

# ***Sterling Commerce***

*An IBM Company*

# **Sterling Supply Collaboration: Configuration Guide**

Release 8.5

*Last updated in HF22*

August 2010



# Copyright Notice

Copyright © 1999 - 2010

Sterling Commerce, Inc.

ALL RIGHTS RESERVED

## STERLING COMMERCE SOFTWARE

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

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

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

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

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

---

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

Copyright © 1999 - 2010

## Third-Party Software

Portions of the Sterling Commerce Software may include products, or may be distributed on the same storage media with products, ("Third Party Software") offered by third parties ("Third Party Licensors"). Sterling Commerce Software may include Third Party Software covered by the following copyrights: Copyright © 2006-2008 Andres Almiray. Copyright © 1999-2005 The Apache Software Foundation. Copyright (c) 2008 Azer Koçulu <http://azer.kodfabrik.com>. Copyright © Einar Lielmanis, einars@gmail.com. Copyright (c) 2006 John Reilly ([www.inconspicuous.org](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

---

## Preface

Intended Audience .....	xxix
Structure .....	xxix
Selling and Fulfillment Foundation Documentation .....	xxxii
Conventions .....	xxxiv

## 1 Introduction

1.1 Business Models.....	2
1.1.1 Multi-Divisional Corporation.....	2
1.1.2 Third-Party Logistics.....	3
1.1.3 Marketplace.....	3
1.2 Supply Collaboration Configuration.....	4
1.2.1 Logistics.....	5
1.2.2 Negotiation.....	5
1.2.3 Vendor.....	5
1.2.4 Order Attributes.....	5
1.2.5 Order Validation.....	5
1.2.6 Instruction Types.....	6
1.2.7 Modification Reasons.....	6
1.2.8 Backorder Reasons.....	6
1.2.9 Process Type Configuration.....	6
1.2.10 Financials.....	7
1.2.11 Purge Criteria.....	7

## 2 Navigating the Applications Manager

2.1	Starting the Applications Manager .....	9
2.2	The Applications Manager Layout.....	10
2.2.1	Application Rules Side Panel .....	12
2.2.1.1	Accessing Configuration Screens.....	13
2.2.1.2	Determining Inheritance.....	14
2.2.1.3	Loading Another Organization's Rules.....	21
2.2.2	Work Area .....	23
2.2.2.1	Search Window .....	23
2.2.2.2	List Window .....	24
2.2.2.3	Details Window .....	25
2.2.2.4	Drag and Drop Window .....	26
2.3	Actions Available in the Applications Manager.....	28
2.3.1	Using the Applications Manager's Lookup Functionality .....	28
2.3.2	Viewing the Document Types Associated with an Application .....	29
2.3.2.1	Adding a Document Type to an Application .....	30
2.3.3	Viewing the User Logged into the Applications Manager .....	31
2.3.4	Using Lists and List Filtering.....	31
2.3.5	Date and Time Entry.....	34
2.3.6	Using On-Line Help .....	34
2.3.7	Troubleshooting Errors .....	34
2.3.8	Using Special Characters .....	35

## 3 Configuring Cross Application Logistics Components

3.1	Defining Logistics Attributes.....	37
3.1.1	Defining Freight Terms.....	37
3.1.1.1	Creating a Freight Term .....	38
3.1.1.2	Modifying a Freight Term.....	40
3.1.1.3	Deleting a Freight Term.....	40
3.1.2	Defining Carrier Modification Reasons.....	41
3.1.2.1	Creating a Carrier Modification Reason .....	41
3.1.2.2	Modifying a Carrier Modification Reason.....	42
3.1.2.3	Deleting a Carrier Modification Reason .....	42
3.1.3	Defining Additional Logistic Rules.....	43

3.2	Defining Delivery Codes .....	45
3.2.1	Creating a Delivery Code .....	45
3.2.2	Modifying a Delivery Code .....	46
3.2.3	Deleting a Delivery Code .....	46
3.3	Defining Shipment Modes .....	46
3.3.1	Creating a Shipment Mode .....	47
3.3.2	Modifying a Shipment Mode .....	48
3.3.3	Deleting a Shipment Mode .....	48
3.4	Defining Outbound Constraints .....	48
3.4.1	Creating a Routing Guide .....	51
3.4.2	Modifying a Routing Guide .....	53
3.4.2.1	Creating a Routing Guide Line .....	54
3.4.2.2	Modifying a Routing Guide Line .....	62
3.4.2.3	Deleting a Routing Guide Line .....	62
3.4.3	Deleting a Routing Guide .....	62

## **4 Configuring Cross Application Negotiation Components**

4.1	Defining Response Actions .....	63
4.1.1	Modifying a Response Action .....	64
4.2	Defining Rejection Reasons .....	64
4.2.1	Creating a Rejection Reason .....	65
4.2.2	Modifying a Rejection Reason .....	65
4.2.3	Deleting a Rejection Reason .....	66

## **5 Configuring Cross Application Vendor Components**

5.1	Defining Vendor Classifications .....	67
5.1.1	Creating a Vendor Classification .....	67
5.1.2	Modifying a Vendor Classification .....	68
5.1.3	Deleting a Vendor Classification .....	69
5.2	Defining Vendor Definitions .....	69
5.2.1	Creating a Vendor Definition .....	69
5.2.2	Modifying a Vendor Definition .....	71
5.2.3	Deleting a Vendor Definition .....	72
5.3	Defining Contact Types .....	72
5.3.1	Creating a Contact Type .....	72

5.3.2	Modifying a Contact Type .....	73
5.3.3	Deleting a Contact Type .....	74

## 6 Configuring a Document's Attributes

6.1	Defining Order Types .....	75
6.1.1	Creating an Order Type .....	76
6.1.2	Modifying an Order Type .....	76
6.1.3	Deleting an Order Type .....	77
6.2	Defining Order Sources .....	77
6.2.1	Creating an Order Source .....	77
6.2.2	Modifying an Order Source .....	78
6.2.3	Deleting an Order Source .....	79
6.3	Defining External References for the Order Level .....	79
6.3.1	Creating an External Reference for the Order Header Level .....	79
6.3.2	Modifying an External Reference for the Order Header Level .....	80
6.3.3	Deleting an External Reference for the Order Header Level .....	81
6.4	Defining External References for the Order Line Level .....	81
6.4.1	Creating an External Reference for the Order Line Level .....	81
6.4.2	Modifying an External Reference for the Order Line Level .....	82
6.4.3	Deleting an External Reference for the Order Line Level .....	83
6.5	Defining Order Address Types .....	83
6.5.1	Creating an Order Address Type .....	83
6.5.2	Modifying an Order Address Type .....	84
6.5.3	Deleting an Order Address Type .....	85
6.6	Defining Line Types .....	85
6.6.1	Creating a Line Type .....	85
6.6.2	Modifying a Line Type .....	86
6.6.3	Deleting a Line Type .....	86
6.7	Defining Other Attributes .....	87
6.7.1	Generating a Prime Line Number for a New Line from a Pre-Configured Number .....	87

## **7 Configuring a Document's Order Validation**

## **8 Configuring a Document's Instruction Types**

8.1	Creating an Instruction Type.....	93
8.2	Modifying an Instruction Type.....	94
8.3	Deleting an Instruction Type.....	95

## **9 Configuring a Document's Modification Reasons**

9.1	Creating a Modification Reason .....	97
9.2	Modifying a Modification Reason.....	99
9.3	Deleting a Modification Reason.....	100

## **10 Configuring a Document's Backorder Reasons**

10.1	Creating a Backorder Reason .....	101
10.2	Modifying a Backorder Reason .....	102
10.3	Deleting a Backorder Reason .....	103

## **11 Configuring a Document's Modification Components**

11.1	Defining Modification Rules.....	106
11.1.1	Changing Modification Rules .....	109
11.2	Defining Custom Modification Types.....	110
11.2.1	Creating a Custom Modification Type .....	111
11.2.2	Modifying a Custom Modification Type.....	113
11.2.3	Deleting a Custom Modification Type.....	113
11.3	Defining Modifications Impacting Pricing.....	114
11.3.1	Adding/Removing a Modification Type for Modifications Impacting Pricing .....	114
11.4	Defining Modifications Requiring Auditing .....	114

## **12 Configuring a Document's Note Reasons**

12.1	Creating a Note Reason.....	117
12.2	Modifying a Note Reason .....	118
12.3	Creating a New Note Reason Based on an Existing One .....	118

12.4	Deleting a Note Reason .....	119
------	------------------------------	-----

## **13 Configuring a Document's Line Relationship Type**

13.1	Defining a Line Relationship Type .....	121
13.2	Modifying a Line Relationship Type .....	122
13.3	Creating a New Line Relationship Type Based on an Existing One .....	123
13.3.1	Deleting a Line Relationship Type .....	123

## **14 Configuring an Inbound Order Document's Fulfillment Specific Components**

14.1	Defining Hold Types .....	126
14.1.1	Creating a Hold Type .....	126
14.1.1.1	Creating an Order Level Hold Type .....	127
14.1.1.2	Creating an Order Line Level Hold Type .....	134
14.1.2	Modifying a Hold Type .....	141
14.1.3	Deleting a Hold Type .....	142
14.2	Defining Process Type Details .....	142
14.3	Process Type Pipeline Configuration .....	142
14.3.1	Defining Pipeline Determination .....	143
14.3.1.1	Condition Variables for Pipeline Determination .....	144
14.3.2	Pipelines .....	144
14.3.3	Transactions .....	146
14.3.4	Statuses .....	150
14.3.5	Conditions .....	153
14.3.6	Actions .....	155
14.4	Defining Transaction Rules .....	155
14.5	Defining Status Inventory Types .....	160
14.5.1	Creating a Status Inventory Type .....	162
14.5.2	Modifying a Status Inventory Type .....	164
14.5.3	Deleting a Status Inventory Type .....	164
14.6	Defining Monitoring Components .....	164
14.6.1	Defining Date Types .....	165
14.6.1.1	Creating a Date Type .....	165
14.6.1.2	Modifying a Date Type .....	166

14.6.1.3	Deleting a Date Type .....	166
14.6.2	Defining Milestones .....	167
14.6.2.1	Creating a Milestone .....	167
14.6.2.2	Modifying a Milestone .....	169
14.6.2.3	Deleting a Milestone .....	170
14.7	Defining Monitoring Events .....	170
14.7.1	Creating an Event Rule .....	170
14.7.2	Modifying an Event .....	174
14.7.3	Deleting an Event.....	174

## **15 Configuring an Order Document's Shipment Specific Components**

15.1	Defining Hold Types.....	176
15.2	Defining Process Type Details .....	176
15.3	Process Type Pipeline Configuration .....	176
15.3.1	Defining Pipeline Determination.....	177
15.3.2	Pipelines .....	178
15.3.3	Transactions.....	180
15.3.4	Statuses .....	183
15.3.5	Conditions.....	185
15.3.6	Actions .....	186
15.4	Defining Monitoring Components .....	187
15.4.1	Defining Date Types .....	187
15.4.1.1	Creating a Date Type .....	188
15.4.1.2	Modifying a Date Type .....	189
15.4.1.3	Deleting a Date Type .....	189
15.4.2	Defining Milestones .....	189
15.4.2.1	Creating a Milestone .....	190
15.4.2.2	Modifying a Milestone .....	192
15.4.2.3	Deleting a Milestone .....	192
15.5	Defining Monitoring Events .....	192
15.5.1	Creating an Event Rule .....	193
15.5.2	Modifying an Event .....	195
15.5.3	Deleting an Event.....	196
15.6	Defining Shipment Preferences.....	196

15.6.1	Over Shipping Preferences .....	196
15.6.1.1	Creating a Shipment Preference .....	197
15.6.1.2	Modifying a Shipment Preference.....	199
15.6.1.3	Deleting a Shipment Preference.....	199
15.6.2	Transaction Rules .....	199

## **16 Configuring an Order Document’s Receipt-Specific Components**

16.1	Defining Process Type Details .....	202
16.2	Process Type Pipeline Configuration .....	202
16.2.1	Defining Pipeline Determination.....	203
16.2.2	Pipelines .....	204
16.2.3	Transactions .....	205
16.2.4	Statuses .....	207
16.2.5	Conditions .....	209
16.2.6	Actions .....	211
16.3	Defining Receipt Preferences.....	211
16.3.1	Creating a Receipt Preference .....	212
16.3.2	Modifying a Receipt Preference.....	214
16.3.3	Deleting a Receipt Preference.....	214
16.4	Defining Receiving Dispositions .....	215
16.4.1	Creating a Receiving Disposition .....	215
16.4.2	Modifying a Receiving Disposition .....	217
16.4.3	Deleting a Receiving Disposition .....	217
16.5	Defining Receiving Discrepancy Reasons.....	218
16.5.1	Creating a Receiving Discrepancy Reason .....	218
16.5.2	Modifying a Receiving Discrepancy Reason.....	220
16.5.3	Deleting a Receiving Discrepancy Reason .....	221

## **17 Configuring an Order Document’s Negotiation-Specific Components**

17.1	Defining Process Type Details .....	224
17.2	Process Type Pipeline Configuration .....	224
17.2.1	Defining Pipeline Determination.....	225

17.2.2	Pipelines .....	226
17.2.3	Transactions .....	228
17.2.4	Statuses .....	230
17.2.5	Conditions .....	232
17.2.6	Actions .....	233

## **18 Configuring a Document's Financial Components**

18.1	Defining Payment Terms .....	235
18.1.1	Creating a Payment Term .....	235
18.1.2	Modifying a Payment Term .....	236
18.1.3	Deleting a Payment Term .....	237
18.2	Defining Charge Definitions .....	237
18.2.1	Creating a Charge Category .....	238
18.2.1.1	Adding a Charge Name Associated with a Charge Category .....	240
18.2.1.2	Modifying a Charge Name Associated with a Charge Category .....	241
18.2.1.3	Deleting a Charge Name Associated with a Charge Category .....	241
18.2.2	Modifying a Charge Category .....	242
18.2.3	Deleting a Charge Category .....	242
18.3	Defining Tax Names .....	242
18.3.1	Creating a Tax Name .....	243
18.3.2	Modifying a Tax Name .....	244
18.3.3	Deleting a Tax Name .....	244
18.4	Defining Additional Payment Rules .....	245

## **19 Configuring a Document's Purge Criteria**

19.1	Modifying an Order Document Type's Purge Criteria Rule .....	249
------	--	-----

## **A Time-Triggered Transaction Reference**

A.1	Running Time-Triggered Transactions .....	255
A.1.1	Steps to Complete Before Scheduling Time-Triggered Transactions ...	255
A.2	Configuring Communication Between an Agent and a JMS Server .....	256
A.2.1	Prerequisites .....	256
A.2.2	Create an Initial Context Factory Code .....	257
A.2.3	Define the Transaction Information .....	258

A.3	Business Process Time-Triggered Transactions .....	259
A.3.1	Asynchronous Request Processor .....	260
A.3.2	Case Insensitive Data Loader .....	262
A.3.3	Change Load Status.....	264
A.3.4	Change Shipment Status .....	266
A.3.5	Close Delivery Plan .....	268
A.3.6	Close Load .....	270
A.3.7	Close Manifest .....	272
A.3.8	Close Order .....	275
A.3.9	Close Receipts .....	277
A.3.10	Close Shipment .....	279
A.3.11	Collect Shipment Statistics .....	281
A.3.12	Consolidate Additional Inventory .....	284
A.3.13	Consolidate To Shipment .....	286
A.3.14	Create Catalog Index .....	290
A.3.15	Create Chained Order .....	294
A.3.16	Create Derived Order .....	296
A.3.17	Create Order Invoice .....	298
A.3.18	Create Shipment Invoice .....	300
A.3.19	ESP Evaluator .....	302
A.3.20	Item Based Allocation .....	305
A.3.21	Mark Load as Trailer Loaded .....	311
A.3.22	Match Inventory.....	313
A.3.23	Payment Collection .....	315
A.3.24	Payment Execution .....	318
A.3.25	Post Inventory Match .....	321
A.3.26	Process Order Hold Type .....	323
A.3.27	Process Work Order Hold Type .....	326
A.3.28	Publish Negotiation Results .....	328
A.3.29	Release.....	330
A.3.30	Route Shipment .....	333
A.3.31	Schedule .....	336
A.3.32	Send Invoice .....	340
A.3.33	Send Item Changes .....	343
A.3.34	Send Customer Changes .....	345

A.3.35	Send Order .....	347
A.3.36	Send Release.....	349
A.3.37	Start Order Negotiation.....	351
A.3.38	Synchronize Colony Map .....	352
A.3.39	Update Best Match Region .....	354
A.3.40	PopulateOwnershipTransferSummary .....	357
A.4	Time-Triggered Purge Transactions.....	358
A.4.1	Purge Strategy.....	359
A.4.2	Configuring Purge Transaction Log Files.....	359
A.4.3	Available Purges.....	360
A.4.3.1	Alert Purge .....	361
A.4.3.2	Capacity Purge.....	364
A.4.3.3	Draft Order History Purge .....	367
A.4.3.4	Draft Order Purge.....	370
A.4.3.5	Delivery Plan Purge .....	374
A.4.3.6	Export Table Purge .....	377
A.4.3.7	Import Table Purge.....	379
A.4.3.8	Inventory Audit Purge .....	382
A.4.3.9	Inventory Purge .....	385
A.4.3.10	Inventory Supply Temp Purge .....	388
A.4.3.11	Item Audit Purge.....	390
A.4.3.12	Load History Purge .....	392
A.4.3.13	Load Purge.....	395
A.4.3.14	Negotiation History Purge .....	398
A.4.3.15	Negotiation Purge.....	400
A.4.3.16	Order History Purge.....	403
A.4.3.17	Order Purge.....	406
A.4.3.18	Order Release Status Purge .....	413
A.4.3.19	Order Status Audit Purge.....	416
A.4.3.20	Organization Audit Purge.....	418
A.4.3.21	Person Info Purge.....	420
A.4.3.22	Person Info History Purge.....	423
A.4.3.23	Picklist Purge .....	426
A.4.3.24	Price List Purge .....	428
A.4.3.25	Purge Catalog Mass Audits.....	430

A.4.3.26	Receipt History Purge.....	432
A.4.3.27	Receipt Purge .....	435
A.4.3.28	Reprocess Error Purge.....	438
A.4.3.29	Reservation Purge .....	440
A.4.3.30	Shipment History Purge.....	442
A.4.3.31	Shipment Purge .....	444
A.4.3.32	Shipment Statistics Purge.....	448
A.4.3.33	Statistics Purge .....	450
A.4.3.34	User Activity Purge .....	453
A.4.3.35	User Activity Audit Purge .....	455
A.4.3.36	Work Order History Purge.....	457
A.4.3.37	Work Order Purge.....	460
A.4.3.38	YFS Audit Purge .....	463
A.4.3.39	YFSInventoryOwnershipAudit Purge .....	466
A.4.3.40	Password Reset Request Purge.....	467
A.4.3.41	User Login Failure Purge.....	469
A.5	Task Queue Syncher Time-Triggered Transactions .....	471
A.5.1	Load Execution Task Queue Syncher .....	472
A.5.2	Order Delivery Task Queue Syncher .....	474
A.5.3	Order Fulfillment Task Queue Syncher.....	475
A.5.4	Order Negotiation Task Queue Syncher .....	476
A.6	Monitors .....	478
A.6.1	Availability Monitor .....	478
A.6.2	Exception Monitor.....	480
A.6.3	Inventory Monitor.....	483
A.6.4	Negotiation Monitor .....	485
A.6.5	Enhanced Order Monitor .....	487
A.6.6	Enhanced Return Monitor .....	491
A.6.7	Real-time Availability Monitor.....	495
A.6.8	Shipment Monitor.....	501
A.6.9	Work Order Monitor .....	504

## **B Order Modification Types**

## **C Condition Builder Attributes**

C.1	Sales Order .....	525
C.1.1	Order Fulfillment .....	525
C.1.2	Order Negotiation.....	528
C.1.3	Outbound Shipment.....	529
C.1.4	Sales Order Receipt .....	530
C.2	Planned Order .....	531
C.2.1	Planned Order Execution .....	531
C.2.2	Planned Order Negotiation .....	531
C.3	Return Order .....	531
C.3.1	Reverse Logistics .....	531
C.3.2	Return Shipment.....	533
C.3.3	Return Receipt.....	534
C.4	Template Order.....	535
C.5	Purchase Order .....	535
C.5.1	Purchase Order Execution.....	535
C.5.2	Purchase Order Negotiation .....	538
C.5.3	Inbound Shipment.....	539
C.5.4	Purchase Order Receipt .....	539
C.6	Transfer Order .....	539
C.6.1	Transfer Order Execution.....	539
C.6.2	Transfer Order Delivery.....	539
C.6.3	Transfer Order Receipt .....	539
C.7	Master Order .....	540
C.7.1	Master Order Fulfillment.....	540
C.8	Load Execution .....	542
C.9	General .....	543
C.10	WMS Putaway.....	545
C.11	WMS Layout Definition .....	545
C.12	WMS Inventory .....	545
C.13	Trailer Loading.....	545
C.14	Task Execution .....	545
C.15	Move Request Execution .....	546

C.16	Manifesting.....	546
C.17	Over Pack Build.....	546
C.18	Count Execution.....	546
C.19	Pack Process.....	548
C.20	Outbound Picking .....	551
C.21	VAS Process .....	551
C.22	Item-Based Allocation (IBA) Order.....	553

## Index

# Preface

---

This manual describes how to use the Applications Manager.

## Intended Audience

This manual is intended for use by system administrators and managers who need to configure the Selling and Fulfillment Foundation rules, participants, users, and business processes as they pertain to their business practices.

## Structure

This manual contains the following sections:

### **Chapter 1, "Introduction"**

This chapter briefly describes the contents of this guide.

### **Chapter 2, "Navigating the Configurator"**

This chapter explains the layout of the Applications Manager, actions you can perform throughout the application, and important concepts you should be aware of before using the application.

### **Chapter 5, "Configuring Cross Application Logistics Components"**

This chapter explains the configuration of components used by different logistics related functionality throughout the business application module.

## **Chapter 4, "Configuring Cross Application Negotiation Components"**

This chapter explains how you can configure rules and common codes that pertain to the negotiation process. Negotiation is the process in which two organizations can settle the conditions of an order document.

## **Chapter 5, "Configuring Cross Application Vendor Components"**

This chapter explains how you can configure trading vendors in the Supply Collaboration module.

## **Chapter 9, "Configuring a Document's Attributes"**

This chapter explains how you can configure common codes as they pertain to order documents viewed in the Application Consoles.

## **Chapter 10, "Configuring a Document's Order Validation"**

This chapter explains how you can define configuration for defaulting Seller and Buyer validation during order creation for a particular Enterprise and document type.

## **Chapter 11, "Configuring a Document's Instruction Types"**

This chapter explains how you can define the common codes used when adding special instructions to an order document.

## **Chapter 12, "Configuring a Document's Modification Reasons"**

This chapter explains how you can define common codes for modification reasons.

## **Chapter 13, "Configuring a Document's Backorder Reasons"**

This chapter explains how you can define common codes for backorder reasons.

## **Chapter 16, "Configuring a Document's Modification Components"**

This chapter explains how you can configure the modification rules and types of a document when it is in a specific status.

## **Chapter 14, "Configuring a Document's Note Reasons"**

This chapter explains how you can configure common codes for note reasons used when modifying an inbound order.

## **Chapter 15, "Configuring a Document's Line Relationship Type"**

This chapter explains how you can configure line relationship types used for linking similar items.

## **Chapter 14, "Configuring an Inbound Order Document's Fulfillment Specific Components"**

This chapter explains how you can configure the fulfillment specific process types used by each inbound order document.

## **Chapter 15, "Configuring an Order Document's Shipment Specific Components"**

This chapter explains how you can configure the shipment specific process types used by each order document.

## **Chapter 16, "Configuring an Order Document's Receipt-Specific Components"**

This chapter explains how you can configure the receipt-specific process types used by each order document.

## **Chapter 17, "Configuring an Order Document's Negotiation-Specific Components"**

This chapter explains how you can configure the negotiation-specific process types used by each order document.

## **Chapter 18, "Configuring a Document's Financial Components"**

This chapter explains how you can define rules and common codes as they pertain to payments and charges given for a given order document.

## **Chapter 20, "Configuring a Document's Purge Criteria"**

This chapter explains the purge criteria business rules that are used to define qualifications around each type of purge. Purges are the process by which old data is removed from the system database.

## **Appendix A, "Time-Triggered Transaction Reference"**

This appendix explains time-triggered transactions that are utilities that perform a variety of individual functions, automatically and at specific time intervals.

## **Appendix B, "Order Modification Types"**

This appendix explains the default order modification types and their associated modification levels.

## **Appendix F, "Condition Builder Attributes"**

This appendix explains the attributes used in the condition builder to build statements for each process type.

# **Selling and Fulfillment Foundation Documentation**

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

- *Selling and Fulfillment Foundation: Release Notes*
- *Selling and Fulfillment Foundation: Installation Guide*
- *Selling and Fulfillment Foundation: Upgrade Guide*
- *Selling and Fulfillment Foundation: Configuration Deployment Tool Guide*
- *Selling and Fulfillment Foundation: Performance Management Guide*
- *Selling and Fulfillment Foundation: High Availability Guide*
- *Selling and Fulfillment Foundation: System Management Guide*
- *Selling and Fulfillment Foundation: Localization Guide*
- *Selling and Fulfillment Foundation: Customization Basics Guide*
- *Selling and Fulfillment Foundation: Customizing APIs Guide*
- *Selling and Fulfillment Foundation: Customizing Console JSP Interface for End User Guide*
- *Selling and Fulfillment Foundation: Customizing the RCP Interface Guide*
- *Selling and Fulfillment Foundation: Customizing User Interfaces for Mobile Devices Guide*
- *Selling and Fulfillment Foundation: Customizing Web UI Framework Guide*

- *Selling and Fulfillment Foundation: Customizing Swing Interface Guide*
- *Selling and Fulfillment Foundation: Extending the Condition Builder Guide*
- *Selling and Fulfillment Foundation: Extending the Database Guide*
- *Selling and Fulfillment Foundation: Extending Transactions Guide*
- *Selling and Fulfillment Foundation: Using Sterling RCP Extensibility Tool Guide*
- *Selling and Fulfillment Foundation: Integration Guide*
- *Selling and Fulfillment Foundation: Product Concepts Guide*
- *Sterling Warehouse Management™ System: Concepts Guide*
- *Selling and Fulfillment Foundation: Application Platform Configuration Guide*
- *Sterling Distributed Order Management™: Configuration Guide*
- *Sterling Supply Collaboration: Configuration Guide*
- *Sterling Global Inventory Visibility™: Configuration Guide*
- *Catalog Management™: Configuration Guide*
- *Sterling Logistics Management: Configuration Guide*
- *Sterling Reverse Logistics™: Configuration Guide*
- *Sterling Warehouse Management System: Configuration Guide*
- *Selling and Fulfillment Foundation: Application Platform User Guide*
- *Sterling Distributed Order Management: User Guide*
- *Sterling Supply Collaboration: User Guide*
- *Sterling Global Inventory Visibility: User Guide*
- *Sterling Logistics Management: User Guide*
- *Sterling Reverse Logistics: User Guide*
- *Sterling Warehouse Management System: User Guide*
- *Selling and Fulfillment Foundation: Mobile Application User Guide*
- *Selling and Fulfillment Foundation: Business Intelligence Guide*

- *Selling and Fulfillment Foundation: Javadocs*
- *Sterling Selling and Fulfillment Suite™: Glossary*
- *Parcel Carrier: Adapter Guide*
- *Selling and Fulfillment Foundation: Multitenant Enterprise Guide*
- *Selling and Fulfillment Foundation: Password Policy Management Guide*
- *Selling and Fulfillment Foundation: Properties Guide*
- *Selling and Fulfillment Foundation: Catalog Management Concepts Guide*
- *Selling and Fulfillment Foundation: Pricing Concepts Guide*
- *Business Center: Item Administration Guide*
- *Business Center: Pricing Administration Guide*
- *Business Center: Customization Guide*
- *Business Center: Localization Guide*

## Conventions

The following conventions may be used in this manual:

Convention	Meaning
. . .	Ellipsis represents information that has been omitted.
< >	Angle brackets indicate user-supplied input.
mono-spaced text	Mono-spaced text indicates a file name, directory path, attribute name, or an inline code example or command.
/ or \	Slashes and backslashes are file separators for Windows, UNIX, and Linux operating systems. The file separator for the Windows operating system is "\" and the file separator for UNIX and Linux systems is "/". The UNIX convention is used unless otherwise mentioned.
<INSTALL_DIR>	User-supplied location of the Selling and Fulfillment Foundation installation directory. This is only applicable for Release 8.0 or later.

Convention	Meaning
<INSTALL_DIR_OLD>	User-supplied location of the Selling and Fulfillment Foundation installation directory (for Release 8.0 or later). <b>Note:</b> This is applicable only for users upgrading from Release 8.0 or later.
<YANTRA_HOME>	User-supplied location of the Sterling Supply Chain Applications installation directory. This is only applicable for Releases 7.7, 7.9, and 7.11.
<YANTRA_HOME_OLD>	User-supplied location of the Sterling Supply Chain Applications installation directory (for Releases 7.7, 7.9, or 7.11). <b>Note:</b> This is applicable only for users upgrading from Releases 7.7, 7.9, or 7.11.
<YFS_HOME>	For Releases 7.3, 7.5, and 7.5 SP1, this is the user-supplied location of the Sterling Supply Chain Applications installation directory. For Releases 7.7, 7.9, and 7.11, this is the user-supplied location of the <YANTRA_HOME>/Runtime directory. For Release 8.0 or above, the <YANTRA_HOME>/Runtime directory is no longer used and this is the same location as <INSTALL_DIR>.
<YFS_HOME_OLD>	This is the <YANTRA_HOME>/Runtime directory for Releases 7.7, 7.9, or 7.11. <b>Note:</b> This is only applicable for users upgrading from Releases 7.7, 7.9, or 7.11.
<ANALYTICS_HOME>	User-supplied location of the Sterling Analytics installation directory. <b>Note:</b> This convention is used only in the <i>Selling and Fulfillment Foundation: Business Intelligence Guide</i> .
<COGNOS_HOME>	User-supplied location of the IBM Cognos 8 Business Intelligence installation directory. <b>Note:</b> This convention is used only in the <i>Selling and Fulfillment Foundation: Business Intelligence Guide</i> .

Convention	Meaning
<MQ_JAVA_INSTALL_PATH>	User-supplied location of the IBM WebSphere® MQ Java components installation directory. <b>Note:</b> This convention is used only in the <i>Selling and Fulfillment Foundation: System Manangement and Administration Guide</i> .
<DB>	Refers to Oracle®, IBM DB2®, or Microsoft SQL Server® depending on the database server.
<DB_TYPE>	Depending on the database used, considers the value oracle, db2, or sqlserver.

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

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



# Introduction

---

This book concentrates on the rules and setup configurations that make up the Supply Collaboration business application module in the Applications Manager. This book is intended for both Hub and Enterprise administrators using the Applications Manager to set up the Selling and Fulfillment Foundation environment. Business analysts should also use this book to plan appropriate business practices as they pertain to Selling and Fulfillment Foundation. Programmers should refer to the *Selling and Fulfillment Foundation: Customization Guide* for information about extending Selling and Fulfillment Foundation. System Integrators should refer to the *Selling and Fulfillment Foundation: Integration Guide* for information about extending or integrating external applications with Selling and Fulfillment Foundation.

---

---

**Important:** This book assumes that you have read and are familiar with the concepts and business functionality detailed in the *Selling and Fulfillment Foundation: Product Concepts Guide*.

---

---

The Applications Manager is a collection of all the rules and setup configurations necessary to implement Selling and Fulfillment Foundation grouped together by business application module. The following business application module configuration groupings are available in the Applications Manager:

- Distributed Order Management
- Global Inventory Visibility
- Catalog Management
- Logistics Management

- Supply Collaboration
- Reverse Logistics
- Warehouse Management
- Application Platform

There is no one specific way to configure the Selling and Fulfillment Foundation environment. Each should be modeled to fit in with your business practices. However, it is not possible to run Selling and Fulfillment Foundation without having the necessary infrastructure components configured.

## 1.1 Business Models

There is no single business model that encompasses the environment in which all the Selling and Fulfillment Foundation applications can be used. Therefore, there is no single way in which to configure your Selling and Fulfillment Foundation environment.

For example, your company might be considered a multi-divisional corporation, a third-party logistics company, or a marketplace business. Each of these business models require a different conceptual approach to the Selling and Fulfillment Foundation configuration.

### 1.1.1 Multi-Divisional Corporation

The **multi-divisional corporation model** is a business corporation whose primary focus is managing purchase and sales activities. A typical multi-divisional corporation can be a buyer, a seller, or both. It could also be a retailer, a manufacturer, or both. Whatever form the multi-divisional corporation takes, it normally has multiple channels with different types of customers, such as, consumers, retailers, dealers, and original equipment manufacturers.

The Internet has significantly altered this picture, with manufacturers able to go online to sell directly to consumers, and the role of suppliers to each retailer changing to include a host of different and hybrid fulfillment models.

In the multi-divisional corporation model, each division might be set up as an Enterprise in Selling and Fulfillment Foundation. This setup allows both segregation of transactions by division and global visibility at the

corporate level. Each Enterprise configures their own business rules, workflow, and transaction processing.

### 1.1.2 Third-Party Logistics

Traditional **third-party logistics** companies provide a range of outsourced services such as warehousing, transportation, and contract manufacturing.

The Internet has dramatically altered supply chain management and the third-party logistics industry. Large companies have begun to recognize the competitive advantage they can gain through the real-time management of their supply chains. These advantages include lower costs and improved customer service. Additionally, new sales channels such as web stores, hand-held devices, and in-store kiosks provide companies new methods of reaching their customers. All of these issues have increased the complexity of the fulfillment process.

Selling and Fulfillment Foundation provides the engine needed to run the operations of a contract fulfillment provider as well as a centralized system for real-time order execution and event-driven problem solving for an entire fulfillment network. It enables fulfillment providers to configure the fulfillment process to meet the needs of their clients.

In the third-party logistics model, each client might be set up as an Enterprise. This setup allows the third-party logistics Hub to have visibility of all transactions in the Hub environment, while the clients that are set up as Enterprises only have visibility to their own transactions. This allows the third-party logistics business to provide unique transaction processing to its clients.

### 1.1.3 Marketplace

A **marketplace** is an online intermediary that connects Buyers and Sellers. Marketplaces eliminate inefficiencies by aggregating offerings from many Sellers or by matching Buyers and Sellers in an exchange or auction. For Buyers, they lower purchasing costs and help them reach new Sellers. For Sellers, they lower sales costs and give them access to new customers. It is a central location, or Hub, where a trusted intermediary integrates both procedures and technology to lower the costs and enhance the effectiveness of Buyer and Seller transactions.

In the marketplace model, each market might be set up as an Enterprise. This setup allows each market to be unique with their own product or service handling.

## 1.2 Supply Collaboration Configuration

Selling and Fulfillment Foundation is a collection of common components used in the Supply Collaboration business application. The Sterling Supply Collaboration business application helps organizations effectively manage purchase order life cycles, including all interactions with external parties. It provides both internal and external users with immediate access to purchase orders, allowing them to negotiate changes, redirect orders and inventory, and resolve unexpected problems or delays.

In the Applications Manager you can use the Supply Collaboration configuration grouping to establish both cross-application and order document-specific rules and attributes. Cross-application rules and attributes can impact other applications, such as Distributed Order Management and Reverse Logistics. Order document specific rules and attributes pertain only to the order document type you are configuring, such as Purchase Order and Transfer Order. You can define different configurations for individual order document types without impacting other applications or order document types.

In the Applications Manager you can use the Supply Collaboration configuration grouping to establish the following aspects of Selling and Fulfillment Foundation for your business application modules:

- [Logistics](#)
- [Negotiation](#)
- [Vendor](#)
- [Order Attributes](#)
- [Order Validation](#)
- [Instruction Types](#)
- [Modification Reasons](#)
- [Backorder Reasons](#)
- [Process Type Configuration](#)
- [Financials](#)

- [Purge Criteria](#)

## 1.2.1 Logistics

**You can configure the components used by different logistics related functionality throughout the Distributed Order Management business application module.**

For more information about Logistics, see [Chapter 5, "Configuring Cross Application Logistics Components"](#).

## 1.2.2 Negotiation

**Negotiation is the process in which two organizations can settle the conditions of an order document. You can** define rules and common codes that pertain to the negotiation process.

For more information about Negotiation, see [Chapter 4, "Configuring Cross Application Negotiation Components"](#).

## 1.2.3 Vendor

You can define the vendors that sell to an organization in the Supply Collaboration module.

For more information about Vendor, see [Chapter 5, "Configuring Cross Application Vendor Components"](#).

## 1.2.4 Order Attributes

**You can** define common codes as they pertain to order documents viewed in the Application Consoles.

For more information about Order Attributes, see [Chapter 9, "Configuring a Document's Attributes"](#).

## 1.2.5 Order Validation

You can define configuration for defaulting the Buyer's vendor validation during order creation.

For more information about Order Validation, see [Chapter 10, "Configuring a Document's Order Validation"](#)

### 1.2.6 Instruction Types

You can define the common codes used when adding special instructions to an order.

For more information about instruction types, see [Chapter 11, "Configuring a Document's Instruction Types"](#).

### 1.2.7 Modification Reasons

You can define common codes for **modification reasons**. These codes define why a modification was made by a user.

For more information about Modification Reasons, see [Chapter 12, "Configuring a Document's Modification Reasons"](#).

### 1.2.8 Backorder Reasons

You can define common codes for **backorder reasons**. These codes describe why an order was backordered.

For more information about Backorder Reasons, see [Chapter 13, "Configuring a Document's Backorder Reasons"](#).

### 1.2.9 Process Type Configuration

To complete an order document's life cycle, each document has a set of different processes that it can go through. These processes are called process types. Every order document has a defined set of process types in Selling and Fulfillment Foundation.

The following process types are defined in Selling and Fulfillment Foundation for the order document types:

- Fulfillment
- Negotiation
- Shipment
- Receipt

You can configure the rules and components that define an order document's process types.

## 1.2.10 Financials

You can configure the components used by the Selling and Fulfillment Foundation financial engine.

For more information about Financials, see [Chapter 18, "Configuring a Document's Financial Components"](#).

## 1.2.11 Purge Criteria

You can define the parameters used when purging order document related records from the system.

For more information about Purge Criteria, see [Chapter 20, "Configuring a Document's Purge Criteria"](#).



# Navigating the Applications Manager

---

This chapter discusses the layout of the Applications Manager, actions you can perform throughout the application, and important concepts you should be aware of before using the application.

## 2.1 Starting the Applications Manager

To access the Applications Manager:

1. Point your browser to  
`http://<hostname>:<portname>/smcfs/console/start.jsp`

where,

- `hostname` is the computer name or IP address of the computer where Selling and Fulfillment Foundation is installed.
- `portnumber` is the listening port of the computer where Selling and Fulfillment Foundation is installed.

The browser displays the Sign In window.

2. Enter your login ID and password and choose the Sign In button. The Console Home Page is displayed.
3. From the menu bar, choose Configuration > Launch Configurator. The Applications Manager opens in a new window.

---

---

**Note:** Additionally, enterprise users who maintain an enterprise can access the Applications Manager by means of `http://<Selling and Fulfillment Foundation installation server>/smcfs/console/login.jsp`.

---

---

---

---

**Note:** If both the Applications Manager and the monitor in the System Management Console are opened at the same time, and if a dialogue window is opened in either application, the other stops responding to user input until that dialogue window is closed. This is due to a bug in the Java platform.

---

---

## 2.2 The Applications Manager Layout

The Applications Manager is a graphical user interface that can be used to configure different aspects of Selling and Fulfillment Foundation. The different configurations are defined by logical groupings called applications that can be accessed from the Configurator's menu bar.

*Figure 2–1 Applications Menu*



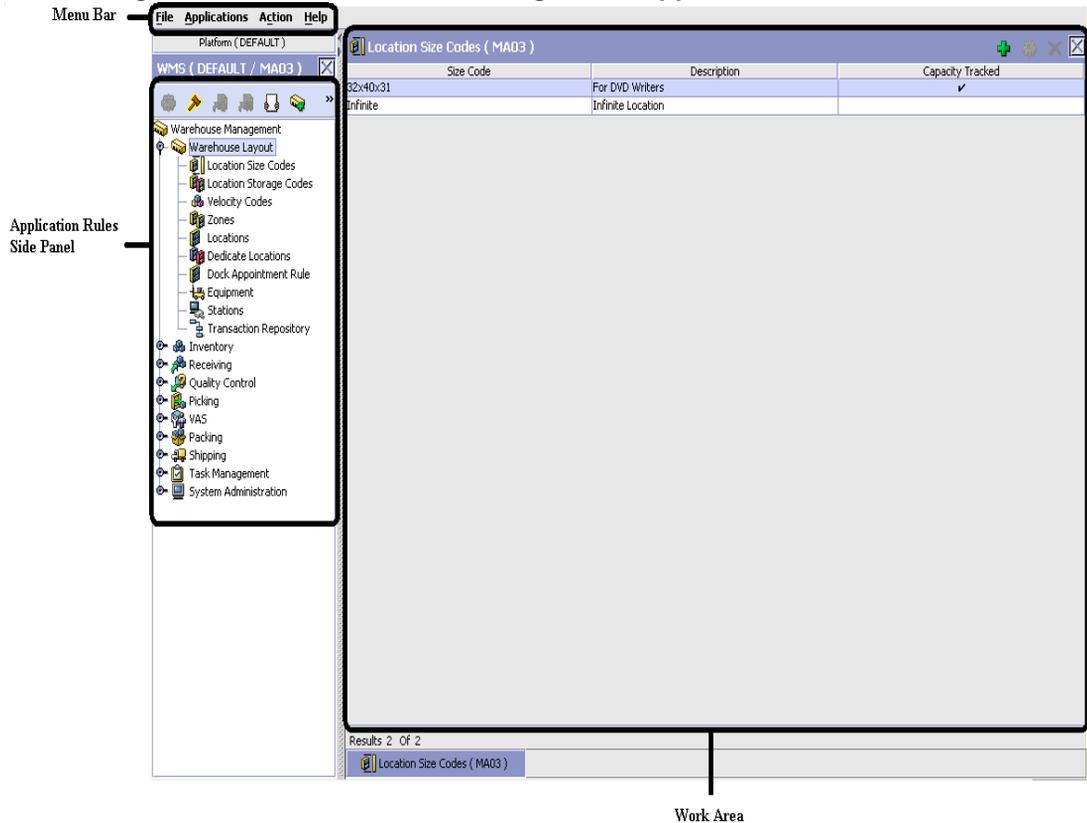
Each application focuses on a particular aspect of Selling and Fulfillment Foundation and contains all of the rules, common codes, and settings necessary for Selling and Fulfillment Foundation to work in a real-world business setting.

The following applications can be configured in this version of Selling and Fulfillment Foundation:

- Distributed Order Management
- Global Inventory Visibility
- Catalog Management
- Logistics Management
- Supply Collaboration
- Reverse Logistics
- Warehouse Management
- Application Platform

When you select the application that you want to configure, the Configurator displays a side panel containing all of the available configuration rules for the selected application and a work area in which these rules can be configured.

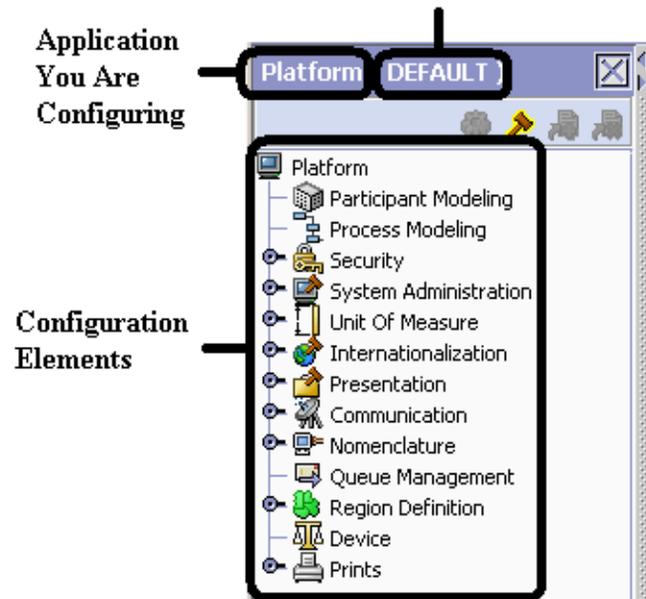
Figure 2–2 The Standard Configurator Application Interface



### 2.2.1 Application Rules Side Panel

The application rules side panel displays a hierarchical tree of elements specific to processes used within the application.

