need. For example, when setting up a Japanese locale, customize the theme to use a font that displays Japanese characters such as Hiragana.

> **Note:** If you use a font that is bigger than the Sterling Multi-Channel Fulfillment Solution default font (Tahoma), it may be necessary to customize the `<INSTALL_DIR>/repository/datatypes/datatypes.xml` file to increase the user interface size of data types that are used for input fields in the Sterling Multi-Channel Fulfillment Solution Consoles. In particular, the user interface size of the "Date" data type should be increased.

2.6.1 Themes

When localizing your Sterling Multi-Channel Fulfillment Solution Configurator user interface themes, you modify the theme-specific XML file. When localizing your Sterling Multi-Channel Fulfillment Solution Consoles user interface themes, you modify the theme-specific CSS file. For example, the following files must be localized for themes:

`<INSTALL_DIR>/repository/xapi/template/merged/resource/<theme>.xml` and `<INSTALL_DIR>/repository/eardata/platform/war/css/<theme>.css`

These files are localized by appending the language and country codes in the file name. For example, if you are using the sapphire theme in a French locale, you must localize the following files:

`<INSTALL_DIR>/repository/xapi/template/merged/resource/sapphire_fr_FR.xml`
`<INSTALL_DIR>/repository/eardata/platform/war/css/sapphire_fr_FR.css`

The following themes are distributed with the Sterling Multi-Channel Fulfillment Solution:

- Earth (`<INSTALL_DIR>/repository/xapi/template/merged/resource/earth.xml` and `<INSTALL_DIR>/repository/eardata/platform/war/css/earth.css`)

- Jade (`<INSTALL_DIR>/repository/xapi/template/merged/resource/jade.xml` and `<INSTALL_DIR>/repository/eardata/platform/war/css/jade.css`)

- Sapphire (`<INSTALL_ DIR>/repository/xapi/template/merged/resource/sapphire.xml` and `<INSTALL_ DIR>/repository/eardata/platform/war/css/sapphire.css`)

To localize a theme:

1. Copy the `<INSTALL_ DIR>/repository/xapi/template/merged/resource/<theme>.xml` file and save it as `<INSTALL_ DIR>/repository/xapi/template/merged/resource/<theme>_ <language>_<country>.xml`.

2. Copy the `<INSTALL_ DIR>/repository/eardata/platform/war/css/<theme>.css` file and save it as `<INSTALL_ DIR>/repository/eardata/platform/war/css/<theme>_ <language>_<country>.css`.

   You must edit the `<INSTALL_ DIR>/repository/eardata/platform/war/css/<theme>_ <language>_<country>.css` file to change the display font for the Sterling Multi-Channel Fulfillment Solution Consoles. In addition, the font name and size for the graph displayed in Inventory Summary screen in the Inventory Console is configured in the `<INSTALL_ DIR>/repository/xapi/template/merged/resource/<theme>_ <language>_<country>.xml` file.

   For example, in the default `sapphire.xml` file, the graph font is configured as:

   ```
   <!-- Font for Inventory Graphs(Axis Titles & Lables) -->
   <Font Name="InvGraphFont" FontName="Tahoma" FontSize="12" />
   <!-- Font for Inventory Graphs -->
   ```

To localize double-byte languages such as Japanese, Sterling Commerce recommends that you edit the `<INSTALL_ DIR>/repository/xapi/template/merged/resource/<theme>_ <language>_<country>.xml` file to use either the MS UI Gothic or SimSun font as follows:

```
<!-- Font for Inventory Graphs(Axis Titles & Lables) -->
   <Font Name="InvGraphFont" FontName="simsun" FontSize="12" />
   <!-- Font for Inventory Graphs -->
```

3. Rebuild the `resources.jar` by running the following command from the `<INSTALL_DIR>/bin` directory:

   `/deployer.sh -t resourcejar`

4. If you are using Weblogic or WebSphere, rebuild the EAR.

> **Note:** If your application server is running on Unix, the valid fonts you can use are stored in the `<JAVA_HOME>/jre/lib/font_properties.<file.encoding>` file.

2.7 Literals

All Sterling Multi-Channel Fulfillment Solution components use a common resource bundle that contains literals displayed on the screens. The Sterling Multi-Channel Fulfillment Solution enables you to customize and localize resource bundles as needed.

In addition, literals used in customized screens have their own resource bundle and should also be considered during the localization process. For more information about localizing your customizations to the Sterling Multi-Channel Fulfillment Solution, see the *Sterling Multi-Channel Fulfillment Solution Customization Guide*.

> **Note:** Literals cannot be localized in the following places:
> - Condition Builder
> - Order/Shipment Monitor
> - Hard-coded literals in APIs

For a complete list of resource bundle literals, along with the screens on which those literals appear, see the `<DocumentationCD>/resource_mapping.htm` file.

2.7.1 Resource Bundles

Sterling Commerce always releases complete resource bundles in the `ycpapibundle.properties` and `yscpapibundle.properties` files with the localized versions of the Sterling Multi-Channel Fulfillment Solution. Incremental updates are not provided. If you localize the Sterling

Multi-Channel Fulfillment Solution, it is your responsibility (or that of your third-party localization company) to compare and validate the differences between the resource bundles shipped with the product to those you have localized.

