

# ***Sterling Commerce***

*An IBM Company*

## **Selling and Fulfillment Foundation: Extending the Condition Builder Guide**

Release 8.5

October 2009



# Copyright Notice

Copyright © 1999 - 2009

Sterling Commerce, Inc.

ALL RIGHTS RESERVED

## STERLING COMMERCE SOFTWARE

\*\*\*TRADE SECRET NOTICE\*\*\*

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

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

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

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

---

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

Copyright © 1999 - 2009

## Third-Party Software

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

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

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

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

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

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

<http://danadler.com/jacob/>

<http://www.dom4j.org>

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

## THE APACHE SOFTWARE FOUNDATION SOFTWARE

The Sterling Commerce Software is distributed with or on the same storage media as the following software products (or components thereof) and java source code files: Xalan version 2.5.2, Cookie.java, Header.java, HeaderElement.java, HttpException.java, HttpState.java, NameValuePair.java, CronTimeTrigger.java, DefaultTimeScheduler.java, PeriodicTimeTrigger.java, Target.java,

TimeScheduledEntry.java, TimeScheduler.java, TimeTrigger.java, Trigger.java, BinaryHeap.java, PriorityQueue.java, SynchronizedPriorityQueue.java, GetOpt.java, GetOptsException.java, IllegalArgumentException.java, MissingOptArgException.java (collectively, "Apache 1.1 Software"). Apache 1.1 Software is free software which is distributed under the terms of the following license:

## License Version 1.1

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

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgement: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).". Alternatively, this acknowledgement may appear in the software itself, if and whenever such third-party acknowledgements normally appear.
4. The names "Commons", "Jakarta", "The Jakarta Project", "HttpClient", "log4j", "Xerces", "Xalan", "Avalon", "Apache Avalon", "Avalon Cornerstone", "Avalon Framework", "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without specific prior written permission. For written permission, please contact [apache@apache.org](mailto:apache@apache.org).
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without the prior written permission of the Apache Software Foundation.

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

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

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

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

```

<install>/jar/commons_collections/3_2/LICENSE.txt,
<install>/jar/commons_digester/1_8/commons-digester-1.8.jar (/META-INF/LICENSE.txt),
<install>/jar/commons_io/1_4/LICENSE.txt,
<install>/jar/commons_lang/2_1/Commons_Lang_License.txt,
<install>/jar/commons_logging/1_0_4/commons-logging-1.0.4.jar (/META-INF/LICENSE.txt),
<install>/jar/commons_net/1_4_1/commons-net-1.4.1.jar (/META-INF/LICENSE.txt),
<install>/jar/smcfs/8.5/lucene-core-2.4.0.jar (/META-INF/LICENSE.txt),
<install>/jar/struts/2_0_11/struts2-core-2.0.11.jar (./LICENSE.txt),
<install>/jar/mesa/gisdav/WEB-INF/lib/Slide_License.txt,
<install>/mesa/studio/plugins/xerces_2_7_license.txt,
<install>/jar/commons_pool/1_2/Commons_License.txt,
<install>/jar/jakarta_oro/2_0_8/JakartaOro_License.txt,
<install>/jar/log4j/1_2_15/LOG4J_License.txt,
<install>/jar/xalan/2_7/Xalan_License.txt,
<install>/jar/soap/2_3_1/Apache_SOAP_License.txt

```

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

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

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

- the W3C consortium (<http://www.w3c.org>) ,
- the SAX project (<http://www.saxproject.org>)

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

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.
- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.
- voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.

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

## Ant-Contrib Software

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

The Apache Software License, Version 1.1

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

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgement:

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

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

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

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

THIS SOFTWARE IS PROVIDED `AS IS' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE ANT-CONTRIB PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The preceding license only applies to the Ant-Contrib Software and does not apply to the Sterling Commerce Software or to any other Third-Party Software.

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

## **DOM4J Software**

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

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

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact [dom4j-info@metastuff.com](mailto:dom4j-info@metastuff.com).
4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.
5. Due credit should be given to the DOM4J Project - <http://www.dom4j.org>

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

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

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

## THE ECLIPSE SOFTWARE FOUNDATION

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

com.ibm.icu.nl1\_3.4.4.v200606220026.jar, org.eclipse.ant.core.nl1\_3.1.100.v200606220026.jar,  
org.eclipse.ant.ui.nl1\_3.2.0.v200606220026.jar, org.eclipse.compare.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.core.boot.nl1\_3.1.100.v200606220026.jar,  
org.eclipse.core.commands.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.core.contenttype.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.core.expressions.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.core.filebuffers.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.core.filesystem.nl1\_1.0.0.v200606220026.jar,  
org.eclipse.core.jobs.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.core.resources.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.core.runtime.compatibility.auth.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.core.runtime.compatibility.nl1\_3.1.100.v200606220026.jar,  
org.eclipse.core.runtime.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.core.variables.nl1\_3.1.100.v200606220026.jar,  
org.eclipse.debug.core.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.debug.ui.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.equinox.common.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.equinox.preferences.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.equinox.registry.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.help.appserver.nl1\_3.1.100.v200606220026.jar,  
org.eclipse.help.base.nl1\_3.2.0.v200606220026.jar, org.eclipse.help.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.help.ui.nl1\_3.2.0.v200606220026.jar, org.eclipse.jdt.apt.core.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.jdt.apt.ui.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.jdt.core.manipulation.nl1\_1.0.0.v200606220026.jar,  
org.eclipse.jdt.core.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.jdt.debug.ui.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.jdt.doc.isv.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.jdt.doc.user.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.jdt.junit4.runtime.nl1\_1.0.0.v200606220026.jar,  
org.eclipse.jdt.launching.nl1\_3.2.0.v200606220026.jar, org.eclipse.jdt.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.jdt.ui.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.jface.databinding.nl1\_1.0.0.v200606220026.jar,  
org.eclipse.jface.nl1\_3.2.0.v200606220026.jar, org.eclipse.jface.text.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ltk.core.refactoring.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ltk.ui.refactoring.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.osgi.nl1\_3.2.0.v200606220026.jar, org.eclipse.osgi.services.nl1\_3.1.100.v200606220026.jar,  
org.eclipse.osgi.util.nl1\_3.1.100.v200606220026.jar, org.eclipse.pde.core.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.pde.doc.user.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.pde.junit.runtime.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.pde.nl1\_3.2.0.v200606220026.jar, org.eclipse.pde.runtime.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.pde.ui.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.platform.doc.isv.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.platform.doc.user.nl1\_3.2.0.v200606220026.jar,

org.eclipse.rcp.nl1\_3.2.0.v200606220026.jar, org.eclipse.search.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.swt.nl1\_3.2.0.v200606220026.jar, org.eclipse.team.core.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.team.cvs.core.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.team.cvs.ssh.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.team.cvs.ssh2.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.team.cvs.ui.nl1\_3.2.0.v200606220026.jar, org.eclipse.team.ui.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.text.nl1\_3.2.0.v200606220026.jar, org.eclipse.ui.browser.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.cheatsheets.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.console.nl1\_3.1.100.v200606220026.jar,  
org.eclipse.ui.editors.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.externaltools.nl1\_3.1.100.v200606220026.jar,  
org.eclipse.ui.forms.nl1\_3.2.0.v200606220026.jar, org.eclipse.ui.ide.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.intro.nl1\_3.2.0.v200606220026.jar, org.eclipse.ui.navigator.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.navigator.resources.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.presentations.r21.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.views.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.views.properties.tabbed.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.workbench.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.ui.workbench.texteditor.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.update.configurator.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.update.core.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.update.scheduler.nl1\_3.2.0.v200606220026.jar,  
org.eclipse.update.ui.nl1\_3.2.0.v200606220026.jar,  
com.ibm.icu\_3.4.4.1.jar,  
org.eclipse.core.commands\_3.2.0.I20060605-1400.jar,  
org.eclipse.core.contenttype\_3.2.0.v20060603.jar,  
org.eclipse.core.expressions\_3.2.0.v20060605-1400.jar,  
org.eclipse.core.filesystem.linux.x86\_1.0.0.v20060603.jar,  
org.eclipse.core.filesystem\_1.0.0.v20060603.jar, org.eclipse.core.jobs\_3.2.0.v20060603.jar,  
org.eclipse.core.runtime.compatibility.auth\_3.2.0.v20060601.jar,  
org.eclipse.core.runtime\_3.2.0.v20060603.jar, org.eclipse.equinox.common\_3.2.0.v20060603.jar,  
org.eclipse.equinox.preferences\_3.2.0.v20060601.jar, org.eclipse.equinox.registry\_3.2.0.v20060601.jar,  
org.eclipse.help\_3.2.0.v20060602.jar, org.eclipse.jface.text\_3.2.0.v20060605-1400.jar,  
org.eclipse.jface\_3.2.0.I20060605-1400.jar, org.eclipse.osgi\_3.2.0.v20060601.jar,  
org.eclipse.swt.gtk.linux.x86\_3.2.0.v3232m.jar, org.eclipse.swt\_3.2.0.v3232o.jar,  
org.eclipse.text\_3.2.0.v20060605-1400.jar,  
org.eclipse.ui.workbench.texteditor\_3.2.0.v20060605-1400.jar,  
org.eclipse.ui.workbench\_3.2.0.I20060605-1400.jar, org.eclipse.ui\_3.2.0.I20060605-1400.jar,  
runtime\_registry\_compatibility.jar, eclipse.exe, eclipse.ini, and startup.jar  
(collectively, "Eclipse Software").