**Figure 2–3 Example of Application Rules Side Panel Organization You Are Defining Rules For**



The application rules side panel also identifies the organization you are configuring rules for and what, if any, rules are inherited from another organization.

You can use the application rules side panel for:

- [Accessing Configuration Screens](#)
- [Determining Inheritance](#)
- [Loading Another Organization's Rules](#)

### **2.2.1.1 Accessing Configuration Screens**

The main purpose of the application rules side panel is to provide an interface to access the application's individual configuration screens. To access a configuration screen, browse through the application tree and double-click on the applicable configuration element, the element's configuration screen displays in the work area.

### 2.2.1.2 Determining Inheritance

In Selling and Fulfillment Foundation, when an Enterprise is created it can inherit all or part of an existing Enterprise's configuration rules. This inheritance is done at the configuration group level. A configuration group is a classification of similar configuration elements. For example, all of the rules and configurations dealing with items are grouped together into one configuration group and all of the rules and configurations dealing with organizations are grouped into another.

An administrator organization is set for every organization defined within the system. Only the administrator organization can modify the rules defined for a particular organization. If a particular organization administers multiple organizations, then they can load the rules of organization that it administers within the application tree. For more information about loading another organization's rules, see [Section 2.2.1.3, "Loading Another Organization's Rules"](#).

Configuration groups are associated with organization levels. Organization levels determine how configuration groups are inherited and which organizations can maintain them. The organization levels defined in Selling and Fulfillment Foundation are:

- Hub Level - Configuration groups that are associated with the Hub organization
- Enterprise Level - Configuration groups that are associated with the individual Enterprise organizations within the Hub environment
- Catalog Organization - Configuration groups that are associated with the organization(s) that maintains the catalog(s) within the Hub environment
- Inventory Organization - Configuration groups that are associated with the organization(s) that maintains the inventory within the Hub environment
- Pricing Organization - Configuration groups that are associated with the organization(s) that maintains the pricing within the Hub environment
- Organization - Configuration groups that are associated with any organization within the Hub environment

---

---

**Note:** The Configurator does not load configuration data and permissions based on Data Access Policies that are described in the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

---

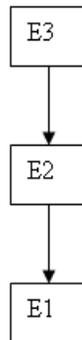
---

### Enhanced Inheritance for Process Models

An Enterprise can inherit the configurations of the following entities from other Enterprises:

- Pipelines
- User Exits
- Services
- Actions
- Conditions
- Statuses
- Transactions
- Events

When an Enterprise inherits these entities from some other Enterprise, the current Enterprise can view the configurations that are inherited from all other Enterprises (including the Hub) in the inheritance hierarchy. In addition, the current Enterprise can view the configurations that are defined for the Hub. For example, consider the following inheritance hierarchy:



In this hierarchy, Enterprise E1 is inheriting from Enterprise E2, which in turn is inheriting from Enterprise E3. Enterprise E1 can view the configurations that are defined for Enterprise E2 and Enterprise E3. In addition, Enterprise E1 can view the configurations that are defined for the Hub.

The following table details the rules used to determine which organizations can maintain a configuration group as defined by the organization level. The table also describes the rules that determine how configuration groups are inherited when an organization is created.

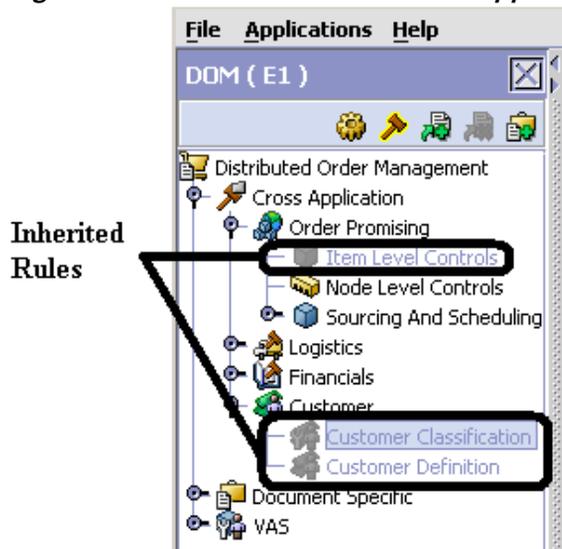
**Table 2–1 Organization Level Rules**

Organization Level	Organizations That Can Modify at this Level...	Inheritance Details
Hub Level	Only the Hub organization can modify configuration groups at the Hub level. All other organizations have read-only access.	All organizations share this information.
Enterprise Level	Only Enterprise organizations can modify configuration groups at the Enterprise level.  Any business transaction requiring Enterprise configuration is picked up from the Enterprise established by the transactional context. For example, order documents have a specific Enterprise.	An Enterprise can inherit this configuration from another Enterprise. Additionally, this configuration can be overridden at a configuration group level.
Catalog Organization	Organizations that are designated as catalog organizations can modify configuration groups at the catalog organization level.	None.
Inventory Organization	Organizations that are designated as inventory organizations can modify configuration groups at the inventory organization level.	None.
Pricing Organizations	Organizations that are designated as pricing organizations can modify configuration groups at the pricing organization level.	None.
Organization	Any organization assigned a role (Seller, Buyer, etc.) can modify configuration groups at the organization level.	None.

**Important:** You cannot inherit from an Enterprise that does not have the same inventory, capacity, and catalog organizations as the organization you are configuring.

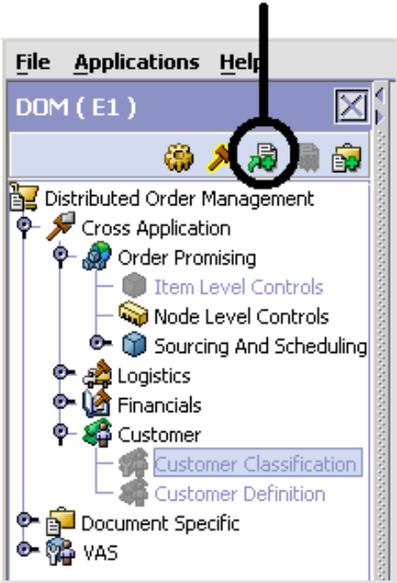
The application rules side panel displays rules that have been inherited as grayed out.

**Figure 2–4** *Inherited Rules in the Application Rules Side Panel*



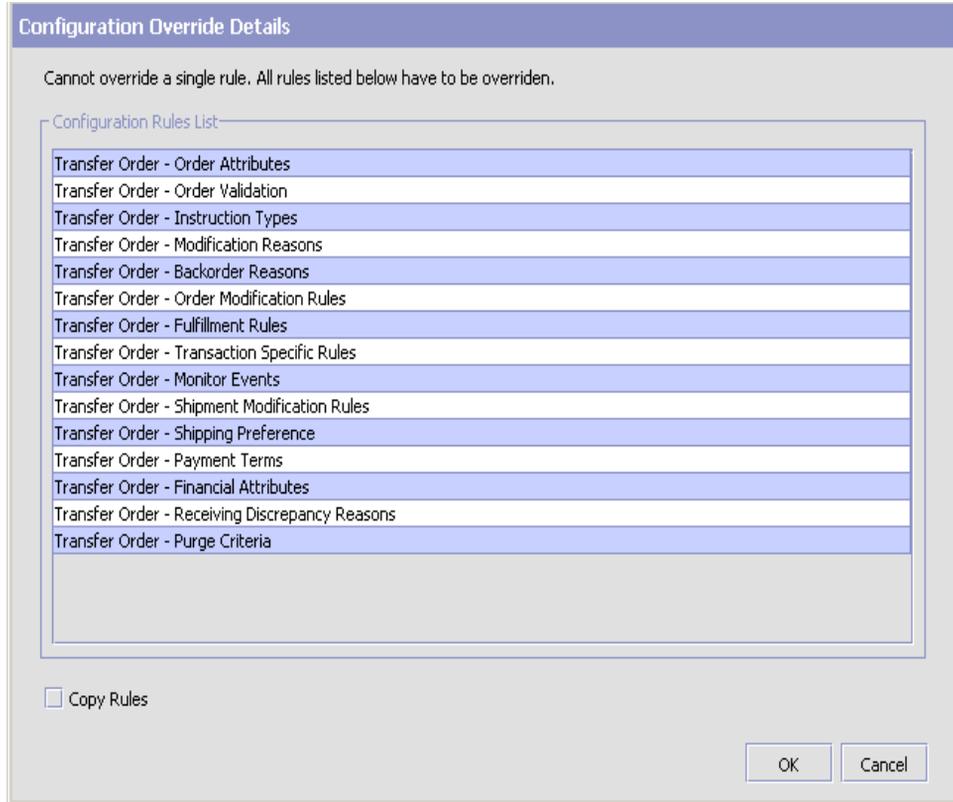
As stated in the table above, depending on the organization you are logged in as, you may be able to override some inherited rules. If a rule can be overridden, the Override Configuration icon becomes available in the application rule side panel when you highlight the rule.

Figure 2–5 *Override Configuration Icon*  
Override Configuration Icon is Available



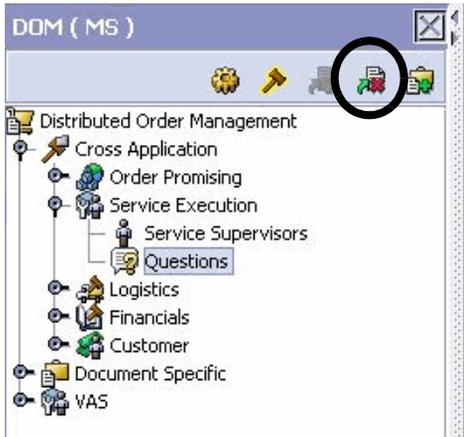
When you choose to override a rule you also override any other rules in the configuration group the rule you are overriding is associated with. When you choose the Override Configuration icon the Configuration Override Details pop-up window displays. This window provides the list of rules that are overridden.

**Figure 2–6 Example of Configuration Override Details Pop-Up Window**



If you override a configuration group and then decide to "re-inherit" the original rules, you can choose the Give Back Configuration Ownership icon. This icon becomes available in the application rules side panel for rules that have been overridden.

Figure 2–7 Give Back Configuration Ownership Icon



When you select the Give Back Configuration Ownership Icon, the Configuration Override Details pop-up window displays. This window provides the list of rules that are re-inherited.

**Important:** If you select the Delete Rules field on the Configuration Override Details pop-up window, you give back rule ownership to the organization you originally inherited from, however you do not retain any of the rules that you inherited from them.

If you do not select this field, you give back rule ownership to the organization you originally inherited from, but you retain the rules that you inherited from them.

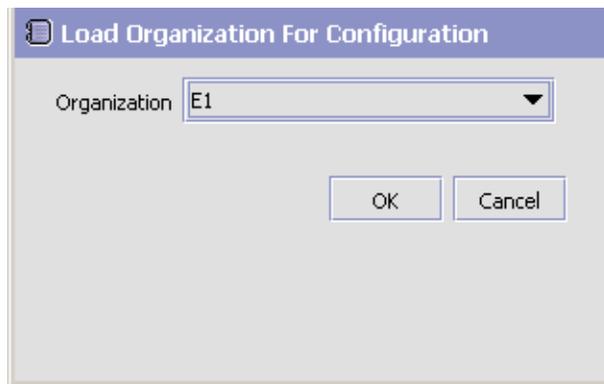
### 2.2.1.3 Loading Another Organization’s Rules

An administrator organization is set for every organization defined within the system. Only the administrator organization can modify the rules defined for a particular organization. If a particular organization administers multiple organizations, then they can load the rules of organization that it administers within the application tree. See [Table 2–1](#) for the rules that determine which organizations you can administer.

**Note:** The rules that are available from the tree in the application rules side panel may vary depending on the type of organization you select and the roles it has been assigned.

To load another organization's rules:

1. From the applicable application rules side panel, choose . The Load Organizations for Configuration pop-up window displays.



2. From Organization, select the organization that you want to work with.
3. Choose OK. The organization's rules display in the application rules side panel.

**Note:** The application rules side panel displays the organization you are working with in parentheses.

## 2.2.2 Work Area

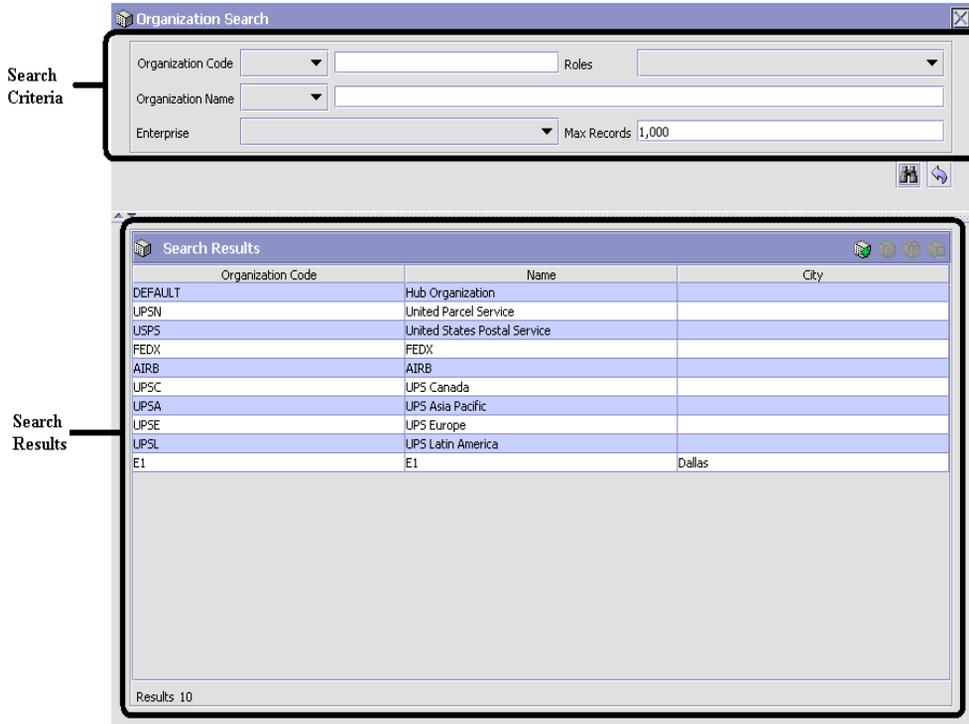
The work area is the main area in which different configuration screens appear. The following are the main types of screens that you can be seen in the work area:

- [Search Window](#)
- [List Window](#)
- [Details Window](#)
- [Drag and Drop Window](#)

### 2.2.2.1 Search Window

A search window provides you with a means to perform a filtered search. The upper panel of a search window offers criteria applicable to the entity you are searching through which you can narrow your search. The lower panel lists the results of a search once it has been performed.

Figure 2–8 Search Window Example



### 2.2.2.2 List Window

When you choose to configure a specific rule or code that does not require a search, the Configurator may display a basic list window of the rules and codes that have previously been configured.

Figure 2–9 List Window Example

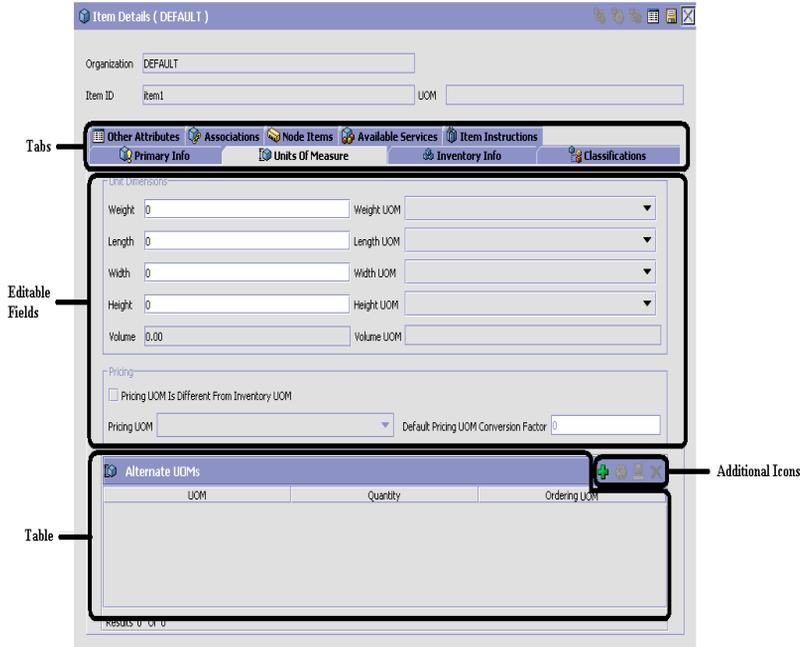
UOM Code	UOM Description
CM	Centimeter
FEET	Feet
IN	Inch
KM	Kilometer
METER	Meter
MILE	Mile

Results 6 Of 6

**2.2.2.3 Details Window**

A details window is the main interface through which a bulk of the configuration is done. A details window can contain editable fields and tables, tabs to configure different aspects of an entity, and additional actions that can be performed on an entity.

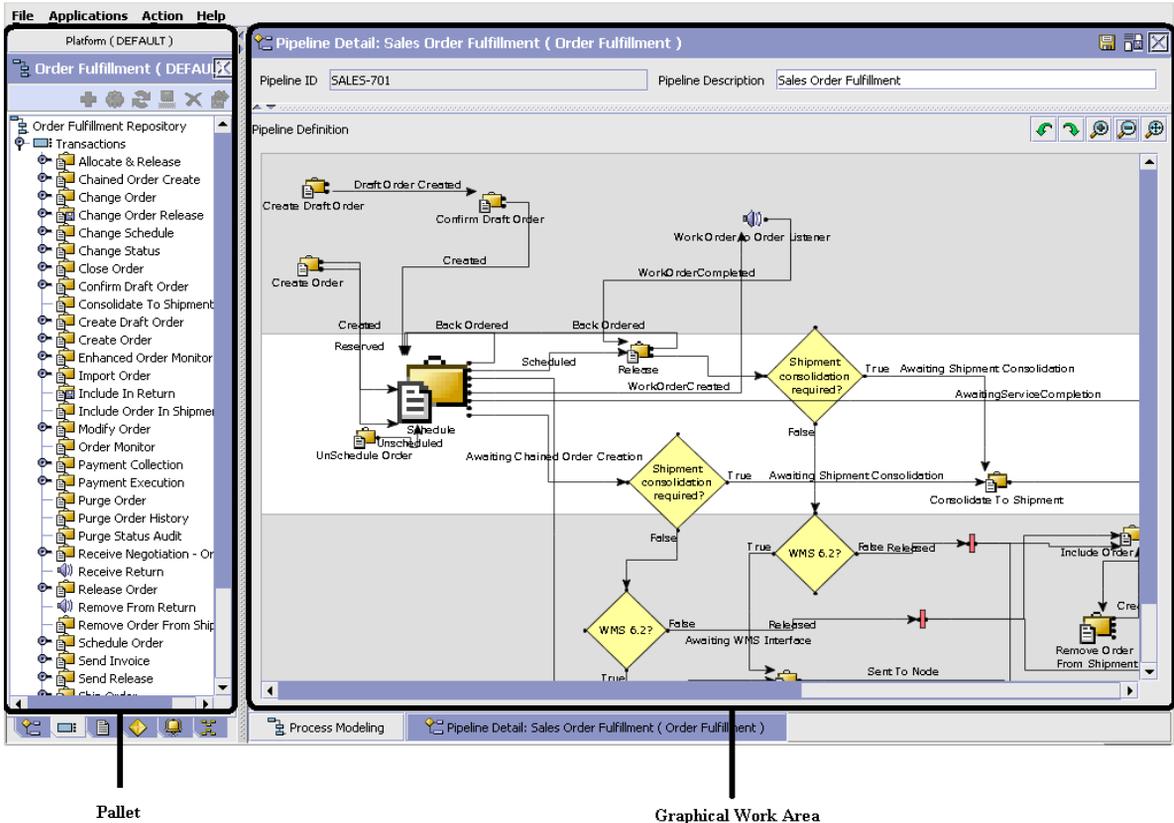
**Figure 2–10 Details Window Example**



## 2.2.2.4 Drag and Drop Window

You can use a graphical drag and drop window to ease the construction of pipelines, pipeline determination, event handlers, status monitoring rules, and services. A drag and drop window consists of a pallet and a graphical work area.

Figure 2–11 Drag and Drop Window Example



To begin building any of these entities, choose a component, such as a transaction, from the pallet. Drag the component into the graphical work area. The transaction is now displayed as a graphical representation of itself.

To connect one component to another, you must drag the mouse from the outgoing port of a component until it forms a connecting line with the incoming port of another component. The links between components can be set up either horizontally or vertically.

To delete components or links, right-click on the component and choose Delete. Once components and links have been established you can move them around by dragging them, the links redraw themselves according to the new position. If you press and hold the CTRL key while dragging a component, the component is copied within the graphical work area.

## 2.3 Actions Available in the Applications Manager

The following actions can be performed throughout the Applications Manager:

- [Using the Configurator's Lookup Functionality](#)
- [Viewing the User Logged into the Configurator](#)
- [Using Lists and List Filtering](#)
- [Using On-Line Help](#)
- [Troubleshooting Errors](#)
- [Using Special Characters](#)

### 2.3.1 Using the Configurator's Lookup Functionality

Throughout the Applications Manager there are many fields that have a lookup functionality to find or create additional records as they pertain to that field. For example, on the Primary Info tab of the Organization Details screen, the Locale field has a lookup functionality to create a new locale from that screen. When you choose the Create New lookup button the Locale Details information displays in a pop-up screen for you to modify.

**Figure 2–12** *Lookup Icon Example*



The information that displays in a lookup field varies depending on how many records you have pertaining to that particular field. When there are 20 or less records, the lookup displays as a drop-down list with a Create New button. When there are between 21 and 75 records, the lookup displays as a drop-down list with a Search button.

When there are more than 75 records, the lookup displays as a text box with a Search button. You can type the value in the text box or search for the value using the Search button. If you enter a value, it is validated when it is saved. You should always type the value as it would appear if it was displayed as a drop-down list. For example, for a currency lookup, you should type the currency description in the text box even though the

currency code is saved in the table. An error displays on save if the user has entered an invalid value.

When you use a lookup for a particular field in the Configurator, you should refer to the corresponding section in this guide to set up the particular information.

### 2.3.2 Viewing the Document Types Associated with an Application

In the Distributed Order Management, Supply Collaboration, Reverse Logistics, and Logistic Management configuration applications, you can view all of the document types associated with the application. Sales Order, Transfer Order, Master Order, and Purchase Order are all examples of document types.

To view an application's associated document types, open the applicable application from the menu and choose  from the application rules side panel. The Associated Document Types window displays displaying a list of all of the document types associated with the application you are working in.

*Figure 2–13 Associated Document Types Window*

Document Type	Description
0004	Template Order
0006	Transfer Order
0001	Sales Order
0007	Master Order

Results 4 of 4

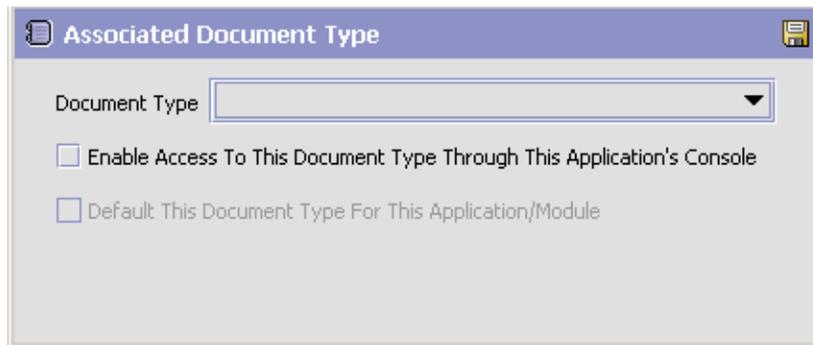
### 2.3.2.1 Adding a Document Type to an Application

You can add a document type that is associated with another application to the application you are currently working in.

**Important:** An added document type's associated screens may be irrelevant to the application you are associating it with.

To add a document type to an application:

1. From the Associated Document Types window, choose . The Associated Document Type pop-up window displays.



2. From Document Type, select the document type that you want to associate with the application.
3. Select Enable Access To This Document Through This Application's Console.
4. Choose .

### 2.3.3 Viewing the User Logged into the Configurator

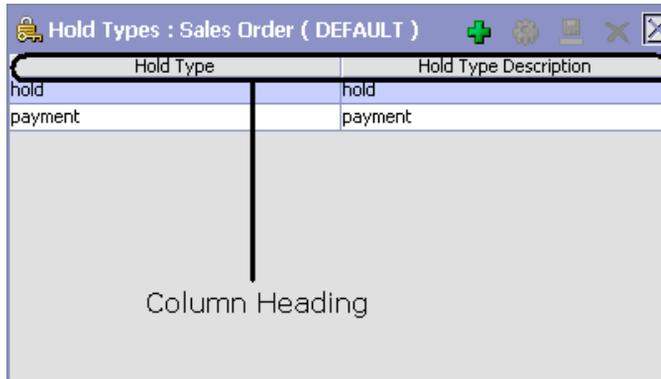
You can view the user logged into the Configurator and their locale at any time. To view this information, move your mouse over the User icon and Locale icons in the bottom right-hand corner of the application to display the tool tips.

### 2.3.4 Using Lists and List Filtering

When viewing any list in the Configurator, it is possible to filter the contents of the list based in criteria that you define. Filtering is

accomplished by right-clicking anywhere on the list's column headings and using the Table Filter Editor associated with the list.

**Figure 2–14** Column Headings in a List



**Figure 2–15 Table Filter Editor Window Example**

**Table 2–2 Table Filter Editor Window**

Field	Description
Apply To Existing Records	Checking this box applies a new filter set of results that have been previously filtered instead of the whole set.
Max Records	Specify the maximum number of records that are to be returned from a filter. The default number is 100
Dynamic Fields	Fields such as "Hold Type" and "Hold Type Description" in <a href="#">Figure 2–15</a> are dynamically populated based on the list you are currently viewing.  These fields can be searched using text strings combined with criteria such as <b>Is</b> , <b>Starts With</b> , or <b>Contains</b> .

**Important:** Search strings are case sensitive. For example, "Item" does not return the same values as "item".

## 2.3.5 Date and Time Entry

Date fields through the Configurator have a calendar icon that can be used to find dates as it pertains to that field. When you click on this icon, a small calendar displays. You can navigate through this calendar to determine the appropriate date. For example, on the Create Calendar window, the Default Effective To field has a calendar icon that you can use to verify the appropriate ship by date to populate the field.

**Figure 2–16 Calendar Icon example**



You can also enter time of day information throughout the Configurator. To do this, double click on the time field, and enter the time of day.

**Figure 2–17 Time Field example**



Time should be entered in a 24 hour time format everywhere throughout the Configurator.

## 2.3.6 Using On-Line Help

You can access the Selling and Fulfillment Foundation Online Help through Help > Online Help.

## 2.3.7 Troubleshooting Errors

You can view the description and cause of any error raised in Selling and Fulfillment Foundation, as well as the actions to troubleshoot it.

To view the Selling and Fulfillment Foundation system error descriptions:

1. From the menu bar, choose Help > Troubleshooting. The Error Search window displays.
2. Enter the applicable search criteria and choose . A list of error codes and their descriptions display.

3. Choose  to view the cause of the error and action to troubleshoot it.

### 2.3.8 Using Special Characters

Throughout the Applications Manager there may be instances where you need to use special characters in data entry. For information about the use of special characters in Selling and Fulfillment Foundation, see the *Selling and Fulfillment Foundation: Customization Basics Guide*.



# 5

## Configuring Cross Application Logistics Components

---

You can configure the components used by different logistics related functionality throughout the business application module.

You can use the Logistics branch for:

- [Defining Logistics Attributes](#)
- [Defining Delivery Codes](#)
- [Defining Shipment Modes](#)
- [Defining Outbound Constraints](#)

### 5.1 Defining Logistics Attributes

**You can** define rules and common codes associated logistics of shipping an order.

You can use the Logistics Attributes branch for:

- [Defining Freight Terms](#)
- [Defining Shipment Modes](#)
- [Defining Carrier Modification Reasons](#)
- [Defining Additional Logistic Rules](#)

#### 5.1.1 Defining Freight Terms

You can define common codes used when associating a freight term to a Carrier. A **freight term** identifies how transportation costs are calculated.

The default freight terms of Selling and Fulfillment Foundation are:

- Cost Insurance and Freight (CIF) - The freight cost is completely paid by either the Seller, the Enterprise, or the Hub.
- Cost and Freight (CFR) - The freight cost is paid by the Buyer and either the Seller, the Enterprise, or the Hub.
- Free On Board (FOB) - The freight cost is paid by the Buyer.

You can use the Freight Terms tab for:

- [Creating a Freight Term](#)
- [Modifying a Freight Term](#)
- [Deleting a Freight Term](#)

### 5.1.1.1 Creating a Freight Term

To create a freight term:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Freight Terms tab.
3. Choose . The Freight Terms Details pop-up window displays.
4. Enter information in the applicable fields. Refer to [Table 5–1](#) for field value descriptions.
5. Enter Choose .

Freight Terms Details

Freight Terms

Short Description

Long Description

Consider Buyer's Routing Guide

---

Charges paid by

Buyer

Shipper

**Table 5–1 Freight Terms Details**

Field	Description
Freight Terms	Enter the name of the freight term.
Short Description	Enter a brief description of the freight term.
Long Description	Enter a more detailed description of the freight term.
Consider Buyer's Routing Guide	<p>Both the Buyer and the Enterprise can establish routing guides (rules for shipping), and Economic Shipping parameters (ESP), which control how items are shipped. In some cases only the Buyer organization has established values for these rules. In other cases, only the enterprise has established values for these rules. If neither is set, then Hub rules are used.</p> <p>In cases where both the Buyer and the Enterprise have set values for these rules, this setting determines whether to apply the Buyer's routing rules before applying the routing rules of the Enterprise. See the <i>Selling and Fulfillment Foundation: Product Concepts Guide</i> for more information about these shipping concepts.</p>
First Buyer then Enterprise	Select to use any shipping rules established by the buyer first. Enterprise rules are applied if no applicable Buyer rule exists.

**Table 5–1 Freight Terms Details**

Field	Description
First Enterprise then Buyer	Select to use any shipping rules established by the enterprise first. Buyer rules are applied if no applicable Enterprise rule exists.
Charges paid by	
Buyer	Select this option if the Buyer pays shipping charges.
Shipper	Select this option if the Shipper pays shipping charges.

### 5.1.1.2 Modifying a Freight Term

To modify a freight term:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Freight Terms tab.
3. Select the applicable freight term and choose . The Freight Terms Details pop-up window displays.
4. Enter the new information in the applicable fields. Refer to [Table 5–1](#) for field value descriptions.
5. Choose .

### 5.1.1.3 Deleting a Freight Term

To delete a freight term:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Freight Terms tab.
3. Select the applicable freight term and choose .

## 5.1.2 Defining Carrier Modification Reasons

You can define common codes that appear in the Reason Code drop-down list when you modify a Carrier. This code should provide a standard reason for modifying a Carrier, such as 'Requested Change' which would be used when the customer requests a change of Carrier.

The default carrier modification reason of Selling and Fulfillment Foundation is:

- Requested Change

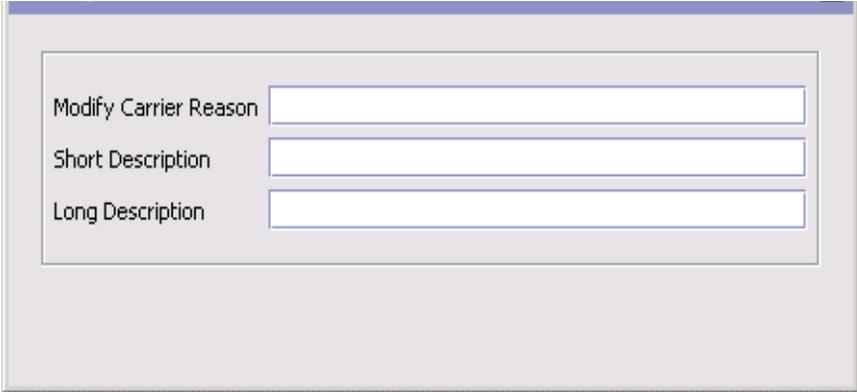
You can use the Modify Carrier Reason tab for:

- [Creating a Carrier Modification Reason](#)
- [Modifying a Carrier Modification Reason](#)
- [Deleting a Carrier Modification Reason](#)

### 5.1.2.1 Creating a Carrier Modification Reason

To create a carrier modification reason:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Modify Carrier Reason tab.
3. Choose . The Modify Carrier Reason Details pop-up window displays.



Modify Carrier Reason	<input type="text"/>
Short Description	<input type="text"/>
Long Description	<input type="text"/>

4. In Modify Carrier Reason, enter the name of the carrier modification reason.
5. In Short Description, enter a brief description of the carrier modification reason.
6. In Long Description, enter a more detailed description of the carrier modification reason.
7. Choose .

### 5.1.2.2 Modifying a Carrier Modification Reason

To modify a carrier modification reason:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Modify Carrier Reason tab.
3. Select the applicable carrier modification reason and choose . The Modify Carrier Reason Details pop-up window displays.
4. In Short Description, enter a brief description of the carrier modification reason.
5. In Long Description, enter a more detailed description of the carrier modification reason.
6. Choose .

### 5.1.2.3 Deleting a Carrier Modification Reason

To delete a carrier modification reason:

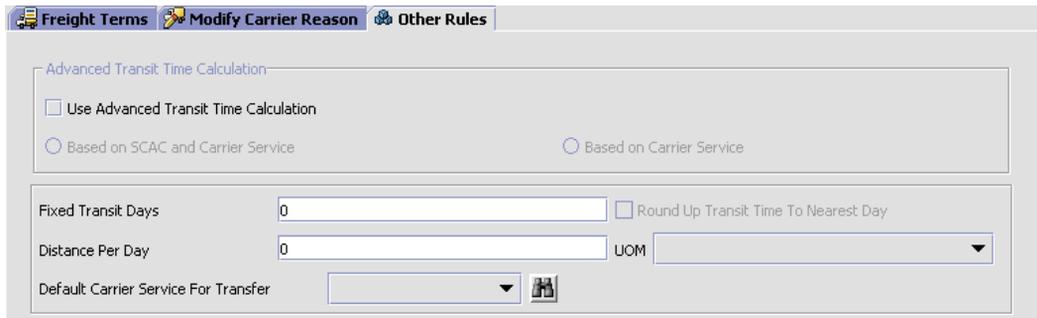
1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Modify Carrier Reason tab.
3. Select the applicable carrier modification reason and choose .

### 5.1.3 Defining Additional Logistic Rules

You can define additional rules that pertain to an order document type.

To define additional logistic rules:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Other Rules tab.



3. Enter information in the applicable fields. Refer to [Table 5–2](#) for field value descriptions.
4. Choose .

**Table 5–2 Other Rules Tab**

Field	Description
Advance Transit Time Calculation	
Use Advanced Transit Time Calculation	Select this field if advance transit time calculation is required when considering ship dates and delivery dates.  Transit time is calculated as Lead Time + Distance Per Day (either from the Distance Per Day for a selected Carrier service).
Based on SCAC and Carrier Service	Select Based on Carrier if you want transit time calculations to be based on the carrier and carrier service being used for an order.

**Table 5–2 Other Rules Tab**

Field	Description
Based on Carrier Service	Select Based on Carrier Service if you want transit time calculations to be based on the specific carrier service being used for an order.
Delivery Lead Time (Days)	<p>Enter the default delivery lead time.</p> <p>Delivery lead time is used to determine when an order line must be shipped based on the requested delivery date. The delivery lead time indicates the amount of time it takes to transport a load from a ship node to a customer. When calculating the delivery date:</p> <ul style="list-style-type: none"> <li>• If neither the ship date or delivery date are provided, the ship date is defaulted to the current days date and the delivery date is defaulted to that date + delivery lead time.</li> <li>• If the ship date is provided but the delivery date is not, the delivery date is defaulted to ship date + delivery lead time.</li> <li>• If the delivery date is provided but the ship date is not, the ship date is defaulted to delivery date - delivery lead time.</li> <li>• If both the ship date and delivery date are provided, this rule is not applied.</li> </ul>
Round Up Transit Time To Nearest Day	If selected, transit time calculations are not specific down to the actual hour. Instead, the system performs the calculations and rounds up to the next available day.
Distance Per Day	Enter the default distance for calculating transit time if a Carrier service is not selected or the service selected does not have a distance per day associated with it.
UOM	Select the distance unit of measure.
Default Carrier Service for Transfer	<p>Select the carrier service you want to use to compute the transfer time between two nodes if they do not have a transfer schedule configured for them.</p> <p>For more information about configuring transfer schedules between nodes, see the <i>Sterling Distributed Order Management: Configuration Guide</i>.</p>

## 5.2 Defining Delivery Codes

You can define common codes used for indicating the delivery code when creating or modifying a Carrier. The **delivery code** identifies the entity that pays for the transportation costs.

The default delivery codes of Selling and Fulfillment Foundation are:

- ENTERPRISE
- MARKETPLACE
- SUPPLIER

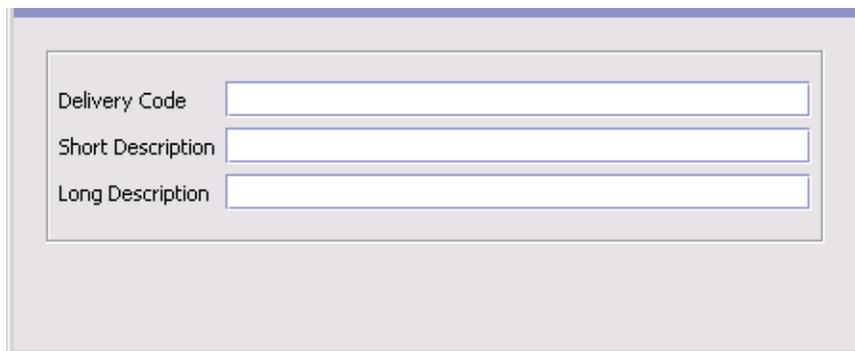
You can use the Delivery Codes branch for:

- [Creating a Delivery Code](#)
- [Modifying a Delivery Code](#)
- [Deleting a Delivery Code](#)

### 5.2.1 Creating a Delivery Code

To create a delivery code:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Delivery Codes. The Delivery Codes window displays in the work area.
2. Choose . The Delivery Code Details pop-up window displays.



The screenshot shows a pop-up window titled "Delivery Code Details" with a light gray background. It contains three text input fields stacked vertically. The first field is labeled "Delivery Code", the second is labeled "Short Description", and the third is labeled "Long Description". Each field has a thin blue border and is currently empty.

3. In Delivery Code, enter the name of the delivery code.
4. In Short Description, enter a brief description of the delivery code.
5. In Long Description, enter a more detailed description of the delivery code.
6. Choose .

### 5.2.2 Modifying a Delivery Code

To modify a delivery code:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Delivery Codes. The Delivery Codes window displays in the work area.
2. Select the applicable delivery code and choose . The Delivery Code Details pop-up window displays.
3. In Short Description, enter a brief description of the delivery code.
4. In Long Description, enter a more detailed description of the delivery code.
5. Choose .

### 5.2.3 Deleting a Delivery Code

To delete a delivery code:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Delivery Codes. The Delivery Codes window displays in the work area.
2. Select the applicable delivery code and choose .

## 5.3 Defining Shipment Modes

You can define common codes used when indicating the ship mode. The **shipment mode** describes how an order is being shipped.

The default shipment modes of Selling and Fulfillment Foundation are:

- TL - Truckload
- LTL - Less-Than Truckload
- PARCEL

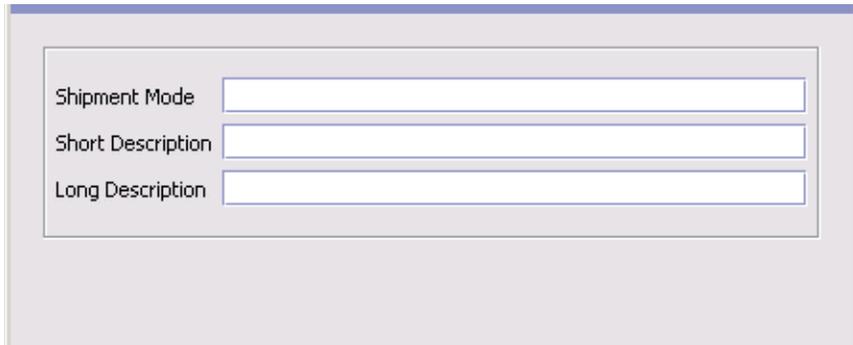
You can use the Shipment Modes tab for:

- [Creating a Shipment Mode](#)
- [Modifying a Shipment Mode](#)
- [Deleting a Shipment Mode](#)

### 5.3.1 Creating a Shipment Mode

To create a shipment mode:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Shipment Modes. The Shipment Modes window displays in the work area.
2. Choose . The Shipment Mode Details pop-up window displays.



The screenshot shows a dialog box titled 'Shipment Mode Details'. It contains three text input fields arranged vertically. The first field is labeled 'Shipment Mode', the second is labeled 'Short Description', and the third is labeled 'Long Description'. Each field is currently empty.

3. In Shipment Mode, enter the name of the shipment mode.
4. In Short Description, enter a brief description of the shipment mode.
5. In Long Description, enter a more detailed description of the shipment mode.
6. Choose .

### 5.3.2 Modifying a Shipment Mode

To modify a shipment mode:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Shipment Modes. The Shipment Modes window displays in the work area.
2. Select the applicable shipment mode and choose . The Shipment Mode Details pop-up window displays.
3. In Short Description, enter a brief description of the shipment mode.
4. In Long Description, enter a more detailed description of the shipment mode.
5. Choose .

### 5.3.3 Deleting a Shipment Mode

To delete a shipment mode:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Shipment Modes. The Shipment Modes window displays in the work area.
2. Select the applicable shipment mode and choose .

## 5.4 Defining Outbound Constraints

Outbound constraints are used to describe conditions that control how shipping is done. These include whether certain items can be shipped together, such as regular and rush orders, whether to use Economic Shipping Parameters, and how routing is performed. You can also use Outbound Constraints for:

- [Creating a Routing Guide](#)
- [Modifying a Routing Guide](#)
- [Deleting a Routing Guide](#)

---

---

**Note:** the Outbound Constraints node does not apply to Reverse Logistics or Supply Collaboration.

---

---

To define outbound constraints:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Outbound Constraints. The Outbound Constraints window displays in the work area.
2. Enter information in the applicable fields. Refer to [Table 5–3](#) for field value descriptions.
3. Choose .

🔍 Outbound Constraints ( DEFAULT )
📄 🗑️

Do not mix in a shipment:

<input type="checkbox"/> Buyer Mark For Node Id	<input type="checkbox"/> Customer PO #
<input type="checkbox"/> Department Code	<input type="checkbox"/> Gift Flag
<input type="checkbox"/> Level of Service	<input type="checkbox"/> Mark For
<input type="checkbox"/> Order #	<input type="checkbox"/> Order Type

Transportation optimization

Economic shipping parameters maintained

Region Schema For Routing  🔍

Routing Guide

Not Maintained   
  Maintained in Sterling Selling and Fulfillment Suite   
  Maintained Externally

🔍 Routing Guides
+ ⚙️ 📄 🗑️

Name	Routing Guide #	Effective Date	Freight Terms

Results: 0 of 0

**Table 5–3 Outbound Constraint Window**

Field	Description
Do not mix in a shipment	<p>If any of the following are selected, separate shipments must be create for items that have different values for these attributes.</p> <p>For example, if Department Code is selected, items that are for different departments can not be included in the same shipment.</p>
Buyer Mark For Node Id	The buyer mark for node id.
Customer PO #	Customer's Purchase Order number.
Department Code	The department for which the item is intended.
Gift Flag	The gift flag.
Level of Service	The level of service on the order.
Mark For	Person for whom this shipment is marked for
Order #	The order number.
Order Type	The order type.
Transportation optimization	
Economic shipping parameters maintained	<p>Economic Shipping Parameters (ESP) are used in shipping consolidation. Select this field to enable the following Economic Shipping Parameters fields.</p> <p>ESP support consolidation of shipments until a weight or volume threshold is met, or until an certain time elapses. By consolidating shipments, shipping costs can be reduced</p> <p>For example, you can set that shipments should be consolidated until the shipment weight is 300 pounds, or 50 cubic feet in volume. To ensure that eventually the shipment is set, you can establish a maximum number of days to wait until the conditions are met.</p> <p>When either the weight, volume or delay shipment threshold is met, the shipment is moved to the next stage in shipping.</p>

**Table 5–3 Outbound Constraint Window**

Field	Description
Delay shipment by not more than ___ Days	Enter the number of days this shipment can be delayed before it should be shipped.  For example, if a value is set for weight threshold of 300 pounds, and this field has been set to 3 days, the shipment is shipped after 3 days, regardless of whether the weight threshold has been met.
Consolidate up to weight threshold of	Enter a weight.
Consolidate up to volume threshold of	Enter a volume
Routing Guide	
Not Maintained	Select this to use manual routing. Shipments are managed in the shipment console, and any routing guides are not consulted.
Maintained in Sterling	Select this to use the Routing Guides maintained in Selling and Fulfillment Foundation to determine how shipments should be routed. See <a href="#">Section 5.4.1, "Creating a Routing Guide"</a> .  In addition to the routing guide maintained here by the enterprise, there may be a routing guide for the buyer organization.  For more information about using both buyer and enterprise routing guides, see <a href="#">Section 5.1.1.1, "Creating a Freight Term"</a> .
Maintained Externally	Select this to indicate that an external routing system is used. The routing guides maintained in Selling and Fulfillment Foundation is not consulted.  Examples of external routing systems include using an integrated Transportation Management System (TMS), or implementing a User Exit which consults with the buyer organization.