The resource bundles of the Sterling Multi-Channel Fulfillment Solution are located in the `<INSTALL_DIR>/resources/ycpapibundle.properties` and `<INSTALL_DIR>/resources/yscpapibundle.properties` files.

To localize the resource bundles:

1. Copy the `<INSTALL_DIR>/resources/ycpapibundle.properties` file and save it as `<INSTALL_DIR>/resources/ycpapibundle_<language>_<country>.properties`.

2. Copy the `<INSTALL_DIR>/resources/yscpapibundle.properties` file and save it as `<INSTALL_DIR>/resources/yscpapibundle_<language>_<country>.properties`.

3. Each resource bundle contains a `<key>=<value>` pair where key is the resource key and value is the literal displayed for the corresponding locale. Replace `<value>` with the translated value.

   - When localizing menus in the Configurator, by default the accelerator key is the first character in a menu item. To specify any other character to be the accelerator key, insert an ampersand (`&`) just before that character.

   - When localizing the console UI, be aware that changing the height or width of the text in the application may affect the layout of the screens. It may be necessary to customize certain screens to achieve optimal layout after the other localization steps are complete. For example, if the resource bundle contains translated literals that are lengthy, you may need to increase the width of the screen in order to accommodate the larger size of the translated literal.

   - Some of the literals that need to be translated in the resource bundles contain data place holders. These data place holders indicate that the literal is displayed with one or more data values within the literal. For example, the application displays the error message "Priority should be greater than X" where X could be any number. Since the location of the "X" within the literal can be different for different languages, the resource bundle uses a place

holder that can be placed anywhere in the literal during translation. The resource bundle entry looks like this:

```
PRIORITY_ERROR_MESSAGE=Priority should be greater than {0}
```

Notice how the "{0}" place holder indicates where the dynamic data value appears in the literal. This "{0}" can be placed any where in the literal. For example,

```
PRIORITY_ERROR_MESSAGE=A number greater than {0} should be entered
```

or

```
PRIORITY_ERROR_MESSAGE={0}: Priority entered should be greater than this
```

are valid possibilities. This gives you the flexibility to translate the literal in any way that the language dictates. Note that multiple place holders may appear in the literal as well. For example, {0}, {1}, {2}, and so forth. Each place holder must exist somwhere in the corresponding translated literal.

- When using literals that contain data place holders, you cannot use single quotation marks. If a single quotation mark is used in conjunction with a place holder, the single quotation mark is not displayed and the place holder is not replaced with its replacement value. In order to avoid this situation, enter two single quotes wherever a single quote is required. For example, the following is invalid:

```
PRIORITY_ERROR_MESSAGE=The primary organization's name is {0}
```

but the following is valid:

```
PRIORITY_ERROR_MESSAGE=The primary organization''s name is {0}
```

- Files should be returned in native format with UTF-8 encoding.

- Properties should be returned in escaped Unicode format with UTF-8 encoding.

4. The default font used is Tahoma. Therefore if you want to display or type Unicode characters, you should localize the theme. This is done by changing the font to Unicode in the theme-specific XML files.

5. Save the modified file. If the file is in UTF-8 format, convert it to ASCII by running the native2ascii command as follows:

```
native2ascii -encoding UTF-8 <source file> <target file>
```

The file should be returned in the following format:

```
<filename>_<2 letter code for language as given by ISO 639>_<2 letter code
for territory as given by ISO 3166>.<file extension>
```

For example, `ycpapibundle.properties` should be returned as
`ycpapibundle_fr_FR.properties` and `yscpapibundle.properties`
should be returned as `yscpapibundle_fr_FR.properties`.

6. If you are customizing the Sterling Multi-Channel Fulfillment Solution,
   save the extended resource bundles as `<INSTALL_
   DIR>/resources/extn/extnbundle_<language>_
   <country>.properties`.

   For example, `ycpapibundle.properties` should be saved as
   `ycpapibundle_fr_FR.properties` and `yscpapibundle.properties`
   should be saved as `yscpapibundle_fr_FR.properties`.

7. For extended tag attributes add the following bundle entry in
   `extnbundle.properties` for each extended tag attribute:

   ```
   Item_Tag_<TagName>=<Tag Name>
   ```

8. Create the resourcejar using the ./deployer.sh -t resourcejar.

9. If you are using Weblogic or WebSphere, rebuild the EAR.

   > **Note:** The Custom_Code_Prefix and Custom_Code_
   > Postfix properties in the `ycpapibundles.properties` file
   > are used to prefix or append identifying literals or
   > extensions to your newly created custom transaction IDs,
   > supply types, demand types, or document types. When
   > you create a new transaction ID, supply type, demand
   > type, or document type, the value specified for these
   > properties is prefixed or appended to each of these types
   > of literals when displayed in the user interface. The default
   > value for the Custom_Code_Prefix is "" (blank) and the
   > Custom_Code_Postfix is ".ex". You may change this value
   > if it does not suit your needs.

2.8 Date and Number Validation

Date and number validations are performed using JavaScript. By default, the Sterling Multi-Channel Fulfillment Solution provides validation for the `en_US` locale (English for the United States).