All Eclipse Software is distributed under the terms and conditions of the Eclipse Foundation Software User Agreement (EFSUA) and/or terms and conditions of the Eclipse Public License Version 1.0 (EPL) or other license agreements, notices or terms and conditions referenced for the individual pieces of the Eclipse Software, including without limitation "Abouts", "Feature Licenses", and "Feature Update Licenses" as defined in the EFSUA .

A copy of the Eclipse Foundation Software User Agreement is found at  
<install\_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/notice.html,  
<install\_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/plugins/notice.html,  
<install\_dir>/SI/repository/rcp/rcpdependencies/gtk.linux.x86/eclipse/notice.html, and  
<install\_dir>/SI/repository/rcp/rcpdependencies/gtk.linux.x86/eclipse/plugins/notice.html.

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

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

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

The Language Pack (NL Pack) piece of the Eclipse Software, is distributed in object code form. Source code is available at [http://archive.eclipse.org/eclipse/downloads/drops/L-3.2\\_Language\\_Packs-200607121700/index.php](http://archive.eclipse.org/eclipse/downloads/drops/L-3.2_Language_Packs-200607121700/index.php). In the event the source code is no longer available from the website referenced above, contact Sterling Commerce at 978-513-6000 and ask for the Release Manager. A copy of this license is located at <install\_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/plugins/epl-v10.htm and <install\_dir>/SI/repository/rcp/rcpdependencies/gtk.linux.x86/eclipse/plugins/epl-v10.html.

The org.eclipse.core.runtime\_3.2.0.v20060603.jar piece of the Eclipse Software was modified slightly in order to remove classes containing encryption items. The org.eclipse.core.runtime\_3.2.0.v20060603.jar was modified to remove the Cipher, CipherInputStream and CipherOutputStream classes and rebuild the org.eclipse.core.runtime\_3.2.0.v20060603.jar.

## **Ehcache Software**

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

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

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

## **EZMorph Software**

The Sterling Commerce Software is also distributed with or on the same storage media as the EZMorph v. 1.0.4 software (Copyright © 2006-2008 Andres Almiray) ("EZMorph Software"). EZMorph Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/jar/ezmorph/1\_0\_4/ezmorph-1.0.4.jar (./LICENSE.txt).

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

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

## **Firebug Lite Software**

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

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

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

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

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

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

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

## **ICE SOFTWARE**

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

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

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

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

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

## **JBOSS SOFTWARE**

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

A copy of the GNU Lesser General Public License is provided at:  
<install\_dir>\jar\jboss\4\_2\_0\LICENSE.html

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

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

jnpserver.jar, jsp-api.jar, servlet-api.jar, tomcat-juli.jar.

The JBoss Software was modified slightly in order to allow the ClientSocketFactory to return a socket connected to a particular host in order to control the host interfaces, regardless of whether the ClientSocket Factory specified was custom or not. Changes were made to org.jnp.server.Main. Details concerning this change can be found at [http://sourceforge.net/tracker/?func=detail&aid=1008902&group\\_id=22866&atid=376687](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](http://sourceforge.net/tracker/?func=detail&aid=1008902&group_id=22866&atid=376687). Source code for all other components of the JBoss Software is located at <http://www.jboss.org>.

## JGO SOFTWARE

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

### U.S. Government Restricted Rights

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

## JSLib Software

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

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

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

The Original Code is Mozdev Group, Inc. code. The Initial Developer of the Original Code is Mozdev Group, Inc. Portions created by Mozdev Group, Inc. are Copyright © 2003 Mozdev Group, Inc. All Rights Reserved. Original Author: Pete Collins <[pete@mozdev.org](mailto:pete@mozdev.org)> one Contributor(s): \_\_\_\_\_ none listed \_\_\_\_\_.

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

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

## Json Software

The Sterling Commerce Software is also distributed with or on the same storage media as the Json 2.2.2 software (Copyright © 2006-2008 Json-lib) ("Json Software"). Json Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/jar/jsonlib/2\_2\_2/json-lib-2.2.2-jdk13.jar.

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

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

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

## Purple Technology

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

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

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

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors and the names "Purple Technology," "Purple Server" and "Purple Chat" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [server@purpletech.com](mailto:server@purpletech.com).

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

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

## Rico Software

The Sterling Commerce Software is also distributed with or on the same storage media as the Rico.js software (Copyright © 2005 Sabre Airline Solutions) ("Rico Software"). Rico Software is free software

which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/repository/eardata/platform/war/ajax/scripts/Rico\_License.txt.

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

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

## Rhino Software

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

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

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

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

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

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

## Sun Microsystems

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

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

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

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

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

The Sterling Commerce Software is also distributed with or on the same storage media as the Web-app\_2\_3.dtd software (Copyright © 2007 Sun Microsystems, Inc.) ("Web-App Software"). Web-App Software is free software which is distributed under the terms of the Common Development

and Distribution License ("CDDL"). A copy of the CDDL is found in <http://kenai.com/projects/javamail/sources/mercurial/show>.

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

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

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

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

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

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

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

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

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

## **WARRANTY DISCLAIMER**

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

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

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

# Contents

---

<b>1</b>	<b>Checklist for Customization Projects</b>	
1.1	Customization Projects.....	13
<b>2</b>	<b>Customizing Condition Builder Fields</b>	
2.1	About Condition Builder .....	17
2.2	Adding Custom Attributes by Process Type .....	17
2.3	Adding Custom Attributes during Condition Definition.....	20
2.4	Providing Condition Properties in Dynamic Condition .....	22
2.5	Providing Condition Cases for an Advanced XML Condition .....	23
<b>3</b>	<b>Creating and Modifying Advanced XML Conditions</b>	
3.1	About Advanced XML Conditions.....	25
3.2	What is Greex? .....	25
3.3	What is Greex Syntax?.....	26
3.4	Writing Custom Greex Functions .....	29
3.5	Localizing the Greex Syntax.....	31
3.6	Logging Information for the Greex Rule.....	31
3.7	The Sterling Greex Editor - Setup Phase 1 .....	31
3.7.1	Installing Prerequisite Software Components.....	32
3.7.2	The Sterling Greex Editor - Setup Phase 2 .....	33
3.7.3	The Sterling Greex Editor - Setup Phase 3 .....	35
3.7.4	Creating a New Advanced XML Condition .....	37
3.8	Creating a Normal Advanced XML Condition".....	38
3.9	Creating a Decision Table Based An Advanced XML Condition .....	42

3.10	Validating an Advanced XML Condition .....	47
3.11	Loading Advanced XML Conditions into the Database.....	47
3.11.1	Steps to Load an Advanced XML Condition .....	48
3.11.1.1	Before Loading an Advanced XML Condition .....	48
3.11.1.2	Steps to Load an Advanced XML Conditions.....	48
3.12	Importing Advanced XML Conditions From the Database.....	51
3.12.1	Before Importing an Advanced XML Condition .....	52
3.12.2	Steps to Import Advanced XML Conditions .....	52
3.13	Modifying an Advanced XML Condition.....	55

## Index

# Checklist for Customization Projects

---

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

## 1.1 Customization Projects

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

### 1. Prepare your development environment

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

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

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

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

### 2. Plan your customizations

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

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

### 3. Extend the Database

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

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

### 4. Make other changes to APIs

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

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

## 5. Customize the UI

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

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

## 6. Extend Transactions

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

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

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

### 7. Build and deploy your customizations or extensions

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

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

# Customizing Condition Builder Fields

---

## 2.1 About Condition Builder

Selling and Fulfillment Foundation condition builder is used as a part of the service definition framework or in pipeline definitions. This can be used if you want to define dynamic conditions but also use some static set of attributes present in the condition builder. The following section explains the use of the fields that can be customized in the condition builder.