### 5.4.1 Creating a Routing Guide

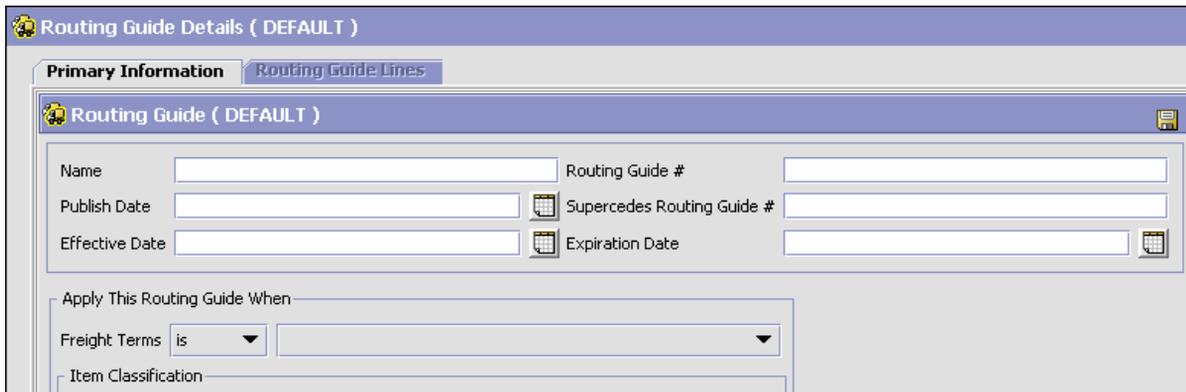
*Routing Guides* are a list of conditions which determine how a shipment should be routed. A routing guide has a time period for which is effective, and conditions for when it should be applied. These conditions are based on Freight Terms and Department.

Each routing guide contains a list of *routing guide lines*, each of which describe detailed conditions for selecting a carrier. The routing guide information is based on data used by VICS (Voluntary InterIndustry Commerce Standards) routing.

To create a routing guide:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Outbound Constraints. The Outbound Constraints window displays in the work area.
2. Select  on the Routing Guides list window. The Routing Guide Details window displays in the work area.
3. Enter information in the applicable fields. Refer to [Table 5–4](#) for field value descriptions.
4. Choose .

**Figure 5–1 Routing Guide Details Window**



**Table 5–4 Routing Guide Details Window**

Field	Descriptions
Name	Enter a name for the routing guide.
Routing Guide #	A number for the routing guide.
Publish Date	When this routing guide is available within the system.

**Table 5–4 Routing Guide Details Window**

Field	Descriptions
Supersedes Routing Guide #	Tracking information. For example, if a minor revision is made to routing guide "1234", you might create a routing guide "1234-A", and enter that it supersedes routing guide "1234". This field is for informational purposes and is not used to determine the effective routing guide.
Effective Date	The start date for applying the routing information in this routing guide. You can use the effective date and expiration date to apply routing guidelines for particular periods of time.
Expiration Date	The end date for applying the routing information in this routing guide.
Apply this Routing Guide when	
Freight Terms	Apply this routing guide when this condition is met. Select <i>is</i> , <i>is in</i> , or <i>is not</i> . Use: <ul style="list-style-type: none"> <li>• <i>is</i> to specify a single Freight Term.</li> <li>• <i>is in</i> to specify a group of Freight Terms, one of which must be matched.</li> <li>• <i>is not in</i> to specify a group of Freight Terms. The routing guide is used if the Freight Term does not match one of these values.</li> </ul>
Item Classification	Items can be classified. <b>Note:</b> This field displays when valid item classifications have been set up for Routing Guide.

### 5.4.2 Modifying a Routing Guide

To modify a routing guide:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Outbound Constraints. The Outbound Constraints window displays in the work area.
2. Select a routing guide in the Routing Guide list window, and select



3. The Routing Guide Details window displays in the work area.
4. Enter information in the applicable fields. Refer to [Table 5–4](#) for field value descriptions.
5. Choose .

### 5.4.2.1 Creating a Routing Guide Line

*Routing guide lines* contain the specific conditions to use when routing a shipment. A routing guide can contain multiple routing guide lines.

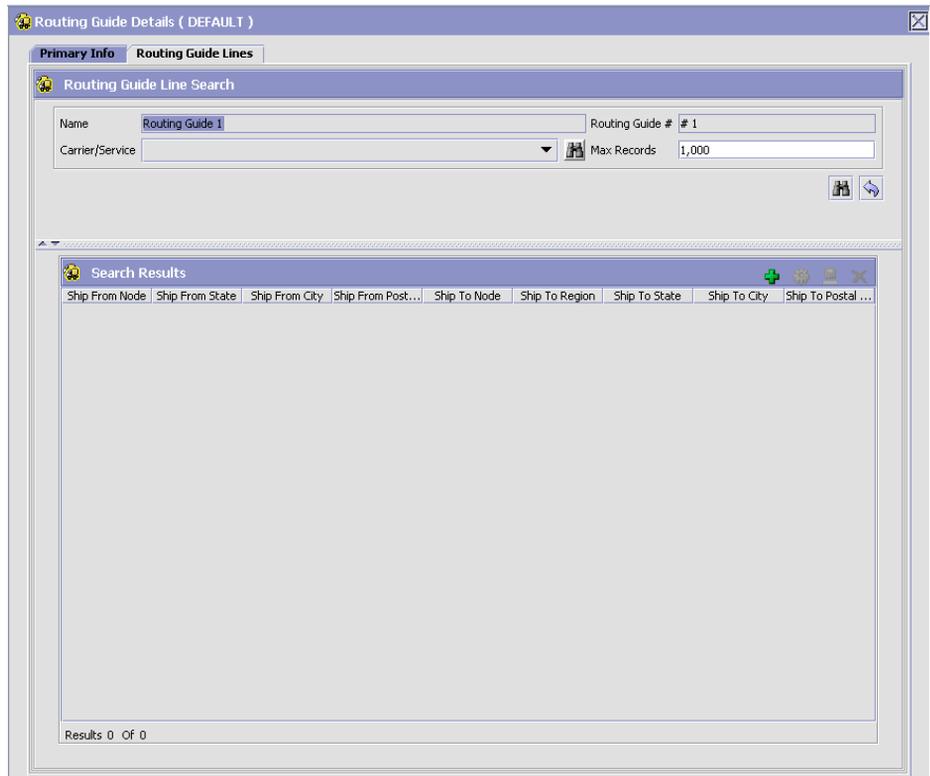
When routing occurs, the shipment is matched against the routing guide lines. Based on the criteria specified, a carrier and carrier service is selected.

When routing results in a change to the shipment destination, the system re-routes, with the revised destination as the factor for routing. This type of configuration is used for consolidator nodes. While routing the second time, system looks for the routing guide entry that contains destination node, but without any other destination parameters filled out (such as address, country, etc.).

To create a routing guide line:

1. From the Routing Guide Details window, select the Routing Guidelines Tab. To have access to the Routing Guidelines Tab, save the information you have entered on the Primary Info Tab.
2. A Routing Guide Line search window displays.

Figure 5–2 Routing Guide Details Window



3. Select . A Routing Guide Line Details screen displays in the work area.
4. Enter information in the applicable fields. Refer to [Table 5–5](#) for field value descriptions.
5. Choose .

Figure 5–3 Routing Guide Line Details Window

**Routing Guide Line Details**

When shipping from:

Node is

When ship from is not node, select the following attribute(s)

Country is  State is

City is  Zip Code is

And shipping to:

Node is   Region

When ship to is not node and region, select the following attribute(s)

Country is  State is

City is  Zip Code is

Consolidator is  Store # is

And weight is in the range:

From 0  LBS To 999,999,999  LBS

And volume is in the range:

From 0  CIN To 999,999,999  CIN

And handling units are in the range:

From 0  To 999,999,999

And if requested carrier service code is

Carrier Service Code

Then ship via:

Priority	Carrier/Service	Break Bulk Node	Contact Specified
Results 0 Of 0			

With overrides:

Override Freight Terms

Override Ship To

**Table 5–5 Routing Guide Line Details**

<p>Setting conditions:</p> <p>In many of the following fields, you can select is, is in, or is not in and then specify a value. Use:</p> <ul style="list-style-type: none"> <li>• <i>is</i> to specify that a single value must be matched</li> <li>• <i>is in</i> to specify a group of values, one of which must be matched.</li> <li>• <i>is not in</i> to specify a group of values. The routing guide line is used if none of these values match.</li> </ul> <p>For example to match any one of a group of states, specify State <i>is in</i> California, Washington, Oregon, Nevada. When assessing the condition, California would match, Florida would not.</p>	
Field	Description
Ship From	
Node	Select the node.
When ship from is not node, select the following attribute(s)	Enter this option if not shipping from the node and then enter one or more of the following conditions.
Country	Select the country name(s).
State	Enter the state name(s).
City	Enter the city name(s).
Zip Code	Enter the zip code or zip code range.
Ship To	
Node	Select the node.
Region	Enter the region.
When ship to is not node and region, select the following attribute(s)	Select this option if not shipping to a node within a specific region and then select one or more of the following conditions.
Country	Select the country name(s).
State	Enter the state name(s).
City	Enter the city name(s).
Zip Code	Enter the zip code or zip code range.
Consolidator	Select the consolidator name(s).

**Table 5–5 Routing Guide Line Details**

Store#	Select the store number(s).
And weight is in the range:	You can match weight. For example, if you want packages that weigh between 100 and 500 pounds to be shipped using a specific carrier, you would specify From as '100' and To as '500'.
From	Enter the minimum value.
To	Enter the maximum value.
And volume is in the range:	You can match volume. For example, if you want packages that are between 3 and 10 cubic feet to be shipped using a specific carrier, you would specify From as '3' and To as '10'.
From	Enter the minimum value.
To	Enter the maximum value.
And handling units are in the range:	Number of containers.
From	Enter the minimum value.
To	Enter the maximum value.
And if requested carrier service code is	
Carrier Service Code	Select a carrier service code.
For more information about defining carrier services, see <a href="#">Section 5.4.2.1.1, "Defining Carrier Services"</a> .	
Then ship via:	
Priority	Indicates the number to give this rule a relative importance.  When a shipment is compared to the routing guide lines, there may be two carrier services that could be used. This priority serves as a tie breaker. The carrier service with the lowest number is used.
Carrier / Service	Indicates the carrier and service code that is desired.
Break Bulk Node	The break bulk node that is close to the buyer.
Contact Specified	Indicates whether the contact details for the shipment is specified.

**Table 5–5 Routing Guide Line Details**

With overrides:	
Override Freight Terms	Select to override the shipment's Freight Term.
Override Ship To	To override the Ship To value, select this field, and then select one of the following. This is only used when performing routing again due to a revised ship to address.
Node	Select the node name.
Consolidator	Select the consolidator name.
Store#	Select the store number.

When the conditions set are assessed, the routing guide line which matches the most conditions is used. For example, imagine there are three routing guide lines:

*Routing guide line A* - What to do when shipping from Massachusetts

*Routing guide line B* - What to do when shipping from Massachusetts, and when shipping from the zip code 01810.

*Routing guide line C* - What to do when shipping from Massachusetts or NY.

If the shipment originates from the zip code 01810, it matches all of these routing guide lines. The actions specified in *Routing guide line B* is used, as more conditions are met (both the state and the zip code).

If the shipment originates from Massachusetts, but not from zip code 01810, then both *Routing guideline A* and *Routing guide line C* match. The priority on the guidelines are used to determine which is used, with the lowest numbered priority being selected. If *Routing guideline A* had a priority number of 3, and *Routing guideline C* had a priority number of 5, *Routing guideline A* is used.

#### **5.4.2.1.1 Defining Carrier Services**

When routing occurs, the shipment is matched against the routing guidelines. Based on the criteria specified, you select a carrier service to use.

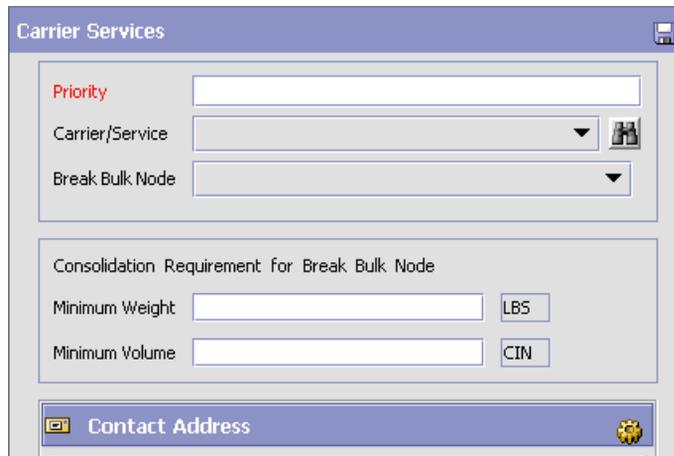
You can use the Carrier Services panel for:

- [Creating a Carrier Service](#)
- [Modifying a Carrier Service](#)
- [Deleting a Carrier Service](#)

### Creating a Carrier Service

To create a carrier service:

1. From the Routing Guidelines Details window, in the Carrier Services panel, select . The Carrier Services window displays.



The image shows a software window titled "Carrier Services". It contains several input fields and dropdown menus. At the top, there is a "Priority" text box. Below it is a "Carrier/Service" dropdown menu with a small icon to its right. Underneath is a "Break Bulk Node" dropdown menu. A section titled "Consolidation Requirement for Break Bulk Node" contains two rows: "Minimum Weight" with a text box and a "LBS" button, and "Minimum Volume" with a text box and a "CIN" button. At the bottom, there is a "Contact Address" section with a small icon to its right.

2. Enter information in the applicable fields. Refer to [Table 5–6](#) for field value descriptions.
3. Choose .

**Table 5–6 Carrier Services**

Fields	Description
Priority	Enter a number to give this rule a relative importance. When a shipment is compared to the routing guide lines, there may be two carrier services that could be used. This priority serves as a tie breaker. The carrier service with the lowest number is used.
Carrier/Service	Select the carrier or service code that is desired.
Break Bulk Node	Select the break bulk node that is close to the buyer.
Contact Address	This is used to specify the address information for the carrier service's contact person. Click  to change the contact Address.

### Modifying a Carrier Service

To modify a carrier service:

1. From the Routing Guidelines Details window, in the Carrier Services panel, select a carrier service from the list in the Carrier Services list window, and select . The Carrier Services window displays.
2. Enter the new information in the applicable fields. Refer to [Table 5–6](#) for field value descriptions.
3. Choose .

### Deleting a Carrier Service

To modify a carrier service:

1. From the Routing Guidelines Details window, in the Carrier Services panel, select a carrier service in the Carrier Services list window and select .
2. Choose .

### 5.4.2.2 Modifying a Routing Guide Line

To modify a routing guide line:

1. From the Routing Guidelines Details window, select the Routing Details Tab. A Routing Guide Line search window displays.
2. Select a routing guide line in the Routing Guide Line list window, and select . The Routing Guide Line Details window displays.
3. Enter the new information in the applicable fields. Refer to [Table 5–5](#) for field value descriptions.
4. Choose .

### 5.4.2.3 Deleting a Routing Guide Line

To delete a Routing Guide Line:

1. From the Routing Guide Lines Details window, select the Routing Details Tab. A Routing Guide Line search window displays.
2. Select a routing guide line in the Routing Guide Line list window, and choose .

## 5.4.3 Deleting a Routing Guide

To delete a routing guide:

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Outbound Constraints. The Outbound Constraints window displays in the work area.
2. Select the applicable Routing Guide and choose .

# Configuring Cross Application Negotiation Components

---

Negotiation is the process in which two organizations can settle the conditions of an order document. You can define rules and common codes that pertain to the negotiation process.

You can use the Negotiation branch for:

- [Defining Response Actions](#)
- [Defining Rejection Reasons](#)

## 4.1 Defining Response Actions

You can define common codes for **response actions** used in the negotiation process. These codes identify specific actions taken when responses to negotiation are made between two organizations.

Following are the Selling and Fulfillment Foundation default response actions:

- 1100 - Offer
- 1200 - CounterOffer
- 1300 - Reject
- 1400 - Remove
- 1500 - Accept

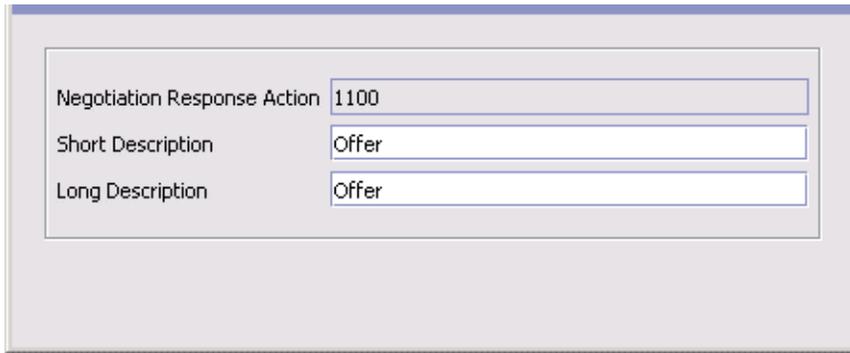
You can use the Negotiation Response Actions branch for:

- [Modifying a Response Action](#)

### 4.1.1 Modifying a Response Action

To modify a response action:

1. From the tree in the application rules side panel, choose Cross Application > Negotiation > Negotiation Response Actions. The Negotiation Response Action window displays in the work area.
2. Select the applicable response action and choose . The Response Action Details pop-up window displays.



Negotiation Response Action	1100
Short Description	Offer
Long Description	Offer

3. In Short Description, enter a brief description of the response action.
4. In Long Description, enter a more detailed description of the response action.
5. Choose .

## 4.2 Defining Rejection Reasons

You can define common codes for **rejection reasons** used in the negotiation process. These codes identify the specific reason an offer in the negotiation process is rejected.

You can use the Negotiation Reject Reasons branch for:

- [Creating a Rejection Reason](#)
- [Modifying a Rejection Reason](#)
- [Deleting a Rejection Reason](#)

## 4.2.1 Creating a Rejection Reason

To create a rejection reason:

1. From the tree in the application rules side panel, choose Cross Application > Negotiation > Negotiation Reject Reasons. The Negotiation Reject Reasons window displays in the work area.
2. Choose . The Rejection Reason Details pop-up window displays.



The screenshot shows a dialog box titled 'Rejection Reason Details'. It contains three text input fields with the following labels: 'Negotiation Reject Reason', 'Short Description', and 'Long Description'. Each field is currently empty.

3. In Rejection Reason, enter the rejection reason.
4. In Short Description, enter a brief description of the rejection reason.
5. In Long Description, enter a more detailed description of the rejection reason.
6. Choose .

## 4.2.2 Modifying a Rejection Reason

To modify a rejection reason:

1. From the tree in the application rules side panel, choose Cross Application > Negotiation > Negotiation Reject Reasons. The Negotiation Reject Reasons window displays in the work area.
2. Select the applicable rejection reason and choose . The Rejection Reason Details pop-up window displays.
3. In Short Description, enter a brief description of the rejection reason.

4. In Long Description, enter a more detailed description of the rejection reason.
5. Choose .

### 4.2.3 Deleting a Rejection Reason

To delete a rejection reason:

1. From the tree in the application rules side panel, choose Cross Application > Negotiation > Negotiation Reject Reasons. The Negotiation Reject Reasons window displays in the work area.
2. Select the applicable rejection reason and choose .

# 5

## Configuring Cross Application Vendor Components

---

You can define trading vendors in the Supply Collaboration module. You can use Vendor branch for:

- [Defining Vendor Classifications](#)
- [Defining Vendor Definitions](#)
- [Defining Contact Types](#)

### 5.1 Defining Vendor Classifications

You can configure the vendor classification codes used to associate with a vendor identification master. For more information about creating a vendor identification master, see [Section 5.2, "Defining Vendor Definitions"](#) on page 69.

You can use the Vendor Classification branch for:

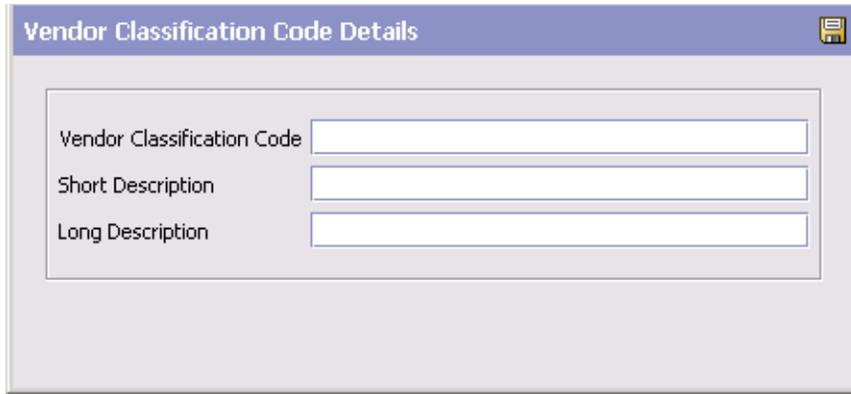
- [Creating a Vendor Classification](#)
- [Modifying a Vendor Classification](#)
- [Deleting a Vendor Classification](#)

#### 5.1.1 Creating a Vendor Classification

To create a vendor classification:

1. From the menu bar, choose Applications > Supply Collaboration. The Supply Collaboration tree displays in the side panel.

2. From the Supply Collaboration tree, choose Cross Application > Vendor > Vendor Classifications. The Vendor Classification Codes window displays in the work area.
3. Choose . The Vendor Classification Code Details pop-up window displays.



The screenshot shows a pop-up window titled "Vendor Classification Code Details". The window has a light blue header bar with the title and a save icon on the right. The main content area is white and contains three text input fields, each with a label to its left: "Vendor Classification Code", "Short Description", and "Long Description". The fields are empty and have a light blue border.

4. In Vendor Classification Code, enter the classification ID code.
5. In Short Description, enter a brief description of the classification ID code.
6. In Long Description, enter a more detailed description of the classification ID code.
7. Choose .

### 5.1.2 Modifying a Vendor Classification

To modify a vendor classification:

1. From the menu bar, choose Applications > Supply Collaboration. The Supply Collaboration tree displays in the side panel.
2. From the Supply Collaboration tree, choose Cross Application > Vendor > Vendor Classifications. The Vendor Classification Codes window displays in the work area.
3. Select the applicable vendor classification code and choose . The Vendor Classification Code Details pop-up window displays.

4. In Short Description, enter a brief description of the classification ID code.
5. In Long Description, enter a more detailed description of the classification ID code.
6. Choose .

### 5.1.3 Deleting a Vendor Classification

To delete a vendor classification:

1. From the menu bar, choose Applications > Supply Collaboration. The Supply Collaboration tree displays in the side panel.
2. From the Supply Collaboration tree, choose Cross Application > Vendor > Vendor Classifications. The Vendor Classification Codes window displays in the work area.
3. Select the applicable vendor classification code and choose .

## 5.2 Defining Vendor Definitions

You can configure vendor definitions used to establish a relationship between an organization and its Seller. When creating a vendor definition you associate an existing Seller organization with a specific vendor ID and classification. The vendor identification uniquely identifies the Seller organization in instances where multiple ERP systems download Seller information into Selling and Fulfillment Foundation.

You can use the Vendor Definition branch for:

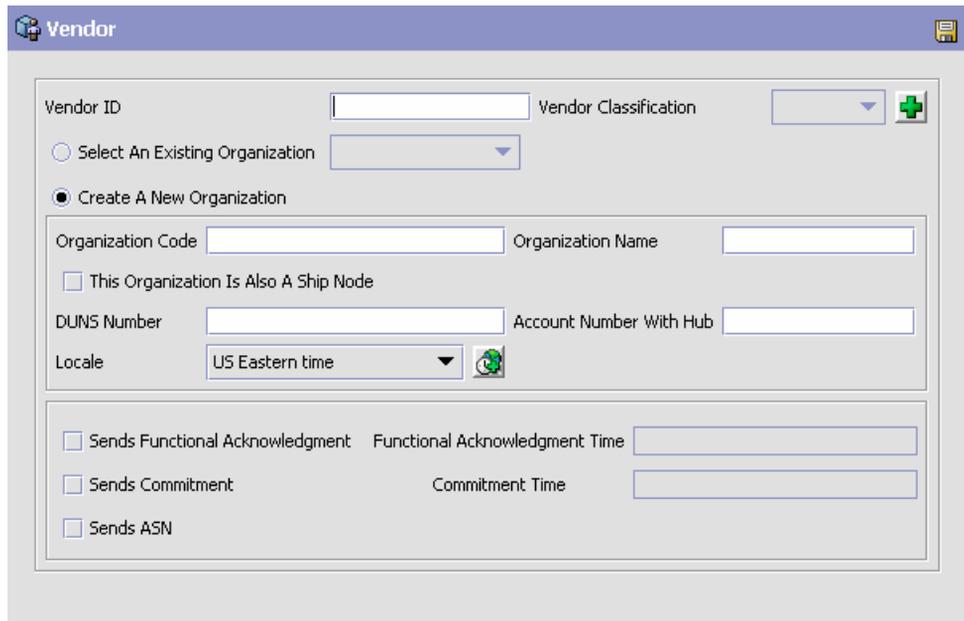
- [Creating a Vendor Definition](#)
- [Modifying a Vendor Definition](#)
- [Deleting a Vendor Definition](#)

### 5.2.1 Creating a Vendor Definition

To create a vendor definition:

1. From the menu bar, choose Applications > Supply Collaboration. The Supply Collaboration tree displays in the side panel.

2. From the Supply Collaboration tree, choose Cross Application > Vendor > Vendor Definitions. The Vendor Search window displays in the work area.
3. Choose . The Vendor pop-up window displays.
4. Enter information into the applicable fields. Refer to [Table 5–1](#) for field value descriptions.
5. Choose .



**Table 5–1 Vendor Pop-Up Window**

Field	Description
Vendor ID	Enter the identifier of the vendor.
Vendor Classification	Select the classification, if applicable.
Select An Existing Organization	Choose this option and select the applicable Seller if you want to associate the vendor with an existing Seller organization.

**Table 5–1 Vendor Pop-Up Window**

Field	Description
Create A New Organization	Choose this option if you want to create a new organization to associate with the vendor.
Organization Code	Enter the organization code.
Organization Name	Enter the name of the organization.
This Organization Is Also A Ship Node	Select this if the new organization is also a ship node.
DUNS Number	Enter the DUNS number of the seller.
Account Number With Hub	Enter the account number of the seller assigned by the Hub organization.
Locale	Select the locale of the seller.
Sends Functional Acknowledgment	Check this box if the supplier sends a functional acknowledgment for a PO.
Functional Acknowledgment Time	Enter the number of hours the supplier took to send the functional acknowledgement for a PO.
Sends Commitment	Check this box if the supplier sends a commitment for a PO.
Commitment Time	Enter the number of hours the supplier took to send the commitment for a PO.
Sends ASN	Check this box if the supplier sends an Advanced Shipment Notice (ASN) for a PO.

## 5.2.2 Modifying a Vendor Definition

To modify a vendor definition:

1. From the menu bar, choose Applications > Supply Collaboration. The Supply Collaboration tree displays in the side panel.
2. From the Supply Collaboration tree, choose Cross Application > Vendor > Vendor Definitions. The Vendor Search window displays in the work area.
3. Enter applicable search criteria and choose . A list of vendors displays.
4. Locate the applicable vendor and choose . The Vendor pop-up window displays.

5. From Vendor Classification, select the Seller's customer classification, if applicable.
6. From Seller Organization, select the Seller organization to associate with the vendor ID.
7. Choose .

### 5.2.3 Deleting a Vendor Definition

To delete a vendor definition:

1. From the menu bar, choose Applications > Supply Collaboration. The Supply Collaboration tree displays in the side panel.
2. From the Supply Collaboration tree, choose Cross Application > Vendor > Vendor Definitions. The Vendor Search window displays in the work area.
3. Enter applicable search criteria and choose . A list of vendors displays.
4. Locate the applicable vendor and choose .

## 5.3 Defining Contact Types

You can configure the contact types when specifying the contact information of a vendor on inbound order notes. For more information about creating a customer, see [Section 5.2, "Defining Vendor Definitions"](#) on page 69.

You can use the Contact Types branch for:

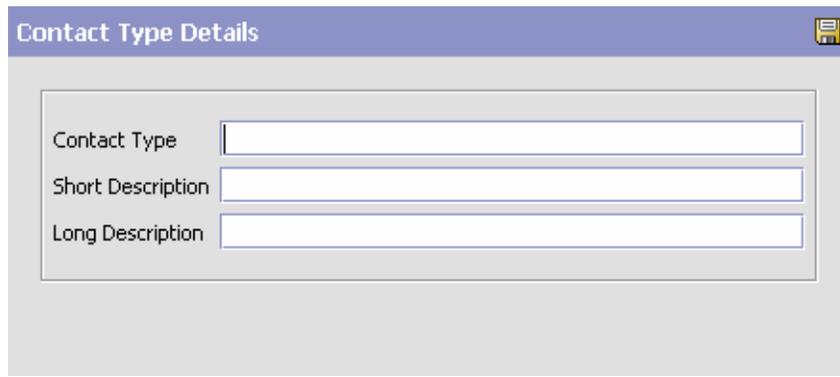
- [Creating a Contact Type](#)
- [Modifying a Contact Type](#)
- [Deleting a Contact Type](#)

### 5.3.1 Creating a Contact Type

To create a contact type:

1. From the tree in the application rules side panel, choose Cross Application > Vendor > Contact Types. The Contact Types window displays in the work area.

2. Click . The Contact Type Details pop-up window displays.



The screenshot shows a pop-up window titled "Contact Type Details". Inside the window, there are three text input fields stacked vertically. The first field is labeled "Contact Type", the second is labeled "Short Description", and the third is labeled "Long Description". Each field is currently empty. There is a close button (represented by a small square icon with a diagonal line) in the top right corner of the window.

3. In Contact Type, enter the contact type.
4. In Short Description, enter a brief description of the contact type.
5. In Long Description, enter a more detailed description of the contact type.
6. Click .

### 5.3.2 Modifying a Contact Type

To modify a contact type:

1. From the tree in the application rules side panel, choose Cross Application > Vendor > Contact Types. The Contact Types window displays in the work area.
2. Select the applicable contact type and click . The Contact Type Details pop-up window displays.
3. In Short Description, enter a brief description of the contact type.
4. In Long Description, enter a more detailed description of the contact type.
5. Click .

### 5.3.3 Deleting a Contact Type

To delete a contact type:

1. From the tree in the application rules side panel, choose Cross Application > Vendor > Contact Types. The Contact Types window displays in the work area.
2. Select the contact type and click .

# Configuring a Document's Attributes

---

**You can** define common codes as they pertain to order documents viewed in the Application Consoles.

You can use the Order Attributes branch for

- [Defining Order Types](#)
- [Defining Order Sources](#)
- [Defining External References for the Order Level](#)
- [Defining External References for the Order Line Level](#)
- [Defining Order Address Types](#)
- [Defining Line Types](#)
- [Defining Other Attributes](#)

## 9.1 Defining Order Types

You can define codes for order types that appear on a document type. This code has no application logic associated with it and can be set up as per your business practices. Examples of order types are Consumer Orders, Service Rep Orders, and Retail Orders.

You can use the Order Types tab for:

- [Creating an Order Type](#)
- [Modifying an Order Type](#)
- [Deleting an Order Type](#)

### 9.1.1 Creating an Order Type

To create an order type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order Types tab.
3. Choose . The Order Type Details pop-up window displays.



The screenshot shows a dialog box titled "Order Type Details". It contains three text input fields, each with a label to its left: "Order Type", "Short Description", and "Long Description". The fields are empty and have a light blue border.

4. In Order Type, enter the name of the order type.
5. In Short Description, enter a brief description of the order type.
6. In Long Description, enter a more detailed description of the order type.
7. Choose .

### 9.1.2 Modifying an Order Type

To modify a order type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order Types tab.
3. Select the applicable order type and choose . The Order Type Details pop-up window displays.

4. In Short Description, enter a brief description of the order type.
5. In Long Description, enter a more detailed description of the order type.
6. Choose .

### 9.1.3 Deleting an Order Type

To delete a order type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order Types tab.
3. Select the applicable order type and choose .

## 9.2 Defining Order Sources

You can define codes for order sources that appear on a document type. This code has no application logic associated with it and can be set up as per your business practices.

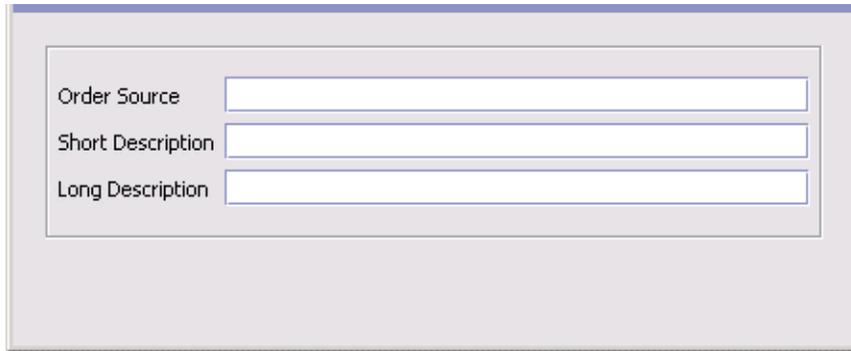
You can use the Order Sources tab for:

- [Creating an Order Source](#)
- [Modifying an Order Source](#)
- [Deleting an Order Source](#)

### 9.2.1 Creating an Order Source

To create an order source:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order Sources tab.
3. Choose . The Order Source Details pop-up window displays.



The screenshot shows a dialog box with a light gray background. Inside, there is a white rectangular area containing three text input fields. The first field is labeled 'Order Source', the second 'Short Description', and the third 'Long Description'. Each label is positioned to the left of its corresponding text box.

4. In Order Source, enter the name of the order source.
5. In Short Description, enter a brief description of the order source.
6. In Long Description, enter a more detailed description of the order source.
7. Choose .

### 9.2.2 Modifying an Order Source

To modify a order source:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order Sources tab.
3. Select the applicable order source and choose . The Order Source Details pop-up window displays.
4. In Short Description, enter a brief description of the order source.
5. In Long Description, enter a more detailed description of the order source.
6. Choose .

### 9.2.3 Deleting an Order Source

To delete a order source:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order Sources tab.
3. Select the applicable order source and choose .

## 9.3 Defining External References for the Order Level

You can define codes for external references that appear on a document type at the order level. This code has no application logic associated with it and can be set up as per your business practices. You can create, modify, and delete external references.

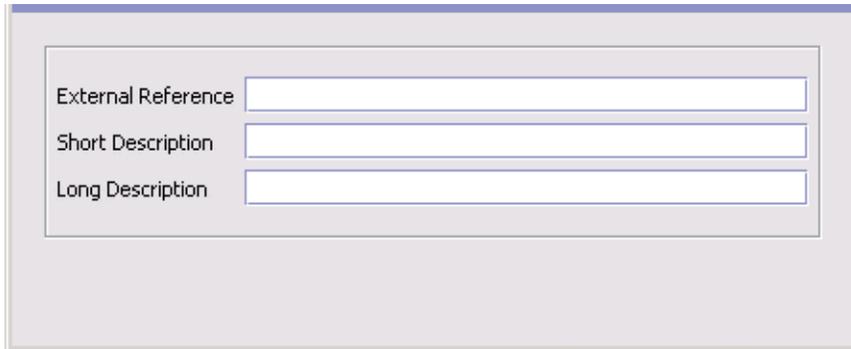
You can use the Order References tab for:

- [Creating an External Reference for the Order Header Level](#)
- [Modifying an External Reference for the Order Header Level](#)
- [Deleting an External Reference for the Order Header Level](#)

### 9.3.1 Creating an External Reference for the Order Header Level

To create an order reference for the order level:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order References tab.
3. From the Order Header External References list choose . The External Reference Details pop-up window displays.



The screenshot shows a dialog box with three text input fields. The first field is labeled 'External Reference', the second is labeled 'Short Description', and the third is labeled 'Long Description'. Each field is currently empty.

4. In External Reference, enter the name of the external reference.
5. In Short Description, enter a brief description of the external reference.
6. In Long Description, enter a more detailed description of the external reference.
7. Choose .

### 9.3.2 Modifying an External Reference for the Order Header Level

To modify an external reference for the order level:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order References tab.
3. In Order Header External References select the applicable external reference and choose . The External Reference Details pop-up window displays.
4. In Short Description, enter a brief description of the external reference.
5. In Long Description, enter a more detailed description of the external reference.
6. Choose .

### 9.3.3 Deleting an External Reference for the Order Header Level

To delete an external reference for the order level:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order References tab.
3. In Order Header External References select the applicable external reference and choose .

## 9.4 Defining External References for the Order Line Level

You can define codes for external references that appear on a document type at the order line level. This code has no application logic associated with it and can be set up as per your business practices. You can create, modify, and delete external references.

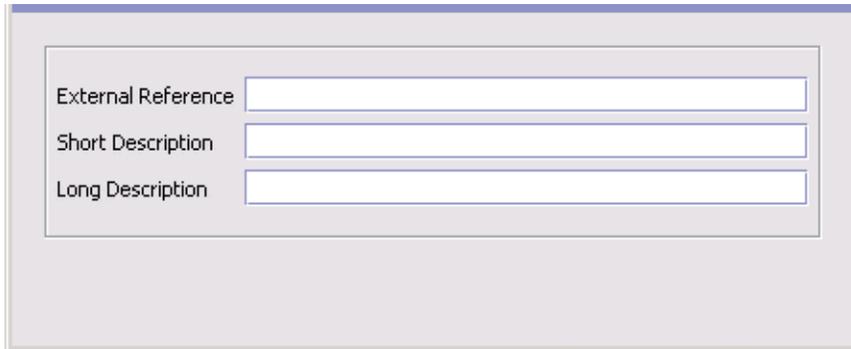
You can use the Order References tab for:

- [Creating an External Reference for the Order Line Level](#)
- [Modifying an External Reference for the Order Line Level](#)
- [Deleting an External Reference for the Order Line Level](#)

### 9.4.1 Creating an External Reference for the Order Line Level

To create an order reference for the order line level:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order References tab.
3. From the Order Line External References list choose . The External Reference Details pop-up window displays.



The screenshot shows a dialog box with three text input fields. The first field is labeled 'External Reference', the second is labeled 'Short Description', and the third is labeled 'Long Description'. Each field is empty and has a light blue border.

4. In External Reference, enter the name of the external reference.
5. In Short Description, enter a brief description of the external reference.
6. In Long Description, enter a more detailed description of the external reference.
7. Choose .

### 9.4.2 Modifying an External Reference for the Order Line Level

To modify an external reference for the order line level:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order References tab.
3. In Order Line External References select the applicable external reference and choose . The External Reference Details pop-up window displays.
4. In Short Description, enter a brief description of the external reference.
5. In Long Description, enter a more detailed description of the external reference.
6. Choose .

### 9.4.3 Deleting an External Reference for the Order Line Level

To delete an external reference for the order level:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order References tab.
3. In the Order Line External References list select the applicable external reference and choose .

## 9.5 Defining Order Address Types

You can define codes for order address types that appear in the Additional Addresses view in the User Interface for a document type.

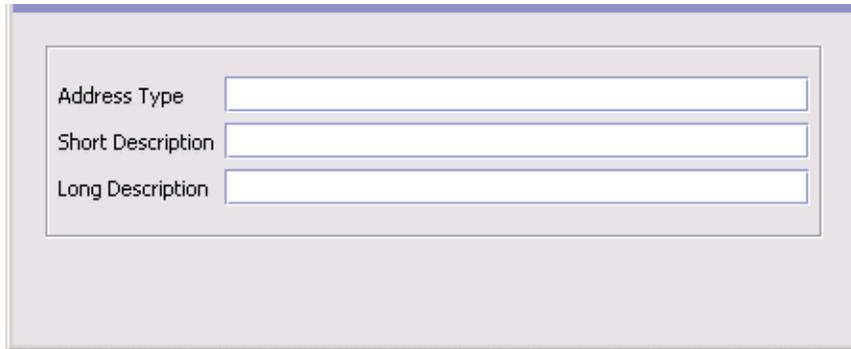
You can use the Order Address Types tab for:

- [Creating an Order Address Type](#)
- [Modifying an Order Address Type](#)
- [Deleting an Order Address Type](#)

### 9.5.1 Creating an Order Address Type

To create an order address type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order Address Types tab.
3. Choose . The Order Address Type Details pop-up window displays.



The screenshot shows a dialog box with a light gray background. Inside, there is a white rectangular area containing three text input fields. The first field is labeled 'Address Type', the second is labeled 'Short Description', and the third is labeled 'Long Description'. Each label is positioned to the left of its corresponding text box.

4. In Order Address Type, enter the name of the order address type.
5. In Short Description, enter a brief description of the order address type.
6. In Long Description, enter a more detailed description of the order address type.
7. Choose .

### 9.5.2 Modifying an Order Address Type

To modify a order address type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order Address Types tab.
3. Select the applicable order address type and choose . The Order Address Type Details pop-up window displays.
4. In Short Description, enter a brief description of the order type.
5. In Long Description, enter a more detailed description of the order type.
6. Choose .

### 9.5.3 Deleting an Order Address Type

To delete a order address type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Order Address Types tab.
3. Select the applicable order address type and choose .

## 9.6 Defining Line Types

You can define codes and for line types that appear on a document type.

You can use the Line Types tab for:

- [Creating a Line Type](#)
- [Modifying a Line Type](#)
- [Deleting a Line Type](#)

### 9.6.1 Creating a Line Type

To create a line type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Line Types tab.
3. Choose . The Line Type Details pop-up window displays.
4. In Line Type, enter the name of the line type.
5. In Description, enter a brief description of the line type.
6. Choose .



The image shows a dialog box with a light gray background. Inside the dialog, there is a white rectangular area containing two text input fields. The first field is labeled 'Line Type' and the second is labeled 'Description'. Both fields are currently empty.

### 9.6.2 Modifying a Line Type

To modify a line type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Line Types tab.
3. Select the applicable line type and choose . The Line Type Details pop-up window displays.
4. In Description, enter a brief description of the line type.
5. Choose .

### 9.6.3 Deleting a Line Type

To delete a line type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Line Types tab.
3. Select the applicable line type and choose .

## 9.7 Defining Other Attributes

You can define other attributes that appear on the document type.

You can use the Others tab for:

- [Generating a Prime Line Number for a New Line from a Pre-Configured Number](#)

### 9.7.1 Generating a Prime Line Number for a New Line from a Pre-Configured Number

Generating a prime line number for a new line from a pre-configured number prevents conflicts between prime line numbers in Selling and Fulfillment Foundation and in an external system when order synchronization occurs.

To specify a pre-configured starting number:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Attributes. The Order Attributes window displays in the work area.
2. Choose the Others tab.
3. In Generate Prime Line No. For New Line Starting From, enter the starting number. The starting prime line number must be a positive integer
4. Choose .



Order Types | Order Sources | Order References | Order Address Types | Line Types | Others

Generate Prime Line Number For New Line Starting From

**Note:** The value entered in the "Generate Prime Line Number for New Line Starting From:" field only affects orders created through the Console UI, not through direct API calls (e.g. createOrder()).



## Configuring a Document's Order Validation

---

You can define the configuration for defaulting Seller and Buyer validation during order creation for a particular Enterprise and document type. This validation is used to determine the Sellers and Buyers available to create an order for, and narrows the search results in the Application Consoles based on the validation type you configured.