2.8.1 Date and Time Validation

Dates can be stored in a standard format but displayed according to the required format. If the date is entered on one of the screens it has to be verified by some means. Date format is specified in the Sterling Multi-Channel Fulfillment Solution Configurator Locale Details screen. For more information, see the *Sterling Multi-Channel Fulfillment Solution Platform Configuration Guide*.

To localize date and time validation:

1. Save the `<INSTALL_ DIR>/repository/eardata/platform/war/yfcscripts/Validation. js` file as `Validation_<language>_<country>.js` (in UTF-8 encoding format) and make modifications to the new file as indicated in the following steps.

2. Ensure that date and time values match the entries specified in the Locale fields in the Sterling Multi-Channel Fulfillment Solution Configurator. For detailed information about the Locale fields and their suggested syntax, see the *Sterling Multi-Channel Fulfillment Solution Platform Configuration Guide*.

3. Change the [`yfcDateFormat`] (MM/dd/yyyy), [`yfcTimeFormat`] (hh:mm:ss) and [`yfcDateTimeFormat`] (MM/dd/yyyy hh:mm:ss) variables to contain the correct date and time format.

   These date and time formats are according to the United States English version. When you create a `Validation.js` file for another language, these formats change accordingly.

Table 2–1   Date and Time Variable Formats

Date Element	Description
yyyy	Four-digit year - for example, 2002
MM	Two-digit month - for example, 05 for May

Table 2–1   Date and Time Variable Formats

Date Element	Description
dd	Two-digit day - for example 01 for the first day of the month
mm	Minutes
HH	Hours
ss	Seconds

4. Localize the popup messages and calendar formats as described in Section 2.10, "Calendar and Message Popups in the Console".

> **Note:**   It is recommended that you use the hour format 'HH' as opposed to 'hh', whenever you localize the Sterling Multi-Channel Fulfillment Solution. 'HH' signifies the 24 hour format, which is the only format supported by the Sterling Multi-Channel Fulfillment Solution.

2.8.2 Number Validation

By default, Java displays the localized number format based on what is specified for `<language>_<country>` in the Sterling Multi-Channel Fulfillment Solution Configurator. In order for the Sterling Multi-Channel Fulfillment Solution UI to validate numeral values, it reads in values as specified in the `Validation_<language>_<country>.js` file. This means that the validation JavaScript file must contain validation logic that is specific to the number format to be used for validation for each locale.

To localize number validation:

1. Save the `<INSTALL_DIR>/repository/eardata/platform/war/yfcscripts/Validation.js` file as `Validation_<language>_<country>.js` (in UTF-8 encoding format).

2. Change the decimal separator and grouping separator as needed. The grouping separator indicates how to format numbers visually; it has no impact on the actual value of the number.

For example, en_US uses a comma (",") and fr_FR uses a non-breaking space (" "). When specifying the separator, use the Unicode literal.

For example, the non-breaking character for fr_FR would be specified as shown below:

```
yfcGroupingSeparator = "\u00a0";
```

3. Localize the exception messages for invalid number format as described in Section 2.7, "Literals".

4. Run the resource deployer from ./deployer.sh -l info -t resourcejar.

5. If you are using Weblogic or WebSphere, rebuild the EAR.

2.9 Templates

Among the templates used by the Sterling Multi-Channel Fulfillment Solution, you can localize e-mail and exception templates.

2.9.1 E-mail Templates

You can store e-mail templates using any character encoding format, but the encoding format must be set by configuring the yfs.email.template.encoding property in the <INSTALL_ DIR>/properties/customer_overrides.properties file.

For additional information about modifying properties and the customer_ overrides.properties file, see the *Sterling Multi-Channel Fulfillment Solution Installation Guide*.

2.9.2 Exception Alert Templates

Exception alert templates enable you to supply additional text to alerts raised. This enables you to make error message more descriptive and easy to understand. They also provide a means of supplying a hyperlink to the resolution screens from the Alert Console.

For example: for any alert created for an order, shipment, or load document type, a hyperlink is created and displays in the "Created For" column on the Alert List screens. You localize the literals displayed in this column by translating them in the DefaultListTemplate.xsl file located in your <INSTALL_

`DIR>/repository/xapi/template/merged/exception_console` directory.

You can store exception alert templates using any character encoding format, but the encoding format must be configured in the `yfs.file.encoding` property in the `<INSTALL_ DIR>/properties/customer_overrides.properties` file. If these properties are not explicitly set in the `<INSTALL_ DIR>/properties/customer_overrides.properties` file, the Sterling Multi-Channel Fulfillment Solution uses UTF-8 character encoding.

For additional information about modifying properties and the `customer_ overrides.properties` file, see the *Sterling Multi-Channel Fulfillment Solution Installation Guide*.

2.10 Calendar and Message Popups in the Console

Calendar and message popup windows contain literals that must be localized.

To localize date formats for calendar popup windows in the Console:

> **Note:** Only the literals in calendar popup windows can be localized. The order of the days cannot be localized. Therefore, the week always starts with Sunday.

1. Copy the `<INSTALL_ DIR>/repository/eardata/platform/war/common/alertmessages.j s` file and save it as `<INSTALL_ DIR>/repository/eardata/platform/war/common/alertmessages_ <language>_<country>.js` (in UTF-8 encoding format).