---

---

**Note:** All the string comparisons made in the condition builder are case sensitive.

---

---

For more information on defining conditions and using condition builder for services, refer to the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

## 2.2 Adding Custom Attributes by Process Type

You can add custom attributes in the statement builder, based on the process types such as order fulfillment, outbound shipment and so forth. This customization is useful when you want to evaluate a condition based on an attribute that is not provided as a pre-defined set in the condition builder. These custom attributes are added in the condition builder when you create a condition. For more information on creating conditions refer to the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

The custom attributes can be added by creating a custom XML file as below:

1. Create a directory named `extensions` in `<INSTALL_DIR>/extensions/global/template/configapi` directory.
2. Create a file named `extn_conditionbuilder.xml` in the `<INSTALL_DIR>/extensions/global/template/configapi` directory. The format of a sample XML file is given in [Example 2–1](#).

---

---

**Note:** For customizing condition builders for order hold types create a file called `holdtype_extn_conditionbuilder.xml` and follow the same procedure outlined in this section.

---

---

---

---

**Note:** You must restart your application server, every time you add conditions in the `extn_conditionbuilder.xml` file, for the changes to appear in the Applications Manager screen.

---

---

The sample file creates conditions for extended order number, shipping information and order details. The extended order details has further sub-elements such as, `CustomItemId` and `ExtnOrderType`.

The `Id` attribute must be identical to the element name. For example, the element `ExtnOrderNo` must contain the attribute `Id="ExtnOrderNo"`.

### **Example 2–1** Sample File for Adding Custom Attributes

```
<CustomConditionAttributes>
  <ProcessType Name="ORDER_FULFILLMENT">
    <ExtnOrderNo Id="ExtnOrderNo" DisplayName="Extended Order Number"
      DataType="Text-40" Type="Leaf"/>
    <ExtnShiptoId Id="ExtnShiptoId" DisplayName="Extended Ship To Id"
      DataType="Text-40" Type="Leaf"/>
    <CustomOrderDetails Id="CustomOrderDetails" DisplayName="Custom Order
      Details">
      <CustomItemId Id="CustomItemId" DisplayName="Custom Item Id"
        DataType="Text-100" Type="Leaf"/>
      <ExtnOrderType Id="ExtnOrderType" DisplayName="Extended Order Type"
        DataType="Text-100" Type="Leaf"/>
    </CustomOrderDetails>
  </ProcessType>
</CustomConditionAttributes>
```

```
</ProcessType>
</CustomConditionAttributes>
```

**Table 2–1 Attributes Details**

Attribute	Description
ProcessType	
Name	Required. Specify the name of the Process type where you are creating the condition.
Elements	
DisplayName	Required. Specify the display name for this attribute in the condition builder screen.
DataType	Required. Specify the data type needed for this attribute. For example, the pre-defined attribute BillTold has a data type of "ID-40".  The data type can be any one of the types provided in the <INSTALL_DIR>/repository/datatypes/datatypes.xml file.
Id	Required. Specify the ID you want this condition to be associated. By default this is same as the element name.
Type	Required. Specify the type of this attribute. For example "Leaf" type identifies the attribute as part of a menu item.  If it is not specified, the attribute is considered to be a menu header leading to a sub menu. In <a href="#">Example 2–1</a> the element CustomOrderDetails does not have a type defined, since it has leading sub-menus such as CustomItemId and ExtnOrderType.

3. The elements like ExtnOrderNo, ExtnShipToId, etc., as shown in [Example 2–1](#), are used to define the conditions when it is saved in the database. Therefore, the element name must be unique for a single process type.
4. This XML file content is merged with the pre-defined static attributes by process type and displayed on the screen as shown below:



5. However, upon clicking **OK** in the statement builder window, the display name of the XML is shown in the condition detail window.
6. The display names are replaced with their element names once the condition is saved. These element names along with their conditions is referred as the condition value in the database table `YFS_CONIDITION`.
7. These custom attributes are shown in the similar way as the current static attributes in the condition builder.

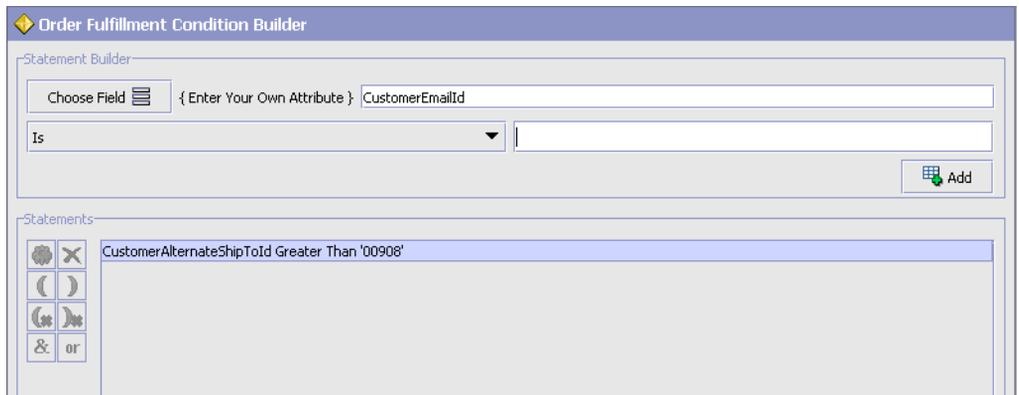
The custom attributes defined in this manner can be used as part of the service definition framework or in pipeline definitions for creating events or conditions. However, if used in pipeline definitions, the condition evaluator does not evaluate these custom attributes.

**Note:** The custom attributes cannot be used to evaluate conditions in pipeline determination rules.

## 2.3 Adding Custom Attributes during Condition Definition

When you define conditions for a particular order, you can optionally choose to add your own custom attributes to be evaluated as part of the condition.

1. The custom attributes can be added in the condition builder by clicking **Choose Field** and selecting **Enter Your Own Attribute** option in the statement builder window.
2. Enter the attribute name you want to be included in the condition as shown below, and click **Add**.



For example, if you want to include an attribute named `CustomerEmailId`, enter that attribute in the text box provided and continue with your statement creation. This attribute is included in evaluating the condition.

---

---

**Note:** The corresponding template for the condition should be extended to include the above attribute.

---

---

---

---

**Note:** This field is limited only to the unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.

---

---

3. Once the name is entered, the rest of the condition can be built in the same way as the pre-defined attributes. For more information on how to create a condition using pre-defined attributes, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

The custom attributes created during condition definition can be used as part of the service definition framework or in pipeline definitions for creating events or conditions.

**Note:** Custom attributes defined in this manner is available only when defining this condition. The attribute entered is not available for re-use in other conditions.

## 2.4 Providing Condition Properties in Dynamic Condition

The condition definition enables you to create static or dynamic or advanced XML condition.

When creating dynamic conditions you must check Dynamic field and mention the class name to be invoked to evaluate the condition. This creation of dynamic condition is extended to include configuration of custom name, value properties. These properties are set into the java class before evaluating the condition.