For example, you are configuring a Hub environment with 10 Enterprises, 50 Sellers, and 100 Buyers. A particular Enterprise only interacts with 10 of the 50 Sellers and 25 of the 100 Buyers as defined in the organization hierarchy. If you set both the Seller and Buyer validations to 'Defined In The Enterprise Hierarchy', when a user creates an order the system verifies that the Seller on the order is one of the 10 Sellers defined in the Enterprise's hierarchy and the Buyer on the order is one of the 25 Buyers defined in the Enterprise's hierarchy. Also, if the user chooses the lookup for either the Seller or Buyer fields, only the Sellers and Buyers defined for the Enterprise appear in the results.

To define an order document's order validation:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Order Validation. The Order Validation pop-up window displays in the work area.
2. Enter information into the applicable fields. Refer to [Table 10–1](#) for field value descriptions.
3. Choose .

 Order Validation 

Seller Validation  ▼

Buyer Validation  ▼

Validate Bill To Id As Customer Id

Validate Vendor Id

Validate Item

**Table 10–1 Order Validation Pop-Up Window**

Field	Description
Seller Validation	<p>Select the type of validation you want to use to verify the Seller on the order document.</p> <p>You can choose from the following options:</p> <ul style="list-style-type: none"><li>• None - No validation is performed for Sellers on an order. All Sellers in the system can be used during order creation. Also, all Sellers in the system display when the Seller lookup is chosen in the Application Consoles.</li><li>• Same As Enterprise - The system validates the Seller on the order is the Enterprise.</li><li>• Defined In The Enterprise Hierarchy - The system validates that the Seller on the order is defined within the Enterprise's organizational hierarchy. Also, only the Sellers defined within the Enterprise's organizational hierarchy display when the Seller lookup is chosen in the Application Consoles. For more information about configuring the organizational hierarchy, see the <i>Selling and Fulfillment Foundation: Application Platform Configuration Guide</i>.</li></ul> <p>Customer Of The Enterprise - The system validates that the Seller on the order has been configured as a customer. Also, only the organizations defined as customers of the Enterprise display when the Seller lookup is chosen in the Application Consoles.</p>

**Table 10–1 Order Validation Pop-Up Window**

Field	Description
Buyer Validation	<p>Select the type of validation you want to use to verify the Buyer on the order document.</p> <p>You can choose from the following options:</p> <ul style="list-style-type: none"> <li>• None - No validation is performed for Buyers on an order. All Buyers in the system can be used during order creation. Also, all Buyers in the system display when the Buyer lookup is chosen in the Application Consoles.</li> <li>• Same As Enterprise - The system validates the Buyer on the order is the Enterprise.</li> <li>• Defined In The Enterprise Hierarchy - The system validates that the Buyer on the order is defined within the Enterprise’s organizational hierarchy. Also, only the Buyers defined within the Enterprise’s organizational hierarchy display when the Buyer lookup is chosen in the Application Consoles. For more information about configuring the organizational hierarchy, see the <i>Selling and Fulfillment Foundation: Application Platform Configuration Guide</i>.</li> <li>• Customer Of The Enterprise - The system validates that the Buyer on the order has been configured as a customer. Also, only the organizations defined as customers of the Enterprise display when the Buyer lookup is chosen in the Application Consoles.</li> </ul>
Validate Bill To ID As Customer ID	<p>Select Validate Bill To ID As Customer ID if you want to validate that the customer ID on an order is defined for the Enterprise.</p>
Validate Vendor ID	<p>Select Validate Vendor ID if you want to validate that the vendor ID on an order is defined for the Enterprise.</p>
Validate Item	<p>Select Validate Item if you want to validate that the product items on the order belong to the Enterprises catalog. Service items, on the other hand, always need to exist within Selling and Fulfillment Foundation.</p>

# 11

## Configuring a Document's Instruction Types

---

You can define the common codes used when adding special instructions to an order document.

The default instruction types of Selling and Fulfillment Foundation are:

- PICK
- PACK
- SHIP
- GIFT
- ORDERING
- OTHER

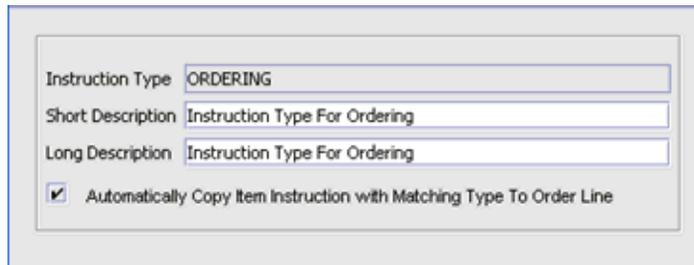
You can use the Instruction Types branch for:

- [Creating an Instruction Type](#)
- [Modifying an Instruction Type](#)
- [Deleting an Instruction Type](#)

### 11.1 Creating an Instruction Type

To create an instruction type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Instruction Types. The Instruction Types window displays in the work area.
2. Choose . The Instruction Type Details pop-up window displays.



Instruction Type: ORDERING

Short Description: Instruction Type For Ordering

Long Description: Instruction Type For Ordering

Automatically Copy Item Instruction with Matching Type To Order Line

3. In Instruction Type, enter the instruction type.
4. In Short Description, enter a brief description of the instruction type.
5. In Long Description, enter a more detailed description of the instruction type.
6. Check Automatically Copy Item Instruction with Matching Type To Order Line to force the system to automatically copy item instructions with matching instruction types to order lines when the items are added onto an order.
7. Choose .

## 11.2 Modifying an Instruction Type

To modify an instruction type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Instruction Types. The Instruction Types window displays in the work area.
2. Select the applicable instruction type and choose . The Instruction Type Details pop-up window displays.
3. In Short Description, enter a brief description of the instruction type.
4. In Long Description, enter a more detailed description of the instruction type.

5. Check Automatically Copy Item Instruction with Matching Type To Order Line to force the system to automatically copy item instructions with matching instruction types to order lines when the items are added onto an order.
6. Choose .

## 11.3 Deleting an Instruction Type

To delete an instruction type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Instruction Types. The Instruction Types window displays in the work area.
2. Select the applicable instruction type and choose .



# 12

## Configuring a Document's Modification Reasons

---

You can define common codes for **modification reasons**. These codes define why a modification was made by a user in the Application Consoles.

**Note:** In addition to modification reasons, the codes that you define are used as hold reasons when you put an order on hold in the Application Consoles.

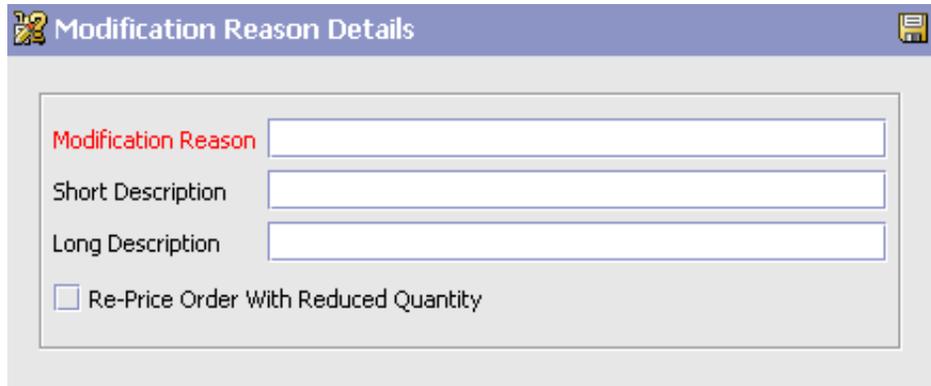
You can use the Modification Reasons branch for:

- [Creating a Modification Reason](#)
- [Modifying a Modification Reason](#)
- [Deleting a Modification Reason](#)

### 12.1 Creating a Modification Reason

To create a modification reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Modification Reasons. The Modification Reasons window displays in the work area.
2. Choose . The Modification Reason Details pop-up window displays.



**Modification Reason Details**

Modification Reason

Short Description

Long Description

Re-Price Order With Reduced Quantity

3. In Modification Reason, enter the modification reason.
4. In Short Description, enter a brief description of the modification reason.
5. In Long Description, enter a more detailed description of the modification reason.
6. If this modification reason requires that the order be re-priced due to a reduced quantity, check the Re-Price Order With Reduced Quantity checkbox.

This flag is applicable only if this modification reason is used for cancellations, where re-pricing needs to occur against a reduced quantity: the quantity against which the order line is re-priced (re-pricing quantity) is adjusted to the reduced quantity. For more information about re-pricing quantity, see the *Selling and Fulfillment Foundation: Javadocs*.

---

---

**Note:** If this modification reason is used for a modification which does not reduce quantity, this flag is not applicable.

---

---

---

---

**Note:** This field does not exist for Load Modification Reasons.

---

---

7. Choose .

## 12.2 Modifying a Modification Reason

To modify a modification reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Modification Reasons. The Modification Reasons window displays in the work area.
2. Select the applicable modification reason and choose . The Modification Reason Details pop-up window displays.
3. In Short Description, enter a brief description of the modification reason.
4. In Long Description, enter a more detailed description of the modification reason.
5. If this modification reason requires that the order be re-priced due to a reduced quantity, check the Re-Price Order With Reduced Quantity checkbox.

This flag is applicable only if this modification reason is used for cancellations, where re-pricing needs to occur against a reduced quantity: the quantity against which the order line is repriced (re-pricing quantity) is adjusted to the reduced quantity. For more information about re-pricing quantity, see the *Selling and Fulfillment Foundation: Javadocs*.

---

---

**Note:** If this modification reason is used for a modification which does not reduce quantity, this flag is not applicable.

---

---

---

---

**Note:** This field does not exist for Load Modification Reasons.

---

---

6. Choose .

## 12.3 Deleting a Modification Reason

To delete a modification reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Modification Reasons. The Modification Reasons window displays in the work area.
2. Select the applicable modification reason and choose .

# 13

## Configuring a Document's Backorder Reasons

---

You can define common codes for **backorder reasons**. These codes describe why an order was backordered.

The default backorder reason of Selling and Fulfillment Foundation is:

- No Stock

You can use the Backorder Reasons branch for:

- [Creating a Backorder Reason](#)
- [Modifying a Backorder Reason](#)
- [Deleting a Backorder Reason](#)

### 13.1 Creating a Backorder Reason

To create a backorder reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Backorder Reasons. The Backorder Reasons window displays in the work area.
2. Choose . The Backorder Reason Details pop-up window displays.



The screenshot shows a window titled "Backorder Reason" with three text input fields. The first field is labeled "Backorder Reason", the second is labeled "Short Description", and the third is labeled "Long Description". Each field is empty and has a light blue border.

3. In Backorder Reason, enter the backorder reason.
4. In Short Description, enter a brief description of the backorder reason.
5. In Long Description, enter a more detailed description of the backorder reason.
6. Choose .

## 13.2 Modifying a Backorder Reason

To modify a backorder reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Backorder Reasons. The Backorder Reasons window displays in the work area.
2. Select the applicable backorder reason and choose . The Backorder Reason Details pop-up window displays.
3. In Short Description, enter a brief description of the backorder reason.
4. In Long Description, enter a more detailed description of the backorder reason.
5. Choose .

## 13.3 Deleting a Backorder Reason

To delete a backorder reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Backorder Reasons. The Backorder Reasons window displays in the work area.
2. Select the applicable backorder reason and choose .



## Configuring a Document's Modification Components

---

You can configure the modification rules and types of a document when it is in a specific status. These rules determine which parts of a document can be modified as well as in which status the modifications can be performed.

If you are using the Distributed Order Management module, you can configure modification components at the following process type levels:

- Fulfillment
- Outbound Logistics

If you are using the Logistics Management module, you can configure modification components at the load process type level.

If you are using the Supply Collaboration module, you can configure modification components at the following process type levels:

- Fulfillment
- Inbound Logistics

If you are using the Reverse Logistics module, you can configure modification components at the following process type levels:

- Fulfillment
- Logistics
- Receipt

You can use the Order Modification branch for:

- [Defining Modification Rules](#)
- [Defining Custom Modification Types](#)
- [Defining Modifications Impacting Pricing](#)

## 16.1 Defining Modification Rules

Most documents flow through a pipeline without requiring any intervention by a customer service representative. However, there are times when modifications are required, such as changing credit card information or quantity. Selling and Fulfillment Foundation supports modification through the Console and APIs. It is critical for you to decide which modifications are allowed for each modification type, modification level, and status combination.

---

---

**Important:** Contemplate business and system integration implications before allowing a modification that is disallowed as part of the system defaults. For example, adding instructions to a sales order document type is disallowed after the release has been sent to the node. If you change the modification to be allowed, the system has no way of communicating the new instruction to the node center because the release has already been sent.

---

---

The modification type indicates the type of modification carried out on a document. Selling and Fulfillment Foundation provides the ability to perform modifications on specific attributes. An example of a modification type is adding an order line to an order.

Modification level indicates the level at which a particular modification type is carried out. These include the following levels:

- Header
- Line
- Release
- Release Line
- Negotiation

- Negotiation Line
- Shipment
- Receipt

For a complete list of the system modification types and their modification levels, see [Appendix B, "Order Modification Types"](#).

Modifications are applied to a particular level and a particular processing status. For example, if modifications are requested for a document at the header level or at the line level, then the order lines, as well as the order release lines, are picked up for validating whether or not modifications are allowed for those order statuses. If modifications are requested at the release or release line level, then order release lines are picked up for validating whether or not modifications are allowed for those order statuses.

You can group modifications in the Modification Rules window by modification type, modification level, or status, by selecting the corresponding grouping from Group By. The Modification Rules window then displays the grouping you have chosen in a hierarchical structure.

All modification rules operate within a certain system-defined range. For instance, for Sales Orders, the Cancel modification on the order entity is always defined to be between the statuses 1000 (Draft Order Created) and 3350 (Included In Shipment). The system never allows a Cancel modification at a status of 3701 (Return Created). On the other hand, you are able to allow modifications between the statuses 1000 and 3350. If an entity is in multiple statuses, the modification is allowed, provided that at least one of the statuses is within the system-defined range.

If you make modifications such as changing a Bill To address after an order has shipped or a return has been created, the changed Bill To address will not be propagated to the shipment, the return order, and so forth.

The following table defines the different settings you can apply to modifications:

**Table 16–1 Order Document Type Rule Modifications**

Field	Description
Status	Indicates each status that is applicable to a modification level and type.
Allow	Indicates whether or not modifications may be made at this modification level and type for the specified status.
Disallow	Indicates that no modifications may be made at this modification level and type for the specified status.
Ignore	Indicates that modifications are ignored at this modification level and type for the specified status.
<p>There are several scenarios to consider for the Allow, Disallow, and Ignore settings:</p> <ul style="list-style-type: none"> <li>• If one line is in status 1 and another line is in status 2 - and both statuses are set to Allow, the modification is allowed.</li> <li>• If one line is in status 1, another line is in status 2, and another is in status 3 - and the 1 and 2 statuses are set to Allow, but the 3 status is set to Disallow, all modifications are disallowed, because one of the currently applied statuses is disallowed.</li> <li>• If one line is in status 1 and one is in the extended status 2 - If the 1 status is set to Allow, but the extended status is set to Ignore (all extended statuses are defaulted to ignore, so that they pick up their base status settings unless you have explicitly overridden the setting) then all modifications are allowed only if the base status is set to allow. If the base status is set to disallow, then all modifications are disallowed.</li> </ul> <p>If all lines are set to Ignore, then all modifications are disallowed, regardless of the base status settings.</p>	

---

---

**Note:** Application Console users can be granted permission to override the modification rules through user group permissions. When a user has been granted this permission, the user can perform a modification that has been disallowed within the Application Consoles. For more information about configuring user group permissions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

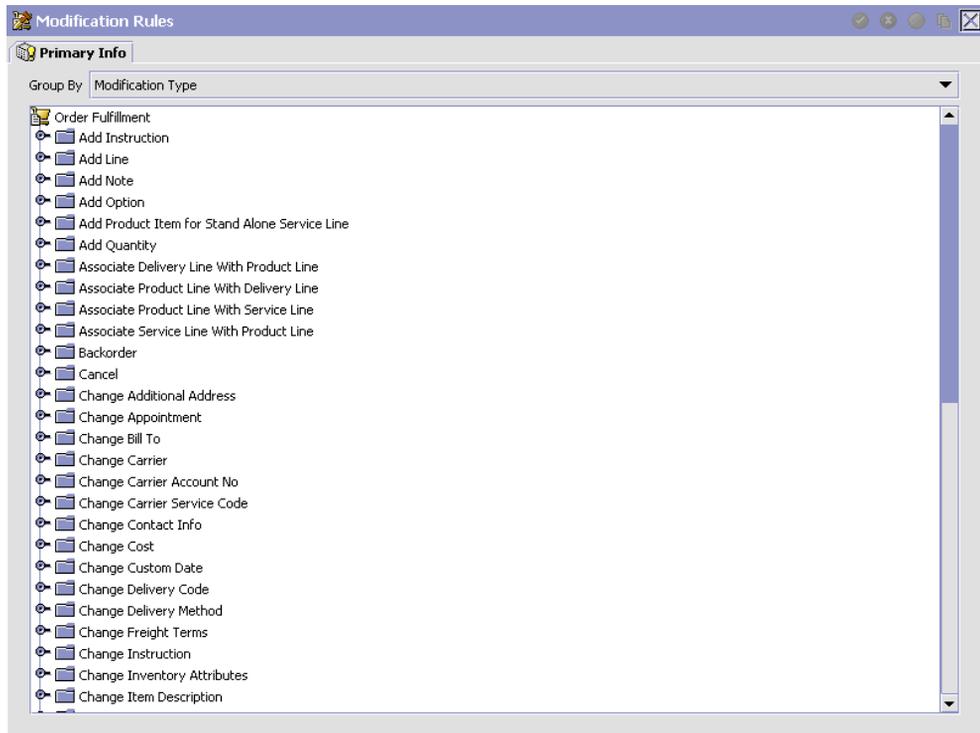
---

---

### 16.1.1 Changing Modification Rules

To change modification rules:

1. From the tree in the application rules side panel, choose Document Specific > *(Document Type)* > *(Process Type)* > *(Process Type)* Modification > *(Process Type)* Modification Rules. The Modification Rules window displays in the work area.



2. Expand the applicable modification types and levels for which you want to set up rules.
3. Right click on the applicable rule and choose allow, disallow, or ignore as per your business practices. Refer to [Table 16–1](#) for field value descriptions.

## 16.2 Defining Custom Modification Types

You can define custom modification types for a process type. Creating a modification type allows you to classify certain attributes (including extended attributes) into one group for which rules that determine when these attributes can and cannot be modified can be defined.

Once created, the custom modification type displays under the modification rules for the business document of the process type you are defining. From there you can decide whether to allow, disallow, or ignore

the custom modification type for a given status. For more information about modification types and rules see [Section 16.1, "Defining Modification Rules"](#).

You can use the Order Modification Types branch for:

- [Creating a Custom Modification Type](#)
- [Modifying a Custom Modification Type](#)
- [Deleting a Custom Modification Type](#)

### 16.2.1 Creating a Custom Modification Type

To create a custom modification type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > (*Process Type*) > (*Process Type*) Modification > (*Process Type*) Modification Types. The Custom Modification List window displays in the work area.
2. From the Custom Modification List, choose . The Custom Modification window displays. Enter information in the applicable fields. Refer to [Table 16–2](#) for field value descriptions.
3. Choose . A pop-up warning you to sign out of the application for changes to take place displays.

The screenshot shows a configuration window for defining custom modification types. At the top, there are several input fields: a dropdown for 'Modification Level', a text box for 'Modification Type', a text box for 'Description', a dropdown for 'Min Allowed Status', and a dropdown for 'Max Allowed Status'. Below these fields are two large, empty list boxes. The left list box is titled 'Available' and the right one is titled 'Subscribed'. Both list boxes have a header row labeled 'XML Attribute Name'. Between the two list boxes are two buttons with right and left arrows, used for moving items from the 'Available' list to the 'Subscribed' list and vice versa.

**Table 16–2 Custom Modification Window**

Field	Description
Modification Level	Select the level of the modification type. For example, Header, Line, or Release.
Modification Type	Enter the name of the modification type.
Description	Enter a brief description of the modification type.
Min. Allowed Status	Select the minimum status the modification type can be performed at.
Max Allowed Status	Select the maximum status the modification type can be performed at.

**Table 16–2 Custom Modification Window**

Field	Description
Available	A list of XML attributes that can be associated with the modification type. To add an available attribute to the modification type, select the attribute you want to add and choose  .
Subscribed	A list of XML attributes that have been associated with the modification type. To remove a subscribed attribute, select the attribute you want to remove and choose  .

## 16.2.2 Modifying a Custom Modification Type

To modify a custom modification type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > (*Process Type*) > (*Process Type*) Modification > (*Process Type*) Modification Types. The Custom Modification List window displays in the work area.
2. From the Custom Modification List, locate the applicable Custom Modification and choose . The Custom Modification window displays.
3. Enter information in the applicable fields. Refer to [Table 16–2](#) for field value descriptions.
4. Choose .

## 16.2.3 Deleting a Custom Modification Type

To delete a custom modification type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > (*Process Type*) > (*Process Type*) Modification > (*Process Type*) Modification Types. The Custom Modification List window displays in the work area.
2. From the Custom Modification List, locate the applicable Custom Modification and choose .

## 16.3 Defining Modifications Impacting Pricing

You can specify whether a modification type impacts pricing on an order. When modifications of these modification types occur, `OrderRepricingUE` is called to update price and charge information at the level indicated for that modification type. For more information about `OrderRepricingUE`, see the *Selling and Fulfillment Foundation: Javadocs*.

### 16.3.1 Adding/Removing a Modification Type for Modifications Impacting Pricing

To specify whether a modification type has pricing impact:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > (*Process Type*) > (*Process Type*) Modification > Modifications Impacting Pricing. The Modifications Impacting Pricing List window displays in the work area.
2. From the Modifications Impacting Pricing List, choose . The Modification Type List window displays.
3. To add a modification type to the Modifications Impacting Pricing list, select the desired modification type(s) from the Modification Types and choose .
4. To remove a modification type from the Modifications Impacting Pricing list, select the desired modification type(s) from the Modification Types and choose .
5. Choose .

## 16.4 Defining Modifications Requiring Auditing

You can specify which modification types will require an audit after being completed.

To specify which modification types require an audit:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Order Modification > Modifications Requiring Auditing. The Modifications Requiring Auditing window displays in the work area.

2. From the Modifications Requiring Auditing window, choose .

**Note:** When opening the Modifications Requiring Auditing screen for the first time, all modification types are listed as requiring audits.

3. To add a modification type to the Modifications Requiring Auditing list, select the desired modification type(s) from the Modification Types column and choose .
4. To remove a modification type from the Modifications Requiring Auditing list, select the desired modification type(s) from the Modification Types column and choose .
5. Choose .



## Configuring a Document's Note Reasons

---

You can define reason codes for entering a note. These codes define why a note was entered by a user in the Console.

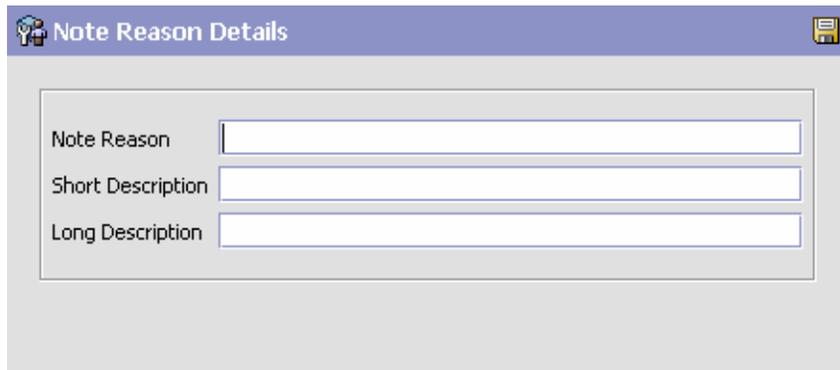
You can use the Note Reasons branch for:

- [Creating a Note Reason](#)
- [Modifying a Note Reason](#)
- [Deleting a Note Reason](#)

### 14.1 Creating a Note Reason

To create a note reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Note Reasons. The Note Reasons window displays in the work area.
2. Choose . The Note Reason Details window displays.



The screenshot shows a window titled "Note Reason Details". The window has a title bar with a gear icon on the left and a save icon on the right. The main content area contains three text input fields, each with a label to its left: "Note Reason", "Short Description", and "Long Description".

3. In Note Reason, enter the note reason as you want it to appear throughout the system.
4. In Short Description, enter a brief description of the note reason.
5. In Long Description, enter a more detailed description of the note reason.
6. Choose .

## 14.2 Modifying a Note Reason

To modify a note reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Note Reasons. The Note Reasons window displays in the work area.
2. Select the applicable appointment failure reason and choose . The Note Reason Details window displays.
3. In Short Description, enter a brief description of the note reason.
4. In Long Description, enter a more detailed description of the note reason.
5. Choose .

## 14.3 Creating a New Note Reason Based on an Existing One

To create a new note reason based on an existing one:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Note Reasons. The Note Reasons window displays in the work area.
2. Select the applicable note reason and choose . The Note Reason Details window displays.
3. Enter information in the applicable fields.
4. Choose .

## 14.4 Deleting a Note Reason

To delete a note reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Note Reasons. The Note Reasons window displays in the work area.
2. Select the applicable appointment failure reason and choose . The Confirmation window displays.
3. Choose OK.



# 15

## Configuring a Document's Line Relationship Type

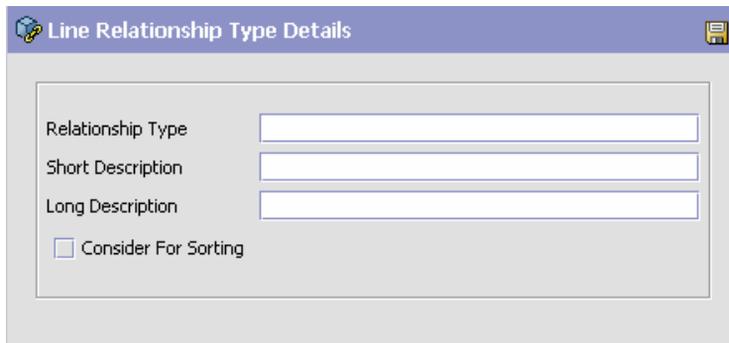
---

You can define the relationship types used when linking two related lines together. These relationships are used to group similar products together on an order.

### 15.1 Defining a Line Relationship Type

To create a line relationship type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Line Relationship Type. The Line Relationship Type window appears in the work area.
2. Choose . The Line Relationship Details window appears.



The screenshot shows a dialog box titled "Line Relationship Type Details". It contains three text input fields for "Relationship Type", "Short Description", and "Long Description". Below these fields is a checkbox labeled "Consider For Sorting".

3. In Relationship Type, enter the relationship type as you want it to appear throughout the system.
4. In Short Description, enter a brief description of the relationship type.
5. In Long description, enter a more detailed description of the relationship type.
6. To enable sorting on this relationship type, check the Consider For Sorting checkbox.
7. Choose .

## 15.2 Modifying a Line Relationship Type

To modify a line relationship type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Line Relationship Type. The Relationship Type window appears in the work area.
2. Select the applicable relationship and choose . The Relationship Type Details window appears.
3. In Short Description, enter a brief description of the relationship type.
4. In Long description, enter a more detailed description of the relationship type.
5. To enable sorting on this relationship type, check the Consider For Sorting checkbox.
6. Choose .

## 15.3 Creating a New Line Relationship Type Based on an Existing One

To create a new line relationship type based on an exiting one.

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Line Relationship Type. The Relationship Type window appears in the work area.
2. Select the applicable relationship type and choose . The Relationship Type Details window appears.
3. Enter information in the applicable fields
4. Choose .

### 15.3.1 Deleting a Line Relationship Type

To delete a line relationship type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Line Relationship Type. The Relationship Type window appears in the work area.
2. Select the applicable relationship type and choose . The Confirmation window appears.
3. Choose OK.



# Configuring an Inbound Order Document's Fulfillment Specific Components

---

To complete an order document's lifecycle, each document has a set of different processes that it can go through. These processes are called process types. Every order document has a defined set of process types in Selling and Fulfillment Foundation.

The following process types are defined in Selling and Fulfillment Foundation for the order document types:

- Fulfillment
- Negotiation
- Shipment
- Receipt

You can configure the rules and components specific to an order document's fulfillment process type.

You can use process type configuration for:

- [Defining Hold Types](#)
- [Process Type Pipeline Configuration](#)
- [Defining Transaction Rules](#)
- [Defining Status Inventory Types](#)
- [Defining Monitoring Components](#)
- [Defining Monitoring Events](#)

## 14.1 Defining Hold Types

Orders and order lines can be placed on hold manually or automatically, by applying a particular hold type. Certain transactions can be configured to not process documents that are on a specific type of hold. Likewise, modification types can be configured to not process documents that are on a specific type of hold. By default, all transactions and modification types are allowed to process all documents for all hold types.

The transactions that can be prevented from processing orders or order lines on a specific type of hold have the checkbox, *This Transaction Can Be Stopped From Processing Orders That Are On Hold*, checked in the Others tab of the transaction details screen. For more information about viewing transaction details, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

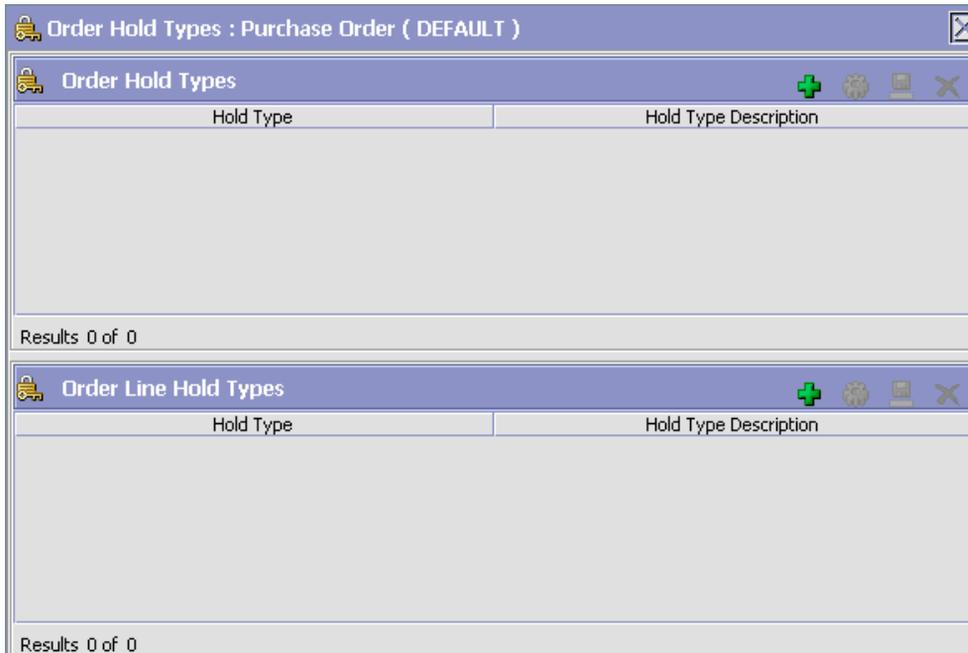
You can use the Hold Types branch in the Applications Manager for:

- [Creating a Hold Type](#)
- [Creating an Order Line Level Hold Type](#)
- [Deleting a Hold Type](#)

### 14.1.1 Creating a Hold Type

Hold types can be created at either the order or order line level. Selecting Hold Types from the application rules side panel displays the Order Hold Types screen.

Figure 14–1 Order Hold Types Screen



This screen provides a list of previously created holds and their associated level.

#### 14.1.1.1 Creating an Order Level Hold Type

To create an order level hold type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Hold Types. The Hold Types window displays in the work area.
2. Click  in the Order Hold Types panel. The Hold Type pop-up window displays.
3. In the Hold Type field, enter the type of the hold.
4. In the Description field, enter the description of the hold type.
5. Enter the information in the applicable fields. For field value descriptions, see [Table 14–1](#), [Table 14–2](#) and [Table 14–3](#).
6. Click .

**Table 14–1 Hold Type Screen, Hold Creation tab**

Field	Description
<b>Hold Created Automatically</b>	
On Draft Order Creation	Check this option to apply this hold type to all orders during draft order creation.
On Draft Order Confirmation	Check this option to apply this hold type to all orders during draft order confirmation.
On Order Creation	Check this option to apply this hold type to all orders during order creation.

Table 14–1 Hold Type Screen, Hold Creation tab

Field	Description
On Resolution Of The Hold Type	<p>Check this option to apply this hold type during the resolution of another hold type. From the drop-down list, select the hold type that, upon resolution, triggers this hold type.</p> <p><b>Note:</b> Selling and Fulfillment Foundation does not check whether or not you are defining a circular hold type definition. For example, if you define hold type B as being applied during the resolution of hold type A, and hold type A as being applied during the resolution of hold type B, you can create an infinite loop that Selling and Fulfillment Foundation does not warn you against.</p>
When The Following Modifications Are Performed	<p>Modification types that automatically apply this hold type to an order.</p> <p>Click  to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available modification types that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the modification types that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>
For All Orders	<p>Select this radio button if the above conditions should be checked for all orders.</p> <p><b>Note:</b> You can only select this option after the created hold has been saved.</p>
Only For Orders Satisfying The Following Condition	<p>Select this radio button if the above conditions should only be checked for orders satisfying a certain condition. Click  to build or modify the condition that is evaluated. For more information about using the condition builder, see the <i>Selling and Fulfillment Foundation: Application Platform Configuration Guide</i>.</p> <p>The available attributes for this condition can be extended. For more information about extending condition attributes, see the <i>Selling and Fulfillment Foundation: Customization Guide</i>.</p> <p><b>Note:</b> You can only select this option after the created hold has been saved</p>

**Table 14–1 Hold Type Screen, Hold Creation tab**

Field	Description
<b>Hold Created Manually</b>	
By All Users	Select this radio button if all user groups can apply this hold to an order.
By Users Who Belong To The Following Groups	<p>Select this radio button if only users belonging to certain user groups can apply this hold to an order.</p> <p>Click  to modify the list of user groups. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available user groups that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the user groups that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>

**Table 14–2** Hold Type Screen, Hold Resolution tab

Field	Description
<b>Hold Resolved Automatically</b>	
The Following Time-Triggered Transaction Will Process Created Holds	From the drop-down list, select the time-triggered transaction that will process the created holds.
The Following Time-Triggered Transaction Will Process Rejected Holds	From the drop-down list, select the time-triggered transaction that will process the rejected holds.

*Table 14–2 Hold Type Screen, Hold Resolution tab*

Field	Description
<b>Hold Resolved Manually</b>	
By All Users	Select this radio button if all user groups can process this hold.
By Users Who Belong To The Following Groups	<p>Select this radio button if only users belonging to certain user groups can process this hold.</p> <p>Click  to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available user groups that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the user groups that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>

 **Hold Type** 

Hold Type  Hold Type Description

**Hold Creation** **Hold Resolution** **Hold Effects**

The Following Transactions Will Be Stopped From Processing Orders On This Hold

 Process Type	Transaction Name
--	------------------

The Following Modifications Are Not Allowed For Orders On This Hold

 Modification Type	Modification Level
---	--------------------

**Table 14–3 Hold Type Screen, Hold Effects tab**

Fields	Description
The Following Transactions Will Be Stopped From Processing Orders On This Hold	Transactions that are disallowed when this hold type is applied to an order.  Click  to modify the list. In the subsequent pop-up window: <ul style="list-style-type: none"> <li>• Use the right arrow to move the available transactions that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the transactions that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>
The Following Modifications Are Not Allowed For Orders On This Hold	Modification types that are disallowed when this hold type is applied to an order.  Click  to modify the list. In the subsequent pop-up window: <ul style="list-style-type: none"> <li>• Use the right arrow to move the available modification types that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the modification types that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>

### 14.1.1.2 Creating an Order Line Level Hold Type

To create an order line level hold type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Hold Types. The Hold Types window displays in the work area.
2. Click  in the Order Line Hold Types panel. The Hold Type pop-up window displays.
3. In the Hold Type field, enter the type of the hold.
4. In the Description field, enter the description of the hold type.
5. Enter the information in the applicable fields. For field value descriptions, see [Table 14–4](#), [Table 14–5](#) and [Table 14–6](#).
6. Click .

**Table 14–4** *Hold Type Screen, Hold Creation tab*

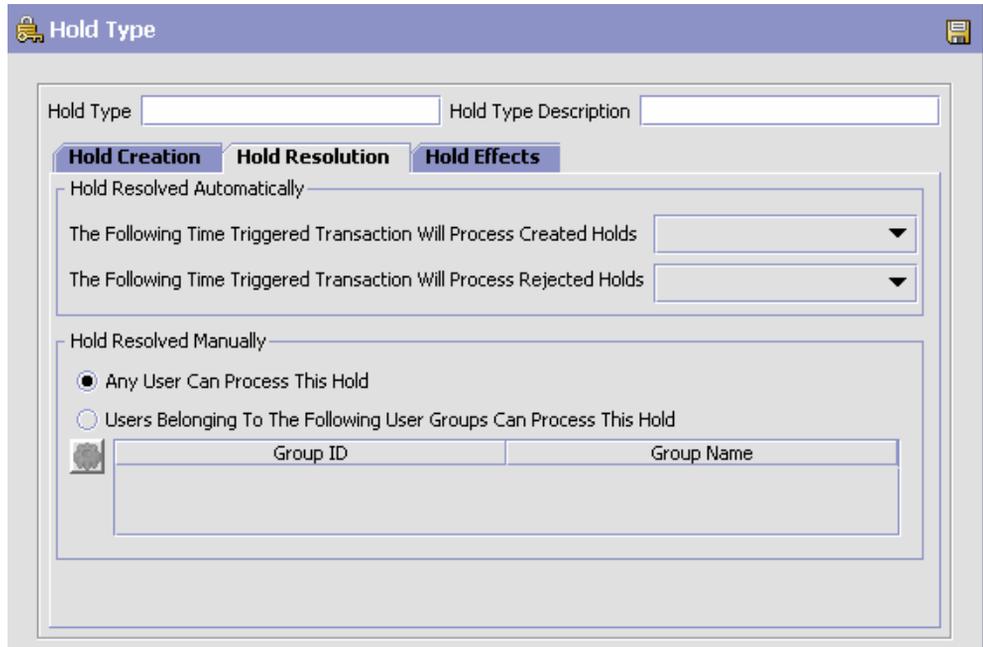
Field	Description
<b>Hold Created Automatically</b>	
On Draft Order Creation With Lines Or Adding Lines To Draft Order	Check this option to apply this hold type to all lines on an order upon entering Draft Order Created status or when a line is added to an order that is already in Draft Order Created status.
On Draft Order Confirmation	Check this option to apply this hold type to a line upon confirmation of a draft order.
On Order Creation Or Adding Lines To An Order	Check this option to apply this hold type to a line upon creation or addition to an order.

**Table 14–4 Hold Type Screen, Hold Creation tab**

Field	Description
On Resolution Of The Hold Type	<p>Check this option to apply this hold type during the resolution of another hold type. From the drop-down list, select the hold type that, upon resolution, triggers this hold type.</p> <p><b>Note:</b> Selling and Fulfillment Foundation do not check whether or not you are defining a circular hold type definition. For example, if you define hold type B as being applied during the resolution of hold type A, and hold type A as being applied during the resolution of hold type B, you could create an infinite loop that Selling and Fulfillment Foundation do not warn you against.</p>
When The Following Modifications Are Performed	<p>Modification types that automatically apply this hold type to an order.</p> <p>Click  to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available modification types that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the modification types that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>
For All Order Lines	<p>Select this radio button if the above conditions should be checked for all order lines.</p> <p><b>Note:</b> You can only select this option after the created hold has been saved.</p>
Only For Order Lines Satisfying The Following Condition	<p>Select this radio button if the above conditions should only be checked for order lines satisfying a certain condition. Click  to build or modify the condition that is evaluated. For more information about using the condition builder, see the <i>Selling and Fulfillment Foundation: Application Platform Configuration Guide</i>.</p> <p>The available attributes for this condition can be extended. For more information about extending condition attributes, see the <i>Selling and Fulfillment Foundation: Customization Guide</i>.</p> <p><b>Note:</b> You can only select this option after the created hold has been saved.</p>

**Table 14–4 Hold Type Screen, Hold Creation tab**

Field	Description
<b>Hold Created Manually</b>	
By All Users	Select this radio button if all user groups can apply this hold to an order.
By Users Who Belong To The Following Groups	<p>Select this radio button if only users belonging to certain user groups may apply this hold to an order.</p> <p>Click  to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available user groups that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the user groups that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>



**Table 14–5** *Hold Type Screen, Hold Resolution tab*

Field	Description
<b>Hold Resolved Automatically</b>	
The Following Time-Triggered Transaction Will Process Created Holds	From the drop-down list, select the time-triggered transaction that will process the created holds.
The Following Time-Triggered Transaction Will Process Rejected Holds	From the drop-down list, select the time-triggered transaction that will process the rejected holds.

**Table 14–5 Hold Type Screen, Hold Resolution tab**

Field	Description
<b>Hold Resolved Manually</b>	
By All Users	Select this radio button if all user groups can process this hold.
By Users Who Belong To The Following Groups	<p>Select this radio button if only users belonging to certain user groups can process this hold.</p> <p>Click  to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available user groups that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the user groups that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>

 **Hold Type** 

Hold Type  Hold Type Description

**Hold Creation** **Hold Resolution** **Hold Effects**

The Following Transactions Will Be Stopped From Processing Orders On This Hold



Process Type	Transaction Name	Hold Effect Level
--------------	------------------	-------------------

The Following Modifications Are Not Allowed For Orders On This Hold



Modification Type	Modification Level
-------------------	--------------------

Table 14–6 Hold Type Screen, Hold Effects tab

Fields	Description
The Following Transactions Will Be Stopped From Processing Orders On This Hold	<p>Transactions that are disallowed when this hold type is applied to an order.</p> <p>Click  to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>Use the right arrow to move the available transactions that you wish to associate with this hold type to the subscribed list.</li> <li>Use the left arrow to unsubscribe the transactions that you wish to disassociate with this hold type and move them back into the available list.</li> </ul> <p>The third column is used to select the effect level of the hold. This determines at whether the transaction will be held at the order or order line level.</p>
The Following Modifications Are Not Allowed For Orders On This Hold	<p>Modification types that are disallowed when this hold type is applied to an order.</p> <p>Click  to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>Use the right arrow to move the available modification types that you wish to associate with this hold type to the subscribed list.</li> <li>Use the left arrow to unsubscribe modification types that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>

### 14.1.2 Modifying a Hold Type

To modify a hold type:

- From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Hold Types. The Hold Types window displays in the work area.
- Select the applicable hold type and click . The Hold Type pop-up window displays. Enter information in the applicable fields. For field value descriptions, see [Table 14–1](#), [Table 14–2](#) and [Table 14–3](#) (for Order Level Holds) or [Table 14–4](#), [Table 14–5](#), and [Table 14–6](#) (for Order Line Level Holds).
- Click .

### 14.1.3 Deleting a Hold Type

To delete a hold type:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Hold Types. The Hold Types window displays in the work area.
2. Select the applicable hold type and click .

## 14.2 Defining Process Type Details

You can define the parameters and templates that distinguish a process type.

For more information about defining process type details, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

## 14.3 Process Type Pipeline Configuration

A **process type pipeline** is a series of transactions and statuses that guide document types, such as a Purchase Order, through a predefined process. A pipeline consists of the different statuses a document goes through during fulfillment, negotiation, shipment, or receipt. You can also set up transactions consisting of events, actions, and conditions, as they pertain to the pipeline you are configuring.

### Repositories

A repository is a logical collection of entities that define the business process workflow.

The following entities are included in a repository:

- Pipelines
- Transactions
- Statuses
- Conditions
- Actions
- Services

Selling and Fulfillment Foundation provides a base repository for each of the system-defined process types. Some of the entities within a repository are copied when creating a new document type. For more information about creating a new document type, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

The process of purchase order fulfillment is modeled through a pipeline. This represents the process configuration that is unique to an organization. An organization may also specify unique processes for each participating Enterprise.

### 14.3.1 Defining Pipeline Determination

**Pipeline determination** is used to set up conditions that affect which pipeline is used during the start of the business process workflow. For example, an organization deals with sales orders that sometimes contain hazardous materials. They have two separate pipelines, one in which orders with order lines without any hazardous materials go through and one in which orders with order lines containing hazardous materials must go through for inspection before continuing through the order process. The organization uses pipeline determination to set up a condition that determines whether or not order lines contain hazardous materials and sends the order line down the correct pipeline.