2. Copy the `<INSTALL_ DIR>/repository/eardata/yantra/war/ysc/scripts/yscalertmess ages.js` file and save it as `<INSTALL_ DIR>/repository/eardata/yantra/war/ysc/scripts/yscalertmess ages_<language>_<country>.js` (in UTF-8 encoding format).

3. Translate the date-related literals using this following as described below:

- `monthArray` - contains the series of literals to use for months of the year

- `weekdayList` - contains the series of literals to use when displaying the literals for days of the week

- `weekdayArray` - contains the series of literals to use to when displaying the shortened literals for days of the week

- `todayString` - displays the word "Today"

4. If you are using Weblogic or WebSphere, rebuild the EAR.

To localize exception messages:

1. Copy the `<INSTALL_ DIR>/repository/eardata/platform/war/common/alertmessages.j s` file and save it as `<INSTALL_ DIR>/repository/eardata/platform/war/common/alertmessages_ <language>_<country>.js,` (in UTF-8 encoding format).

2. Copy the `<INSTALL_ DIR>/repository/eardata/yantra/war/ysc/scripts/yscalertmess ages.js` file and save it as `<INSTALL_ DIR>/repository/eardata/yantra/war/ysc/scripts/yscalertmess ages_<language>_<country>.js` (in UTF-8 encoding format).

3. Edit your exception messages in the files as needed.

2.11 Images

Images and icons are stored in JAR files. A separate JAR file can be used for each locale. The Sterling Multi-Channel Fulfillment Solution Configurator and Sterling Multi-Channel Fulfillment Solution Consoles both use their own mechanisms for reading in images.

2.11.1 Application Consoles Images

To localize images used in the Sterling Multi-Channel Fulfillment Solution Consoles:

1. Copy the `<INSTALL_ DIR>/repository/eardata/yantra/war/yfscommon/sscapiconsbe.j ar` file to a new `<INSTALL_ DIR>/repository/eardata/yantra/war/yfscommon/sscapiconsbe_ <language>_<country>.jar` file.

2. Within your custom JAR file, add or remove images as needed.

3. If you are customizing and localizing the Sterling Multi-Channel Fulfillment Solution, your custom images are read in from the `<INSTALL_ DIR>/repository/eardata/yantra/extn/war/icons/sscapiconsbe. jar` file. Therefore, you must copy the images from the `<INSTALL_ DIR>/repository/eardata/yantra/extn/war/icons/sscapiconsbe. jar` file and make the necessary locale-specific changes in the images.

4. Save the changed images in `<INSTALL_ DIR>/repository/eardata/yantra/extn/war/icons/sscapiconsbe_ <language>_<country>.jar` file.

> **Note:** The warning message that displays when you click on the Sterling Multi-Channel Fulfillment Solution icon in the top right-hand corner of the Sterling Multi-Channel Fulfillment Solution Consoles user interface, is text included in an image file. You can localize this message by following the instructions in this section to localize the `/console/icons/about_box_back.gif` file.

2.11.2 Sterling Multi-Channel Fulfillment Solution Configurator Images

To localize images used in the Sterling Multi-Channel Fulfillment Solution Configurator:

1. Copy the images from the `<INSTALL_DIR>/repository/eardata/yantra/war/yfscommon/sscapicons.jar` file and make the necessary locale-specific changes to the images.

2. Save the changed images in `<INSTALL_DIR>/repository/eardata/yantra/war/yfscommon/sscapicons_<language>_<country>.jar` file.

> **Note:** The warning message that displays at the bottom of the Sterling Multi-Channel Fulfillment Solution Configurator About box when you select Help > About from the Sterling Multi-Channel Fulfillment Solution Configurator menu is text included in an image file. You can localize this message by following the instructions in this section to localize the `/com/yantra/ycp/ui/icons/about.gif` file.

2.12 Localizing the Greex File

A Greex file, also known as advanced XML file, contains the advanced XML condition or Greex Rule that is defined by a user. By localizing the Greex file, you can localize an advanced XML condition or Greex Rule.

To localize the Greex file:

1. Create the BundleResolver class and implement the following methods within the class:

- getString(String key) method and return the localized strings.

    For example, if you want to localize the Greex file using properties files:

```
public class MyBundleResolver implements BundleReslover
{
    Properties prop = new Properties();
      public MyBundleResolver()
      {
        //read and initialize the property file
      }
      public  String getString(String key)
      {
          Return prop.getProperty(key);
      }
 }
```

2. Register the BundleResolver class with the GreexContext using registerBundle() method. For example:

```
public class MyApp
{
  GreexContext  ctx = new GreexContext();
  ctx.registerBundle(new MyBundleResolver())
}
```

3

Localizing the Sterling Rich Client Application

This chapter provides steps for localizing the Sterling Rich Client application for both languages that are not yet provided and languages that are supported by the Sterling Multi-Channel Fulfillment Solution out of the box.

3.1 Before You Begin

Before you begin localizing a Sterling Rich Client application, keep the information in this section in mind.

International Standards Organization (ISO) Codes

Sterling Rich Client Platform uses locale-related codes as specified by the ISO. Throughout this guide, locale is represented by <language>_<country>. Locale is comprised of <language>, a 2-character lower-case ISO-639 code, and <country>, a 2-character upper-case ISO-3166 code.