To enable these extended attributes, the dynamic condition class should implement a `YCPDynamicConditionEx` interface. The definition of this interface is as follows:

```
public interface YCPDynamicConditionEx {
    boolean evaluateCondition(YFSEEnvironment env, String name, Map mapData,
        Document doc);
    void setProperties(Map map);
}
```

The parameter name passed to the interface refers to the Condition Name configured during definition.

For more information about the interface and parameters, see the *Selling and Fulfillment Foundation: Javadocs*.

## 2.5 Providing Condition Cases for an Advanced XML Condition

When you are modifying decision table-based advanced XML conditions, you can extend an advanced XML condition to include different attributes of custom cases. These attributes are set before you evaluate the decision table-based advanced XML condition of type.

Condition ID: GreexTest  
 Condition Name: GreexTest  
 Condition Group: Default

Static  Dynamic  Advanced XML

**Advanced XML** | Source View

If  
 OrderType IS WEB  
 LineQuantity SUM IS GREATER THAN 200  
 Then Return 7  
 Else Return 0

Condition Cases		
WEB	200	7
WEB	100	5
STORE	500	7
CALLCENTER	250	3

You can add or modify existing cases defined for a decision table-based advanced XML condition. You can modify the different attributes of a condition case. After you modify a condition case and click the Save button, the new values of the attribute reflects in both Advanced XML and Source View screen.

The default return value defined for a decision table-based advanced XML condition displays as a hyperlink on the Advanced XML screen. If none of the condition cases are satisfied, the Greex engine returns the default value. Click the hyperlink to edit the default return value and specify the new default return value for the advanced XML condition. The pop-up screen displays the old value. You can also enter new values. The new value reflects in the Advanced XML screen as well as in the Source View screen when you click the Save button in the pop-up screen.

# Creating and Modifying Advanced XML Conditions

---

## 3.1 About Advanced XML Conditions

This section explains how to modify an advanced XML condition, which is also known as the Greex rule. You can only assign new values to the modifiable parameters of an advanced XML condition. You can also localize or log information about an advanced XML condition or Greex rule.

## 3.2 What is Greex?

The Greex framework enables you to define advanced XML conditions based on the Greex syntax. The Greex framework provides a declarative mechanism to use an input XML document in order to evaluate advanced XML conditions.

Some of the salient features of the Greex framework are:

- It is XML aware.
- It supports namespaces.
- It contains a set of libraries.
- It can return an XML element, a String, or Boolean value.

The Greex framework contains the Greex library, which is comprised of a set of functions you can call on input data through XPath expressions. Constants can be used in function parameters also.

### 3.3 What is Greex Syntax?

Using the Greex syntax you can create advanced XML conditions or Greex rules. An advanced XML condition is used to evaluate certain conditions on the input data. The Greex syntax is an XML based. This style of condition evaluation enables you to use the input data in multiple ways, rather than just a Boolean output. The Greex syntax provides Greex constructs that are capable of being nested using multiple IF, and ELSE blocks, and also allows to group expressions using an AND or OR operator. Each expression comprises one or more function calls. You can include functions in a nested loop, which means parameters to functions can be other function calls. These contain basic IF ELSE conditions.

[Table 3–1](#) describes various elements of an If/Else construct.

**Table 3–1 If/Else Condition's Element List**

Element	Description
Condition	The Condition element provides a logical grouping of all expressions that need to be evaluated by operating on an input XML.
Return	<p>Every condition defined under the If/Else construct must return a value. A condition can return an XML element, string, or boolean value. It returns the appropriate value for an If/Else condition in the Return element. For example,</p> <ul style="list-style-type: none"> <li>• If you want to return an XML element, the Return element can be: <pre data-bbox="691 621 1262 817"> &lt;Return&gt;     &lt;Value output="&amp;lt;Order         type="&amp;quot;Web&amp;quot;         discount="&amp;quot;5&amp;quot;"/&amp;gt;" /&gt; &lt;/Return&gt; </pre> </li> <li>• If you want to return a string, the Return element can be: <pre data-bbox="691 904 1248 1055"> &lt;Return&gt;     &lt;Value output="Draft Order         Created"&gt; &lt;/Return&gt; </pre> </li> <li>• If you want to return a boolean value, the Return element can be: <pre data-bbox="691 1142 1162 1251"> &lt;Return&gt;     &lt;Value output="true"&gt; &lt;/Return&gt; </pre> </li> </ul>

The Condition element contains various Expression elements. Each Expression element defines the expression that you want to evaluate for

the specified condition. [Table 3–2](#) describes various elements of a condition.

**Table 3–2 Condition's Element List**

Element	Description
Expression	<p>The Expression element contains the expressions that you want to evaluate for a condition to satisfy. To make function calls in the expression, prefix the function name with "fn:". For example, define the following Expression element which makes a function call:</p> <pre>&lt;Expression&gt;fn:equals(@ZipCode, "01876")&lt;/Expression&gt;</pre> <p>You can also pass functions as parameters to other functions. For example, you can define the following Expression element to pass functions as parameters to other functions:</p> <pre>&lt;Expression&gt;fn:intGreater(fn:count(OrderLines/OrderLine), "100")&lt;/Expression&gt;</pre>
Group	<p>This is an optional element. If you want to evaluate a set of expressions together, you must group these set of expressions together.</p> <p>The Group element's "op" attribute indicates the operation you want to perform on the set of expressions. The valid values are: "or" and "and". You can define more than one expression in a Group element. If the Group element's op value is "and", then the condition satisfies only if ALL expressions that are part of the Group element evaluates to "true". Likewise, if the Group element's "op" attribute is "or", then the condition satisfies only if ONE of the expressions that are part of the Group element evaluates to "true".</p> <p>You can create any level of nested Group and Expression elements.</p> <p>Note: If you want to evaluate a single expression, define a single Expression element under the Condition element, without creating the Group element.</p>

A sample IF ELSE construct for defining an advanced XML condition or Greex rule can be:

```

<If>
  <Condition name="isWebOrder?">
    <Group op="and">
      <Expression>fn:!equals(@orderType, "WEB")</Expression>
      <Expression>fn:equals(address::@ZipCode, "01876")</Expression>
    </Group>
  </Condition>
  <Return>
    <Value output="&lt;Order type=&quot;Web&quot;
      discount=&quot;5&quot;/&gt;" />
  </Return>
</If>
<Else>
  <Return>
    <Value output="&lt;Order type=&quot;Catalog&quot;
      discount=&quot;2&quot;/&gt;" />
  </Return>
</Else>

```

### 3.4 Writing Custom Greex Functions

The Greex framework provides a set of defined functions as a part of the Greex library. You can also write custom Greex functions, if necessary.

To write a custom Greex function:

1. Provide implementation for your custom library by implementing all functions of the `com.yantra.ycp.greex.library.LibraryFunction` interface. You must implement the following functions:
  - `Object invoke(GreexContext ctx, List params)`
  - `boolean validateParams(List params)`
  - `String getName()`
  - `String getDescription()`
  - `String getReturnType()`
  - `String[] getParamTypes()`
2. Register your custom library with Greex framework by calling the `registerFunction(LibraryFunction function)` method of the `LibraryFunctionFactory` class.

For example, to register your custom java class such as `MyCustomLibrary`, call the following `registerFunction()` method:

```
LibraryFunctionFactory.getInstance().registerFunction(new  
MyCustomLibrary());
```

3. Create a jar for the custom library that you created in [Step 1](#) and add it to the application server classpath.
4. Load the custom library to the `LibraryFunctionFactory` of Greex by creating a custom Greex initializer servlet (such as `MyCustomGreexInitializer`), which can be a normal servlet. To load this function with Greex framework, specify the name of the servlet in the `<INSTALL_DIR>/repository/eardata/platform/descriptors/weblogic/WAR/WEB-INF/web.xml` file. For example, if your servlet class is `com.servlet.MyCustomGreexInitializer`, add the following entry in the `web.xml` file:

```
<servlet>  
  <servlet-name>MyCustomGreexInitializer</servlet-name>  
  <servlet-class>com.servlet.MyCustomGreexInitializer</servlet-class>  
  <load-on-startup>4</load-on-startup>  
</servlet>
```

5. Copy the custom jar that you created in [Step 3](#) to the following locations:
  - `<INSTALL_DIR>/jar/smcfs`
  - `<INSTALL_DIR>/repository/eardata/platform/war/yfscommon`

---

---

**Note:** Make sure that the custom jar is signed before copying it to this location. You must also update the `<INSTALL_DIR>/repository/eardata/platform/war/yfscommon/jarlist.txt` file.