When you expand the Pipeline Determination branch, the components displayed depends on what role you are logged in as. If you are logged in as a Hub role, the Hub Rule displays. If you are logged in as an Enterprise role, both the Hub Rule and My Rule components display. Double-click on the applicable node to display the pipeline determination rules.

---

---

**Note:** If you are logged in as an Enterprise role, the Hub Rule screen is grayed out and cannot be modified.

---

---

Drag conditions and pipelines into the work area to construct pipeline determination rules. A single pipeline or condition must be the root. Conditions cannot link back to an earlier component in the chain and a pipeline cannot be linked to twice.

---

---

**Note:** When configuring pipeline determination for an order document type pipeline, please note that pipeline determination is only considered when adding a line or creating an order. When changes are made to draft orders pipeline determination does not occur.

---

---

### 14.3.1.1 Condition Variables for Pipeline Determination

For a list of the condition variables that can be used at the inbound order header and inbound order line level for pipeline determination, refer to [Appendix F, "Condition Builder Attributes"](#).

## 14.3.2 Pipelines

For more information about configuring pipelines, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

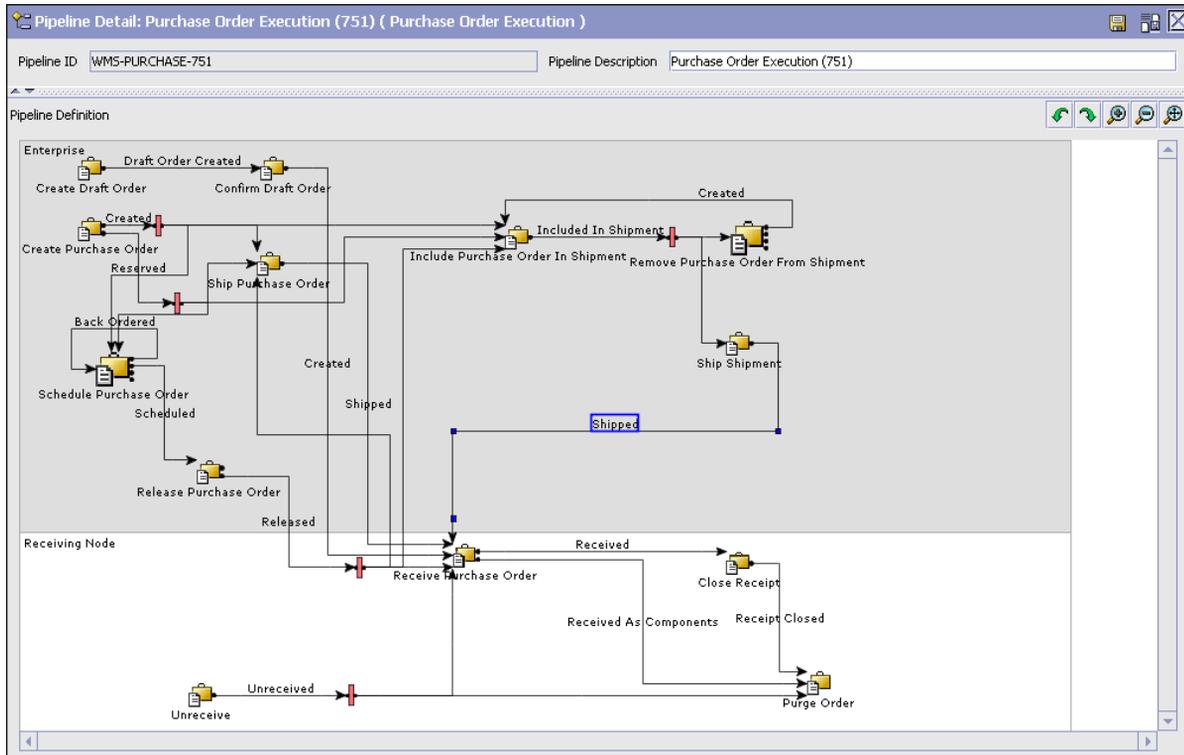
To view the purchase order pipeline details,

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Fulfillment Process Model. The Purchase Order Execution window displays.



2. In the Purchase Order Execution window, choose Purchase Order Execution Repository > Pipelines > Purchase Order Execution.
3. The Pipeline Detail: Purchase Order Execution (Purchase Order Execution) window displays.

For more information about creating and modifying a pipeline, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



### 14.3.3 Transactions

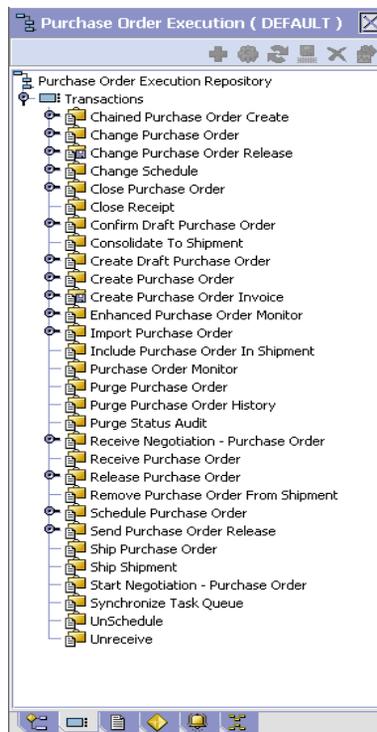
Every process type has a set of base transactions defined for it. A transaction is a logical unit of work that is necessary for performing activity within Selling and Fulfillment Foundation. Base transactions are predefined transactions that contain information about how the transaction behaves, such as how many copies of a transaction can be kept in a process type and whether or not it can have configurable base pick and drop statuses. Base transactions can be used to create new transactions. These transactions can be changed within the limits defined in the base transaction.

For more information about transactions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the transaction details for a purchase order pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Fulfillment Process Model. The Purchase Order Execution window displays.
2. In the Purchase Order Execution window, choose .
3. The Transactions tab window displays.

For more information about creating and modifying transactions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 14–7 Purchase Order Pipeline - Transactions Tab Window**

Field	Description
Chained Purchase Order Create	This transaction represents a point in the pipeline where a chained purchase order is created.
Change Purchase Order	This transaction represents any modifications that may be made to a purchase order.
Change Purchase Order Release	This transaction represents any modifications that may be made to a purchase order release.
Change Schedule	This transaction represents any modifications that may be made to the scheduling determinations for a purchase order or order line.
Close Purchase Order	This transaction represents a purchase order being closed.
Close Receipt	This transaction represents a purchase order receipt being closed.
Confirm Draft Purchase Order	This transaction represents a draft purchase order is manually confirmed and considered an actual purchase order in the system.
Consolidate To Shipment	This transaction the process of finding a shipment into which a given purchase order release can be included.
Create Draft Purchase Order	This transaction represents the creation of a draft purchase order in the system.
Create Purchase Order	This transaction represents the creation of a purchase order in the system.
Create Purchase Order Invoice	This transaction represents the creation of an invoice for an existing purchase order.
Enhanced Purchase Order Monitor	This transaction represents the an alternate purchase order monitor featuring an advanced set of parameters used to monitor purchase orders in the system.
Import Purchase Order	This transaction represents the process of importing a purchase order that has already been processed to some extent by an external system.
Include Purchase Order In Shipment	This transaction represents the process of creating a shipment. This transaction is internally invoked the confirmShipment API. See the <i>Selling and Fulfillment Foundation: Javadocs</i> for more information.

**Table 14–7 Purchase Order Pipeline - Transactions Tab Window**

Field	Description
Purchase Order Monitor	This transaction represents the basic set of parameters used to monitor orders in the system.
Purge Purchase Order	This transaction represents a purchase order that can be purged moved from the tables into history tables.
Purge Purchase Order History	This transaction represents the process of purging purchase orders from the history tables and removing them from the system.
Purge Status Audit	This transaction represents the process of removing purchase order status audit data from the system.
Receive Negotiation	This transaction represents receiving negotiation requests from the Seller on the purchase order.
Receive Purchase Order	This listener transaction represents the process of receiving a purchase order at the receiving node.
Release Purchase Order	This transaction represents the process of releasing purchase orders to specific ship nodes, making sure that the scheduled ship nodes have enough inventory to process the order.
Remove Purchase Order From Shipment	This transaction represents the process of removing a purchase order from an existing shipment.
Schedule Purchase Order	This transaction represents the process of scheduling purchase orders to specific ship nodes making sure that the scheduled ship nodes have enough inventory to process the order.
Send Purchase Order Release	This transaction represents the process of dispatching releases to ship nodes.
Ship Purchase Order	This transaction is internally invoked the confirmShipment API. See the <i>Selling and Fulfillment Foundation: Javadocs</i> for more information.
Ship Shipment	This transaction is internally invoked the confirmShipment API. See the <i>Selling and Fulfillment Foundation: Javadocs</i> for more information.
Start Negotiation - Purchase Order	This transaction represents the process of initiating the negotiation process with the Seller on an order.
Synchronize Task Queue	This transaction represents the process of synchronizing the purchase order fulfillment task queue.

**Table 14–7 Purchase Order Pipeline - Transactions Tab Window**

Field	Description
Unschedule Order	This transaction represents the process of unscheduling a purchase order that has already been scheduled to a ship node.
Unreceive	This transaction represents the process of unreceiving a purchase order that has already been received by a receiving node.

### 14.3.4 Statuses

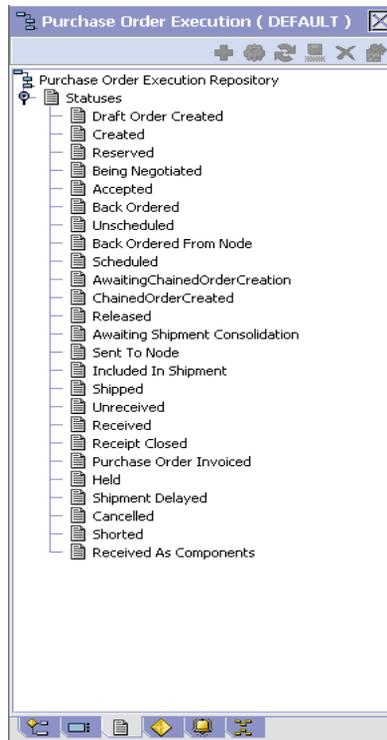
**Statuses** are the actual states that a document moves through in the pipeline. A transaction can contain two types of statuses, a drop status and a pickup status. A document is moved into a **drop status** when the events and conditions of a transaction have been completed. A **pickup status** takes the document from the previous drop status and moves it through the next transaction. Created and Scheduled are examples of statuses.

For more information about statuses, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the status details of a purchase order pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Fulfillment Process Model. The Purchase Order Execution window displays.
2. In the Purchase Order Execution window, choose .
3. The Statuses tab window displays.

For more information about creating and modifying statuses, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 14–8 Order Fulfillment Pipeline - Statuses Tab Window**

Field	Description
Draft Order Created	This indicates that a draft purchase order has been created.
Created	This indicates that a purchase order has been created.
Reserved	This indicates the purchase order has been created, but it is not ready to schedule yet.
Being Negotiated	This indicates that the purchase order is undergoing the negotiation process with the Buyer on the order.
Accepted	This indicates that a purchase order has been manually accepted.
Backordered	This indicates that a purchase order is backordered until sufficient inventory is available.

**Table 14–8 Order Fulfillment Pipeline - Statuses Tab Window**

Field	Description
Unscheduled	This indicates that the purchase order has been removed from Scheduled status and any inventory that has been reserved for the order at the scheduled node(s) is canceled.
Backordered From Node	This indicates that the purchase order has been created and released to the node, but the node does not have enough inventory to fulfill the order.
Scheduled	This indicates that the applicable node(s) have the inventory to fulfill the purchase order and can be scheduled for release.
Awaiting Chained Order Creation	This indicates that the purchase order is waiting on the creation of a chained order.
Awaiting Shipment Consolidation	This indicates the purchase order is waiting on the consolidation of its shipments.
Chained Order Created	This indicates that a chained purchase order must be created and sent to the applicable node.
Released	This indicates that there is enough inventory to schedule to the purchase order for fulfillment. The order is released to the Application Consoles, the Sterling Warehouse Management System, or another third-party warehouse management system.
Sent To Node	This indicates that the purchase order is sent in the form of a purchase order release.
Included In Shipment	This indicates that the purchase order is included in a shipment.
Shipped	This indicates that the purchase order has been shipped
Unreceived	This indicates that a purchase order that was originally received at a receiving node needs to be re-received.
Received	This indicates that the purchase order has been received by the receiving node and is in Receipt In Progress status in the Purchase Order Receipt pipeline.

**Table 14–8 Order Fulfillment Pipeline - Statuses Tab Window**

Field	Description
Receipt Closed	This indicates that the purchase order has been fully received and the receipt can be removed from the system.  <b>Note:</b> A purchase order moves into Receipt Closed status after a purchase order receipt moves into Received status in the Purchase Order Receipt pipeline.
Purchase Order Invoiced	This indicates that an invoice has been created for a purchase order.
Held	This indicates that the purchase order is being held and no modifications can be made until it is released from the hold.
Shipment Delayed	This indicates that all or part of the purchase order shipment has been delayed.
Cancelled	This indicates that the purchase order has been canceled.
Shorted	This indicates that the purchase order contains less quantity than requested.
Received As Components	This indicates that the purchase order has been received as one or more individual components.

### 14.3.5 Conditions

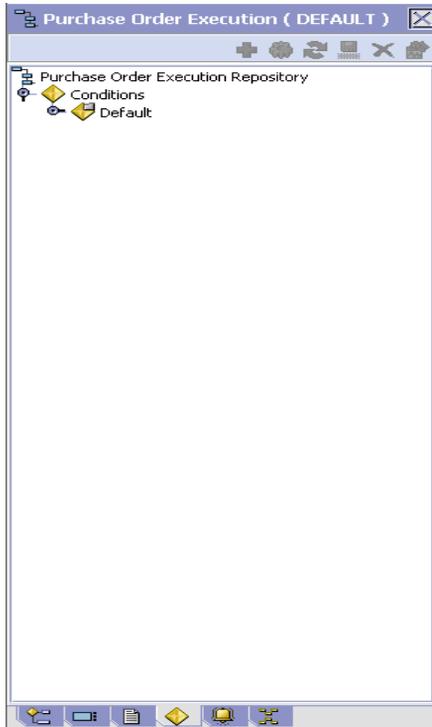
A **condition** matches document type attributes against decision points and routes the documents to different paths based on the specified attribute and value combinations. The document type attributes against which conditions can be created are predefined in Selling and Fulfillment Foundation. You can use these attributes in any combination or you can create conditions that run the appropriate application logic for specific circumstances.

For more information about conditions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the condition details of a purchase order pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Fulfillment Process Model. The Purchase Order Execution window displays.
2. In the Purchase Order Execution window, choose .
3. The Conditions tab window displays.

For more information about creating and modifying conditions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 14–9 Purchase Order Pipeline - Conditions Tab Window**

Field	Description
Conditions	Displays conditions that are specific to the purchase order pipeline, if any.

### 14.3.6 Actions

An **action** is a process or program that is triggered by an event. These processes and programs send user alert notifications and automatically resolve issues.

For example, when an order is released (the event), you can set an action to send the customer an e-mail.

For more information about Actions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the action details of a purchase order pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Fulfillment Process Model. The Purchase Order Execution window displays.
2. In the Purchase Order Execution window, choose .
3. The Actions tab window displays.

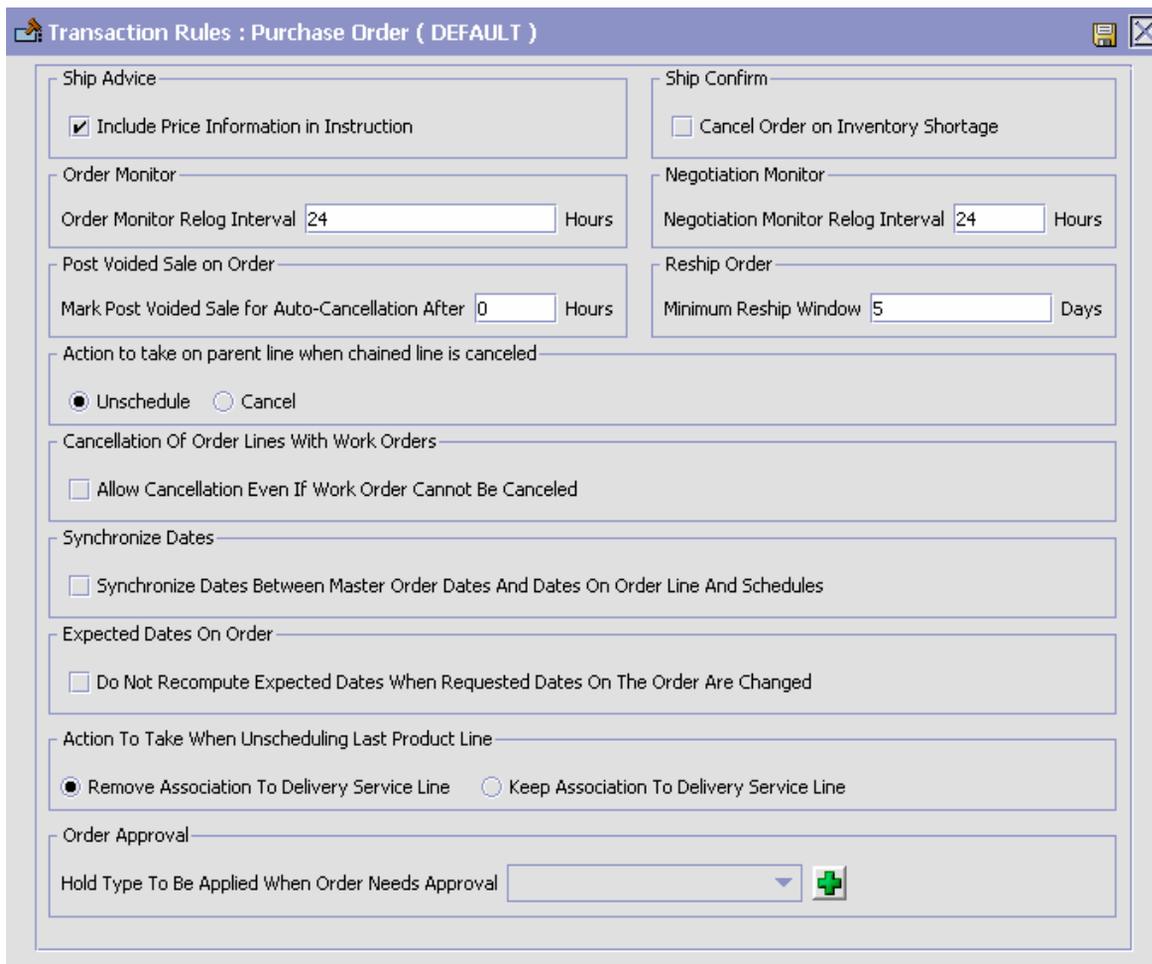
For more information about creating and modifying actions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

## 14.4 Defining Transaction Rules

You can define additional rules for shipment advice, shipment confirmation, order entry, order monitoring, and negotiation monitoring.

To define additional transaction rules:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Transaction Specific Rules. The Transaction Rules window displays.
2. Enter information in the applicable fields. Refer to [Table 14–7](#) for field value descriptions.
3. Choose .



**Transaction Rules : Purchase Order ( DEFAULT )**

**Ship Advice**

Include Price Information in Instruction

**Order Monitor**

Order Monitor Relog Interval  Hours

**Post Voided Sale on Order**

Mark Post Voided Sale for Auto-Cancellation After  Hours

**Ship Confirm**

Cancel Order on Inventory Shortage

**Negotiation Monitor**

Negotiation Monitor Relog Interval  Hours

**Reship Order**

Minimum Reship Window  Days

Action to take on parent line when chained line is canceled

Unschedule  Cancel

Cancellation Of Order Lines With Work Orders

Allow Cancellation Even If Work Order Cannot Be Canceled

Synchronize Dates

Synchronize Dates Between Master Order Dates And Dates On Order Line And Schedules

Expected Dates On Order

Do Not Recompute Expected Dates When Requested Dates On The Order Are Changed

Action To Take When Uncheduling Last Product Line

Remove Association To Delivery Service Line  Keep Association To Delivery Service Line

Order Approval

Hold Type To Be Applied When Order Needs Approval  

Table 14–10 Transaction Rules Window

Field	Description
<b>Ship Advice</b>	
Include Price Information in Instruction	<p>When selected, the system sends down price information on the order as a part of the ship advice instructions.</p> <p>This is a DCS-specific parameter. Key price-related elements from Selling and Fulfillment Foundation are sent to DCS as instructions of type SHC (shipping and handling charges at order header level), PRM (discount amount at the order header level), and STX (tax at the order header level).</p>
<b>Ship Confirm</b>	
Cancel Order on Inventory Shortage	When selected, items are canceled or backordered in case of inventory shortage.
<b>Order Monitor</b>	
Order Monitor Relog Interval	<p>Enter the number of hours after which the order monitor raises an action if a document type remains in the same status in a pipeline.</p> <p>The Inventory Monitor and Order Monitor run at pre-defined (scheduled) intervals. Once an alert is raised, the same alert should not be raised over and over again at every run. Relog intervals control how soon after the previous alert the next alert should be triggered.</p> <p><b>Important:</b> This field has no impact on the Enhanced Order Monitor.</p>
<b>Negotiation Monitor</b>	
Negotiation Monitor Relog Interval	Enter the number of hours after which the Negotiation Monitor raises an action if a document type remains in the same status in a negotiation pipeline.
<b>Post Voided Sale on Order</b>	
Mark Post Voided Sale for Auto-Cancellation After	Enter the number of hours based on which the auto-cancel date is set on the order.

**Table 14–10 Transaction Rules Window**

Field	Description
<b>Reship Order</b>	
Minimum Reship Window	Enter the minimum number of days allowed to pass after an order line has been shipped before an order line may need to be reshipped.
<b>Action to take on parent line when chained line is canceled</b>	
Unschedule	When selected the unschedule action is performed on the parent order line when a chained order line is canceled. An unscheduled parent line is synonymous with a backordered line. For more information about chained orders, see the <i>Selling and Fulfillment Foundation: Application Platform Configuration Guide</i> .
Cancel	When selected the parent line is canceled when a chained order line is canceled. For more information about chained orders, see the <i>Selling and Fulfillment Foundation: Application Platform Configuration Guide</i> .
<b>Cancellation of Order Lines with Work Orders</b>	
Allow Cancellation Even If Work Order Cannot Be Cancelled	<p>An order may have generated a work order to customize the item for the customer. In some scenarios, the work order cannot be cancelled. For example, the work order cannot be cancelled because the work order has already been completed, or because the work order is performed by an organization that does not accept work order cancellations.</p> <p>By default, the order associated with the work order cannot be cancelled. Select 'Allow Cancellation Even If Work Order Cannot Be Cancelled' to permit the parent orders to be cancelled if the work order cannot be cancelled.</p>

Table 14–10 Transaction Rules Window

Field	Description
<b>Synchronize Dates</b>	
Synchronize Dates Between Master Order Dates And Dates On Order Line And Schedules	<p>Check this box to synchronize the requested dates and the expected ship dates for the Order, Order Header, Order Line, and Order Line Schedules.</p> <p>The requested dates synchronize with the requested ship, requested deliver, and cancel dates on the order line or header.</p> <p>The expected dates synchronize with the order schedules.</p> <p><b>Note:</b> If this rule is not chosen, the synchronization between dates is not possible.</p>
<b>Expected Dates On Order</b>	
Do Not Recompute Expected Dates When Requested Dates On The Order Are Changed	<p>Check this box when you do not want the requested dates on the order line to be synchronized with the expected dates on the order line schedule.</p> <p><b>Note:</b> The dates are synchronized only during the line creation.</p> <p><b>Note:</b> Scheduling should not be used on orders which have expected and requested dates that are not synchronized.</p>
<b>Action To Take When Unsheduling Last Product Line</b>	
Remove Association To Delivery Service Line	When selected, if the last product line on a work order is cancelled, the association with the delivery line is removed.
Keep Association To Delivery Service Line	When selected, if the last product line on a work order is cancelled, the association with the delivery line is kept.
<b>Order Approval</b>	
Hold Type To Be Applied When Order Needs Approval	<p>Select the hold type you want to be applied when an order requires approval.</p> <p><b>Note:</b> The hold is triggered internally by the system, and therefore, should not be set to automatically apply in the hold configuration.</p>

## 14.5 Defining Status Inventory Types

You can define how and when inventory is updated for Sellers and Buyers tracking inventory, on a status-by-status basis. The Status Inventory Types table is used to associate statuses with specific supply and demand types according to organization. When an order moves through the statuses of a given fulfillment pipeline the values corresponding to the Buyer supply type and Seller demand type associated with the original status are decreased and the values for the status the order is moving into are increased.

### Example

Assume you have the following records in the Status Inventory Type table:

*Table 14–11 Sample Status Inventory Type Records*

Status	Buyer Supply Type	Seller Demand Type	Seller Supply Type	Increment Seller Supply
1100	Purchase Order Placed	Open Order	Onhand	N
3200	Purchase Order Released	Released	Onhand	N
3700	Intransit			Y

When an order with a line item quantity of 10 is created in Created (1100) status, the Purchase Order Placed supply record is updated with a quantity of 10. A Open Order demand type with a quantity of 10 is created for the Seller.

In this example, if a quantity of 3 is moved into Released (3200) status, the Purchase Order Placed supply record is decreased by 3 and a new supply record with a quantity of 3 is created for the Purchase Order Released supply type. The Open Order demand record is also decreased by 3 and a new demand record is with a quantity of 3 is created for the Released demand type.

When the order moves from Released (3200) status to Shipped (3700) status, the Buyer's supply is decreased for the Purchase Order Released supply type and increased for Intransit. The Seller's demand is decreased

for the Released demand type. However, the demand type is not increased for a new type, because the Seller Demand Type associated with the Shipped (3700) status is blank.

In the above configuration, the Increment Seller Supply flag is set to 'Y' and the Seller's supply type for the Shipped (3700) status is Onhand. The Increment Seller Supply flag indicates that the Seller's supply must be adjusted when moving any quantity into the Shipped (3700) status.

The value in the Seller Supply Type column indicates the supply type that should be updated, in this example, Onhand. Since the record for the Released (3200) status has the Onhand Seller supply type associated with it and the Shipped (3700) status record has a blank Seller supply type associated with it, the Onhand Seller supply type decreases when moving from Released (3200) status to Shipped (3700) status. The Seller supply type is not increased with this status move because the value in the Seller Supply Type column for the Shipped (3700) status is blank.

To view a process type's status inventory types, from the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Status Inventory Types. The Status Inventory window displays. Refer to [Table 14–12](#) for assistance.

Status Inventory Types : Sales Order ( DEFAULT )

Orders Without Chaining   Orders With Chained Children   Procurement Orders

Orders Without Chaining

Status	Buyer Supply Type	Seller Supply Type	Update Seller Supply	Seller Demand Type
Draft Order Reserved	Purchase Order Placed	Onhand	N	Reserved Order
Created	Purchase Order Placed	Onhand	N	Open Order
Reserved	Purchase Order Placed	Onhand	N	Reserved Order
Being Negotiated	Purchase Order Placed	Onhand	N	Open Order
Accepted	Purchase Order Placed	Onhand	N	Open Order
Backordered	Purchase Order Backorder	Onhand	N	Backorder
Unscheduled	Purchase Order Backorder	Onhand	N	Backorder
Scheduled	Purchase Order Scheduled	Onhand	N	Scheduled
Awaiting Chained Order Cre...	Purchase Order Released	Onhand	N	Allocated
Work Order Created	Purchase Order Scheduled	Onhand	N	Scheduled
Released	Purchase Order Released	Onhand	N	Allocated
Sent To Node	Purchase Order Released	Onhand	N	Allocated
Included In Shipment	Purchase Order Released	Onhand	N	Allocated
Shipped	In Transit		Y	
Cancelled		Onhand	N	

**Table 14–12 Status Inventory Types Window**

Field	Description
Orders Without Chaining/Orders With Chained Children/Procurement Orders	Select the Orders Without Chaining tab to view the status inventory types for orders that flow through the process type pipeline without having any associated chained orders.  Select the Orders With Chained Children tab to view the status inventory types of orders having associated drop-ship chained orders.  Select the Procurement Orders tab to view the status inventory types of procurement orders.
Status	The order document's status.
Buyer Supply Type	The Buyer's supply type associated with the order document's status.
Seller Supply Type	The Seller's supply type associated with the order document's status.
Update Seller Supply	Indicates if inventory updates are made when an order document moves into the associated status.
Seller Demand Type	The Seller's demand type associated with the order document's status.

You can use the Status Inventory Types branch for:

- [Creating a Status Inventory Type](#)
- [Modifying a Status Inventory Type](#)
- [Deleting a Status Inventory Type](#)

### 14.5.1 Creating a Status Inventory Type

To create a status inventory type:

1. In the Status Inventory Types window, choose . The Status Inventory Type Details window displays.
2. Enter information in the applicable fields. Refer to [Table 14–13](#) for field level descriptions.
3. Choose .

The screenshot shows a window titled 'Status Inventory Type Details Window'. It contains the following fields:

- Status:** A dropdown menu.
- Buyer Supply Type:** A dropdown menu.
- Seller Supply Type:** A dropdown menu.
- Update Seller Supply:** A checkbox.
- Seller Demand Type:** A dropdown menu.

**Table 14–13** *Status Inventory Type Details Window*

Field	Description
Status	Select the order document status that you want to associate inventory types with.
Buyer Supply Type	Select the Buyer supply type that you want to associate with the order document status.
Seller Supply Type	Select the Seller supply type that you want to associate with the order document status.
Update Seller Supply	Select this field if you want inventory updates to be performed on the associated inventory types when the order document enters this status. <b>Note:</b> If you are integrating with Sterling WMS, this field must be selected and you must specify the Seller Supply Type.
Seller Demand Type	Select the Seller demand type that you want to associate with the order document status.

## 14.5.2 Modifying a Status Inventory Type

To modify a status inventory type:

1. In the Status Inventory Types window, locate the applicable status inventory type and choose . The Status Inventory Type Details window displays.
2. Enter information in the applicable fields. Refer to [Table 14–13](#) for field level descriptions.
3. Choose .

## 14.5.3 Deleting a Status Inventory Type

To delete a status inventory type, locate the applicable status inventory type in the Status Inventory Types window and choose .

---

---

**Note:** Default status inventory types which are originally shipped with Selling and Fulfillment Foundation cannot be deleted.

---

---

## 14.6 Defining Monitoring Components

You can define the components used to measure and report unexpected conditions and delays in the order document's lifecycle. For more information about using these components to configure monitoring rules, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To define a process type's monitoring components, from the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Order Monitoring. The Monitoring window displays.

You can use the Monitoring window for:

- [Defining Date Types](#)
- [Defining Milestones](#)

## 14.6.1 Defining Date Types

You can define custom date types. These dates automatically appear in the configuration screen and the Order/Shipment Dates window in the Application Console.

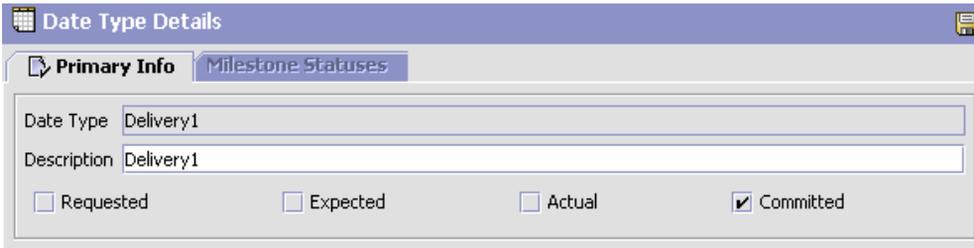
You can use the Date Types tab for:

- [Creating a Date Type](#)
- [Modifying a Date Type](#)
- [Deleting a Date Type](#)

### 14.6.1.1 Creating a Date Type

To create a date type:

1. In the Monitoring window, choose the Date Types tab.
2. From the Date Types list, choose . The Date Type Details window displays.
3. Enter information in the applicable fields. Refer to [Table 14–14](#) for field level descriptions.
4. Choose .



**Table 14–14** *Date Type Details Window*

Field	Description
Date Type	Enter the name of the date type.
Description	Enter a brief description of the date type.
Requested	Select this field to indicate if the date type represents a date requested by a Buyer, user, etc.

**Table 14–14 Date Type Details Window**

Field	Description
Expected	Select this field to indicate if the date type represents a date the system expects or has calculated something to occur.
Actual	Select this field to indicate if the date type represents the actual date.

### 14.6.1.2 Modifying a Date Type

To modify a date type:

1. In the Monitoring window, choose the Date Types tab.
2. From the Date Types list, locate the applicable date type and choose . The Date Type Details window displays.
3. Enter information in the applicable fields. Refer to [Table 14–14](#) for field level descriptions.
4. Choose .

### 14.6.1.3 Deleting a Date Type

To delete a date type:

**Note:** The following system dates cannot be deleted:

- Delivery Date
- Order Date
- Ship Date

1. In the Monitoring window, choose the Date Types tab.
2. From the Date Types list, locate the applicable date type and choose .

## 14.6.2 Defining Milestones

You can configure applicable statuses in a process type to be milestones. A milestone is a type of date that Selling and Fulfillment Foundation automatically determines when an order moves from one status to another. A milestone represents a significant point in the processing lifecycle that can be used as a criterion for monitoring. Milestones can be defined at the order, order line, order release, and order release line levels.

---

---

**Note:** A milestone can be reached whenever there is a change in an order line. Selling and Fulfillment Foundation marks a milestone as reached if an order line reaches a status marked as a milestone. However, there may be times that only part of an order line reaches a particular status defined as milestone.

---

---

You can use the Milestones tab for:

- [Creating a Milestone](#)
- [Modifying a Milestone](#)
- [Deleting a Milestone](#)

### 14.6.2.1 Creating a Milestone

To create a milestone:

1. In the Monitoring window, choose the Milestones tab.
2. From the Milestones list, choose . The Milestone Details window displays.
3. Enter information in the applicable fields. Refer to [Table 14–15](#) for field level descriptions.
4. Choose .

**Table 14–15 Milestone Details**

Field	Description
<b>Primary Info</b>	
Date Type	Enter the name of the milestone being created. <b>Note:</b> You cannot use date types you have created on the date type tab. You must create a unique name for the milestone.
Description	Enter a brief description of the milestone.
Requested	Select this field to indicate if the milestone represents a date requested by a Buyer, user, etc.
Expected	Select this field to indicate if the milestone represents a date the system expects or has calculated something to occur.
Actual	This field is not applicable for milestones.
Milestone Statuses	You can add statuses to associate with the milestone by selecting  and entering information in the applicable fields. <b>Note:</b> This tab can only be accessed once the Primary Info tab has been filled out and saved.
Date Type	The date type if any associated with the milestone.
Status	Select the status you want use to indicate the milestone has been reached.

**Table 14–15 Milestone Details**

Field	Description
Level	<p>Select Order to indicate this status must be reached at the order header level.</p> <p>Select Order Line to indicate that this status must be reached at the order line level.</p> <p>Select Order Release to indicate that this status must be reached at the order release level.</p>
Quantity Type	<p>Select Initial to indicate that the milestone is met when any quantity at the above selected level moves into the status.</p> <p>Select Complete to indicate that the milestone is met when all quantity at the above selected level moves into the status.</p>

### 14.6.2.2 Modifying a Milestone

---



---

**Important:** If modifications are made to an existing milestone, the changes are only applied to new orders. Existing orders for which milestone records have already been created are not considered.

---



---

To modify a milestone:

1. In the Monitoring window, choose the Milestones tab.
2. From the Milestones list, locate the applicable milestone and choose . The Milestone Details window displays.
3. Enter information in the applicable fields. Refer to [Table 14–15](#) for field level descriptions.
4. Choose .

### 14.6.2.3 Deleting a Milestone

To delete a milestone:

1. From the Monitoring window, choose the Milestones tab.
2. From the Milestones list, locate the applicable milestone and choose .

## 14.7 Defining Monitoring Events

Events are used in instances where the Enhanced Order Monitor may raise multiple alerts of the same type. For example, if an order with multiple lines that are shipped together has a shipment delay and you have configured the Enhanced Order Monitor to raise alerts when shipments are delayed at the line level, an alert of the same type would be raised against each line in the order. You can create rules to aggregate all of these similar alerts and raise one "root cause".

You can use the Monitor Events tab for:

- [Creating an Event Rule](#)
- [Modifying an Event](#)
- [Deleting an Event](#)

### 14.7.1 Creating an Event Rule

To create an event rule:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Monitor Events. The Monitor Events window displays.
2. From the Monitor Events list, choose . The Monitor Events Details window displays.
3. Enter information in the applicable fields. Refer to [Table 14–16](#) for field level descriptions.
4. Choose .

Event Id

Description

Requires Realert      Realert Interval

Automatically Resolve Alerts

---

Event Identified By

Order       Order Line       Order Release

Ship Node       Seller Organization       Buyer Organization

Service To Be Invoked

---

Aggregate And Invoke Service For

Order       Order Line       Order Release

Ship Node       Seller Organization       Buyer Organization

**Table 14–16 Monitor Event Details Pop-Up Window**

Field	Description
Event Id	Enter the event ID.
Description	Enter a brief description of the event.
Requires Realert	Select this field if you want users to be re-alerted if the issue has not been resolved within a certain timeframe.

**Table 14–16 Monitor Event Details Pop-Up Window**

Field	Description
Realert Interval	If you selected Requires Realert, enter the interval (in hours) that re-alerts should be sent.
Automatically Resolve Alerts	Select this field to automatically remove an alert if the alert condition is no longer valid. Once the alert is resolved, a new alert is raised if the condition is detected again.
Event Identified By	
Order	Select this field if you want two or more alert conditions to be treated the same if they belong to the same order. <b>Note:</b> This field can be selected in conjunction with Order Line and/or Order Release fields.
Order Line	Select this field if you want two or more alert conditions to be treated the same if they belong to the same order line. <b>Note:</b> This field can be selected in conjunction with Order and/or Order Release fields.
Order Release	Select this field if you want two or more alert conditions to be treated the same if they belong to the same order release. <b>Note:</b> This field can be selected in conjunction with Order and/or Order Line fields.
Ship Node	Select this field if you want two or more alert conditions to be treated the same if they belong to the same ship node. <b>Note:</b> This field must be used in conjunction with the Order, Order Line, and/or Order Release fields.
Seller Organization	Select this field if you want two or more alert conditions to be treated the same if they belong to the same Seller. <b>Note:</b> This field must be used in conjunction with the Order, Order Line, and/or Order Release fields.
Buyer Organization	Select this field if you want two or more alert conditions to be treated the same if they belong to the same Buyer. <b>Note:</b> This field must be used in conjunction with the Order, Order Line, and/or Order Release fields.

**Table 14–16 Monitor Event Details Pop-Up Window**

Field	Description
Service To Be Invoked	Select the alert service to be invoked should the event consolidation rule conditions be met.
Aggregate And Invoke Service For	
Order	Select this field if you want only one alert to be raised for an order when alert conditions are detected.
Order Line	Select this field if you want only one alert to be raised per order line when alert conditions are detected.
Order Release	Select this field if you want only one alert to be raised for an order release when alert conditions are detected.
Ship Node	Select this field if you want only one alert to be raised for a particular ship node when alert conditions are detected.
Seller Organization	Select this field if you want only one alert to be raised for a particular Seller when alert conditions are detected.
Buyer Organization	Select this field if you want only one alert to be raised for a particular Buyer when alert conditions are detected.

---



---

**Note:** In most cases the attributes that identify an event should be a subset of the attributes that specify event aggregation.

---



---

## 14.7.2 Modifying an Event

To modify an event rule:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Monitor Events. The Monitor Events window displays.
2. From the Monitor Events list, select the applicable event rule and choose . The Monitor Event Details window displays.
3. Enter information in the applicable fields. Refer to [Table 14–16](#) for field level descriptions.
4. Choose .

## 14.7.3 Deleting an Event

To delete an event rule:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Monitor Events. The Monitor Events window displays.
2. From the Monitor Events list, select the applicable event rule and choose .

# Configuring an Order Document's Shipment Specific Components

---

To complete an order document's lifecycle, each document has a set of different processes that it can go through. These processes are called process types. Every order document has a defined set of process types in Selling and Fulfillment Foundation.

The following process types are defined in Selling and Fulfillment Foundation for the order document types:

- Fulfillment
- Negotiation
- Shipment
- Receipt
- Receipt

You can configure the rules and components specific to an order document's shipment process type.

You can use process type configuration for:

- [Defining Hold Types](#)
- [Process Type Pipeline Configuration](#)
- [Defining Monitoring Components](#)
- [Defining Monitoring Events](#)
- [Defining Shipment Preferences](#)

## 15.1 Defining Hold Types

Shipments can be placed on hold manually or automatically by applying a particular hold type. Certain transactions can be configured to ensure that shipments put on hold are not processed. Likewise, modification types can be configured to ensure shipments that are on hold are not processed. By default, all transactions and modification types are allowed to process all documents for all hold types.

To prevent transactions from processing shipments that are put on hold, in the Others tab in the Transaction Detail screen, check the "This Transaction Can Be Stopped From Processing Shipments That Are On Hold" box. For more information about viewing transaction details, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To create, modify, and delete hold types, from the tree in the application rules side panel, choose Document Specific > (Document Type) > Inbound Logistics > Hold Types. For more information about defining hold types, see the *Sterling Logistics Management: Configuration Guide*.

## 15.2 Defining Process Type Details

You can define the parameters and templates that distinguish a process type.

For more information about defining process type details, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

## 15.3 Process Type Pipeline Configuration

A **process type pipeline** is a series of transactions and statuses that guide document types, such as a Sales Order, through a predefined process. A pipeline consists of the different statuses a document goes through during fulfillment, negotiation, shipment, or receipt. You can also set up transactions consisting of events, actions, and conditions, as they pertain to the pipeline you are configuring.

### Repositories

A repository is a logical collection of entities that define the business process workflow.

The following entities are included in a repository:

- Pipelines
- Transactions
- Statuses
- Conditions
- Actions
- Services

Selling and Fulfillment Foundation provide a base repository for each of the system defined process types. Some of the entities within a repository are copied when creating a new document type. For more information about creating a new document type, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

The process of purchase order shipment is modeled through a pipeline. This represents the process configuration that is unique to an organization. An organization may also specify unique processes for each participating Enterprise.

### 15.3.1 Defining Pipeline Determination

**Pipeline determination** is used to set up conditions that affect which pipeline is used during the start of the business process workflow. For example, an organization deals with sales orders that sometimes contain hazardous materials. They have two separate pipelines, one in which orders with order lines without any hazardous materials go through and one in which orders with order lines containing hazardous materials must go through for inspection before continuing through the order process. The organization uses pipeline determination to set up a condition that determines whether or not order lines contain hazardous materials and sends the order line down the correct pipeline.

When you expand the Pipeline Determination branch, the components displayed depends on what role you are logged in as. If you are logged in as a Hub role, the Hub Rule displays. If you are logged in as an Enterprise role, both the Hub Rule and My Rule components display. Double-click on the applicable node to display the pipeline determination rules.

---

---

**Note:** If you are logged in as an Enterprise role, the Hub Rule screen is grayed out and cannot be modified.

---

---

Drag conditions and pipelines into the work area to construct pipeline determination rules. A single pipeline or condition must be the root. Conditions cannot link back to an earlier component in the chain and a pipeline cannot be linked to twice.

---

---

**Note:** When configuring pipeline determination for an order document type pipeline, please note that pipeline determination is only considered when adding a line or creating an order. When changes are made to draft orders pipeline determination does not occur.