> **Note:** For complete steps to localize the Sterling Rich Client Application, see the Implementation Guide specific to your PCA.

3.2 User Interface Themes

The user interface theme files specify screen colors, display fonts, and images to use. Display fonts are dependent on what languages need to be supported. Some fonts may not support all languages. When setting

up a theme, choose a font that displays the specific language you need or choose an image that displays for a particular locale. For example, when setting up a Japanese locale, customize the theme to use a font that displays Japanese characters such as Hiragana.

3.2.1 Themes

When localizing your Sterling Rich Client user interface themes, you must localize the Sterling RCP-specific theme files and the Sterling Rich Client application-specific theme files.

The Sterling RCP contains `com.yantra.yfc.rcp.common_<theme_ name>.ythm` theme files located in the `com.yantra.yfc.rcp.common_ <version>` directory within the `<INSTALL_ DIR>/repository/rcp/rcpclient/com.yantra.yfc.rcp_<version>.zip` file.

The Sterling Rich Client application contains `<Plug-in_ id>_<theme_ name>.ythm` theme files located in the `plugins/com.yantra.pca.ycd.rcp_<version>` directory within the `<INSTALL_DIR>/repository/rcp/rcpclient/com20.zip` file.

The theme file is localized by appending the language and country codes in the file name. For example, if you are localizing the sapphire theme file in a French locale of the Sterling RCP, modify the com.yantra.yfc.rcp.common_sapphire.ythm file as:

`com.yantra.yfc.rcp.common_sapphire_fr_FR.ythm`

To localize the sapphire theme file in a French locale of the Sterling Rich Client application, modify the <Plug-in_id>_sapphire.ythm file as:

`<Plug-in_id>_sapphire_fr_FR.ythm`

For example, if you are localizing the sapphire theme file in a French locale of the Sterling COM PCA, modify the com.yantra.pca.ycd.rcp_ sapphire.ythm file as:

`com.yantra.pca.ycd.rcp_sapphire_fr_FR.ythm`

3.2.1.1 Types of themes

There are two types of themes: Fonts and Images.

Fonts

The theme entries for Font theme are specified in the Font, ForegroundColor, BackgroundColor elements under the ThemeEntry element. There are three types of theme entries for the Font theme:

1. Font—The Font element contains the data for font's height, name, and style. Table 3–1 describes the attributes of the Font element.

Table 3–1   Font Element Attribute List

Attribute	Description
Height	Enter the height of the font.
Name	Enter the font name. For example, For example, Tahoma, Arial, and so forth.
Style	Enter the font style. For example, NORMAL, BOLD, and so forth.

2. ForegroundColor and BackgroundColor—The ForegroundColor element describes the color of the text that appears. The BackgroundColor element describes the background color in which the text appears. Available attributes for both elements are Red, Blue, and Green – standard RGB values. The RGB values have to be entered in the decimal color code (0 - 255).

Images

The theme entries for Image theme are specified in the Image element under the ThemeEntry element. In the Image element, specify the path for the locale specific image to be used in the theme in the Path attribute.

> **Note:**   To store the locale specific images:
>
> 1. Create an icons folder in the <INSTALL_ DIR>/repository/rcp/extn directory.
> 2. Put all the locale specific images in this newly created <INSTALL_DIR>/repository/rcp/extn/icons folder.