---

---

- `<INSTALL_DIR>/platformrcp/5_0/othertools/com.yantra.yfc.other.tools.zip/com.yantra.ide.rcptools.core_1.1.0/lib`

## 3.5 Localizing the Greex Syntax

Based on the requirement, you can appropriately localize the Greex syntax. For more information about localizing the Greex syntax, see the *Selling and Fulfillment Foundation: Localization Guide*.

## 3.6 Logging Information for the Greex Rule

You can log information for the Greex rule or an advanced XML condition at different levels. The different levels of logging information is described in GreexLogConstants. You can log information ranging from Apache's log4j framework to a simple System.out.println().

To log information for the Greex rule:

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

```
log(GreexLogData data) method and log information as needed.
For example:
public class MyLogger implements GreexLogger
{
    public void log(GreexLogData data)
    {
        System.out.println("Message:: "+data.getMessage());
        System.out.println("Severity:: "+data.getSeverity());
    }
}
```

2. Register the GreexLogger class with the GreexContext using the registerLogger() method. For example,

```
public class MyApp
{
    GreexContext ctx = new GreexContext();
    Ctx.registerLogger(new MyLogger(), GreexLogConstants.GREEX_DEBUG);
}
```

## 3.7 The Sterling Greex Editor - Setup Phase 1

Using the Greex editor you can conveniently create and modify an advanced XML file, which is also referred as the \*.greex file. The advanced XML file is used to define new advanced XML conditions to evaluate them on the input data.

### 3.7.1 Installing Prerequisite Software Components

This section describes the various software components required for creating an advanced XML condition or a Greex rule using the Sterling Greex Editor. Before creating an advanced XML condition using Sterling Greex Editor, ensure that you have already installed the following software components:

- **Eclipse SDK**

Install the Eclipse SDK version that Sterling Commerce supports. For more information about the Eclipse SDK version, see the *Selling and Fulfillment Foundation: Installation Guide*.

- **Eclipse Related plug-in**

Install the Eclipse Modelling Framework (EMF) plug-in version that Sterling Commerce supports. For more information about the Eclipse EMF plug-in versions, see the *Selling and Fulfillment Foundation: Installation Guide*.

- **Java Development Kit (JDK)**

Install the JDK version that Sterling Commerce supports. For more information about the JDK version, see the *Selling and Fulfillment Foundation: Installation Guide*.

---

---

**Note:** If you are using a browser with JRE version lower than 1.5, the exception `java.security.AccessControlException` displays when you try to open the Greex rule using the Applications Manager.

To resolve this error, in the `<JRE_HOME>/lib/security/java.policy` file, add the following property under the default permission for all domains section:

```
permission java.util.PropertyPermission
"java.home", "read";
```

where `<JRE_HOME>` is the JRE installation directory.

Now, close all the browser instances and again log in to Selling and Fulfillment Foundation.

---

---

- **Sterling Greex plug-in**

Install the Sterling Other Tools plug-in that Sterling Commerce supports. For more information about the Sterling Greex plug-ins version, see the *Selling and Fulfillment Foundation: Installation Guide*.

This plug-in is shipped along with Selling and Fulfillment Foundation that can be found at:

<INSTALL\_DIR>/platformrcp/5\_0/othertools

---



---

**Note:** If you are installing a new version of the Sterling Other Tools plug-ins or updating the previous versions you must clean the cached build information in Eclipse.

To clean this information, start Eclipse with the "-clean" option:

1. Right-click on the Eclipse shortcut and select Properties from the pop-menu. The Properties window displays.
  2. In Target, enter the command line argument "-clean" at the end. For example, "C:\Eclipse 3.2\eclipse\eclipse.exe" -clean.
  3. Start Eclipse.
- 
- 

### Installing the Sterling Other Tools plug-in

To install the Sterling Other Tools plug-in:

Copy all of the folders within the the <INSTALL\_DIR>/platformrcp/5\_0/othertools directory to the <ECLIPSE\_HOME>/plugins folder.

<ECLIPSE\_HOME> refers to the Eclipse SDK installation directory.

Restart Eclipse SDK in order to allow the newly installed plug-ins to be found.

## 3.7.2 The Sterling Greex Editor - Setup Phase 2

Before you start working with the Greex Editor, you must create a java project, which acts as a container for the Greex Model wizard.

To create a java project:

1. Start Eclipse SDK.

2. From the menu bar, select File > New > Project.... The New Project window displays.
3. From the list of wizards, under Java category, select the Java Project.
4. Click Next. The New Java Project window displays.

Project name:

**Contents**

Create new project in workspace  
 Create project from existing source

Directory:

**JRE**

Use default JRE (Currently 'jre1.4.2\_11') [Configure JREs...](#)  
 Use a project specific JRE:

**Project layout**

Use project folder as root for sources and class files  
 Create separate source and output folders [Configure default...](#)

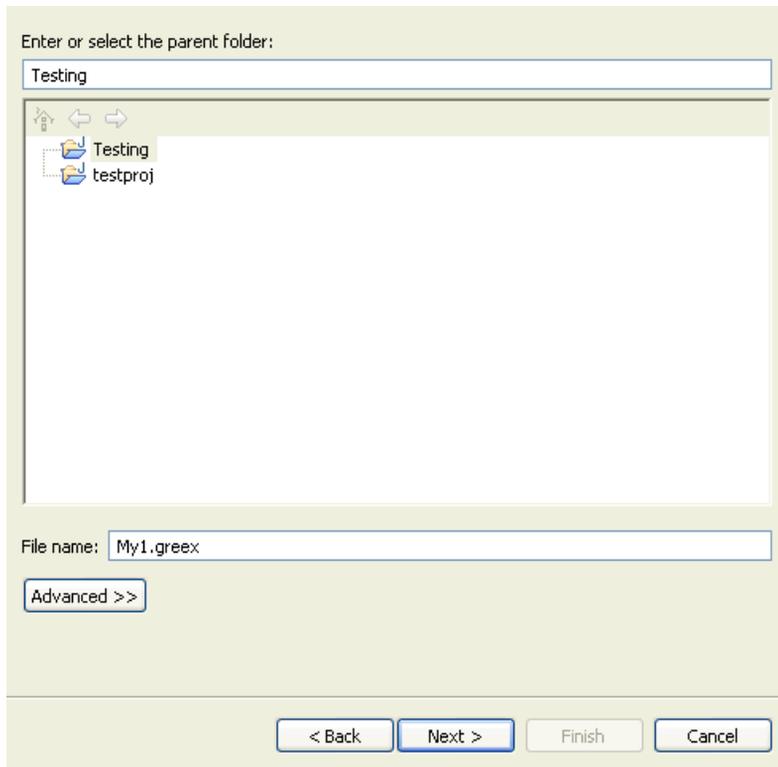
5. In Project name, enter the name of the new java project.
6. Click Next. The Java Settings page displays.
7. Click Finish. The new java project is created.

### 3.7.3 The Sterling Greex Editor - Setup Phase 3

After creating the new java project, run the Greex Model wizard on top of the new java project that you created. The Greex Model wizard either creates an empty \*.greex file. You can create an advanced XML condition or Greex rule in the \*.greex file.

To run the Greex Model wizard:

1. Start Eclipse SDK.
2. From the menu bar, select Window > Show View > Navigator. The java project displays in the Navigator view.
3. Expand the java project that you created. For more information about creating a java project, see [Section 3.7.2, "The Sterling Greex Editor - Setup Phase 2"](#).
4. Right-click on the folder where you want to store the \*.greex file and select New > Other from the pop-up menu. The New window displays.
5. From the list of wizards, select Rich Client Platform Wizards > Rich Client Platform Greex Model.
6. Click Next. The New window displays.



7. In File name, enter the name of the \*.greex file.
8. Click Next. The Greex Model window displays.

Model Object

Greex Rule

XML Encoding

UTF-8

< Back   Next >   Finish   Cancel

9. In Model Object, select Greex Rule.
10. In XML Encoding, select UTF-8.
11. Click Finish. The \*.greex file is created in the specified folder.