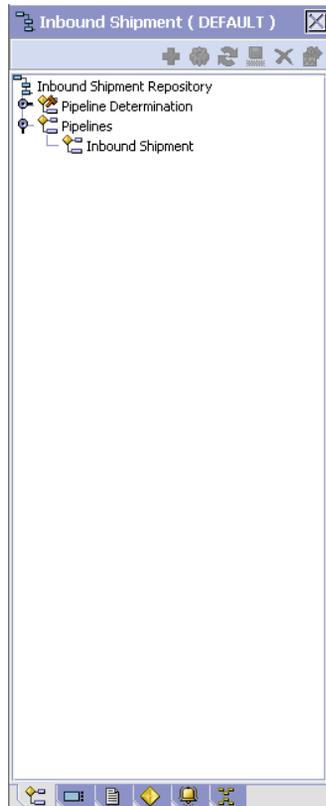
---

---

### 15.3.2 Pipelines

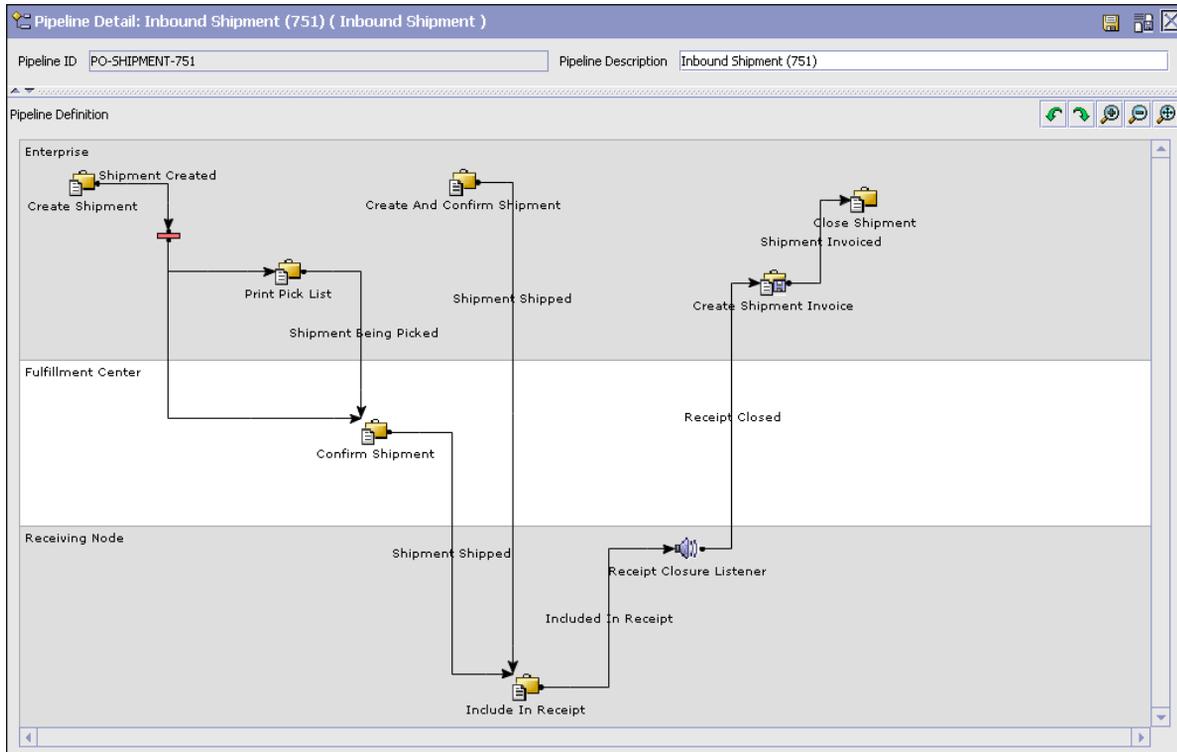
^ To view the inbound shipment pipeline details,

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Inbound Logistics > Shipment Process Model. The Inbound Shipment window displays.



2. In the Inbound Shipment window, choose Inbound Shipment Repository > Pipelines > Inbound Shipment.
3. The Pipeline Detail: Inbound Shipment (Inbound Shipment) window displays.

For more information about creating and modifying a pipeline, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



### 15.3.3 Transactions

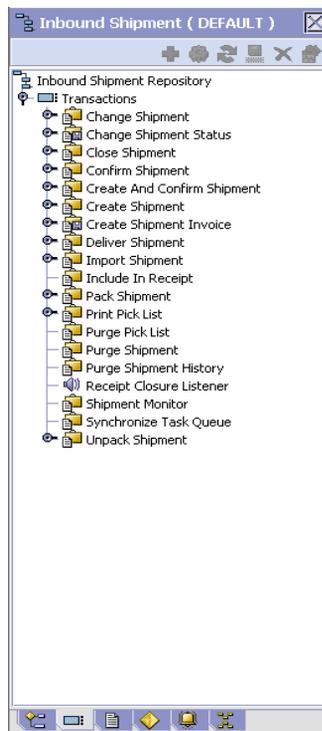
Every process type has a set of base transactions defined for it. A transaction is a logical unit of work that is necessary for performing activity within Selling and Fulfillment Foundation. Base transactions are predefined transactions that contain information about how the transaction behaves, such as how many copies of a transaction can be kept in a process type and whether or not it can have configurable base pick and drop statuses. Base transactions can be used to create new transactions. These transactions can be changed within the limits defined in the base transaction.

For more information about transactions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the transaction details for an inbound shipment pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Inbound Logistics > Shipment Process Model. The Inbound Shipment window displays.
2. In the Inbound Shipment window, choose .
3. The Transactions tab window displays.

For more information about creating and modifying transactions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 15–1 Inbound Shipment Pipeline - Transactions Tab Window**

<b>Field</b>	<b>Description</b>
Change Shipment	This transaction represents any modifications that may be made to a shipment.
Change Shipment Status	This transaction represents any modifications that may be made involving an order or order line's status.
Close Shipment	This transaction represents a shipment being closed and archived in the system.
Confirm Shipment	This transaction represents a shipment is manually confirmed and shipped.
Create And Confirm Shipment	This transaction represents the process of creating a shipment and shipping it.
Create Shipment	This transaction represents the creation of a shipment in the system.
Create Shipment Invoice	This transaction represents the creation of a shipment invoice.
Deliver Shipment	This transaction represents a shipment being delivered.
Import Shipment	This transaction represents the process of importing a shipment that has already been processed to some extent by an external system.
Include In Receipt	This transaction represents the process of including a shipment in an existing receipt.
Pack Shipment	This transaction represents the process of packing a shipment.
Print Pick List	This transaction represents the process of printing a pick list.
Purge Pick List	This transaction represents a pick list that can be purged from the system.
Purge Shipment	This transaction represents the process of moving shipments to the history tables.
Purge Shipment History	This transaction represents the process of purging shipments from the history tables and removing them from the system.

**Table 15–1 Inbound Shipment Pipeline - Transactions Tab Window**

Field	Description
Receipt Closure Listener	This transaction listens to the inbound receipt process and identifies whether the receipt is closed for the shipment.
Shipment Monitor	This transaction represents the process of monitoring shipments in the system based on defined parameters.
Synchronize Task Queue	This transaction represents the process of synchronizing the order fulfillment task queue.
Unpack Shipment	This transaction indicates that a shipment that has moved through the Pack Shipment transaction is unpacked.

### 15.3.4 Statuses

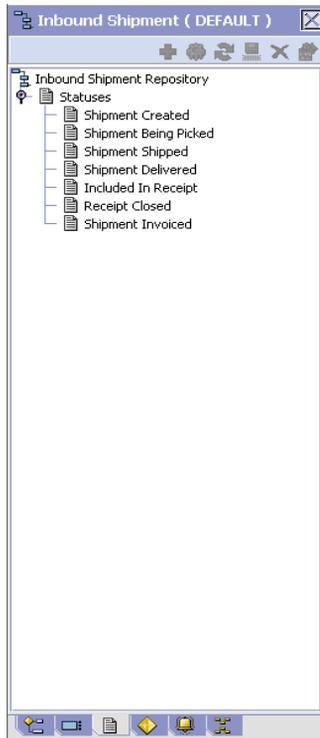
**Statuses** are the actual states that a document moves through in the pipeline. A transaction can contain two types of statuses, a drop status and a pickup status. A document is moved into a **drop status** when the events and conditions of a transaction have been completed. A **pickup status** takes the document from the previous drop status and moves it through the next transaction. Created and Scheduled are examples of statuses.

For more information about statuses, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the status details of an inbound shipment pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Inbound Logistics > Shipment Process Model. The Inbound Shipment window displays.
2. In the Inbound Shipment window, choose .
3. The Statuses tab window displays.

For more information about creating and modifying statuses, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 15–2 Inbound Shipment Pipeline - Statuses Tab Window**

Field	Description
Shipment Created	This indicates that a shipment has been created.
Shipment Being Picked	This indicates that the line items are physically being picked in preparation for shipment.
Shipment Shipped	This indicates that the shipment has been shipped to the ship to address.
Shipment Delivered	This indicates that the shipment has been delivered to the ship node address.
Included In Receipt	This indicates that the shipment has been included in the receipt.

**Table 15–2 Inbound Shipment Pipeline - Statuses Tab Window**

Field	Description
Receipt Closed	This indicates the receipt for the shipment has been closed.
Shipment Invoiced	This indicates that an invoice has been created for the shipment.

### 15.3.5 Conditions

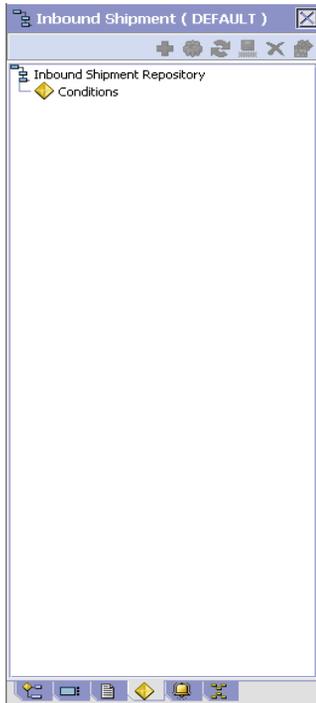
A **condition** matches document type attributes against decision points and routes the documents to different paths based on the specified attribute and value combinations. The document type attributes against which conditions can be created are predefined in Selling and Fulfillment Foundation. You can use these attributes in any combination or you can create conditions that run the appropriate application logic for specific circumstances.

For more information about conditions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the condition details of an inbound shipment pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Inbound Logistics > Shipment Process Model. The Inbound Shipment window displays.
2. In the Inbound Shipment window, choose .
3. The Conditions tab window displays.

For more information about creating and modifying conditions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 15–3 Inbound Shipment Pipeline - Conditions Tab Window**

Field	Description
Conditions	Displays conditions that are specific to the inbound shipment pipeline, if any.

### 15.3.6 Actions

An **action** is a process or program that is triggered by an event. These processes and programs send user alert notifications and automatically resolve issues.

For example, when an order is released (the event), you can set an action to send the customer an e-mail.

For more information about actions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the action details of an inbound shipment pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Inbound Logistics > Shipment Process Model. The Inbound Shipment window displays.
2. In the Inbound Shipment window, choose .
3. The Actions tab window displays.

For more information about creating and modifying actions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

## 15.4 Defining Monitoring Components

You can define the components used to measure and report unexpected conditions and delays in the order document's lifecycle. For more information about using these components to configure monitoring rules, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To define monitoring components, from the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Outbound Logistics > Shipment Monitoring. The Monitoring window displays.

You can use the Monitoring window for:

- [Defining Date Types](#)
- [Defining Milestones](#)

### 15.4.1 Defining Date Types

You can define custom date types. These dates automatically appear in the configuration screen and the Order/Shipment Dates window in the Application Console.

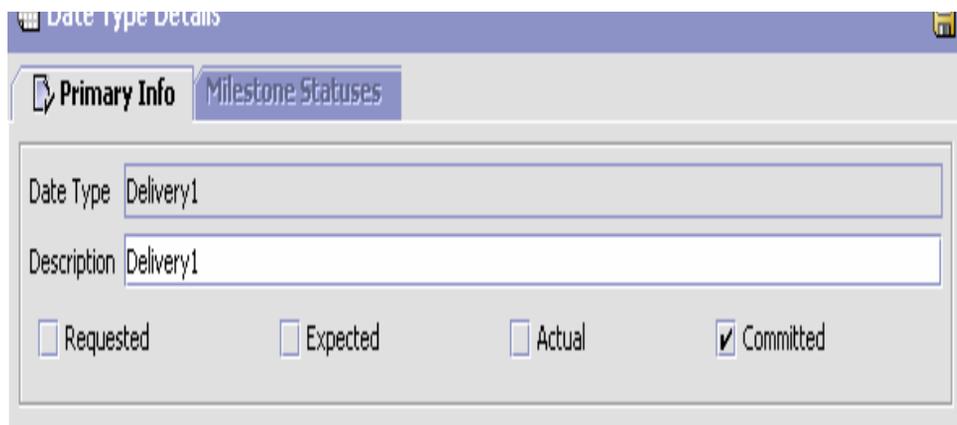
You can use the Date Types tab for:

- [Creating a Date Type](#)
- [Modifying a Date Type](#)
- [Deleting a Date Type](#)

### 15.4.1.1 Creating a Date Type

To create a date type:

1. In the Monitoring window, choose the Date Types tab.
2. From the Date Types list, choose . The Date Type Details window displays.
3. Enter information in the applicable fields. Refer to [Table 15–4](#) for field level descriptions.
4. Choose .



**Table 15–4** *Date Type Details Window*

Field	Description
Date Type	Enter the name of the date type.
Description	Enter a brief description of the date type.
Requested	Select this field to indicate if the date type represents a date requested by a Buyer, user, etc.
Expected	Select this field to indicate if the date type represents a date the system expects or has calculated something to occur.
Actual	Select this field to indicate if the date type represents the actual date.

### 15.4.1.2 Modifying a Date Type

To modify a date type:

5. In the Monitoring window, choose the Date Types tab.
6. From the Date Types list, locate the applicable date type and choose . The Date Type Details window displays.
7. Enter information in the applicable fields. Refer to [Table 15–4](#) for field level descriptions.
8. Choose .

### 15.4.1.3 Deleting a Date Type

To delete a date type:

**Note:** The following system dates cannot be deleted:

- Delivery Date
- Ship Date

1. In the Monitoring window, choose the Date Types tab.
2. From the Date Types list, locate the applicable date type and choose .

## 15.4.2 Defining Milestones

You can configure applicable statuses in a process type to be milestones. A milestone is a type of date. Selling and Fulfillment Foundation automatically determines when an order moves from one status to another. A milestone represents a significant point in the processing lifecycle that can be used as a criterion for monitoring. Milestones can be defined at the order, order line, order release, and order release line levels.

**Note:** A milestone can be reached whenever there is a change in an order line. Selling and Fulfillment Foundation marks a milestone as reached if an order line reaches a status marked as a milestone. However, there may be times that only part of an order line reaches a particular status defined as milestone.

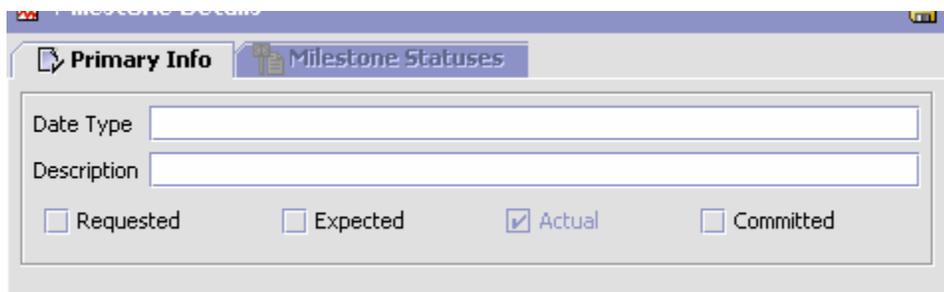
You can use the Milestones tab for:

- [Creating a Milestone](#)
- [Modifying a Milestone](#)
- [Deleting a Milestone](#)

### 15.4.2.1 Creating a Milestone

To create a milestone:

1. In the Monitoring window, choose the Milestones tab.
2. From the Milestones list, choose . The Milestone Details window displays.
3. Enter information in the applicable fields. Refer to [Table 15–5](#) for field level descriptions.
4. Choose .



The screenshot shows a software window titled "Milestone Details". It has two tabs: "Primary Info" (active) and "Milestone Statuses". Under the "Primary Info" tab, there are two text input fields: "Date Type" and "Description". Below these fields are four radio button options: "Requested", "Expected", "Actual" (which is selected), and "Committed".

Table 15–5 Milestone Details

Field	Description
Date Type	Enter the name of the milestone being created. <b>Note:</b> You cannot use date types you have created on the date type tab. You must create a unique name for the milestone.
Description	Enter a brief description of the milestone.
Requested	Select this field to indicate if the milestone represents a date requested by a Buyer, user, etc.
Expected	Select this field to indicate if the milestone represents a date the system expects or has calculated something to occur.
Actual	This field is not applicable for milestones.
Milestone Statuses	You can add statuses to associate with the milestone by selecting  and entering information in the applicable fields. <b>Note:</b> This tab can only be accessed once the Primary Info tab has been filled out and saved.
Date Type	The date type if any associated with the milestone.
Status	Select the status you want use to indicate the milestone has been reached.
Level	Select Order to indicate this status must be reached at the order header level. Select Order Line to indicate that this status must be reached at the order line level. Select Order Release to indicate that this status must be reached at the order release level.
Quantity Type	Select Initial to indicate that the milestone is met when any quantity at the above selected level moves into the status. Select Complete to indicate that the milestone is met when all quantity at the above selected level moves into the status.

### 15.4.2.2 Modifying a Milestone

**Important:** If modifications are made to an existing milestone, the changes are only applied to new orders. Existing orders for which milestone records have already been created are not considered.

To modify a milestone:

1. In the Monitoring window, choose the Milestones tab.
2. From the Milestones list, locate the applicable milestone and choose . The Milestone Details window displays.
3. Enter information in the applicable fields. Refer to [Table 15–5](#) for field level descriptions.
4. Choose .

### 15.4.2.3 Deleting a Milestone

To delete a milestone:

1. From the Monitoring window, choose the Milestones tab.
2. From the Milestones list, locate the applicable milestone and choose .

## 15.5 Defining Monitoring Events

Events are used in instances where the Order Monitor may raise multiple alerts of the same type. For example, if an order with multiple lines that are shipped together has a shipment delay and you have configured the Order Monitor to raise alerts when shipments are delayed at the line level, an alert of the same type would be raised against each line in the order. You can create rules to aggregate all of these similar alerts and raise one "root cause".

You can use the Monitor Events tab for:

- [Creating an Event Rule](#)
- [Modifying an Event](#)
- [Deleting an Event](#)

### 15.5.1 Creating an Event Rule

To create an event rule:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Monitor Events. The Monitor Events window displays.
2. From the Monitor Events list, choose . The Monitor Events Details window displays.
3. Enter information in the applicable fields. Refer to [Table 15–6](#) for field level descriptions.
4. Choose .

The screenshot shows a configuration window for defining monitoring events. It contains the following fields and options:

- Event Id:** A text input field.
- Description:** A text input field.
- Requires Realert:** A checkbox, currently unchecked.
- Realert Interval:** A text input field, visible only if 'Requires Realert' is checked.
- Automatically Resolve Alerts:** A checkbox, currently unchecked.
- Event Identified By:** A dropdown menu with 'Shipment' selected.
- Service To Be Invoked:** A dropdown menu.
- Aggregate And Invoke Service For:** A dropdown menu with 'Shipment' selected.

**Table 15–6** *Monitor Event Details Pop-Up Window*

Field	Description
Event Id	Enter the event ID.
Description	Enter a brief description of the event.
Requires Realert	Select this field if you want users to be re-alerted if the issue has not been resolved within a certain timeframe.

**Table 15–6 Monitor Event Details Pop-Up Window**

Field	Description
Realert Interval	If you selected Requires Realert, enter the interval (in hours) that re-alerts should be sent.
Automatically Resolve Alerts	Select this field to automatically remove an alert if the alert condition is no longer valid. Once the alert is resolved, a new alert is raised if the condition is detected again.
Event Identified By	
Shipment	Select this field if you want two or more alert conditions to be treated the same if they belong to the same shipment.
Service To Be Invoked	Select the alert service to be invoked should the event consolidation rule conditions be met.
Aggregate And Invoke Service For	
Shipment	Select this field if you want only one alert to be raised for a shipment when alert conditions are detected.

**Note:** In most cases the attributes that identify an event should be a subset of the attributes that specify event aggregation.

## 15.5.2 Modifying an Event

To modify an event rule:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Monitor Events. The Monitor Events window displays.
2. From the Monitor Events list, select the applicable event rule and choose . The Monitor Event Details window displays.
3. Enter information in the applicable fields. Refer to [Table 15–6](#) for field level descriptions.
4. Choose .

### 15.5.3 Deleting an Event

To delete an event rule:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Fulfillment > Monitor Events. The Monitor Events window displays.
2. From the Monitor Events list, select the applicable event rule and choose .

## 15.6 Defining Shipment Preferences

Shipment preferences can be created to enable over shipment of products, or allow the creation of shipments without order information in the system.

Shipment Preferences are divided into two sets:

- [Over Shipping Preferences](#)
- [Transaction Rules](#)

### 15.6.1 Over Shipping Preferences

Over shipment is the ability to ship more than an ordered quantity. Over shipment tolerance definitions can be configured using the following criteria:

- Line Type
- Seller Organization Code
- CustomerVendor Classification/BuyerSeller Organization Code
- Item Classification/Item ID

During shipment, if a shipping preference has not been configured that matches the criteria of the shipment line, over shipment is not allowed. Otherwise, over shipment within the specified percentage is allowed.

You can use the Shipping Preference branch for:

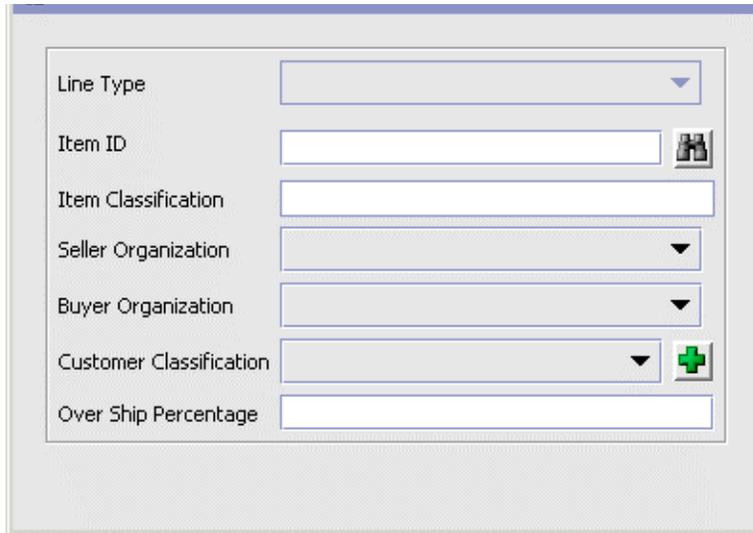
- [Creating a Shipment Preference](#)
- [Modifying a Shipment Preference](#)
- [Deleting a Shipment Preference](#)

### 15.6.1.1 Creating a Shipment Preference

To create a shipment preference:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Inbound Logistics > Shipping Preference. The Shipping Preference Search window displays.
2. In the Shipping Preference window, choose the Over Shipping Preferences tab. The Shipping Preference Search panel displays.

3. In the Search Results panel, choose . The Shipping Preference Details pop-up window displays.
4. Enter information into the applicable fields. Refer to [Table 15–7](#) for field value descriptions.
5. Choose .



The screenshot shows a configuration window with the following fields:

- Line Type: A dropdown menu.
- Item ID: A text input field with a help icon.
- Item Classification: A text input field.
- Seller Organization: A dropdown menu.
- Buyer Organization: A dropdown menu.
- Customer Classification: A dropdown menu with a plus icon.
- Over Ship Percentage: A text input field.

**Table 15–7 Shipping Preference Details**

Field	Description
Line Type	Select the line type you want to allow over shipment for.
Item ID	Enter the item ID of the item you want to allow over shipment for, if applicable.
Item Classification	Enter the item classification group you want to allow over shipment for, if applicable. For more information about item classification, see the <i>Catalog Management: Configuration Guide</i> .
Seller Organization	Select the Seller organization that you want to allow to over ship.
Buyer Organization	Select the Buyer organization that you want to be able to receive over shipments.
Customer Classification	Select the customer classification that you want to be able to receive over shipments, if applicable.
Over Ship Percentage	Enter the percentage allowed for over shipment.

### 15.6.1.2 Modifying a Shipment Preference

To modify a shipment preference:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Inbound Logistics > Shipping Preference. The Shipping Preference Search window displays.
2. In the Shipping Preference window, choose the Over Shipping Preferences tab. The Shipping Preference Search panel displays.
3. Enter the applicable search criteria and choose . A list of preferences displays.
4. Select the applicable preference and choose . The Shipping Preference Details pop-up window displays.
5. Enter information into the applicable fields. Refer to [Table 15-7](#) for field value descriptions.
6. Choose .

### 15.6.1.3 Deleting a Shipment Preference

To delete a shipment preference:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Inbound Logistics > Shipping Preference. The Shipping Preference Search window displays.
2. In the Shipping Preference window, choose the Over Shipping Preferences tab. The Shipping Preference Search panel displays.
3. Enter the applicable search criteria and choose . A list of preferences displays.
4. Select the applicable preference and choose .

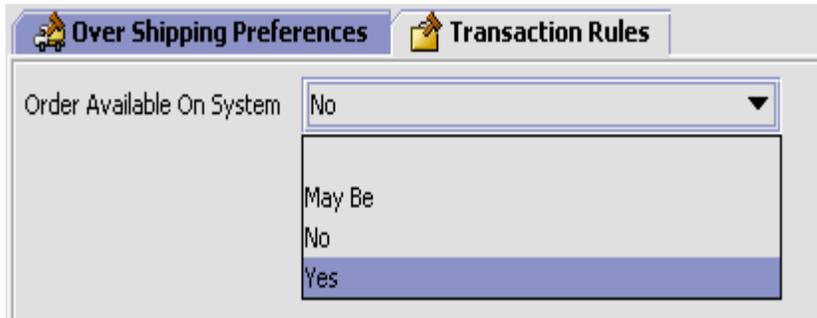
## 15.6.2 Transaction Rules

Transaction Rules define whether the system allows creation of shipments without order information on the system.

To define transaction rules:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Outbound Logistics > Shipping Preference. The Shipping Preference Search window displays.

2. In the Shipping Preference Search window, choose the Transaction Rules tab.
3. Enter information in the applicable fields. Refer to [Table 15–8](#) for field level descriptions.
4. Choose .



**Table 15–8 Transaction Rules Tab**

Field	Description
Order Available On System	<p>Select the appropriate option from the drop-down list to ensure that the shipments are either created against existing orders or not. The options are:</p> <ul style="list-style-type: none"> <li>• May Be - Select this option if the orders might be available on the system.</li> <li>• No - Select this option if the orders are not available on the system.</li> <li>• Yes - Select this option if the orders are available on the system.</li> </ul>

# 16

## Configuring an Order Document's Receipt-Specific Components

---

To complete an order document's lifecycle, each document has a set of different processes that it can go through. These processes are called process types. Every order document has a defined set of process types in Selling and Fulfillment Foundation.

The following process types are defined in Selling and Fulfillment Foundation for the order document types:

- Fulfillment
- Negotiation
- Shipment
- Receipt

You can configure the rules and components specific to an order document's receipt process type.

You can use process type configuration for:

- [Process Type Pipeline Configuration](#)
- [Defining Receipt Preferences](#)
- [Defining Receiving Dispositions](#)
- [Defining Receiving Discrepancy Reasons](#)

## 16.1 Defining Process Type Details

You can define the parameters and templates that distinguish a process type.

For more information about defining process type details, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

## 16.2 Process Type Pipeline Configuration

A **process type pipeline** is a series of transactions and statuses that guide document types, such as a Sales Order, through a predefined process. A pipeline consists of the different statuses a document goes through during fulfillment, negotiation, shipment, or receipt. You can also set up transactions consisting of events, actions, and conditions, as they pertain to the pipeline you are configuring.

### Repositories

A repository is a logical collection of entities that define the business process workflow.

The following entities are included in a repository:

- Pipelines
- Transactions
- Statuses
- Conditions
- Actions
- Services

Selling and Fulfillment Foundation provides a base repository for each of the system-defined process types. Some of the entities within a repository are copied when creating a new document type. For more information about creating a new document type, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

The process of purchase order receipt is modeled through a pipeline. This represents the process configuration that is unique to an organization. An organization may also specify unique processes for each participating Enterprise.

## 16.2.1 Defining Pipeline Determination

**Pipeline determination** is used to set up conditions that affect which pipeline is used during the start of the business process workflow. For example, an organization deals with sales orders that sometimes contain hazardous materials. They have two separate pipelines, one in which orders with order lines without any hazardous materials go through and one in which orders with order lines containing hazardous materials must go through for inspection before continuing through the order process. The organization uses pipeline determination to set up a condition that determines whether or not order lines contain hazardous materials and sends the order line down the correct pipeline.

When you expand the Pipeline Determination branch, the components displayed depends on what role you are logged in as. If you are logged in as a Hub role, the Hub Rule displays. If you are logged in as an Enterprise role, both the Hub Rule and My Rule components display. Double-click on the applicable node to display the pipeline determination rules.

---

---

**Note:** If you are logged in as an Enterprise role, the Hub Rule screen is grayed out and cannot be modified.

---

---

Drag conditions and pipelines into the work area to construct pipeline determination rules. A single pipeline or condition must be the root. Conditions cannot link back to an earlier component in the chain and a pipeline cannot be linked to twice.

---

---

**Note:** When configuring pipeline determination for an order document type pipeline, please note that pipeline determination is only considered when adding a line or creating an order. When changes are made to draft orders pipeline determination does not occur.

---

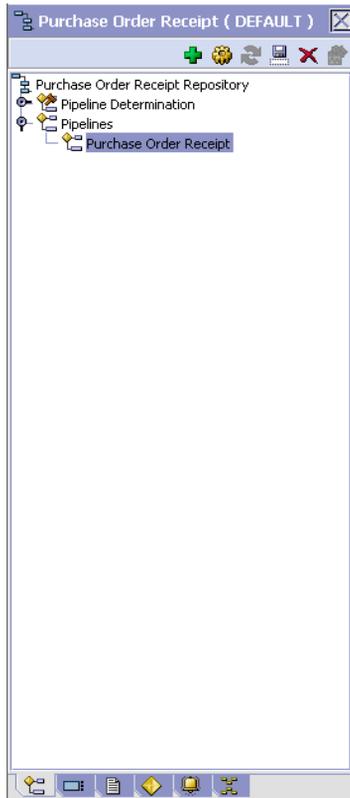
---

## 16.2.2 Pipelines

For more information about configuring pipelines, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

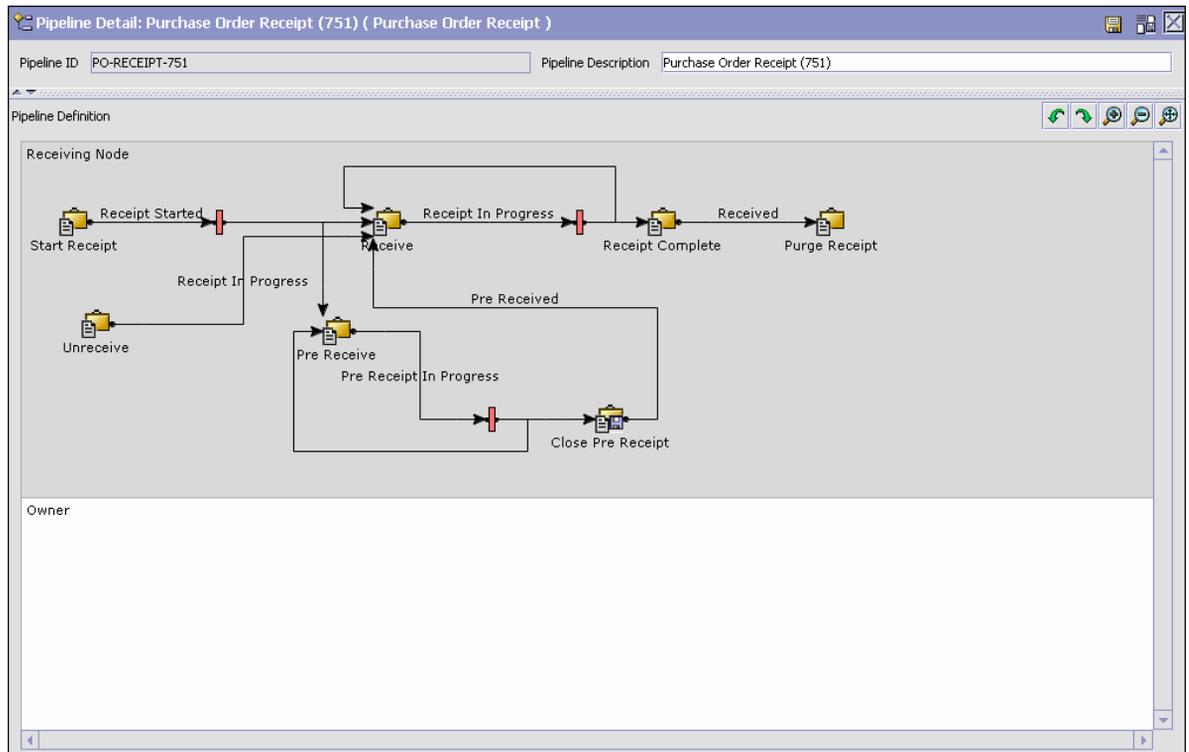
To view the purchase order receipt pipeline details:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receipt Process Model. The Purchase Order Receipt window displays.



2. In the Purchase Order Receipt window, choose Purchase Order Receipt Repository > Pipelines > Purchase Order Receipt.
3. The Pipeline Detail: Purchase Order Receipt (Purchase Order Receipt) window displays.

For more information about creating and modifying a pipeline, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



### 16.2.3 Transactions

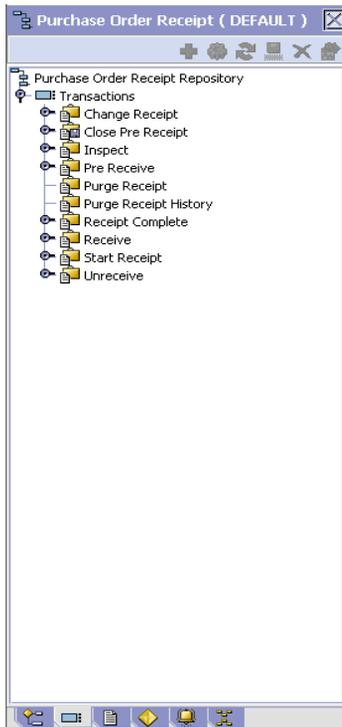
Every process type has a set of base transactions defined for it. A transaction is a logical unit of work that is necessary for performing activity within Selling and Fulfillment Foundation. Base transactions are predefined transactions that contain information about how the transaction behaves, such as how many copies of a transaction can be kept in a process type and whether or not it can have configurable base pick and drop statuses. Base transactions can be used to create new transactions. These transactions can be changed within the limits defined in the base transaction.

For more information about transactions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the transaction details for an purchase order receipt pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receipt Process Model. The Purchase Order Receipt window displays.
2. In the Purchase Order Receipt window, choose .
3. The Transactions tab window displays.

For more information about creating and modifying transactions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 16–1 Purchase Order Receipt Pipeline - Transactions Tab Window**

Field	Description
Change Receipt	This transaction represents any modifications that may be made to a purchase order receipt.
Close Pre-Receipt	This transaction represents the completion of the pre-receiving process.
Inspect	This transaction represents the purchase order receipt being manually inspected.
Pre-Receive	This transaction represents the pre-receipt of a container. The contents of the container are not known by the system until the actual receipt is performed.
Purge Receipt	This transaction represents the process of moving purchase order receipts to the history tables.
Purge Receipt History	This transaction represents the process of purging purchase order receipts from the history tables and removing them from the system.
Receipt Complete	This transaction represents the completion of a receipt.
Receive	This transaction represents the process of receiving a purchase order receipt.
Start Receipt	This transaction indicates the start of the purchase order receipt process.
Unreceive	This transaction represents the reversal of any performed receipt. This is used to correct any receipt that may have been recorded incorrectly.

## 16.2.4 Statuses

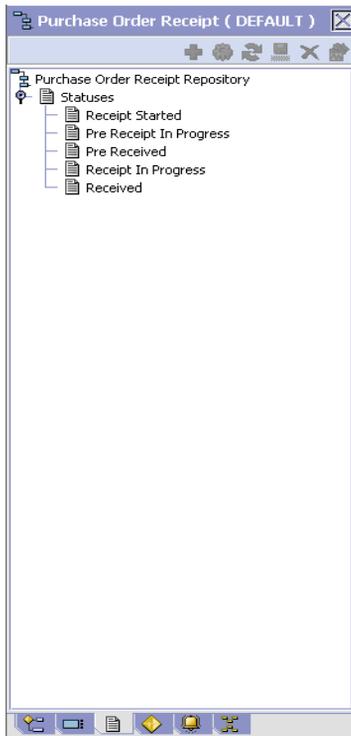
**Statuses** are the actual states that a document moves through in the pipeline. A transaction can contain two types of statuses, a drop status and a pickup status. A document is moved into a **drop status** when the events and conditions of a transaction have been completed. A **pickup status** takes the document from the previous drop status and moves it through the next transaction. Created and Scheduled are examples of statuses.

For more information about statuses, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the status details of a purchase order receipt pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receipt Process Model. The Purchase Order Receipt window displays.
2. In the Purchase Order Receipt window, choose .
3. The Statuses tab window displays.

For more information about creating and modifying statuses, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 16–2 Return Receipt Pipeline - Statuses Tab Window**

Field	Description
Receipt Started	This indicates that a purchase order receipt has started.
Pre-Receipt In Progress	This indicates that the purchase order receipt is being pre-received and cannot continue in the pipeline until pre-receipt is complete.
Pre-Received	This indicates that the purchase order receipt has been pre-received.
Receipt In Progress	This indicates that the purchase order receipt is being received and cannot continue in the pipeline until marked as received.  <b>Note:</b> When the purchase order receipt is moved to Receipt In Progress status, the purchase order is moved to Received status in the Purchase Order Execution pipeline.
Received	This indicates that the purchase order receipt has been received.  <b>Note:</b> When the purchase order receipt is moved to Received status, the purchase order is moved to Receipt Closed status in the Purchase Order Execution pipeline.

## 16.2.5 Conditions

A **condition** matches document type attributes against decision points and routes the documents to different paths based on the specified attribute and value combinations. The document type attributes against which conditions can be created are predefined in Selling and Fulfillment Foundation. You can use these attributes in any combination or you can create conditions that run the appropriate application logic for specific circumstances.

For more information about conditions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the condition details of a purchase order receipt pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receipt Process Model. The Purchase Order Receipt window displays.

2. In the Purchase Order Receipt window, choose .
3. The Conditions tab window displays.

For more information about creating and modifying conditions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 16–3 Purchase Order Receipt Pipeline - Conditions Tab Window**

Field	Description
Conditions	Displays conditions that are specific to the purchase order receipt pipeline, if any.

## 16.2.6 Actions

An **action** is a process or program that is triggered by an event. These processes and programs send user alert notifications and automatically resolve issues.

For example, when an order is released (the event), you can set an action to send the customer an e-mail.

For more information about actions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the action details of a purchase order receipt pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receipt Process Model. The Purchase Order Receipt window displays.
2. In the Purchase Order Receipt window, choose .
3. The Actions tab window displays.

For more information about creating and modifying actions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

## 16.3 Defining Receipt Preferences

Receipt preferences can be created to enable over receipt of products in the system. Over receipt is the ability to receive more than an ordered quantity. Over receipt tolerance definitions can be configured using the following criteria:

- Line Type
- Seller Organization Code
- CustomerVendor Classification/BuyerSeller Organization Code
- Item Classification/Item ID

During receipt, if a receiving preference has not been configured that matches the criteria of the receipt line, over receipt is not allowed. Otherwise, over receipt within the specified percentage is allowed.

---

---

**Note:** Sterling Commerce recommends that at a minimum you should create a default receiving preference that can be used across all nodes.

---

---

You can use the Receiving Preference branch for:

- [Creating a Receipt Preference](#)
- [Modifying a Receipt Preference](#)
- [Deleting a Receipt Preference](#)

### 16.3.1 Creating a Receipt Preference

To create a receipt preference:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receiving Preference. The Receiving Preference Search window displays.
2. Choose . The Receiving Preference Details pop-up window displays.
3. Enter information into the applicable fields. Refer to [Table 16–4](#) for field value descriptions.
4. Choose .

**Table 16–4 Receiving Preference Details**

Field	Description
Line Type	Select the line type you want to allow over receipt for.
Item ID	Enter the item ID of the item you want to allow over receipt for, if applicable.
Item Classification	Enter the item classification group you want to allow over receipt for, if applicable. For more information about item classification, see the <i>Catalog Management: Configuration Guide</i> .
Seller Organization	Select the Seller organization that you want to allow to over receive.
Buyer Organization	Select the Buyer organization that you want to be able to receive over receipts.
Over Receipt Percentage	Enter the percentage allowed for over receipt.

**Table 16–4 Receiving Preference Details**

Field	Description
QC Required	Select this field if you require a quality control check upon receipt.
Case Content Entry Required	Select this field if you require the contents of cases to be entered into the system upon receipt.

### 16.3.2 Modifying a Receipt Preference

To modify a receipt preference:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receiving Preference. The Receiving Preference Search window displays.
2. Enter the applicable search criteria and choose . A list of preferences displays.
3. Select the applicable preference and choose . The Receiving Preference Details pop-up window displays.
4. Enter information into the applicable fields. Refer to [Table 16–4](#) for field value descriptions.
5. Choose .

### 16.3.3 Deleting a Receipt Preference

To delete a receipt preference:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receiving Preference. The Receiving Preference Search window displays.
2. Enter the applicable search criteria and choose . A list of preferences displays.
3. Select the applicable preference and choose .

## 16.4 Defining Receiving Dispositions

You can define common codes for receiving dispositions used when handling a receipt. This common code identifies what happens to items for the document type when they are received.

You can use the Receiving Disposition branch for:

- [Creating a Receiving Disposition](#)
- [Modifying a Receiving Disposition](#)
- [Deleting a Receiving Disposition](#)

### 16.4.1 Creating a Receiving Disposition

To create a receiving disposition:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receiving Disposition. The Return Disposition window displays.
2. Choose . The Receiving Disposition Details pop-up window displays.
3. Enter information into the applicable fields. Refer to [Table 16–5](#) for field value descriptions.
4. Choose .

Receiving Disposition

Short Description

Product Class  

Damaged  Final Disposition

**Disposition Transitions : Purchase Order ( DEFAULT )** 

	Receiving Disposition
*	<input type="text"/>

Results 0 Of 0

**Table 16–5 Receiving Disposition Details**

Field	Description
Receiving Disposition	Enter the name of the receiving disposition.
Short Description	Enter a brief description of the receiving disposition.
Product Class	Select a product class to associate with received items, if applicable. For example, by you could assign the product class of Returned to any returned items.
Damaged	Select Damaged if the receiving disposition is used for handling damaged items.

**Table 16–5 Receiving Disposition Details**

Field	Description
Final Disposition	Select Final Disposition if the receiving disposition is to be used as final disposition for the receipt. Final Disposition marks the disposition code as final, and does not allow any further disposition transitions.
Disposition Transitions	<b>This displays the existing Receiving Disposition codes that are available for associating a transition with the Receiving Disposition being created or modified.</b>
Receiving Disposition	Existing Receiving Disposition Code available for transition association.

## 16.4.2 Modifying a Receiving Disposition

To modify a receiving disposition:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receiving Disposition. The Return Disposition window displays.
2. Select the applicable receiving disposition and choose . The Receiving Disposition Details pop-up window displays.
3. Enter information into the applicable fields. Refer to [Table 16–5](#) for field value descriptions.
4. Choose .

## 16.4.3 Deleting a Receiving Disposition

To delete a receiving disposition:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receiving Disposition. The Return Disposition window displays.
2. Select the applicable receiving disposition and choose .

## 16.5 Defining Receiving Discrepancy Reasons

You can define codes to specify reasons for any discrepancies that may occur during the receipt of a shipment or return.

The different types of receiving discrepancies that occur are:

- **Over Receipt**—Occurs when a receiving node receives additional quantity than the quantity expected.
- **Under Receipt**—Occurs when a receiving node receives less than the expected quantity for the receipt.
- **Damaged Receipt**—Occurs when the receiving disposition code indicates that a damaged product has been received.

**Note:** You can define multiple reason codes for a given discrepancy type. For example, if you receive any shipment with a quantity of 10, which is less than the expected quantity, then for the Under Receipt discrepancy type, you can specify two different reasons for the receipt, such as 6 units SHORT\_SHIPMENT and 4 units CARRIER\_FAULT.

You can use the Receiving Discrepancy Reasons branch for:

- [Creating a Receiving Discrepancy Reason](#)
- [Modifying a Receiving Discrepancy Reason](#)
- [Deleting a Receiving Discrepancy Reason](#)

### 16.5.1 Creating a Receiving Discrepancy Reason

To create a receiving discrepancy reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receiving Discrepancy Reasons. The Receiving Discrepancy Reasons window displays.