The sample data from the <Plug-in_ id>_<theme_name>.ythm file is given below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Theme id="sapphire">
  <ThemeEntry Name="Label">
     <Font Height="9" Name="Tahoma" Style="NORMAL"/>
     <ForegroundColor Blue="0" Green="0" Red="0"/>
     <BackgroundColor Blue="255" Green="255" Red="255"/>
  </ThemeEntry>
  <ThemeEntry Name="CComboEditor">
     <Font Height="8" Name="Tahoma" Style="NORMAL"/>
     <BackgroundColor Blue="255" Green="255" Red="255"/>
     <ForegroundColor Blue="0" Green="0" Red="0"/>
  </ThemeEntry>
  <ThemeEntry Name="Text">
     <Font Height="8" Name="Tahoma" Style="NORMAL"/>
     <ForegroundColor Blue="0" Green="0" Red="0"/>
     <BackgroundColor Blue="255" Green="255" Red="255"/>
  </ThemeEntry>
<ThemeEntry Name="Composite">
     <Font Height="10" Name="Tahoma" Style="NORMAL"/>
     <BackgroundColor Blue="255" Green="255" Red="255"/>
     <ForegroundColor Blue="198" Green="146" Red="140"/>
</ThemeEntry>
<ThemeEntry Name="DateLookup">
     <Image Path="/icons/calendar.gif"/>
</ThemeEntry>
<ThemeEntry Name="HeaderTriangle">
     <Image Path="/icons/header_triangle.jpg"/>
</ThemeEntry>
```

3.2.1.2 Localizing a Theme

To localize the client application theme files:

1. To localize a Sterling RCP theme file, go to the `<INSTALL_ DIR>/repository/rcp/rcpclient/` directory. Unzip the `com.yantra.yfc.rcp_<version>.zip` file. All Sterling RCP-specific theme files are located in the `<unzip_ dir>/com.yantra.yfc.rcp.common_<version>` directory.

   where <unzip_dir> refers to the directory where you have unzipped the file.

   The theme files are in the following format:

   com.yantra.yfc.rcp.common_<theme_name>.ythm

where <theme_name> refers to the name of a particular theme. By default, Sterling RCP provides three different theme files named sapphire, jade, and earth.

Copy the com.`yantra.yfc.rcp.common_<theme_name>.ythm` file to the `resources` directory of the `extensions` folder that you created. For more information about creating an extensions folder, see the *Sterling Multi-Channel Fulfillment Solution Installation Guide*.

2. To localize a theme file of the Sterling Rich Client application, go to the `<INSTALL_DIR>/repository/rcp/rcpclient/` directory.

Unzip the `<PCA_APPLICATION_ID_VERSION>.zip` file and go to the following directory:

`<unzip_dir>/plugins/<Plug-in_id>_<version>`

where `<unzip_dir>` refers to the directory where you have unzipped the `<PCA_APPLICATION_ID_VERSION>.zip` file.

In the `<Plug-in_id>_<version>` directory, there is a `<PCA_ APPLICATION_ID_VERSION>.jar` file. The theme file is in the root directory of this jar and is called `<Plug-in_id>_<theme_name>.ythm`.

Copy the <Plug-in_id>_<theme_name>.ythm file to the resources directory of the extensions folder that you created.

For example, if you are localizing a theme file of the Sterling COM PCA, unzip the `<INSTALL_ DIR>/repository/rcp/rcpclient/com20.zip` file.

Go to the following directory: `<unzip_ dir>/plugins/com.yantra.pca.ycd.rcp_<version>`

where `<unzip_dir>` refers to the directory where you have unzipped the `com20.zip` file.

In the `com.yantra.pca.ycd.rcp_<version>` directory, there is a `com20.jar` file. The theme file is in the root directory of this jar and is called `com.yantra.pca.ycd_<theme>.ythm`.

Copy the `com.yantra.pca.ycd_<theme>.ythm` file to the `resources` directory of the `extensions` folder that you created.

3. Rename the `<Plug-in_ id>_<theme_name>.ythm` files to `<Plug-in_ id>_<theme_name>_<locale_name>.ythm` and `com.yantra.yfc.rcp.common_<theme_name>.ythm` files to `com.yantra.yfc.rcp.common_<theme_name>_<locale_name>.ythm`.

> The `<locale_name>`=lang_cc where `lang` refers to language code and `cc` refers to the country code. For example, `com.yantra.pca.ycd_ sapphire_en_US.ythm`.

4. Modify the theme entries in the theme file based on the user's locale. For more information on theme entries in the theme file, see Section 3.2.1.1, "Types of themes".

5. Create a new icons folder and store all images or icons, which you want to localize in this folder. Now, copy the icons folder to the `<RCP_ EXTN_FOLDER>/resources` directory.

6. If you are using Weblogic or WebSphere, rebuild the EAR.

Note: For information on localizing theme files for PCAs, see the Implementation Guide specific to your PCA.

3.3 Literals

All the Sterling Rich Client applications use a resource bundle that contains literals or text displayed on the screens. The Sterling RCP enables you to customize and localize resource bundles based on user's locale.

In addition, literals used in customized screens have their own resource bundle and should also be considered during the localization process.

3.3.1 Resource Bundles

All Sterling Rich Client application plug-ins contain bundle files named as `<Plug-in_id>_<name>.properties`. The bundle file contains `<key>-<value>` pairs, specify the resource such as control text, string, and so forth that needs to be localized in the <Key> and the localized string or text in the <value>.

`<key>=<value>`

`my_name=Sterling Rich Client Application`

The resource bundles of a Sterling Rich Client application are located in the application's zip file.

For example, the resource bundle for the Sterling COM PCA is located within the `<INSTALL_DIR>/repository/rcp/rcpclient/com20.zip` file.

Unzip `com20.zip` file, and go to the following directory:

`<unzip_dir>/plugins/com.yantra.pca.ycd.rcp_<version>`

where `<unzip_dir>` refers to the directory where you have unzipped the `com20.zip` file.

Within this directory, there is a `com20.jar` file. The bundle files are in the root directory of this jar and are called `com.yantra.pca.ycd_bundle.properties` and `com.yantra.pca.ycd_Messages_bundle.properties`.

To localize the resource bundles:

1. Copy the `<Plug-in_id>_<name>.properties` file from the Sterling Rich Client application plug-in's directory of the application which you want to localize and also from the Sterling RCP plug-in directory to the `resources` directory of the extensions folder that you created.

    > **Note:** Because every Sterling Rich Client application is dependent on the Sterling RCP plug-in, whenever you want to localize a Sterling Rich Client application, you must modify the `<Plug-in_id>_<name>.properties` file of both Sterling RCP plug-in and Sterling Rich Client application plug-in directories. The Sterling RCP plug-in directory is located in the `<INSTALL_DIR>/repository/rcp/rcpclient/com.yantra.yfc.rcp_<version>.zip` file. To localize the Sterling COM PCA bundle file, follow the instructions provided in the example below.

    For example, if you are localizing the bundle file of the Sterling COM PCA, the `<INSTALL_DIR>/repository/rcp/rcpclient/com20.zip` file will contain the Sterling COM PCA plug-in directory called `com.yantra.pca.ycd.rcp_<version>` directory.

    Unzip `com20.zip` file, and go to the following directory:

    `<unzip_dir>/plugins/com.yantra.pca.ycd.rcp_<version>`

where `<unzip_dir>` refers to the directory where you have unzipped the `com20.zip` file.

Within this directory, there is a `com20.jar` file. The bundle file is in the root directory of this jar and is called `com.yantra.pca.ycd_<name>.properties`.

Therefore, copy the <Plug-in_id>_<name>.properties file from both the Sterling COM PCA plug-in directory as well as from the Sterling RCP plug-in directory to the resources directory of the extensions folder that you created. For more information about creating the extensions folder, refer to the *Sterling Multi-Channel Fulfillment Solution Installation Guide*.

2. Rename the `<Plug-in_id>_<name>.properties` files to `<Plug-in_id>_<name>_<locale_name>.properties`. The `<locale_name>=lang_cc` where `lang` refers to language code and `cc` refers to the country code. For example, `com.yantra.pca.ycd_bundle_en_US.properties`.

3. Each resource bundle contains a <key>=<value> pair where key is the resource key and value is the literal displayed for the corresponding locale. Replace the <value> with the translated value.

> **Note:** By default Sterling RCP localizes:
> - Text on Labels
> - Table Column names
> - Descriptions in Combo Boxes
> - Text on Buttons
> - Tab Folder items
> - Groups names
>
> Sterling RCP does not localize text in the text boxes and keys used for identification such as ItemId or a resource key.

The sample bundle entries from the `*.properties` file are given below:

```
Credit_Card_#=Credit Card
View_Details=View &Details
```

```
Customer_Address=Customer Address
Save=&Save
Ship_To_Address=Ship To Address
Address=Address
Close=&Close
```

where entries on the left represent the resource key and entries on the right represent the translated value that is displayed for each control, text, or string based on the user's locale.

4. If you want to get the localized value for any key use the following method:

```
YRCPlatformUI.getString(String bundleKey);
```

It returns the localized string as the output.

> **Note:** For information on localizing resource bundles for PCAs, see the Implementation Guide specific to your PCA.

3.3.1.1 Eclipse Resource Bundles

To localize the eclipse platform resource bundles:

1. Modify the Sterling Rich Client application's `*.ini` file to provide the appropriate program arguments to use the language pack. You can find the `*.ini` file for the Sterling Rich Client application in the `<INSTALL_DIR>/bin/rcpdrop/<OPERATING_SYSTEM>/<PCA_DIR>/` directory.

where `<INSTALL_DIR>` refers to the directory where you have installed the Sterling Multi-Channel Fulfillment Solution.

For example, if you want to run the Sterling COM PCA in debug mode, edit the `<INSTALL_DIR>/bin/rcpdrop/<OPERATING_SYSTEM>/com/com.ini` file.

2. In the `*.ini` file, add any one of the following arguments:

- Program arguments