### 3.7.4 Creating a New Advanced XML Condition

After creating the \*.greex file, you can define an advanced XML condition or Greex rule. You can create two types of advanced XML conditions:

- Normal advanced XML condition
- Decision Table based advanced XML condition.

## 3.8 Creating a Normal Advanced XML Condition"

After creating the \*.greex file, you can define a normal advanced XML condition or Greex rule.

**Note:** Make sure that you **do not** add any comments in the advanced XML condition or Greex rule of any type.

The normal advanced XML condition is useful in scenarios where you need to define multiple condition criteria for an advanced XML condition you need to define multiple condition criteria and each condition criteria is associated with different attributes. These multiple condition criteria are defined using nested IF and ELSE constructs. For example, you may want to create a normal advanced XML condition for the condition criteria such as:

If Ordertype="WEB" and OrderQty="100", then TaxExemptFlag="N".

Else

If OriginalTotalAmout="1000" and OrderLine>"5", then  
discount="10" TaxExemptFlag="Y".

Else TaxExemptFlag="N".

In this case, we have multiple condition criteria and each condition criteria has different attributes associated with it. The first condition criteria has Ordertype, OrderQty, and Discount associated with it. Whereas, the second condition criteria has OriginalTotalAmount, OrderLine, and TaxExemptFlag associated with it.

These condition criteria are defined in the \*.greex file using nested IF AND ELSE constructs. A sample \*.greex file for the advanced XML condition is as follows:

```
<GreexRule desc="Determine Discount and Tax Exemption based on
          various attributes"
          id="getDiscountTaxExempt"
          name="Get Discount and Tax Exempt"
          returnType="Xml"
          type="">
<If>
  <Condition name="isDiscount20?">
    <Group op="and">
```

```

    <Expression>fn:equals(@orderType, &quot;WEB&quot;)</Expression>
    <Expression>fn:equals(@orderQty, &quot;100&quot;)</Expression>
  </Group>
</Condition>
<Return>
  <Value output="&lt;Order TaxExemptFlag=&quot;N&quot;"/>" />
</Return>
</If>
<Else>
  <If>
    <Condition name="TaxExempted?">
      <Group op="and">
        <Expression>fn:equals(@OriginalTotalAmout, &quot;1000&quot;)</Expression>
        <Expression>fn:intGreater(@orderLine,&quot;5&quot;)</Expression>
      </Group>
    </Condition>
    <Return>
      <Value output="&lt;Order discount=&quot;10&quot;
        TaxExemptFlag=&quot;Y&quot;"/>" />
    </Return>
  </If>
  <Else>
    <Return>
      <Value output="&lt;Order discount=&quot;20&quot;
        TaxExemptFlag=&quot;N&quot;"/>" />
    </Return>
  </Else>
</Else>
</GreexRule>

```

To create a normal advanced XML condition:

1. Expand the java project that you created.
2. In the Project Explorer hierarchy, select the \*.greex file. Right-click and select Open With > Greex Model Editor from the pop-up menu.
3. Expand the tree structure that appears in the Greex Editor. Click on a node from the tree. All child leaves of the node are listed. The Document Root element contains the Greex Rule root element.
4. Select the Greex Rule root element in the Properties view. Enter the values for various attributes. See [Table 3–5](#) for the description of various attributes of the Greex Rule root element.

- In the Properties view, you can view various properties of the selected element.

To open the Properties View:

From the menu bar, select Window > Show View > Other.... From the list of views under Basic, select Properties.

**Table 3–3 Greex Rule Element’s Attribute List**

Attribute	Description
Desc	Enter the description of the Greex rule.
Id	Enter a unique identifier of the Greex rule.
Name	Enter the name of the Greex rule.
Return Type	Enter the return type of the Greex rule. For a normal Greex rule, the valid values are "Xml", "String", and "Boolean". The normal Greex rule can return an XML document, string, or boolean value.
Type	By default, a Greex rule is a normal Greex rule.

- Under the Greex Rule root element, create a new IF ELSE construct element, as needed. Right-click on the Greex Rule root element and select New Child > If/Else from the pop-up menu. You can create any level of nesting of IF and ELSE constructs.
- Select the If/Else element. In the Properties view, enter the name of the If element in the Name property.
- Under the If/Else element, create a new Condition child element. Right-click on the If/Else element and select New Child > Condition from the pop-up menu.
- Select the Condition element. In the Properties view, enter the name of the Condition element in the Name property.
- As every condition must return a value, under the If/Else element, create a new Return element and specify the appropriate return value for the associated condition. Right-click on the If/Else element and select New Child > Return from the pop-up menu.
- Right-click on the Return element and select New Child > Value from the pop-up menu. The Enter the value pop-up window displays.

12. Enter the value that you want to return if the IF condition satisfies.
13. Select the Return element. In the Properties view, in the Default property, specify the default value (if necessary) that you want to return if the IF condition is not satisfied. In the Output property, specify the value that you want to return if the IF condition satisfies.
14. Under the Condition element, create a new Expression element to specify the expressions that you want to evaluate for the condition to satisfy. Right-click on the Condition element and select New Child > Expression from the pop-up menu. The Edit Expression pop-up window displays.
15. In Expression, enter the expression you want to evaluate. You can make function calls in the expression by prefixing the function name with "fn:". You can also pass functions to other functions.

---

---

**Note:** Press Ctrl+Space and select the expression from the drop-down list.

---

---

16. (Optional) If you want to evaluate a set of expressions together, you must group them. Under the Condition element, create a new Group element to group a set of expressions together. Right-click on the Condition element and select New Child > Group from the pop-up menu.

Now, you can add more than one expression to this group. Right-click on the Group element and select New Child > Expression from the pop-up menu.

You can also add a new Group element to an existing Group element. Right-click on the Group element and select New Child > Group from the pop-up menu.

---

---

**Note:** You can create any level of nested Group and Expression elements.

---

---

17. (Optional) Select the Group element. In the Properties view, specify the operation that you want to perform on the set of expressions in the Op property. The valid values are: "or" and "and".

If you specify the `Op` property as "or", a condition is satisfied if any of the expressions specified in the group is "true".

If you specify the `Op` property as "and", a condition is satisfied only if all expressions specified in the group is "true".

18. Click Save.

### 3.9 Creating a Decision Table Based An Advanced XML Condition

After creating the \*.greex file, you can define a normal advanced XML condition or Greex rule.

**Note:** Make sure that you **do not** add any comments in the advanced XML condition or Greex rule of any type.

Decision table based advanced XML conditions are useful in scenarios where you have multiple nested condition criteria to be defined for an advanced XML condition but each condition criteria has same attributes associated with it. In such cases, you can write just one condition and have a table of parameters that works like a switch statement. In the decision table based advanced XML condition you can define multiple nested condition criteria in a single IF construct. Hence, there is no ELSE construct in case of decision table based advanced XML conditions. The IF construct contains an array of constant values instead of one constant value as used to be in case of normal advanced XML conditions. The IF construct has a table of parameters that works like a switch statement.

For example, you may want to create a decision table based advanced XML condition for the condition criteria such as:

If `Ordertype="WEB"` and `OrderLineQty="200"`, then `Discount="5"`.

Else

If `Ordertype="STORE"` and `OrderLineQty="500"`, then `Discount="7"`

Else

If `Ordertype="CALL"` and `OrderLineQty="250"`, then `Discount="3"`.

Else

default="0"

Table 3–4 describes the above scenario in the form of a decision table.

**Table 3–4 Decision Table**

Order Type	Order Line Quantity	Discount
WEB	200	5
STORE	500	7
CALLCENTER	250	3
default	0	

In this case, we have multiple condition criteria but each condition criteria has the same attributes associated with it. All the condition cases have Ordertype, OrderLineQty, and Discount attributes associated with it. Therefore, this advanced XML condition has only one IF construct and it contains an array of constant values.

**Note:** For a decision table based Greex rule, it is mandatory to give a default return value. This value is returned if no IF condition gets satisfied.