2. Choose . The Receiving Discrepancy Reason Details pop-up window displays.
3. Enter information in the applicable fields. For field value descriptions, see [Table 16–6](#).
4. Choose .

**Table 16–6 Receiving Discrepancy Reason Details**

Fields	Description
Discrepancy Reason Code	Enter the name of the discrepancy reason code as you want it to appear throughout the system.
Discrepancy Reason Description	Enter a brief description of the discrepancy reason.
Discrepancy Reference	Enter any additional reference information according to your business practices.
<b>Discrepancy Type Group</b>	
Over Receipt	Choose this option if you want the discrepancy reason to identify scenarios in which a receiving node receives more than the expected quantity.

**Table 16–6 Receiving Discrepancy Reason Details**

Fields	Description
Under Receipt	Choose this option if you want the discrepancy reason to identify scenarios in which a receiving node receives less than the expected quantity.
Damaged Receipt	Choose this option to identify scenarios in which a receiving node receives items with a receiving disposition identifying them as damaged.
Requires Invoice Adjustment	Check this box if a monetary adjustment need to be made when a receipt discrepancy is associated with the discrepancy reason.
<b>Invoice Adjustment Type Group</b>	
Credit	Choose this option if the adjustment amount results in a credit invoice.
Debit	Choose this option if the adjustment amount results in a debit invoice.
Invoice Line Reference	If you select the Requires Invoice Adjustment option, you must enter a name for the adjustment. This name can be used when multiple adjustment invoices are created for the same order line. In such situations, the invoices are split into different invoice lines only if the invoices have different invoice line references.

## 16.5.2 Modifying a Receiving Discrepancy Reason

To modify a receiving discrepancy reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receiving Discrepancy Reasons. The Receiving Discrepancy Reasons window displays.
2. Select the receiving discrepancy reason and choose . The Receiving Discrepancy Reason Details pop-up window displays.
3. Enter information in the applicable fields. For field value descriptions, see [Table 16–6](#).
4. Choose .

### 16.5.3 Deleting a Receiving Discrepancy Reason

To delete a receiving discrepancy reason:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Receipt > Receiving Discrepancy Reasons. The Receiving Discrepancy Reasons window displays.
2. Select the receiving discrepancy reason and choose .



## Configuring an Order Document's Negotiation-Specific Components

---

**Important:** Be aware that return fulfillment requires sourcing configuration. Sourcing configuration is accessible through the Distributed Order Management configuration grouping. For more information about configuring sourcing, see the *Sterling Distributed Order Management: Configuration Guide*.

To complete an order document's lifecycle, each document has a set of different processes that it can go through. These processes are called process types. Every order document has a defined set of process types in Selling and Fulfillment Foundation.

The following process types are defined in Selling and Fulfillment Foundation for the order document types:

- Fulfillment
- Negotiation
- Shipment
- Receipt
- Receipt

You can configure the rules and components specific to an order document's negotiation process type.

You can use process type configuration for:

- [Defining Process Type Details](#)
- [Process Type Pipeline Configuration](#)

### 17.1 Defining Process Type Details

You can define the parameters and templates that distinguish a process type.

For more information about defining process type details, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

### 17.2 Process Type Pipeline Configuration

A **process type pipeline** is a series of transactions and statuses that guide document types, such as a Sales Order, through a predefined process. A pipeline consists of the different statuses a document goes through during fulfillment, negotiation, shipment, or receipt. You can also set up transactions consisting of events, actions, and conditions, as they pertain to the pipeline you are configuring.

#### Repositories

A repository is a logical collection of entities that define the business process workflow.

The following entities are included in a repository:

- Pipelines
- Transactions
- Statuses
- Conditions
- Actions
- Services

Selling and Fulfillment Foundation provides a base repository for each of the system-defined process types. Some of the entities within a repository are copied when creating a new document type. For more information about creating a new document type, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

The process of negotiation is modeled through a pipeline. This represents the process configuration that is unique to an organization. An organization may also specify unique processes for each participating Enterprise.

### 17.2.1 Defining Pipeline Determination

**Pipeline determination** is used to set up conditions that affect which pipeline is used during the start of the business process workflow. For example, an organization deals with sales orders that sometimes contain hazardous materials. They have two separate pipelines, one in which orders with order lines without any hazardous materials go through and one in which orders with order lines containing hazardous materials must go through for inspection before continuing through the order process. The organization uses pipeline determination to set up a condition that determines whether or not order lines contain hazardous materials and sends the order line down the correct pipeline.

When you expand the Pipeline Determination branch, the components displayed depends on what role you are logged in as. If you are logged in as a Hub role, the Hub Rule displays. If you are logged in as an Enterprise role, both the Hub Rule and My Rule components display. Double-click on the applicable node to display the pipeline determination rules.

---

---

**Note:** If you are logged in as an Enterprise role, the Hub Rule screen is grayed out and cannot be modified.

---

---

Drag conditions and pipelines into the work area to construct pipeline determination rules. A single pipeline or condition must be the root. Conditions cannot link back to an earlier component in the chain and a pipeline cannot be linked to twice.

---

---

**Note:** When configuring pipeline determination for an order document type pipeline, please note that pipeline determination is only considered when adding a line or creating an order. When changes are made to draft orders pipeline determination does not occur.

---

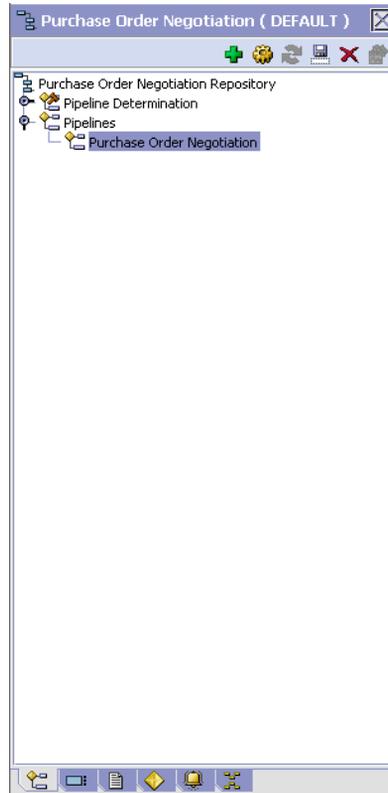
---

### 17.2.2 Pipelines

For more information about configuring pipelines, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

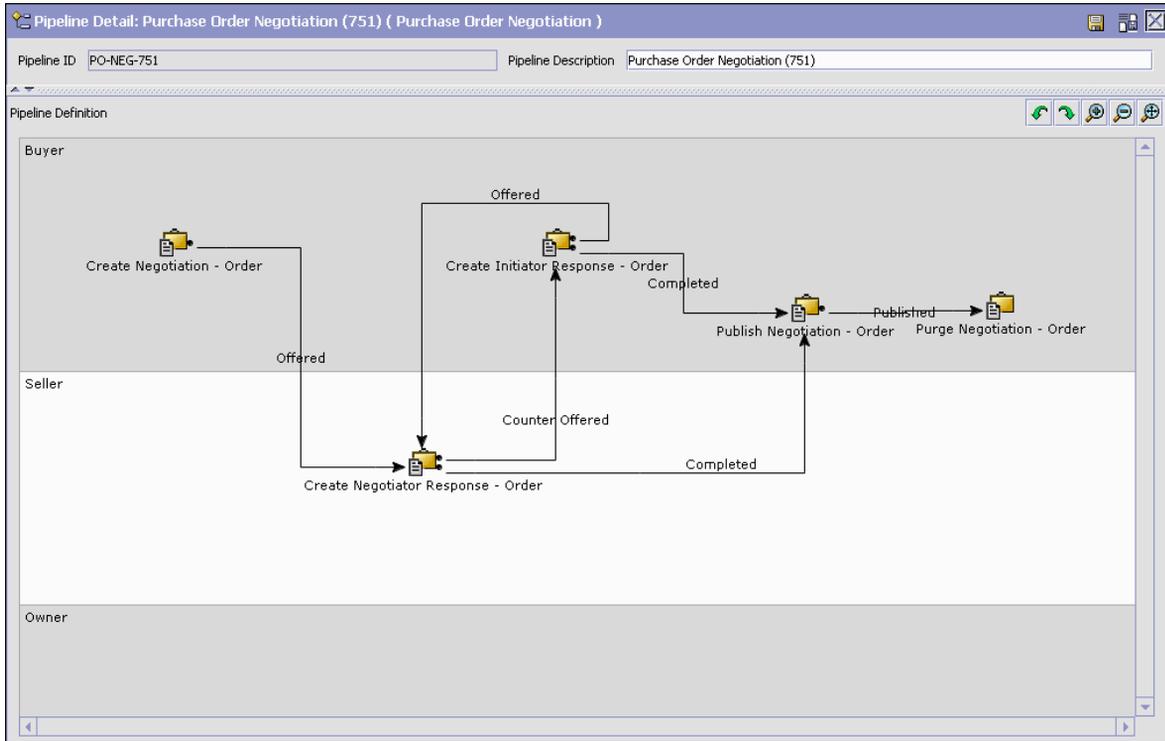
To view the negotiation pipeline details,

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Negotiation > Negotiation Process Model. The Purchase Order Negotiation window displays.



2. In the Purchase Order Negotiation window, choose Purchase Order Negotiation Repository > Pipelines > Purchase Order Negotiation.
3. The Pipeline Detail: Purchase Order Negotiation (Purchase Order Negotiation) window displays.

For more information about creating and modifying a pipeline, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



### 17.2.3 Transactions

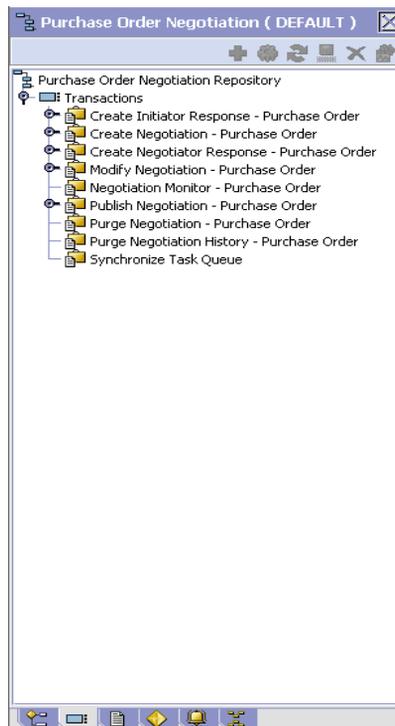
Every process type has a set of base transactions defined for it. A transaction is a logical unit of work that is necessary for performing activity within Selling and Fulfillment Foundation. Base transactions are predefined transactions that contain information about how the transaction behaves, such as how many copies of a transaction can be kept in a process type and whether or not it can have configurable base pick and drop statuses. Base transactions can be used to create new transactions. These transactions can be changed within the limits defined in the base transaction.

For more information about transactions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the transaction details for a negotiation pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Negotiation > Negotiation Process Model. The Purchase Order Negotiation window displays.
2. In the Purchase Order Negotiation window, choose .
3. The Transactions tab window displays.

For more information about creating and modifying transactions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 17–1 Negotiation Pipeline - Transactions Tab Window**

Field	Description
Create Initiator Response	This transaction represents the creation of a negotiation response from the initiating organization.
Create Negotiation	This transaction represents a the creation of a negotiation for a purchase order.
Create Negotiator Response	This transaction represents the creation of a negotiation response from the negotiating organization.
Modify Negotiation	This transaction represents any modifications made to a negotiation.
Negotiation Monitor	This transaction represents the process of monitoring negotiations in the system based on defined parameters.
Publish Negotiation	This transaction represents the process of publishing the negotiated terms to an order.
Purge Negotiation	This transaction represents the process of moving negotiations to the history tables.
Purge Negotiation History	This transaction represents the process of purging negotiations from the history tables and removing them from the system.
Synchronize Task Queue	This transaction represents the process of synchronizing the negotiation fulfillment task queue.

## 17.2.4 Statuses

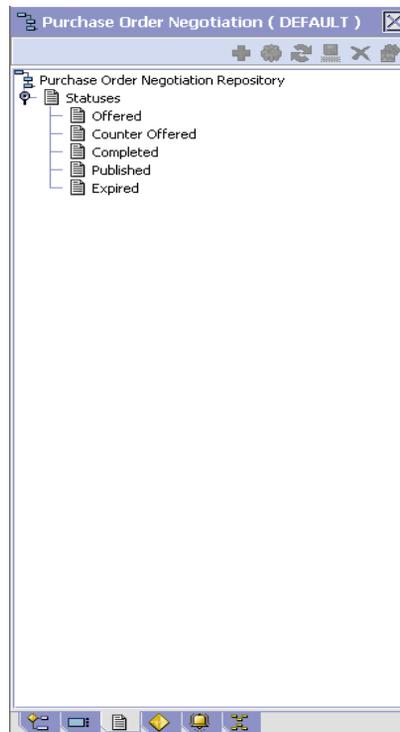
**Statuses** are the actual states that a document moves through in the pipeline. A transaction can contain two types of statuses, a drop status and a pickup status. A document is moved into a **drop status** when the events and conditions of a transaction have been completed. A **pickup status** takes the document from the previous drop status and moves it through the next transaction. Created and Scheduled are examples of statuses.

For more information about statuses, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the status details of a negotiation pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Negotiation > Negotiation Process Model. The Purchase Order Negotiation window displays.
2. In the Purchase Order Negotiation window, choose .
3. The Statuses tab window displays.

For more information about creating and modifying statuses, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 17–2 Negotiation Pipeline - Statuses Tab Window**

Field	Description
Offered	This indicates that a set of negotiation terms have been offered.
Counter Offered	This indicates that a revised set of negotiation terms have been counter offered.
Completed	This indicates that the terms of the negotiation have been agreed upon and are ready to be published to the order.
Published	This indicates that the negotiation results have been published to the order.
Expired	This indicates that the negotiation terms were never agreed upon.

## 17.2.5 Conditions

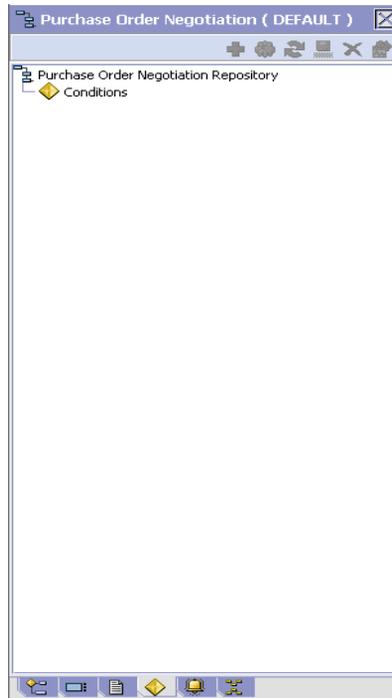
A **condition** matches document type attributes against decision points and routes the documents to different paths based on the specified attribute and value combinations. The document type attributes against which conditions can be created are predefined in Selling and Fulfillment Foundation. You can use these attributes in any combination or you can create conditions that run the appropriate application logic for specific circumstances.

For more information about conditions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the condition details of a negotiation pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Negotiation > Negotiation Process Model. The Purchase Order Negotiation window displays.
2. In the Purchase Order Negotiation window, choose .
3. The Conditions tab window displays.

For more information about creating and modifying conditions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.



**Table 17–3** *Negotiation Pipeline - Conditions Tab Window*

Field	Description
Conditions	Displays conditions that are specific to the negotiation pipeline, if any.

## 17.2.6 Actions

An **action** is a process or program that is triggered by an event. These processes and programs send user alert notifications and automatically resolve issues.

For example, when an order is released (the event), you can set an action to send the customer an e-mail.

For more information about actions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

To view the action details of an outbound shipment pipeline:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Negotiation > Negotiation Process Model. The Purchase Order Negotiation window displays.
2. In the Purchase Order Negotiation window, choose .
3. The Actions tab window displays.

For more information about creating and modifying actions, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

# 18

## Configuring a Document's Financial Components

---

You can define rules and common codes as they pertain to payments and charges for a given order document.

You can use the Financial Attributes branch for:

- [Defining Payment Terms](#)
- [Defining Charge Definitions](#)
- [Defining Tax Names](#)
- [Defining Additional Payment Rules](#)

### 18.1 Defining Payment Terms

You can define common codes for **payment terms** that you may have with your customers. These terms are pre-defined methods of payment.

You can use the Payment Terms tab for:

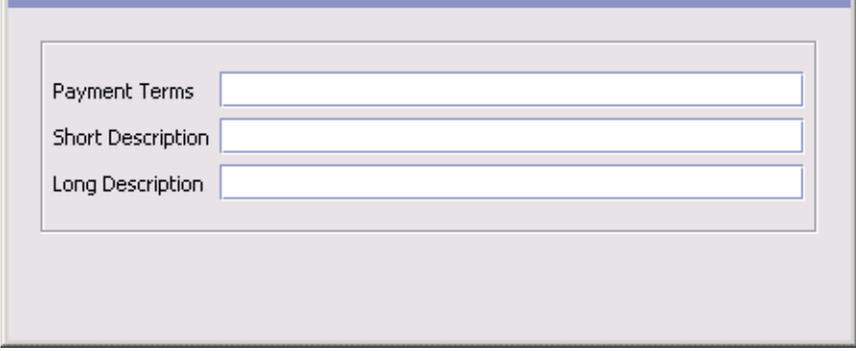
- [Creating a Payment Term](#)
- [Modifying a Payment Term](#)
- [Deleting a Payment Term](#)

#### 18.1.1 Creating a Payment Term

To create a payment term:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Payment Terms. The Payment Terms window displays in the work area.

2. Choose . The Payment Term Details pop-up window displays.



Payment Terms

Short Description

Long Description

3. In Payment Term, enter the name of the payment term.
4. In Short Description, enter a brief description of the payment term.
5. In Long Description, enter a more detailed description of the payment term.
6. Choose .

### 18.1.2 Modifying a Payment Term

To modify a payment term:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Payment Terms. The Payment Terms window displays in the work area.
2. Select the applicable payment term and choose . The Payment Term Details pop-up window displays.
3. In Short Description, enter a brief description of the payment term.
4. In Long Description, enter a more detailed description of the payment term.
5. Choose .

### 18.1.3 Deleting a Payment Term

To delete a payment term:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Payment Terms. The Payment Terms window displays in the work area.
2. Select the applicable payment term and choose .

## 18.2 Defining Charge Definitions

You can define **charge definitions** that you can associate with orders and invoices by creating charge categories. These categories contain a group of related charge names that can be used when the particular category is used. When adding a charge to an order header or an order line, you must use the charge categories that you have defined here. The charge name that is used on the order header or on the order line may or may not be defined, depending on the Validate Charge Name rule in the additional payment rules. For more information on this rule, refer to [Section 18.4, "Defining Additional Payment Rules"](#).

Following are the Selling and Fulfillment Foundation default charge definitions:

- Shipping
- Handling
- Personalization
- Discount

---

---

**Note:** The default charge definitions are only available to the Hub organization at the time of installation. Any Enterprises that are created must create their own charge definitions.

---

---

Use the Charge Definitions tab for:

- [Creating a Charge Category](#)
- [Modifying a Charge Category](#)
- [Adding a Charge Name Associated with a Charge Category](#)
- [Modifying a Charge Name Associated with a Charge Category](#)
- [Deleting a Charge Name Associated with a Charge Definition](#)
- [Deleting a Charge Definition](#)

### 18.2.1 Creating a Charge Category

To create a charge category:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Financial Attributes. The Financial window displays in the work area.
2. Choose the Charge Definitions tab.
3. Choose . The Charge Category Details window displays.

Charge Category Details : Sales Order : Handling ( DEFAULT )

Charge Category: Handling

Description: Handling

Billable  Discount

Charge Name	Description
Handling	Handling

Results 1 Of 1

4. In Charge Category, enter the name of the charge category.
5. In Description, enter a brief description of the charge category.
6. Select Billable if the charge is billable. Non-billable charges are not considered in order totals, but do appear in invoices.
7. Select Is Fee (or Discount if applied to a pickup request) if the charge you are creating is a discount charge type.
8. Choose .

---

---

**Note:** Charge categories cannot be localized. For more information about localization, see the *Selling and Fulfillment Foundation: Localization Guide*.

---

---

You can use the Charge Category Details window for:

- [Adding a Charge Name Associated with a Charge Category](#)
- [Modifying a Charge Name Associated with a Charge Category](#)
- [Deleting a Charge Name Associated with a Charge Category](#)

### 18.2.1.1 Adding a Charge Name Associated with a Charge Category

Charge names are names of the actual charges included in the charge definition.

---

---

**Note:** Sterling Commerce recommends that you clearly specify between order charges and discount charges when naming a charge. In the Application Consoles both order charges and discount charges appear on the same screens and drop-down menu. There is no way for the user to distinguish which is an order charge and which is a discount charge other than its naming convention.

---

---

To add a charge name to a charge category:

1. In the Charge Category Details window, choose . The Charge Name Details pop-up window displays.

The screenshot shows a dialog box titled "Charge Name Details". It has a title bar with a flag icon on the left and a save icon on the right. The main area contains three text input fields stacked vertically. The first field is labeled "Charge Category" and contains the text "Shipping". The second field is labeled "Charge Name" and is empty. The third field is labeled "Description" and is empty.

2. In Charge Name, enter the charge name.
3. In Description, enter a brief description of the charge name.
4. Choose .

**Note:** Charge names cannot be localized. For more information about localization, see the *Selling and Fulfillment Foundation: Localization Guide*.

### 18.2.1.2 Modifying a Charge Name Associated with a Charge Category

To modify a charge category's charge name:

1. In the Charge Category Details window, select the applicable charge name and choose . The Charge Name Details pop-up window displays.
2. In Description, enter a brief description of the charge name.
3. Choose .

### 18.2.1.3 Deleting a Charge Name Associated with a Charge Category

To delete a charge category's charge name select the applicable charge name in the Charge Category Details window and choose .

## 18.2.2 Modifying a Charge Category

To modify a charge category:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Financial Attributes. The Financial window displays in the work area.
2. Choose the Charge Definitions tab.
3. Select the applicable charge category and choose . The Charge Category Details window displays.
4. In Description, enter a brief description of the charge category.
5. Select Billable if the charge is billable. Non-billable charges are not considered in order totals, but do appear in invoices.
6. Select Discount if the charge you are creating is a discount charge type.
7. Choose .

## 18.2.3 Deleting a Charge Category

To delete a charge definition:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Financial Attributes. The Financial window displays in the work area.
2. Choose the Charge Definitions tab.
3. Select the applicable charge category and choose .

## 18.3 Defining Tax Names

You can define common codes for tax names. **Tax names** are any specific taxes that may pertain to orders and invoices.

Selling and Fulfillment Foundation understands three different types of taxes: a tax against a price, against a charge, or a flat tax.

- A tax against a price is an additional cost for a percentage of the price of the order line.
- A tax against a charge is an additional cost for a percentage of an existing charge on the order header, or order line. When adding a tax against a charge, the charge category must be one that already exists on the order header, or on the order line.
- A flat tax is a fixed tax applied on an order, independently of any charge, or price.

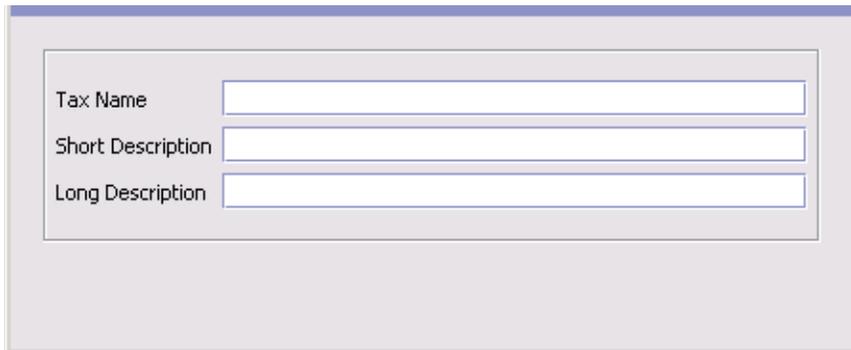
You can use the Tax Names tab for:

- [Creating a Tax Name](#)
- [Modifying a Tax Name](#)
- [Deleting a Tax Name](#)

### 18.3.1 Creating a Tax Name

To create a tax name:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Financial Attributes. The Financial window displays in the work area.
2. Choose the Tax Names tab.
3. Choose . The Tax Name Details pop-up window displays.



The screenshot shows a pop-up window titled "Tax Name Details" with a light gray background. Inside the window, there is a white rectangular area containing three text input fields. The first field is labeled "Tax Name", the second is labeled "Short Description", and the third is labeled "Long Description". Each label is positioned to the left of its corresponding input field.

4. In Tax Name, enter the name of the tax name.
5. In Short Description, enter a brief description of the tax name.
6. In Long Description, enter a more detailed description of the tax name.
7. Choose .

### 18.3.2 Modifying a Tax Name

To modify a tax name:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Financial Attributes. The Financial window displays in the work area.
2. Choose the Tax Names tab.
3. Select the applicable tax name and choose . The Tax Name Details pop-up window displays.
4. In Short Description, enter a brief description of the tax name.
5. In Long Description, enter a more detailed description of the tax name.
6. Choose .

### 18.3.3 Deleting a Tax Name

To delete a tax name:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Financial Attributes. The Financial window displays in the work area.
2. Choose the Tax Names tab.
3. Select the applicable tax name and choose .

## 18.4 Defining Additional Payment Rules

You can set up payment collection rules that are used when an order is sent for payment authorization.

To define additional payment rules:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Financials > Financial Attributes. The Financial window displays in the work area.
2. Choose the Others tab.
3. Enter information in the applicable fields.
4. Choose .



## Configuring a Document's Purge Criteria

---

**Purge Criteria** business rules are used to define qualifications around each type of purge. **Purges** are the process by which old data is removed from the system database. Purges minimize the number of unused database records to increase search efficiency and reduce the size of the required physical disk. In Purge Criteria Rules, default purge rules are provided. These can be modified for your system operations.

[Table 20–1](#) lists the purge rules provided for order document types in Selling and Fulfillment Foundation.

*Table 20–1 Order Document Type Purge Rules*

Rule	Description	Default Retention Days
PRG_SHIP_STATS	Purges shipment statistics and archives them in the history tables.	30
STATUSAUDITPRG	Purges order age alerts (if you have configured the system to trigger alerts when the order document type stays in a particular status for a specified time period).	30
NEGOTIATIONPRG	Purges negotiation information and archives it in the history tables.	30
NEGOTIATIONHISTPRG	Purges negotiation information from the negotiation history tables.	30

**Table 20–1 Order Document Type Purge Rules**

<b>Rule</b>	<b>Description</b>	<b>Default Retention Days</b>
RECEIPTPRG	Purges receipt information and archives it in the history tables.	30
RECEIPTHISTPRG	Purges receipt information from the receipt history tables.	30
ORDERHISTPRG	Purges order information from the order history tables.	30
ORDERPRG	Purges order information and archives it in the history tables.	30
ORDER_RELEASE_STATUS_PURGE	Purges order release status records with a quantity of 0.	30
PICKLISTPRG	Purges pick list information.	30
SHIPMENTHISTPRG	Purges shipment information from the shipment history tables.	30
SHIPMENTPRG	Purges shipment information and archives it in the history tables.	30
DRAFTORDERNOLINEPRG	Purges draft orders that do not have any order lines.	30
DRAFTORDERNOLINEHISTPRG	Purges draft orders without any order lines from the history table.	30
DRAFTORDERHISTPRG	Purges draft order information from the draft order history tables.	30
DRAFTORDERPRG	Purges draft order information and archives it in the history tables.	30

## 20.1 Modifying an Order Document Type's Purge Criteria Rule

To modify an order document type's purge criteria rule:

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Purge Criteria. The Purge Criteria List window displays in the work area.
2. Enter information in the applicable fields. Refer to [Table 20–2, "Purge Criteria Details Pop-up Window"](#) for field value descriptions.
3. Choose .

**Table 20–2 Purge Criteria Details Pop-up Window**

Field	Description
Purge Code	Identifies a purge program. This is a system defined code.
Description	Describes the type of purge.
Rollback Segment	Defines the rollback segment that should be explicitly used for the purge transaction qualified by the purge code. This is useful when there are huge logical data sets that have to be purged. This is optional and used for order related purges.

**Table 20–2 Purge Criteria Details Pop-up Window**

Field	Description
Retention Days	Enter the number of days the data is to be retained in the database (going backwards from the time the program runs). Make sure that your table size takes into account the number of retention days entered here.
Write to Log File	Check this box if you want purged data written to a log file. The log file can be backed up and used as a journal at a later date.
Log File Name	<p>Enter a log file name. The log file is created in the directory specified in the <code>yfs.purge.path</code> property. If this is not passed, it defaults to the value specified in the <code>yfs.properties</code> file. If a variable is introduced, then the <code>yfs.purge.path</code> is ignored.</p> <p>To override this property, add an entry for it in the <code>&lt;INSTALL_DIR&gt;/properties/customer_overrides.properties</code> file. For additional information about overriding properties using the <code>customer_overrides.properties</code> file, see the <i>Selling and Fulfillment Foundation: Properties Guide</i>.</p> <p>For more information about using variables for the log file directory, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i>.</p> <p>For information about filename limitations related to internationalization, see the <i>Selling and Fulfillment Foundation: Localization Guide</i>.</p>
<p><b>Additional Purge Criteria</b></p> <p>These parameters are used to override the order history purge retention days. This override is configured based on the line types within each order defined at the enterprise and document type levels.</p> <p><b>Note:</b> These additional parameters can be defined only for order history purge (ORDHISTPRG) criteria.</p>	

**Table 20–2 Purge Criteria Details Pop-up Window**

Field	Description
Line Type	Select the line types from the drop-down list. For more information about defining line types, see the <i>Sterling Distributed Order Management: Configuration Guide</i> .
Additional Retention Days	Enter the additional number of days (apart from the retention days specified by the order history purge) the data is to be retained in the database. Make sure that your table size takes into account the number of retention days entered here.  <b>Note:</b> To be considered for additional retention days, the order line must have at least some quantity that is not cancelled or shorted.

---



---

**Note:** The history purge date cannot be reset when you restore the order after it was purged. For example, if an order is purged with a history purge date of 20070801 and when the order is restored in the year 2006, the history purge date still remains as 20070801.

---



---

The following example provides an use-case of the line type purge in an order placement scenario:

**Example 20–1 Line Type Purge**

An order is placed with the following 4 order lines:

- Order Line 1 - Television
- Order Line 2 - 2 year Television service plan with Line Type as 2YR\_WARRANTY. Therefore, the additional retention days are 721.
- Order Line 3 - Stereo
- Order Line 4 - 4 year Stereo service plan with Line Type as 4YR\_WARRANTY. Therefore, the additional retention days are 1451.

Assume that the order is set to be purged after 30 days. On day 1, the order moves into a purgeable status. On day 30, the order is purged to the history table. The purge history date is set as:

Today + 10 + Maximum(721, 1491) = 1491 days, where 10 is the number of retention days for the history purge.

On day 40, the history purge agent does not pick up this order to purge, since the purge history date is set. Rather, the order is purged from the history on day 1491.

## Time-Triggered Transaction Reference

---

Selling and Fulfillment Foundation provides a collection of time-triggered transactions, which are utilities that perform a variety of individual functions, automatically and at specific time intervals.

Time-triggered transactions perform repetitive actions on a scheduled basis, typically performing database updates, raising events, or calling APIs. One type of transaction, monitors, are designed to watch for processes or circumstances that are out of bounds and then raise alerts. Often, but not always, they retrieve tasks from the task queue or work from the pipeline.

Some transactions enable you to collect statistical data regarding the application's health. This data is collected periodically, using the value specified for the `yantra.statistics.persist.interval` attribute in the `yfs.properties` file. By default, statistics collection set to `on`. To override this property, add an entry in the `<INSTALL_DIR>/properties/customer_overrides.properties` file. For additional information about overriding properties using the `customer_overrides.properties` file, see the *Selling and Fulfillment Foundation: Properties Guide*.

For more information about statistics persistence, see the *Selling and Fulfillment Foundation: Performance Management Guide*. For more information about the specific statistics parameters used, see the applicable time-triggered transactions.

The time-triggered transactions described in this appendix are unique transactions, that may or may not be document type specific. For document specific transactions, the nomenclature helps define which unique transaction it is based on: a transaction ID is in the format `Unique_Transaction_ID.Document_Type_Code`. For example, the transaction ID for Purge Return is `PURGE.0003`, indicating that it is based on the unique transaction `PURGE`, for document type `0003`, which is

Return Order. Therefore, in order to be able to configure Purge Return, you should look for the PURGE transaction ID in this appendix, which is Order Purge.

Selling and Fulfillment Foundation provides the following types of time-triggered transactions:

- [Business Process Time-Triggered Transactions](#) - responsible for processing
- [Time-Triggered Purge Transactions](#) - clear out data that may be discarded after having been processed
- [Task Queue Syncher Time-Triggered Transactions](#) - update the task queue repository with the latest list of open tasks to be performed by each transaction, based on the latest pipeline configuration.
- [Monitors](#) - watch and send alerts for processing delays and exceptions

Selling and Fulfillment Foundation tracks the following statistics for each time-triggered transaction:

- `ExecuteMessageCreated` - The number of jobs added to the JMS queue in a given time interval.
- `ExecuteMessageSuccess` - The number of jobs that were run successfully in a given time interval.
- `ExecuteMessageError` - The number of jobs that failed to run in a given time interval.
- `GetJobsProcessed` - The number of `GetJob` messages that were processed in a given time interval.

---

---

**Note:** Some of the statistics collected and tracked in Release 8.5 for time-triggered transactions, monitors, and integration and application servers may change with the next release of Selling and Fulfillment Foundation.

---

---

## A.1 Running Time-Triggered Transactions

All time-triggered transactions are threadable. This means that you can run multiple instances of a transaction within a single process. For more information about running time-triggered transactions, see the *Selling and Fulfillment Foundation: Installation Guide*. For more information about fine-tuning system performance while running them concurrently, see the *Selling and Fulfillment Foundation: Performance Management Guide*.

### A.1.1 Steps to Complete Before Scheduling Time-Triggered Transactions

Before running and scheduling a time-triggered transaction, ensure that you have completed the following:

1. Configure a JMS Connection Factory to correlate with the QCF name configured for the time-triggered transaction. The Selling and Fulfillment Foundation factory defaults include the `AGENT_QCF` as the JMS Connection Factory. For more information about configuring JMS, see the documentation for your specific application server.
2. Configure JMS Server Destinations to correlate with the group or individual name of the time-triggered transaction. The Selling and Fulfillment Foundation factory defaults include the `DefaultAgentQueue` as the server destination.

---

---

**Note:** Do not put a dot (.) in the name of a JMS Server Destination, for example, 'A.0001'. If you do, Selling and Fulfillment Foundation is unable to communicate with it.

---

---

3. Using the Applications Manager, configure each time-triggered transaction required for your business process as described in the section entitled "Defining Transactions" in the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*. Each set of time-triggered transaction criteria parameters must ensure the appropriate association of a JMS Agent Server.

## A.2 Configuring Communication Between an Agent and a JMS Server

Setting up communication between an agent (time-triggered transaction) and a remote JMS server requires that you do some prerequisite setup on your JMS system, then do some configuration within the application, which consists of the following procedures:

- If an initial context factory code for your JMS system is not provided with the application, you must create one. See "Create an Initial Context Factory Code" on page 341 for the list of codes that are provided.
- Defining the transaction details – the time-triggered transaction, or agent, must be edited to include connection information for your JMS system and the initial context factory you create. See [Section A.2.3, "Define the Transaction Information"](#).

For more information about time-triggered transactions and how they fit into the larger picture of application business process modeling, see the *Configuring Process Models* chapter. Also see the *Configuring Alert Queues* chapter for additional information about queues and agents.

### A.2.1 Prerequisites

Before starting, complete these tasks for your JMS Server. See your JMS Server documentation for more information about performing these tasks.

1. Configure the JMS Queue Connection Factory (QCF) and queues on your JMS server.
2. Configure the JNDI representation of the queues on your JMS server.

Ensure that you have the following information available from these tasks:

- JNDI name for each queue
- JNDI QCF lookup
- JMS location - the provider URL for the JMS server

Once you have completed the preceding tasks, complete the next two procedures in the order shown. These are both done in the application.

## A.2.2 Create an Initial Context Factory Code

Using an Initial Context Factory (ICF) class enables remote Java clients to connect to your application. This class is provided by the application vendor. The application uses ICF codes to identify these when setting up agents. Initial context factory codes are predefined in the application for the following JMS vendors:

- WebSphere MQ (for MQSeries accessed through a WebSphere iiop URL)
- File (for MQSeries accessed through a file URL, as with WebLogic)
- WebLogic (for WebLogic JMS)
- Jboss (for JBoss JMS)

If you are using a JMS server that is **not** in the preceding list (for example, ActiveMQ), you must create an initial context factory code for it in the application:

1. Open the Configurator. From the tree in the application rules side panel, choose System Administration > Initial Context Factory Codes. The Initial Context Factory Codes window displays in the work area.
2. Select the + icon to create a new initial context factory code. The Initial Context Factory window is displayed.
3. In the Initial Context Factory field, enter the name of the class provided by your JMS vendor. For example, for ActiveMQ, the class name is `org.apache.activemq.jndi.ActiveMQInitialContextFactory`.
4. In the Short Description field, enter a descriptive name, up to 40 characters. Make note of this name, because you will use it in the next procedure (see [Section A.2.3, "Define the Transaction Information"](#)). For ActiveMQ, enter **ActiveMQ**.
5. In the Long Description field, enter a more detailed description for the initial context factory, up to 100 characters.
6. Save the new initial context factory code and close the window.

For more information about ICFs, see *Creating an Initial Context Factory Code*.

### A.2.3 Define the Transaction Information

For the JMS server to communicate with the application, there must be a time-triggered transaction configured with the JMS server and ICF information.

1. Open the Applications Manager. From the tree in the application rules side panel, double-click Process Modeling. The Process Modeling window displays in the work area.
2. Select the desired tab, then Base Document Type, then double-click Process Type.
3. Double-click the transaction that corresponds to the agent to be run.
4. Select the Time Triggered tab.
5. Create or select an existing Agent Criteria Definition to edit.
6. The Agent Criteria Details screen is displayed. Select the Runtime Properties tab.
7. Select an existing Agent Server from the list or create your own (recommended).
8. Select an existing Alert Queue from the list or create your own.
9. In the JMS Queue Name field, enter the JNDI name for the queue that you created. See [Section A.2.1, "Prerequisites"](#).
10. Enter the desired number of threads the agent should run (recommended not to exceed 5 threads - if more than 5 are needed, start another agent in its own JVM).
11. Select the Initial Context Factory code you created. See [Section A.2.2, "Create an Initial Context Factory Code"](#).
12. In the QCF Lookup field, enter the JNDI QCF lookup for the queue that you created (this is the Queue Connection Factory created for the applicable JMS Server). See [Section A.2.1, "Prerequisites"](#).
13. Enter the Provider URL. This is the location where the JMS system resides, and is JMS vendor specific.
14. Select whether the agent should trigger itself (recommended) and at what interval (in minutes) or use an external trigger (triggeragent.sh in the `<install_dir>/install/bin` directory).

15. See Setting up the JMS Security Properties for information about setting the JMS Security option.
16. Leave the Criteria Parameters tab values at the default values.
17. Save the Agent Criteria Details and close the window.
18. Launch the agent in its own JVM by executing the `startagentserver.sh/cmd` script in the `<install_dir>/install/bin` directory.

For additional information defining transactions and about this procedure, see the sections *Defining Transactions* and *Specifying a Transaction as Time-Triggered* in the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

## A.3 Business Process Time-Triggered Transactions

This section provides an alphabetical list of all business process transactions.

---

---

**Note:** Some of the statistics collected and tracked in Release 8.5 for time-triggered transactions, monitors, and integration and application servers may change with the next release of Selling and Fulfillment Foundation.

---

---

---

---

**Note:** All Business Process Time-Triggered Transactions have a `CollectPendingJobs` criteria parameter. If this parameter is set to `N`, the agent does not collect information about the pending jobs pertaining to this monitor. This pending job information is used for monitoring the monitor in the System Management Console.

By default, `CollectPendingJobs` is set to `Y`. It can be helpful to set it to `N` if one particular time-triggered transaction is performing a significant amount of `getPendingJobs` queries, and the overhead cost is too high.

---

---

### A.3.1 Asynchronous Request Processor

This transaction completes any API request or service request in offline mode. It picks up the API messages or service messages from the YFS\_ASYNC\_REQ table and invokes the corresponding API or service. The messages can be inserted into the YFS\_ASYNC\_REQ table using the createAsyncRequest API. Some of the business transactions in the Sterling Warehouse Management System also insert the messages into the YFS\_ASYNC\_REQ table.

#### Attributes

Following are the attributes for this time-triggered transaction:

**Table A–1 Asynchronous Request Processor Attributes**

Attribute	Value
Base Transaction ID	ASYNC_REQ_PROCESSOR
Base Process Type	General
Abstract Transaction	No

#### Criteria Parameters

Following are the criteria parameters for this transaction:

**Table A–2 Asynchronous Request Processor Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Lead Days	Number of days before the present date the agent will purge the records. If left blank or specified as 0 (zero), it defaults to 30.

**Table A–2 Asynchronous Request Processor Parameters**

Parameter	Description
Maximum Error Count	Maximum number of times the record is processed if an exception is thrown. Once the number of unsuccessful attempts equals this number, that record is not processed further by the agent. If left blank or specified as 0 (zero), it defaults to 20.
Reprocess Interval In Minutes	Time in minutes after which the transaction will be reprocessed - after it has been processed and has thrown an exception.
ColonyID	Required in a multischema deployment where the YFS_ASYNC_REQ table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None

**Pending Job Count**

None

**Events Raised**

The following events are raised by this time-triggered transaction:

**Table A–3 Events Raised by the Asynchronous Request Processor**

Transaction/Event	Key Data	Data Published*	Template Support?
HAS_EXCEPTIONS	None	YCP_ASYNC_REQ_PROCESSOR.HAS_EXCEPTIONS.html	Yes
<p>*These files are located in the following directory:                      &lt;INSTALL_DIR&gt;/xapidocs/api_javadocs/XSD/HTML</p>			

### A.3.2 Case Insensitive Data Loader

The Case Insensitive Data Loader agent migrates data from columns marked CaseInsensitiveSearch to shadow columns. The agent uses the transaction criteria to identify the records that need to be updated and then converts the original column values to lowercase values in the shadow columns. For more information about enabling case insensitive searches, refer to the *Selling and Fulfillment Foundation: Extending the Database Guide*.

#### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–4 Case Insensitive Data Loader Attributes**

Attribute	Value
Base Transaction ID	DATA_LOADER
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

**Criteria Parameters**

The following are the criteria parameters for this transaction:

*Table A–5 Case Insensitive Data Loader Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. <ul style="list-style-type: none"> <li>• If left blank or the number specified is less than 10000, it defaults to 5000.</li> <li>• If the number specified is greater than 10000, then that value is used.</li> </ul>
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
TableType	Required in a multischema deployment when a table may exist in multiple schemas. Valid Values: CONFIGURATION, TRANSACTION, MASTER. If set to CONFIGURATION, the agent runs for the records associated with tables that have TableType as CONFIGURATION. If set to TRANSACTION, the agent runs for the records associated with tables that have TableType as TRANSACTION.
Table Name	Required. The table name for the records to be migrated to shadow columns.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None.

**Pending Job Count**

None.

**Events Raised**

None.

**A.3.3 Change Load Status**

This transaction is equivalent to the `changeLoadStatus()` API. For detailed information about this transaction, see the *Selling and Fulfillment Foundation: Javadocs*.

To be configured as part of your load processing pipeline, this transaction can be used whenever an automatic change in the status of a load is required. This automatic change could represent exporting load information to load planning software or transmission to the load's carrier.

**Note:** This transaction should be configured to work from the task queue.

**Attributes**

The following are the attributes for this time-triggered transaction:

*Table A-6 Change Load Status Attributes*

Attribute	Value
Base Transaction ID	CHANGE_LOAD_STATUS
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	Yes
APIs Called	<code>changeLoadStatus()</code>

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–7** *Change Load Status Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–8** *Change Load Status Statistics*

Statistic Name	Description
NumLoadsChanged	Number of loads whose status was changed.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the CurrentDate value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the `changeLoadStatus()` API in the *Selling and Fulfillment Foundation: Javadocs*.