  ```
  -nl
  <locale_code>
  ```

- VM arguments

  ```
  -Duser.language=<language_code>
  ```

  Note: You must enter the program arguments before the VM arguments. Also, make sure that the program arguments are separated by a line break.

A

Localizable XML Attributes

The Table A–1, "Localizable Factory Setup XML Attributes" table below enables you to determine the XML file name associated with the specific XML attributes. This table lists the factory default XML attributes that can be localized along with the associated database table name. The full name of the XML file can be derived from the database table name using the following procedure.

To derive a list of XML files:

1. Find the database table that corresponds with the XML attribute you want to translate.

2. Using the database table name, append ".xml" to it and prepend the appropriate prefix for the module to which it belongs. Module prefixes are as follows:

   - BI—Business Intelligence

   - INV—Inventory Synchronization

   - OMD—Order Management, Supply Collaboration, and Reverse Logistics

   - OMP—Order Management Platform

   - OMR—Order Management Returns

   - OMS—Order Management Services

   - RDT—Rapid Deployment

   - REF—Reference implementation

   - VAS—Value Added Services

   - WMS—Warehouse Management Services

- YCM—Product Management
- YCP/PLT—Platform
- YCS—Carrier Service
- YDM—Delivery Management
- YSC—Supply Chain

For example, the Actionname attribute corresponds with the YFS_ ACTION database table, which uses the YCP_YFS_ACTION.xml file.

Table A–1   Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_ACTION	Actionname
YFS_ACTIVITY	Description
YFS_ACTIVITY_CONSTRAINT	ConstraintDescription
YFS_ADAPTER	AdapterDescription
YFS_ADJUSTMENT_REASON	Description
YFS_ALLOCATION_RULE	Description
YFS_ANSWER_OPTION	Description
YFS_ATP_RULES	AtpRuleName
YFS_BARCODE_TRANSLATION	Description
YFS_BASE_ACTIVITY_GROUP	ActivityGroupName
YFS_BASE_DOCUMENT_ PARAMS	Description
YFS_BASE_EVENT	EventName
YFS_BASE_PACK_DO_NOT_MIX	FieldNameDesc
YFS_BASE_PROCESS_TYPE	ProcessTypeName
YFS_BASE_PURGE_CRITERIA	PurgeCodeDescription
YFS_BASE_SHIP_CONSTRAINTS	FieldNameDesc
YFS_BASE_TRANSACTION	BaseTranname

Table A–1   Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_BUSINESS_DOCUMENT	DocumentDescription
	DocumentName
YFS_CALENDAR	CalendarDescription
YFS_CARRIER_SERVICE	CarrierServiceDesc
YFS_CATEGORY	ShortDescription
	Description
YFS_CATEGORY_DOMAIN	ShortDescription
	Description
YFS_CHARGE_CATEGORY	Description
YFS_CHARGE_NAME	Description
YFS_CLASSIFICATION_ PURPOSE	ClassificationPurposeDesc
YFS_COMMON_CODE	CodeLongDescription
	CodeShortDescription
	Note: For entries where CODE_TYPE starts with CODE_TYPE_LIST, you must change the CODE_DESCRIPTION property in the `ycpapibundle.properties` and `yscpapibundle.properties` files.
YFS_CONDITION	ConditionID
	ConditionName
YFS_COUNT_STRATEGY	Description
YFS_CURRENCY	CurrencyDescription
YFS_DATE_TYPE	Description
YFS_DEVICE_SUB_TYPE	Description
YFS_DEVICE_TYPE	Description
YFS_DOCUMENT_FORMAT	FormatDescription
YFS_DOCUMENT_PARAMS	Description

Table A–1   Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_ENTERPRISE	Enterprisename
YFS_EQUIPMENT_TYPE	Description
YFS_ERROR_CAUSE_ACTION	Cause
	Action
YFS_ERROR_CODE	ErrorMessage
YFS_EVENT	EventName
YFS_EXCEPTION_TYPE	ExceptionTypeDescription
YFS_EXECUTION_EXCEPTION	ExceptionShortDescription
	ExceptionLongDescription
YFS_FREIGHT_TERMS	Description
	ShortDescription
YFS_HOLD_TYPE	HoldTypeDescription
YFS_INVENTORY_DEMAND_ TYPE	Description
YFS_INVENTORY_STATUS	Description
YFS_INVENTORY_SUPPLY_TYPE	Description
YFS_ITEM_UOM_MASTER	Description
YFS_LOCALE	LocaleDescription
YFS_LOCATION_SIZE	Description
YFS_MONITOR_TYPE	Field1Name, Field2Name, Field3Name, Field4Name, Field5Name, Field6Name, Field7Name, Field8Name, Field9Name, Field10Name
YFS_NEGOTIATION_RULE	NegotiationRuleId
YFS_NODE_TYPE	NodeTypeDescription
YFS_ORDER_LINE_TYPE	LineTypeDesc

Table A–1   Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_ORGANIZATION	OrganizationName
	Note: OrganizationCode and OrganizationKey cannot be localized.
YFS_PAYMENT_TYPE	PaymentTypeDescription
YFS_PICK_STRATEGY	Description
YFS_PIPELINE	PipelineDescription
	PipelineId
YFS_PLA_ACTIVITY_DETER	Description
YFS_PLA_ZONE_SET	Description
YFS_PRINT_DOCUMENT	PrintDocumentDescription
YFS_PROCESS_TASK_TYPE	ProcessTaskTypeDesc
YFS_PROCESS_TYPE	ProcessTypeName
	Description
YFS_PRODUCTIVITY_TYPE	Description
	ProductivityType
YFS_PURGE_CRITERIA	PurgeCodeDescription
	ErrFileName
	LogFileName
YFS_QUESTION	QuestionText
YFS_QUEUE	QueueDescription
	Note: QueueKey and QueueId cannot be localized.
YFS_REGION_LEVEL	AddressFieldAlias
YFS_REGION_MATCH_PREF	AddressFieldAlias
YFS_RETRIEVAL_STRATEGY	Description
YFS_RETURN_DISPOSITION	Description

Table A–1   Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_ROLE	RoleDescription
	Note: RoleID and RoleKey cannot be localized.
YFS_ROLE_DOCUMENT	DocumentName
	DocumentDescription
YFS_RULES	RuleSetFieldDescription
YFS_SCAC	ScacDesc
	Note: Scac and ScacKey cannot be localized.
YFS_SCAC_AND_SERVICE	ScacAndService
	ScacAndServiceDesc
YFS_SERVICE_TYPE	Description
YFS_SHIPMENT_GROUP	Description
YFS_SPECIAL_SERVICES	SpecialServicesDescription
YFS_STATUS	Description
	StatusName
YFS_STATUS_MODIFICATION_TYPE	ModificationLevelScreenName
	ModificationTypeScreenName
YFS_TASK_TYPE	TaskTypeName
YFS_TRANSACTION	Tranname
YFS_UOM	UomDescription
	Note: If the YFS_UOM.xml is changed, the YFS_UOM_CONVERSION.xml should also have corresponding changes.
YFS_USER	Username
YFS_USER_EXIT_IMPL	ImplementationNotes
YFS_USER_GROUP	UsergroupName

Table A–1   Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_WAVE_SIZE_CONSTRAINT	Description
YFS_ZONE	Description
YFS_COUNT_PROGRAM	CountProgramName
YFS_COUNT_PROGRAM_COND	Description

Index