These condition criteria are defined in the \*.greex file using IF constructs. A sample \*.greex file for the decision table-based advanced XML condition is as follows:

```
<GreexRule desc="Determine Discount based on some attributes"
  id="getDiscount"
  name="Get Discount"
  returnType="String"
  type="DecisionTable">>
<If>
  <Condition name="isDiscount5?">
    <Group op="and">
      <Expression>fn:equals(@orderType,
        &quot;WEB|STORE|CALLCENTER&quot;)</Expression>
      <Expression>fn:equals(@orderLineQty, &quot;200|500|250&quot;)</Expression>
    </Group>
  </Condition>
```

```

<Return>
  <Value default="0" output="5|7|3"/>
</Return>
</If>
</GreexRule>

```

To create a new decision table based advanced XML condition:

1. Expand the java project that you created.
2. In the Project Explorer hierarchy, select the \*.greex file. Right-click > Open With > Greex Model Editor.
3. Expand the tree structure that appears in the Greex editor panel. Click on a node from the tree, all the child leaves of the node are listed. The Document Root element contains the root element Greex Rule.
4. Select the root element Greex Rule and in the Properties view, enter the values for various attributes. [Table 3–5](#) describes various attributes of the root element Greex Rule.

In the Properties view, you can view various properties of the selected element. To open the Properties View, from the menu bar, select Window > Show View > Other.... From the list of views under Basic, select Properties.

**Table 3–5 Greex Rule Element’s Attribute List**

Attribute	Description
Desc	Enter the description of the Greex rule.
Id	Enter the unique identifier of the Greex rule
Name	Enter the name of the Greex rule.
Return Type	Enter the return type of the Greex rule. For a decision table based Greex rule, the only valid value is "String". The decision table based Greex rule can only return a String.
Type	Enter the Greex rule type. By default, it is a normal Greex rule. To make it a decision table based Greex rule, select "DecisionTable" from the drop-down list.

5. Under the root element Greex Rule, create a new IF construct element as per the requirement. Right-click on the root element Greex Rule and select New Child > If from the pop-up menu.

---



---

**Note:** A decision table based Greex rule cannot have an ELSE construct. It can only have a single IF construct.

---



---

6. Select the If element and in the Properties view, enter the name of the If element in the Name property.
7. Under the If element, create a new Condition child element. Right-click on the If element and select New Child > Condition from the pop-up menu.
8. Select the Condition element and in the Properties view, enter the name of the Condition element in the Name property.
9. As every condition must return a value, under the If element, create a new Return element and specify the appropriate return value for the associated condition in the Value element. Right-click on the If element and select New Child > Return from the pop-up menu.
10. Right-click on the Return element and select New Child > Value from the pop-up menu. The Enter the value pop-up window displays.
11. Enter the value which you want to return if the IF condition gets satisfied.
12. Select the Return element. In the Properties view, in the Default property, specify the default value which you want to return if the none of the IF condition does not gets satisfied and in the Output property, specify the value which you want to return if the IF condition gets satisfied.

---



---

**Note:** For a decision table based Greex rule, it is mandatory to give a default return value.

---



---

For a decision table based Greex rule, you can define an array of constant values in the value element. The multiple values are separated using the "|" operator. For example, 5|7|3.

13. Under the Condition element, create a new Expression element to specify the expression(s) that you want to evaluate for the condition

to be satisfied. Right-click on the `Condition` element and select `New Child > Expression` from the pop-up menu. The `Edit Expression` pop-up window displays.

14. In `Expression`, enter the expression that you want to evaluate. You can make function calls in the expression by prefixing the function name with "fn:". You can also pass functions to other functions.

---

---

**Note:** You can press `Ctrl+Space` and select the expression from the drop-down list.

---

---

For a decision table based Greex rule, you can define an array of constant values for a particular XML attribute in the `Expression` element. The multiple values are separated using the "|" operator. For example, `fn:equals(@orderType, "WEB|STORE|CALLCENTER")`.

15. (Optional) Only if you want to evaluate a set of expressions together, you must group these set of expressions. Under the `Condition` element, create a new `Group` element to group a set of expressions together. Right-click on the `Condition` element and select `New Child > Group` from the pop-up menu.

Now, you can add more than one expressions to this group. Right-click on the `Group` element and select `New Child > Expression` from the pop-up menu.

You can also add a new `Group` element to a `Group` element. Right-click on the `Group` element and select `New Child > Group` from the pop-up menu.

---

---

**Note:** Any level of nesting of `Group` and `Expression` elements is allowed.

---

---

16. (Optional) Select the `Group` element and in the `Properties` view, specify the operation that you want to perform on the set of expressions in the "op" property. The valid values are: "or" and "and".

If you specify the `Op` property as "or" then a condition gets satisfied if any of the expressions specified in the group evaluates to true.

If you specify the `Op` property as "and" then a condition gets satisfied only if all of the expressions specified in the group evaluates to true.

17. Click Save.

## 3.10 Validating an Advanced XML Condition

After creating an advanced XML condition, you must validate its syntax and structure. For example, if an advanced XML condition is of decision table type, it must contain only one IF construct and no ELSE constructs. Therefore, you must validate an advanced XML condition before loading it into the database.

To validate an advanced XML condition:

1. Expand the java project that you created.
2. In the Project Explorer hierarchy, select the \*.greex file. Right-click and select Open With > Greex Model Editor from the pop-up menu.
3. Expand the Greex rule that appears in the Greex editor panel. Click on a node from the tree. All child leaves of the node are listed. The Document Root element contains the Greex Rule root element.
4. Select the Greex Rule root element and right-click. Select Validate Greex Rule from the pop-up menu.

If the Greex rule that you created uses correct syntax and structure, the message "Greex rule validation succeeded" displays. Otherwise, the system displays an appropriate error message.

## 3.11 Loading Advanced XML Conditions into the Database

After creating an advanced XML condition or Greex Rule using the Rich Client Platform Greex Editor, load the \*.greex file into the database. This file contains the newly created advanced XML condition that you want to load into the database. Using the Applications Manager, you can modify appropriate parameters of the advanced XML condition (if necessary).

---

**Note:** Before loading an advanced XML condition or Greex rule into the database, you must validate the advanced XML condition for a valid structure or syntax. For more information about validating an advanced XML condition, see [Section 3.10, "Validating an Advanced XML Condition"](#).

---

## 3.11.1 Steps to Load an Advanced XML Condition

### 3.11.1.1 Before Loading an Advanced XML Condition

1. Modify the `<ECLIPSE_HOME>/plugins/com.yantra.ide.othertools.greex.editor_<VERSION_NUMBER>/lib/greexide.properties` file and specify the JDBC connection properties based on the database you are using. `<ECLIPSE_HOME>` is the directory where Eclipse SDK is installed. `<VERSION_NUMBER>` is the current version number of the Greex editor plug-in.
2. Depending on the database, copy the required database driver jar file to the `<ECLIPSE_HOME>/plugins/com.yantra.ide.othertools.greex.editor_<VERSION_NUMBER>/lib` directory. For example, if you are using an Oracle database, copy the `ojdbc14.jar` file.
3. Edit the `<ECLIPSE_HOME>/plugins/com.yantra.ide.othertools.greex.editor_<VERSION_NUMBER>/META-INF/MANIFEST.MF` file and add the relative path (relative to the `<ECLIPSE_HOME>/plugins/com.yantra.ide.othertools.greex.editor_<VERSION_NUMBER>` directory) of the required database driver jar file in the `Bundle-ClassPath` property.

For example, if you are using an Oracle database, add `lib/ojdbc14.jar` to the `Bundle-ClassPath` property, as follows:

```
Bundle-ClassPath: greexeditor.jar,
lib/ojdbc14.jar
```

4. If Eclipse SDK is already running, restart Eclipse in clean mode, as follows:

```
eclipse.exe -clean
```

Loading an Advanced XML Condition

### 3.11.1.2 Steps to Load an Advanced XML Conditions

To load an advanced XML condition into the database:

1. Start Eclipse SDK.

2. From the menu bar, select Window > Show View > Navigator. The java project displays in the Navigator view.
3. Expand the java project that you created. For more information about creating a java project, see [Section 3.7.2, "The Sterling Greex Editor - Setup Phase 2"](#).
4. Right-click the folder that contains all \*.greex files, which contains an advanced XML condition that you want to load into the database. Select Greex > Export Greex Rule to Database from the pop-up menu. The Export Greex Rule to Database window displays.

Process Type Key 20030708142438541

Owner Key AIRB

< Back Next > Finish Cancel

**Table 3–6** *Select the Process Type and Owner Key Window*

Field	Description
Process Type Key	Select the key of the appropriate process type from the drop-down list.
Owner Key	Select the key of the appropriate owner from the drop-down list.

5. Click Next. The Select Greex Rules window displays.

Process Type Key And Owner Key

Select the greex rules that are to be inserted with the Process Type Key 20030708142438541 and Owner Key AIRB. The rules which cannot be inserted into the database will be disabled. The reason for disabling them can be seen as the tooltip of the corresponding "Condition Id" label.

Select All Unselect All

List of greex files

<input checked="" type="checkbox"/>	test.greex	Condition Id	test
-------------------------------------	------------	--------------	------

Commit To Database  Create Factory Settings XML File

Factory Settings File  ...

? < Back Next > Finish Cancel

**Table 3–7 Select Greex Rules Window**

Field	Description
Process Type Key and Owner Key	Displays the process type and owner keys that you selected.
Select All	Click this button if you want to select all *.greex files listed in List of Greex Files panel.
Unselect All	Click this button if you want to unselect all *.greex files listed in the List of Greex Files panel.
<b>List of Greex files</b>	
Select the *.greex files that you want to load into the database.	
Connection Id	Displays the unique identifier of a Greex rule defined in each *.greex file that you select.
Commit To Database	Choose this option to commit the selected Greex rules to the database.
Create Factory Settings XML File	Choose this option to create the factory setup XML file for the selected Greex rules.
Factory Settings File	Enter the path and name of the new factory settings XML file in which you want to store the YFS_CONDITION Factory Settings.  Click  . The Select XML File to Store the YFS_CONDITION Factory Settings pop-up window displays. Select the factory settings XML file in which you want to store the YFS_CONDITION Factory Settings.

6. Click Finish.

### 3.12 Importing Advanced XML Conditions From the Database

You can import the existing advanced XML conditions or Greex rules from the database to your local machine. This is useful if you want to modify the structure of an advanced XML condition. For example, adding new expressions or IF and ELSE constructs to an existing advanced XML condition.

### 3.12.1 Before Importing an Advanced XML Condition

1. Modify the `<ECLIPSE_HOME>/plugins/com.yantra.ide.othertools.greex.editor_<VERSION_NUMBER>/lib/greexide.properties` file and specify the JDBC connection properties based on the database you are using. `<ECLIPSE_HOME>` is the directory where Eclipse SDK is installed. `<VERSION_NUMBER>` is the current version number of the Greex editor plug-in.
2. Depending on the database, copy the required database driver jar file to the `<ECLIPSE_HOME>/plugins/com.yantra.ide.othertools.greex.editor_<VERSION_NUMBER>/lib` directory. For example, if you are using an Oracle database, copy the `ojdbc14.jar` file.
3. Edit the `<ECLIPSE_HOME>/plugins/com.yantra.ide.othertools.greex.editor_<VERSION_NUMBER>/META-INF/MANIFEST.MF` file and add the relative path (relative to the `<ECLIPSE_HOME>/plugins/com.yantra.ide.othertools.greex.editor_<VERSION_NUMBER>` directory) of the required database driver jar file in the `Bundle-ClassPath` property.

For example, if you are using an Oracle database, add `lib/ojdbc14.jar` to the `Bundle-ClassPath` property, as follows:

```
Bundle-ClassPath: greexeditor.jar,
lib/ojdbc14.jar
```

4. If Eclipse SDK is already running, restart Eclipse in clean mode, as follows:
 

```
eclipse.exe -clean
```

Importing an Advanced XML Condition

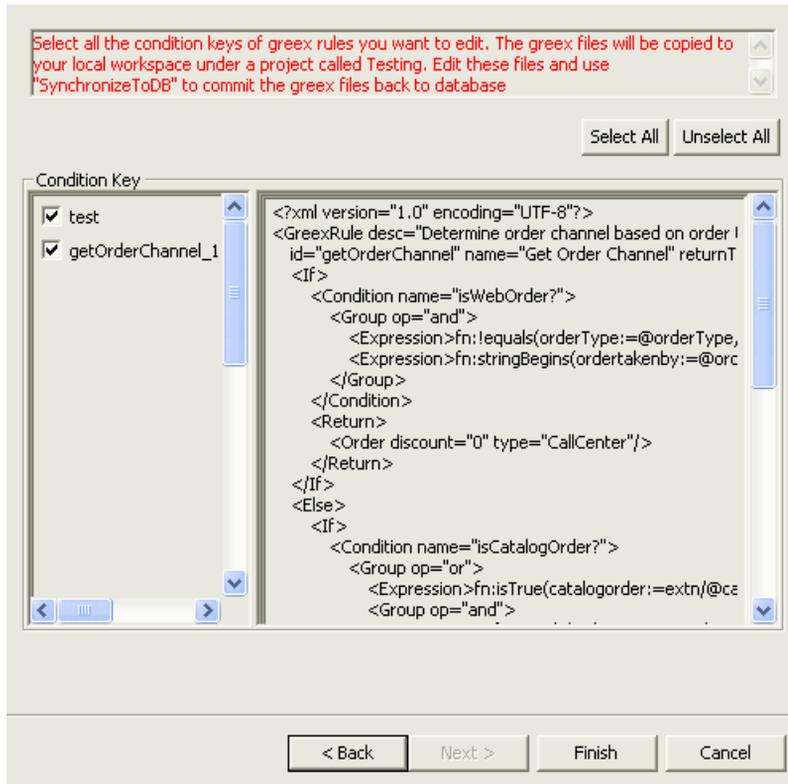
### 3.12.2 Steps to Import Advanced XML Conditions

To import advanced XML conditions from the database:

1. Start Eclipse SDK.
2. From the menu bar, select `Window > Show View > Navigator`. The java projects display in the Navigator view.

3. Right-click the java project in which you want to import the \*.greex files from the database.
4. Select Greex > Import Greex Rule from Database from the pop-up menu. The Import Greex Rule from Database window displays. Each \*.greex file contains an advanced XML condition or Greex rule.

For more information about creating a java project, see [Section 3.7.2, "The Sterling Greex Editor - Setup Phase 2"](#).



**Table 3–8 Select Greex Rules Window**

Field	Description
Select All	Click this button if you want to select all *.greex files listed in the Condition Key panel.
Unselect All	Click this button if you want to unselect all *.greex files listed in the Condition Key panel.

**Table 3–8 Select Greex Rules Window**

Field	Description
Condition Key	Displays the list of condition identifiers specified for each *.greex file loaded in the database. If you position the mouse on a particular condition identifier, in the right panel, the contents of the selected advanced XML condition or Greex rule displays. For example, Greex rule or the advanced XML condition identifier, name, description, and so forth. All IF and ELSE constructs defined for that advanced XML condition also displays. Select or unselect the appropriate *.greex files that you do not want to copy to your local machine.

5. Click Finish. The selected \*.greex files are imported into the java project.
6. Open the \*.greex files using the Sterling Greex editor and modify the advanced XML condition or Greex rule as needed.
7. Reload the modified \*.greex files into the database. For more information about reloading the advanced XML conditions into the database, see [Section 3.11, "Loading Advanced XML Conditions into the Database"](#).

### 3.13 Modifying an Advanced XML Condition

Using the Applications Manager, you can only change the modifiable parameters of an advanced XML condition. You cannot change the structure of an advanced XML condition. For more information about modifying advanced XML conditions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



## C

---

- condition builder, 17
  - adding custom attributes, 17
    - creating `extn_conditionbuilder.xml` file, 18
    - during condition definition, 20 to 21
    - process type based, 17 to 20
  - providing condition properties, 22
    - defining `YCPDynamicConditionEx` interface, 22
- customization checklist, 13
- customizing
  - condition builder, 17 to 23





