### A.3.4 Change Shipment Status

This transaction is equivalent to the `changeShipmentStatus()` API. For detailed information about this transaction, see the *Selling and Fulfillment Foundation: Javadocs*.

To be configured as part of your shipment processing pipeline, this transaction can be used whenever an automatic change in the status of a shipment is required. For example, this automatic change could represent exporting shipment information to a warehouse management system or to transmit an Advance Shipping Notice to the buyer.

**Note:** This transaction should be configured to work from the task queue.

#### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–9** *Change Shipment Status Attributes*

Attribute	Value
Base Transaction ID	CHANGE_SHIPMENT_STATUS
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	Yes
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–10** *Change Shipment Status Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–11** *Create Chained Order Statistics*

Statistic Name	Description
NumShipmentsChanged	Number of shipments whose status was changed.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the `changeShipmentStatus()` API in the *Selling and Fulfillment Foundation: Javadocs*.

### A.3.5 Close Delivery Plan

To boost system performance, this transaction serves as a temporary purge until the Delivery Plan Purge deletes delivery plan-related data (see [Section A.4.3.5, "Delivery Plan Purge"](#)).

This transaction picks all delivery plans that do not have any of their loads or shipments still open and marks the `deliveryplan_closed_flag='Y'`. This flag indicates no further operations are possible on the plan.

This transaction corresponds to the base transaction close delivery plan (CLOSE\_DELIVERY\_PLAN) in the load pipeline.

Any enterprise using the Console must schedule purge jobs.

#### Attributes

The following are the attributes for this time-triggered transaction:

*Table A–12 Close Delivery Plan Attributes*

Attribute	Value
Base Transaction ID	CLOSE_DELIVERY_PLAN
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None

#### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table A–13 Close Delivery Plan Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.

**Table A–13 Close Delivery Plan Criteria Parameters**

Parameter	Description
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

The following statistics are tracked for this transaction:

**Table A–14 Close Delivery Plan Statistics**

Statistic Name	Description
NumDeliveryPlansClosed	Number of delivery plans closed.

**Pending Job Count**

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

**Events Raised**

The following events are raised by this time-triggered transaction:

**Table A–15 Events Raised by Close Delivery Plan Transaction**

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	delivery_ plan_dbd.txt	YDM_CLOSE_ DELIVERY_ PLAN.ON_ SUCCESS.xml	Yes

However, note that the template name would read <TransactionId>.ON\_SUCCESS.xml.

### A.3.6 Close Load

To boost system performance, this transaction serves as a temporary purge until the Load Purge deletes load-related data (see [Section A.4.3.13, "Load Purge"](#)).

This transaction corresponds to the base transaction Close Load (CLOSE\_LOAD) in the load pipeline.

If you use the Load processing pipeline, you must schedule this transaction. Only closed loads are picked up by the purge transaction. Therefore, it is required that this transaction be made part of the pipeline and scheduled to run at the end of the day.

**Note:** This transaction should be made part of the pipeline. In addition, it should be configured to work from the task queue.

#### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–16 Close Load Attributes**

Attribute	Value
Base Transaction ID	CLOSE_LOAD
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–17 Close Load Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–18 Close Load Statistics**

Statistic Name	Description
NumLoadsClosed	Number of loads closed.

### Pending Job Count

For this transaction the pending job count is the number of open delivery plans, which are not associated to any open loads and open shipments.

### Events Raised

The following events are raised by this time-triggered transaction:

**Table A–19 Events Raised by the Close Load Transaction**

Transaction/Event	Data Published	Template Support?
ON_SUCCESS	YDM_CLOSE_LOAD_PLAN.ON_SUCCESS.xml	Yes

However, note that the template name would read `<TransactionId>.ON_SUCCESS.xml`.

### A.3.7 Close Manifest

This time-triggered transaction sets the manifest's `MANIFEST_CLOSED_FLAG` flag to 'Y' and updates the manifest status to `CLOSED`. This time-triggered transaction confirms all the shipments that are pending confirmation, and closes the manifest.

---

---

**Note:** If the Close Manifest Agent is triggered without any criteria, it closes all the candidate manifests across all ShipNodes.

---

---

The `yfs.closemanifest.online` property in the `yfs.properties_ysc_ext.in` file is used to set this time-triggered transaction to work in online or offline mode.

- **Online mode:** In the online mode, the close manifest transaction runs as usual, confirming all shipments in the manifest and then closing the manifest.
- **Offline mode:** In the offline mode, the close manifest transaction triggers an agent and changes the manifest status to 'Closure Requested'. When the agent runs, it confirms either each shipment of the manifest, or closes the manifest, in an execution call.

The mode of operation (online or offline) is decided on the basis of the value specified for the `yfs.closemanifest.online` property in the `yfs.properties_ysc_ext.in` file. To override this property, add an entry for it in the `<INSTALL_DIR>/properties/customer_overrides.properties` file. For additional information about overriding properties using the `customer_overrides.properties` file, see the *Selling and Fulfillment Foundation: Properties Guide*.

The default out-of-the-box shipped property causes the Close Manifest transaction to run in online mode.

---

---

**Note:** In instances where the Close Manifest transaction is run in offline mode, ensure that all Agent Criteria defined for the transaction are configured properly.

---

---

**Attributes**

The following are the attributes for this time-triggered transaction:

*Table A–20 Close Manifest Attributes*

Attribute	Value
Base Transaction ID	CLOSE_MANIFEST
Base Document Type	General
Base Process Type	Manifesting
Abstract Transaction	No
APIs Called	confirmShipment()

**Criteria Parameters**

The following are the criteria parameters for this transaction:

*Table A–21 Close Manifest Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by WMS time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ShipNode	Optional. Ship node for which the Close Manifest needs to be run. If not passed, then all ship nodes are monitored.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following are statistics are tracked for this transaction:

**Table A–22** *Close Manifest Statistics*

Statistic Name	Description
NumShipmentsConfirmed	Number of shipments confirmed.
NumManifestsClosed	Number of manifests closed.
NumManifestsErrored	Number of manifests errored.
NumShipmentsErrored	Number of shipments errored.

## Pending Job Count

For this transaction the pending job count is the sum of open manifests and shipments belonging to manifests (with MANIFEST\_STATUS='1200').

## Events Raised

The following events are raised by this time-triggered transaction:

**Table A–23** *Events Raised by the Close Manifest Transaction*

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	manifest_ dbd.txt	YDM_CLOSE_ MANIFEST.ON_ SUCCESS.xml	Yes

## A.3.8 Close Order

This time-triggered transaction sets the order's ORDER\_CLOSED flag to 'Y' and raises the ON\_SUCCESS event. These actions are only performed when the entire ORDER\_QTY for all the order lines reaches the configured pickup status. If an order has ORDER\_CLOSED set to 'Y', it is not picked up for monitoring.

**Note:** The Close Order agent must be configured along with the Purge transaction in the pipeline.

---



---

**Note:** The Close Order agent must be run before running the Monitor agent in order to avoid alerts getting raised for cancelled orders.

---



---



---



---

**Note:** Many of this transaction's elements and attributes are template-driven. Refer to the XML for element level details.

---



---

### Attributes

The following are the attributes for this time-triggered transaction:

*Table A-24 Close Order Attributes*

Attribute	Value
Base Transaction ID	CLOSE_ORDER
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–25 Close Order Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–26 Close Order Statistics**

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumOrdersClosed	Number of orders closed.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table, if tasks on hold are not ready to be processed.

### Events Raised

The following events are raised by this time-triggered transaction:

*Table A–27 Events Raised by the Close Order Transaction*

Transaction/Event	Data Published	Template Support?
ON_SUCCESS	YFS_CLOSE_ORDER.ON_SUCCESS.xml	Yes

### A.3.9 Close Receipts

This time-triggered transaction closes receipts using the receiving rule specified.

#### Attributes

The following are the attributes for this time-triggered transaction:

*Table A–28 Close Receipts Attributes*

Attribute	Value
Base Transaction ID	RECEIPT_COMPLETE
Base Document Type	Order
Base Process Type	Receipt (Purchase Order Receipt, Return Receipt, Transfer Order Receipt, Sales Order Receipt)
Abstract Transaction	No
APIs Called	None
User Exits Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–29 Close Receipts Criteria Parameters**

Parameter	Description
Action	Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Enterprise for which the Close Receipts needs to be run. If not passed, then all enterprises are monitored.
Node	Mandatory. Node for which the Close Receipts needs to be run.
AgentCriteriaGroup	Used to classify nodes. This value can be accepted by WMS time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–30 Close Receipts Statistics**

Statistic Name	Description
NumReceiptsClosed	Number of receipts closed.

### Pending Job Count

For this transaction the pending job count is the number of Receipts that can be closed (with OPEN\_RECEIPT\_FLAG='Y').

### Events Raised

The following events are raised by this time-triggered transaction:

**Table A-31 Events Raised by the Close Receipts Transaction**

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	receipt_dbd.txt	YFS_RECEIPT_COMPLETE.ON_SUCCESS.xml	Yes

**Troubleshooting Tip:** When multiple inbound shipments are received into the same location, and the inventory received is not license plated, an error message, "There is no inventory for put away at the SourceLocation" displays. The solution to this problem lies in one of these steps:

- Manually create move requests for receipts that you already received. For more information about creating move requests, refer to the *Sterling Warehouse Management System: User Guide*.
- For receipts that are expected to be received, ensure that the inventory is license plated and that you don't receive inbound shipments and inventory for put away into the same location.

### A.3.10 Close Shipment

To boost system performance, this transaction serves as a temporary purge until the Shipment Purge deletes all shipment-related data (see [Section A.4.3.31, "Shipment Purge"](#)).

This transaction picks all shipments eligible to be closed, based on the pipeline configuration for pickup for transaction CLOSE\_SHIPMENT, and marks the shipment\_closed\_flag='Y'. This flag indicates no further operations are possible on the shipment. There is no status change

involved. This transaction can be configured in the pipeline so that it picks up either Shipped or Delivered status.

This transaction corresponds to the base transaction close shipment (CLOSE\_SHIPMENT) in the shipment pipeline.

**Note:** This transaction should be made part of the pipeline. In addition, it should be configured to work from the task queue.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–32 Close Shipment Attributes**

Attribute	Value
Base Transaction ID	CLOSE_SHIPMENT
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–33 Close Shipment Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

**Table A–33 Close Shipment Criteria Parameters**

Parameter	Description
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following are statistics are tracked for this transaction:

**Table A–34 Close Shipment Statistics**

Statistic Name	Description
NumShipmentsClosed	Number of shipments closed.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

### Events Raised

The following events are raised by this time-triggered transaction:

**Table A–35 Events Raised by the Close Shipment Transaction**

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	shipment_ dbd.txt	YDM_CLOSE_ SHIPMENT.ON_ SUCCESS.xml	Yes

## A.3.11 Collect Shipment Statistics

Collect Shipment Statistics is a time-triggered transaction which can be invoked to process the shipments, and generate information required for the Daily Shipment Report.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–36 Collect Shipment Statistics Attributes**

Attribute	Value
Transaction Name	Collect Shipment Statistics
Transaction ID	COLLECT_STATISTICS
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None
User Exits Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–37 Collect Shipment Statistics Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Node	Required. The warehouse management ship node for which records are being processed.

**Table A–37 Collect Shipment Statistics Criteria Parameters**

Parameter	Description
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by WMS time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

The following statistics are tracked for this transaction:

**Table A–38 Statistics for Collect Shipment Statistics**

Statistic Name	Description
NumDaysStatisticsCollected	Number of days for which shipment statistics have been collected.

**Pending Job Count**

For this transaction the pending job count is the number of days for which shipment statistics needs to be collected. The number of days is calculated as the difference (in days) between the current date and the last date when shipment statistics was collected.

**Events Raised**

The following events are raised by this time-triggered transaction:

**Table A–39 Events Raised by the Collect Shipment Statistics Transaction**

Transaction/Event	Data Published	Template Support?
ON_SUCCESS	YDM_COLLECT_STATISTICS.ON_SUCCESS.xml	No

### A.3.12 Consolidate Additional Inventory

The Consolidate Additional Inventory time-triggered transaction consolidates supply and demand from the YFS\_INVENTORY\_SUPPLY\_ADDNL and YFS\_INVENTORY\_DEMAND\_ADDNL tables. Consolidation is performed by summing up the quantities of additional supply and demand in the YFS\_INVENTORY\_SUPPLY and YFS\_INVENTORY\_DEMAND tables.

If no matching supply or demand is found, a new supply or demand is created with the sum quantity of the changes in the YFS\_INVENTORY\_SUPPLY\_ADDNL and YFS\_INVENTORY\_DEMAND\_ADDNL tables. After the changes are applied, the records in the YFS\_INVENTORY\_SUPPLY\_ADDNL and YFS\_INVENTORY\_DEMAND\_ADDNL tables that were used in the consolidation process, are deleted.

#### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-40 Consolidate Additional Inventory Attributes**

Attribute	Value
Base Transaction ID	CONSOLIDATE_ADDNL_INV
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the parameters for this transaction:

**Table A-41 Consolidate Additional Inventory Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of inventory item records (whose additional supplies and demands are consolidated_ to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where the YFS_INVENTORY_SUPPLY_ADDNL and YFS_INVENTORY_DEMAND_ADDNL tables may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A-42 Consolidate Additional Inventory Statistics**

Statistic Name	Description
NumInventorySupplyAddnlsProcessed	Number of additional inventory supply records processed in the consolidation.
NumInventoryDemandAddnlsProcessed	Number of additional inventory demand records processed in the consolidation.
NumInventoryDemandDtlsProcessed	Number of inventory demand details records processed in the consolidation.

### Pending Job Count

For this transaction the pending job count is the number of distinct inventory items in the YFS\_INVENTORY\_SUPPLY\_ADDNL and YFS\_INVENTORY\_DEMAND\_ADDNL tables, multiplied by two.

### Events Raised

None.

### A.3.13 Consolidate To Shipment

This is a task queue based transaction in the order pipeline that corresponds to base transaction CONSOLIDATE\_TO\_SHIPMENT. This transaction finds a shipment into which a given order release can be included. If it finds an existing shipment, it calls `changeShipment()` API. Otherwise, it calls the `createShipment()` API.

To find the existing shipments it matches ShipNode, ShipTo Address, SellerOrganizationCode, Carrier, DocumentType and so forth, of the Order Release with that of existing shipments. List of attributes it matches is actually based on Document Template for Document Type of the Order.

This transaction is applicable only to the shipments in one of the following Statuses:

- Shipment Created
- ESP Check Required
- On ESP Hold
- Released from ESP Hold
- Released For Routing
- Awaiting Routing
- Shipment Routing
- Sent To Node
- Shipment Being Picked

**Troubleshooting Tip:** To successfully consolidate an Order Release to an existing shipment, the Add Line and related modification types on shipment in its current status should be allowed.

For more information, see the details provided under the `createShipment()`, `changeShipment()`, and `releaseOrder()` APIs in the *Selling and Fulfillment Foundation: Javadocs*.

**Note:** This transaction is a part of the Order Fulfillment pipeline. In addition, it should be configured to work from the task queue.

**Note:** Order releases with GIFT\_FLAG set to Y are never consolidated with any other release.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-43 Consolidate to Shipment Attributes**

Attribute	Value
Base Transaction ID	CONSOLIDATE_TO_SHIPMENT
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No

**Table A–43 Consolidate to Shipment Attributes**

Attribute	Value
APIs Called	<code>createShipment()</code> and <code>changeShipment()</code>
User Exits	<ul style="list-style-type: none"> <li>• It calls <code>beforeConsolidateToShipment</code> in <code>com.yantra.ydm.japi.ue</code>.</li> <li>• <code>YDMBeforeConsolidateToShipment</code> for each release before it begins processing.</li> <li>• After it finds the shipments, it calls <code>determineShipmentToConsolidateWith</code> in <code>com.yantra.ydm.japi.ue.YDMdetermineShipmentToConsolidateWith</code>. For more information, see the <i>Selling and Fulfillment Foundation: Javadocs</i>.</li> </ul>

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–44 Consolidate to Shipment Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to <code>Get</code> , the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

### Pending Job Count

**Table A-45 Consolidate to Shipment Statistics**

Statistic Name	Description
NumOrderReleasesConsolidated	Number of order releases consolidated.

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

## Events Raised

The following events are raised by this time-triggered transaction:

**Table A-46 Events Raised by the Consolidate to Shipment Transaction**

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	shipment_dbd.txt	YDM_CONSOLIDATE_TO_SHIPMENT.ON_SUCCESS.xml	Yes

**Note:** This transaction also raises events as specified under the `createShipment()` and `changeShipment()` APIs in the *Selling and Fulfillment Foundation: Javadocs*.

However, note that the template name would read `<TransactionId>.ON_SUCCESS.xml`. The XML and DTD depicted above represent the output that the abstract transaction CONSOLIDATE\_TO\_SHIPMENT transaction is capable of generating.

### A.3.14 Create Catalog Index

The Create Catalog Index transaction builds the Apache Lucene index file that is used by catalog search. This index file enhances search performance by storing denormalized item data that has been extracted from the Selling and Fulfillment Foundation database.

The Create Catalog Index transaction can be configured to perform the following tasks:

- Run either a scheduled index build or user-initiated index build
- Build either a full or incremental index file
- Activate the index file

#### The Index Building Process

The Create Catalog Index transaction provides an agent for index building. Index building is a multi-thread process in which the index building agent extracts item and item-related information from the active selling catalog in the Selling and Fulfillment Foundation database and writes this information to multiple files. The files identify the item data that should be included in the final index. After the agent finishes writing the files, it merges them into the final index file.

The multi-thread process provides the advantage of parallel processing. Large amounts of database data are segmented and processed simultaneously, which is faster and more scalable than sequentially processing one long file.

When writing information to multiple files, the index building agent performs the following tasks for each item before looping to the next item:

- Queries the Selling and Fulfillment Foundation database for data about the item.
- Uses information from the XML configuration file and extension file to determine the data that be retrieved from the query.
- Retrieves relevant data from the Selling and Fulfillment Foundation database.
- Creates a Lucene document for the item.

After the transaction creates a Lucene document for each item, the transaction writes the documents to the index file based on the organization and the organization’s locales.

**Attributes**

Table A–47 displays the attributes for the Create Catalog Index transaction.

**Table A–47 Create Catalog Index**

Attribute	Value
Base Transaction ID	Create_Catalog_Index
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YCMParseAssetUE

**Criteria Parameters**

Table A–48 displays the criteria parameters for the Create Catalog Index transaction.

**Table A–48 Create Catalog Index**

Parameter	Description
Organization Code	Required. The organization code of the catalog organization or subcatalog organization that maintains the search index.
Number of Messages	Required. Number of messages to use when building the index file.  Selling and Fulfillment Foundation processes only one message per thread. For example, if Number of Messages is set to 10 and Threads is set to 3, Selling and Fulfillment Foundation processes only 3 messages at a time. For more information about fine-tuning system performance, see the <i>Selling and Fulfillment Foundation: Performance Management Guide</i> .

*Table A-48 Create Catalog Index*

Parameter	Description
Incremental Build	<p>Y or N.</p> <p>Y to rebuild the existing index file. If you specify Y, Selling and Fulfillment Foundation rebuilds the index based on the last successful index build.</p> <p>N to build a full index file.</p> <p>This parameter is ignored for user-initiated index builds. However, if scheduled builds are configured, ensure that you specify whether you want a full or incremental index build.</p>
Category Domain	<p>Optional. The catalog from which the index is built. The active selling catalog of the catalog organization or subcatalog organization is the default. If scheduled builds are configured, ensure that you specify a catalog.</p>
Auto Activate	<p>Y or N. Optional.</p> <p>Y to activate the index after building the index file.</p> <p>The default is N.</p>

**Table A–48 Create Catalog Index**

Parameter	Description
Auto Insert Search Index Trigger	<p>Y or N. Optional.</p> <p>Y to enable scheduled builds of the catalog index file. The agent refers to information stored in the YFS_SEARCH_INDEX_TRIGGER table to determine when to run the scheduled index build. Specify the type of index build, whether full or incremental, in the agent criteria.</p> <p>N to enable user-initiated builds of the catalog index file. The agent continuously queries the YFS_SEARCH_INDEX_TRIGGER table to determine whether an index build is indicated. If a user starts an index build from the Business Center, the status setting in the table changes to Scheduled, triggering the agent to build the index. The user specifies the type of index build, whether full or incremental, from the Business Center.</p> <p>After a scheduled or user-initiated build runs, the user can activate the index from the Business Center. Alternatively, the agent can be configured to automatically activate the index.</p> <p>To allow both scheduled and user-initiated index builds, configure the transaction to include two instances of the agent. Configure one instance to trigger user-initiated builds and the second instance to trigger scheduled index builds.</p>
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

Table A–49 shows the statistics that are tracked for the Create Catalog Index transaction.

**Table A–49 Create Catalog Index**

Statistic Name	Description
SearchIndicesBuilt	Number of search indices that have been built.

**Pending Job Count**

None.

**Events Raised**

None.

**A.3.15 Create Chained Order**

This transaction creates one or more chained orders from an order whose `OrderHeaderKey` is stored in the task queue object. Chainable lines of the order can also be added to existing chained orders, instead of creating new chained orders with these lines. The existing chained orders must be identified by the `determineChainedOrderForConsolidation` user exit. If the user exit is not implemented, or if the user exit returns a blank document, one or more new chained orders are created.

For more information about the creation of chained orders, see the information provided under the `createChainedOrder()` API and the `YFSDetermineChainedOrderForConsolidation` user exit in the *Selling and Fulfillment Foundation: Javadocs*.

This transaction should be invoked after order scheduling.

**Attributes**

The following are the attributes for this time-triggered transaction:

*Table A–50 Create Chained Order Attributes*

Attribute	Value
Base Transaction ID	CHAINED_ORDER_CREATE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	createChainedOrder()

**Criteria Parameters**

The following are the criteria parameters for this transaction:

*Table A–51 Create Chained Order Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Note:** If there are 2 orders being processed and the first order creates a  
*Table A-52 Create Chained Order Statistics*

Statistic Name	Description
NumOrdersProcessed	Number of orders processed for creating chained order.
NumOrdersCreated	Number of chained orders created.

chained order, the DetermineChainedOrderForConsolidation user exit causes the lines of the 2nd order to be added to the first order. The number of chained orders created is counted as 2.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

### Events Raised

This transaction raises events as specified under the createChainedOrder() API in the *Selling and Fulfillment Foundation: Javadocs*.

## A.3.16 Create Derived Order

This transaction creates one or more derived orders from an order whose OrderHeaderKey is stored in the task queue object. For existing derived orders, you can add derivable lines or create new derived orders with these lines. The existing derived orders must be identified by the determineDerivedOrderForConsolidation user exit. If the user exit is not implemented or if the user exit returns a null document, new derived orders are created. For more information about the creation of derived orders, see the details provided under the createDerivedOrder() API and YFSDetermineDerivedOrderForConsolidation user exit in the *Selling and Fulfillment Foundation: Javadocs*.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–53** *Create Derived Order Attributes*

Attribute	Value
Base Transaction ID	DERIVED_ORDER_CREATE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	<code>createDerivedOrder()</code>

**Note:** The TransactionKey posted in the task queue object must be an instance of the Abstract Transaction DERIVED\_ORDER\_CREATE for the ProcessType associated with the Order. Otherwise, an exception is thrown.

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–54** *Create Derived Order Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–55** *Create Derived Order Statistics*

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumOrdersCreated	Number of derived orders created.

**Note:** If there are 2 orders being processed and the first order creates a derived order, the `DetermineChainedOrderForConsolidation` user exit causes the lines of the 2nd order to be added to the first order. The number of derived orders created is counted as 2.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the `AVAILABLE_DATE` value less than or equal to (`<=`) the current date value in the `YFS_Task_Q` table.

### Events Raised

This transaction raises events as specified under the `createDerivedOrder()` API in the *Selling and Fulfillment Foundation: Javadocs*.

## A.3.17 Create Order Invoice

This transaction creates one or more invoices from an order whose `OrderHeaderKey` is stored in a task queue object. The `createOrderInvoice()` API is called for the `OrderHeaderKey`.

Configure this transaction in the pipeline only after all processing that can impact quantity or price has been completed. Post invoice creation, the line quantity cannot be reduced below the invoiced quantity.

**Note:** Both the Create Order Invoice and Create Shipment Invoice transactions can create invoices for an Order. When configuring your pipeline, ensure that only *one* of these two transactions is configured to create invoices for a particular order line. For more information, see [Section A.3.18, "Create Shipment Invoice"](#).

**Attributes**

The following are the attributes for this time-triggered transaction:

**Table A–56 Create Order Invoice Attributes**

Attribute	Value
Base Transaction ID	CREATE_ORDER_INVOICE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	createOrderInvoice()

**Criteria Parameters**

The following are the criteria parameters for this transaction:

**Table A–57 Create Order Invoice Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

*Table A–58 Create Order Invoice Statistics*

Statistic Name	Description
NumOrderInvoicesCreated	Number of order invoices created.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the `AVAILABLE_DATE` value less than or equal to (`<=`) the current date value in the `YFS_Task_Q` table.

### Events Raised

This transaction raises events as specified under the `createOrderInvoice()` API in the *Selling and Fulfillment Foundation: Javadocs*.

## A.3.18 Create Shipment Invoice

Invoicing is mandatory if an order requires payment processing. Invoicing occurs if the following conditions are met:

- Invoicing is enabled at the document parameter level.
- The Seller requires payment processing.

This transaction creates one or more invoices for the shipment whose `ShipmentKey` is stored in the task queue object. The `createShipmentInvoice()` API is called for the `ShipmentHeaderKey`.

This transaction should be configured in the shipment pipeline only after the shipment has reached a shipped status.

**Note:** Both the Create Order Invoice and Create Shipment Invoice can create invoices for an order. When configuring your pipeline, ensure that only *one* of these two transactions is configured to create invoices for a particular order line. See [Section A.3.17, "Create Order Invoice"](#).

**Attributes**

The following are the attributes for this time-triggered transaction:

**Table A–59 Create Shipment Invoice Attributes**

Attribute	Value
Base Transaction ID	CREATE_SHIPMENT_INVOICE
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	Yes
APIs Called	createShipmentInvoice()

**Criteria Parameters**

The following are the criteria parameters for this transaction:

**Table A–60 Create Shipment Invoice Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–61 Create Shipment Invoice Statistics**

Statistic Name	Description
NumShipmentInvoicesCreated	Number of shipment invoices created.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the `AVAILABLE_DATE` value less than or equal to (`<=`) the current date value in the `YFS_Task_Q` table.

## Events Raised

This transaction raises events as specified under the `createShipmentInvoice()` API in the *Selling and Fulfillment Foundation: Javadocs*.

## A.3.19 ESP Evaluator

The ESP Evaluator time-triggered transaction verifies whether a shipment meets certain economic shipping parameters (ESP). ESP can be configured either for buyer or enterprise, with the freight terms on the shipment determining which one is used.

If the configuration is defined to hold shipment for ESP, the shipment when created is held for ESP (with status *On ESP Hold*). This task queue based time-triggered transaction evaluates the shipment for ESP, and passes it on to the next step in the shipment pipeline if the criteria (weight and volume limits, plus maximum days of hold up) are met. The shipment status is now set to *Released from ESP hold*, and routing processing begins.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–62** *ESP Evaluator Attributes*

Attribute	Value
Base Transaction ID	ESP_EVALUATOR.0001
Base Document Type	Order
Base Process Type	Outbound Shipment
Abstract Transaction	No
APIs Called	None
User Exits Called	getNodeMinimumNotificationTime

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–63** *ESP Evaluator Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
EnterpriseCode	Optional. Enterprise for which the ESP Evaluator needs to be run. If not passed, then all enterprises are monitored.
Number of Records to Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
Node	Required. The warehouse management ship node for which records are being processed.

**Table A–63 ESP Evaluator Criteria Parameters**

Parameter	Description
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by WMS time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None.

**Pending Job Count**

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

**Events Raised**

The following events are raised by this time-triggered transaction:

**Table A–64 Events Raised by ESP Evaluator Transaction**

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	shipment_dbd.txt	ESP_EVALUATOR.ON_SUCCESS.xml	Yes

### A.3.20 Item Based Allocation

The Item Based Allocation transaction allocates unpromised and promised demands of existing orders to more suitable supplies based upon inventory items and nodes which have been triggered for the Item Based Allocation process in the YFS\_IBA\_TRIGGER table.

The Item Based Allocation agent obtains and processes all Item Based Allocation triggers from the YFS\_IBA\_TRIGGER table that meet the following conditions:

- IBA\_RUN\_REQUIRED = "Y"
- LAST\_IBA\_PROCESSED\_TS was 'x' hours before current time, where 'x' is from the 'Item Based Allocation Agent Execution Interval (in hours)' rule in the Installation rules. For more information about installation rules, refer to the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*. This rule is used to indicate the interval that the Item Based Allocation agent should not reprocess the triggers in the YFS\_IBA\_TRIGGER table, which were processed earlier. This prevents the IBA agent from over-processing the item and node combination in the given time interval to avoid any high loads on the system.
- PROCESSING\_BY\_AGENT="N" or PROCESS\_OVER\_BY\_TS is before the current timestamp. The PROCESSING\_BY\_AGENT field is used to prevent the picking up of the IBA trigger which is being processed by another instance of the agent.

If InventoryOrganizationCode is specified in the agent criteria, only the IBA trigger with inventory items of that inventory organization is retrieved.

For each triggered item and node combination, the agent finds all of the applicable order lines or order line reservations that contain the item and node and tries to move their unpromised and promised demands to more suitable available supplies based on user-configured IBA selection rules or FIFO (First-In-First-Out) IBA selection rules.

Selling and Fulfillment Foundation creates new positive order line reservations with the matched supply's first ship date and negative order line reservations for the existing demand ship date. Once all orders are processed, they are placed on hold to be rescheduled if changes are detected in the order line reservations.

---

---

**Note:** The following configuration is required for the Item Based Allocation process:

- The Use Item Based Allocation rule needs to be enabled.
  - Item and node need to have Item Based Allocation Allowed enabled.
  - A hold type is required to be set up for the change order line reservations modification type so that the order can be placed on hold for rescheduling. For more information, refer to the *Selling and Fulfillment Foundation: Javadocs*.
- 
- 

---

---

**Note:** The 'When a line is backordered, backorder against the highest priority ship node' rule should be checked in order to reallocate backordered demand. For more information, see the Fulfillment Rules section in the *Sterling Distributed Order Management: Configuration Guide*.

---

---

Before processing the Item Based Allocation logic, the Item Based Allocation agent updates the following fields on the Item Based Allocation trigger:

- PROCESSING\_BY\_AGENT = "Y". This indicates that an instance of the agent is currently processing this trigger.
- PROCESS\_OVER\_BY\_TS = current time + 1 hr. This indicates the expected time that the agent should finish with processing this IBA trigger. One hour is the fixed window and cannot be changed. Selling and Fulfillment Foundation treats the PROCESSING\_BY\_AGENT flag as "N" regardless of the actual value when current timestamp is after this timestamp.
- IBA\_RUN\_REQUIRED = "N". This resets the IBA\_RUN\_REQUIRED flag back to "N".

### **Obtaining a List of Demands Based on Applicable Order Release Statuses and Order Line Reservations to be Allocated**

A list of demands is derived from applicable order release statuses and order line reservations, which have the item and node in the IBA trigger. The following types of demands are retrieved:

- Demands of chained orders
- Demands of orders with chained order already created
- Demands of orders with procurement node but chained order creation is not yet created
- Demands of orders without procurement node
- Demands from order line reservations

The demand quantity is derived based on the order release status quantity with the status from the Status Inventory Type configuration that has a demand type, which considers the supply type with 'Use Consider Demand Type for Item Based Allocation' enabled. For more information, refer to the *Sterling Global Inventory Visibility: Configuration Guide*.

### **Obtaining a List of Available Supplies for Allocation**

Selling and Fulfillment Foundation obtains the available supply based on the availability of the item at the node by ignoring unpromised and promised demands. If the inventory organization maintains its inventory externally, the external availability can be read by the `YFSGetExternalInventoryUE` user exit. Only the availability of supplies that consider the 'Demand Type Look for Availability during Item Based Allocation' are used in the allocation logic. For more information, refer to the *Sterling Global Inventory Visibility: Configuration Guide*.

---

---

**Note:** Allocated demands should be matched with the same supplies as "Demand to look for during release".

---

---

### Matching Demands Against Supplies in FIFO (First-In-First-Out) Order

Selling and Fulfillment Foundation sorts the list of available supplies in the order of the first shippable date (ETA), and matches the obtained list of demands using the top-down logic (unlike the normal matching logic for obtaining availability, where matches are based on the closest ETA). Demands are allocated in the following orders:

- Demands of chained orders - first based on user-configured sequencing rules, and then in ascending order of order creation date. (These types of demands are matched based on the closest ETA to avoid any changes in the chained orders).
- Demands of orders with a chained order already created - first based on user-configured sequencing rules, then in ascending order of product availability date. (These types of demands are matched based on the closest ETA to avoid any changes in the orders).
- Demands of orders for which procurement node and chained order creation is imminent (within the advanced notification time window) - first based on user-configured sequencing rules, then in order of order creation date.
- Demands of orders without a procurement node and within the release window (advanced notification time window) - first based on user-configured sequencing rules, then in order of order creation date.
- Demands from order line reservations on the order lines in the order of requested reservation date, and left-over demands (outside of the advanced notification time window) of orders with or without a procurement node, first based on user-configured sequencing rules and then in the order of order creation date.
- Demands from inventory reservations in the order of ship date.

Notice that different types of demands are given different priorities based on their significance. The demands of chained orders or orders related to chained orders are treated with a higher priority than the demands of normal orders. Furthermore, the demands with a ship date within the advanced notification time window also have a higher priority than the demands with a date outside of the advanced notification time window.

### Updating Order Reservations for the Matched Demands

After matching the available supply and demand in user-configured sequencing and then in FIFO order, the system builds up a list of order line reservation changes and inventory demand changes (corresponding to the order line reservation changes) and summarize them to optimize the number of order reservation updates and inventory updates. Negative order line reservations are added for the matched demands. Positive order reservations are added for the matched demands with the product availability date set to the matched supplies' first ship date.

After the Item Based Allocation agent completes its tasks for an Item Based Allocation trigger, it updates the fields of the trigger with the following values:

- IBA\_REQUIRED = "N"
- LAST\_IBA\_PROCESSED\_TS = current timestamp.
- PROCESS\_OVER\_BY\_TS = current timestamp.
- PROCESSING\_BY\_AGENT = "N"

The Item Based Allocation agent should be used in conjunction with the rescheduling process as the rescheduling process reschedules the affected orders by utilizing the order line reservations created by the Item Based Allocation process.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-65** *Item Based Allocation Attributes*

Attribute	Value
Base Transaction ID	ITEM_BASED_ALLOCATION
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	changeOrder – for updating the order line reservations created as part of the Item Based Allocation process.
User Exits Called	None

**Criteria Parameters**

The following are the criteria parameters for this transaction:

**Table A-66 Item Based Allocation Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
InventoryOrganizationCode	The inventory organization code of the inventory items which are processed by the Item Based Allocation agent. If provided, only the IBA triggers with the inventory item that belongs to this inventory organization are processed.
ColonyID	Required in a multischema deployment where the YFS_IBA_TRIGGER table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–67** *Item Based Allocation Statistics*

Statistic Name	Description
NumOrdersProcessed	Number of orders processed by the Item Based Allocation agent.
NumOrdersRequiredReschedule	Number of orders required rescheduling as the result of Item Based Allocation process.

### Pending Job Count

None.

### Events Raised

This transaction raises events as specified under the changeOrder API in the *Selling and Fulfillment Foundation: Javadocs*.

## A.3.21 Mark Load as Trailer Loaded

This is a time-triggered transaction which works on “Load pipeline”.

This time-triggered transaction gets records from the Task Q. This transaction is used to mark the load as trailer loaded when all containers for the load are on the trailer.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–68** *Mark Load As Trailer Loaded Attributes*

Attribute	Value
Base Transaction ID	MARK_AS_TRAILER_LOADED
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No

**Table A–68 Mark Load As Trailer Loaded Attributes**

Attribute	Value
APIs Called	None
User Exits Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–69 Mark Load As Trailer Loaded Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ReprocessInterval	Optional. Reprocess Interval is the time taken to reprocess the load.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–70 Mark Load As Trailer Loaded Statistics**

Statistic Name	Description
NumLoadsChanged	Number of trailer loads changed.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

**Events Raised**

None.

**A.3.22 Match Inventory**

Match Inventory processes all pending records in the YFS\_INVENTORY\_SHIPMENT table. Pending records have a smaller number in POSTED\_QUANTITY than in QUANTITY.

Each pending record is matched against the receipt records in YFS\_INVENTORY\_RECEIPT table by applying the inventory cost determination logic. The unit cost at which the sales and receipt data are matched is also posted in YFS\_INVENTORY\_MATCH table.

Use this transaction if any of the configured ship nodes maintain inventory cost.

**Attributes**

The following are the attributes for this time-triggered transaction:

*Table A-71 Match Inventory Attributes*

Attribute	Value
Base Transaction ID	INVENTORY_MATCH
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A-72 Match Inventory Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
InventoryOrganizationCode	Optional. Valid inventory owner organization. Organization to process in this run. If not passed, all inventory organizations are processed.
CutOffDate	Optional. If passed, records are matched up to this date. Defaults to all unmatched records in Database.
ColonyID	Required in a multischema deployment where the YFS_INVENTORY_SHIPMENT, YFS_INVENTORY_RECEIPT, and the YFS_INVENTORY_MATCH tables may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A-73 Match Inventory Statistics**

Statistic Name	Description
NumInventoryShipmentsProcessed	Number of inventory shipments processed.
NumInventoryMatchesInserted	Number of inventory matches inserted.

### Pending Job Count

For this transaction the pending job count is the number of distinct inventory items that exist in the YFS\_INVENTORY\_SHIPMENT table where the QUANTITY value is not equal to the POSTED\_QUANTITY value.

### Events Raised

None.

## A.3.23 Payment Collection

This transaction requests credit validation for orders that are pending authorization or charging.

Use this transaction for creating authorization and charge requests.

**Note:** This transaction works in combination with the Payment Execution transaction. Although this transaction can run independent of that transaction, authorization and collection occurs *only* after the Payment Execution dependencies are met. For more details, see [Section A.3.24, "Payment Execution"](#).

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–74** *Payment Collection Attributes for Sales Orders*

Attribute	Value
Base Transaction ID	PAYMENT_COLLECTION
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	requestCollection()

**Table A-75 Payment Collection Attributes for Return Orders**

Attribute	Value
Base Transaction ID	PAYMENT_COLLECTION.0003
Base Document Type	Order
Base Process Type	Reverse Logistics
Abstract Transaction	No
APIs Called	requestCollection()

**Criteria Parameters**

The following are the criteria parameters for this transaction:

**Table A-76 Payment Collection Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. The enterprise for which the transaction needs to be run. If left blank, orders for all enterprises are processed. If specified, only orders for that enterprise are processed.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table A-77 Payment Collection Statistics*

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumChargeReqsCreated	Number of charge requests created.
NumAuthorizationReqsCreated	Number of authorization requests created.

## Pending Job Count

For this transaction the pending job count is the number of orders in the appropriate payment statuses with the value of the AUTHORIZATION\_EXPIRATION\_DATE is less than or equal to ( $\leq$ ) the current date. The appropriate payment statuses for such orders are:

- AWAIT\_PAY\_INFO
- AWAIT\_AUTH
- REQUESTED\_AUTH
- REQUEST\_CHARGE
- AUTHORIZED, INVOICED
- PAID
- RELEASE\_HOLD
- FAILED\_AUTH
- FAILED\_CHARGE
- VERIFY
- FAILED

## Events Raised

The following events are raised by this time-triggered transaction:

**Table A-78 Events Raised by the Payment Collection Transaction**

Transaction/Event	Key Data	Data Published	Template Support?
INCOMPLETE_PAYMENT_INFORMATION	modifyOrder_dbd.txt	YFS_PAYMENT_COLLECTION.INCOMPLETE_PAYMENT_INFORMATION.xml	Yes
PAYMENT_STATUS	YFS_PAYMENT_COLLECTION.PAYMENT_STATUS_dtd.txt	YFS_PAYMENT_COLLECTION.PAYMENT_STATUS.xml	Yes
REQUEST_PAYMENT_STATUS		YFS_PAYMENT_COLLECTION.REQUEST_PAYMENT_STATUS.xml	Yes
ON_LIABILITY_TRANSFER	modifyOrder_dbd.txt	YFS_PAYMENT_COLLECTION.ON_LIABILITY_TRANSFER.xml	Yes
ON_INVOICE_COLLECTION	order_dbd/txt	YFS_CREATE_ORDER_INVOICE.ON_INVOICE_COLLECTION.xml	Yes

### A.3.24 Payment Execution

This transaction processes all requests that are pending authorization and charging.

**Note:** Use this time-triggered transaction for processing all authorization and charge requests.

This transaction requires interfacing with a product that provides financial services.

**Attributes**

The following are the attributes for this time-triggered transaction:

**Table A–79 Payment Execution Attributes for Sales Orders**

Attribute	Value
Base Transaction ID	PAYMENT_EXECUTION
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	executeCollection()
User Exits Called	collectionCreditCard, collectionOthers, collectionCustomerAcct

**Table A–80 Payment Execution Attributes for Return Orders**

Attribute	Value
Base Transaction ID	PAYMENT_EXECUTION.0003
Base Document Type	Order
Base Process Type	Reverse Logistics
Abstract Transaction	No
APIs Called	executeCollection()
User Exits Called	collectionCreditCard, collectionOthers, collectionCustomerAcct

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–81 Payment Execution Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ChargeType	Type of credit card process. Valid values are: <ul style="list-style-type: none"> <li>• AUTHORIZATION - Validates the credit card account</li> <li>• CHARGE - Applies the charge to the credit card</li> </ul>
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–82 Payment Execution Statistics**

Statistic Name	Description
NumAuthTransProcessed	Number of authorization transaction processed.
NumAuthTransSuccessfullyProcessed	Number of successful returns from user exit for authorization transaction processed.
NumChargeTransProcessed	Number of charge transaction processed.
NumChargeTransSuccessfullyProcessed	Number of successful returns from user exit for charge transaction processed.

**Table A–82 Payment Execution Statistics**

Statistic Name	Description
NumCollectionValidations	Number of successful returns from the invoked validate collection user exits.
NumCreditCardCollections	Number of credit card collections.
NumCustomerAccountCollections	Number of successful returns from the customer account collection user exits.
NumOtherCollections	Number of successful returns from the other collection user exits.

### Pending Job Count

For this transaction the pending job count is the number of open charge and authorization transactions.

### Events Raised

The following events are raised by this time-triggered transaction:

**Table A–83 Events Raised by Payment Execution Transaction**

Transaction/Event	Key Data	Data Published	Template Support?
CHARGE_FAILED	modifyOrder dbd.txt	PAYMENT_EXECUTION_ CHARGE_FAILED_ dbd.txt	No

This transaction raises events as specified under the `executeCollection()` API in the *Selling and Fulfillment Foundation: Javadocs*.

## A.3.25 Post Inventory Match

This transaction processes all open records in `YFS_INVENTORY_MATCH` table and posts the records to a financial system. An open record in the `YFS_INVENTORY_MATCH` table has the status of 01. After posting, the status is changed to 02.

Use this transaction if any of the configured ship nodes maintain inventory cost.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–84 Post Inventory Match Attributes**

Attribute	Value
Base Transaction ID	POST_INVENTORY_MATCH
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–85 Post Inventory Match Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where the YFS_INVENTORY_MATCH table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

*Table A–86 Post Inventory Match Statistics*

Statistic Name	Description
NumInventoryMatchPosted	Number of inventory match records posted.

### Pending Job Count

For this transaction the pending job count is the number of inventory matches with an open status.

### Events Raised

The following events are raised by this time-triggered transaction:

*Table A–87 Events Raised by the Post Inventory Match Transaction*

Transaction/Event	Key Data	Data Published	Template Support?
POST_INVENTORY_MATCH	POST_INVENTORY_MATCH_dbd.txt	YFS_postInventoryMatch_output.xml	No

## A.3.26 Process Order Hold Type

You can create a time-triggered transaction, derived from the PROCESS\_ORDER\_HOLD\_TYPE abstract transaction. It can be configured as the processing transaction for one or more hold types. If an order is associated with a hold type that has a transaction configured as the processing transaction, a record is created in the YFS\_TASK\_Q table for processing that transaction.

When the processing transaction is triggered, it checks the hold types that it can process based on the hold type configuration. If no hold types can be processed, the YFS\_TASK\_Q record is deleted. If some hold types can be processed, the processOrderHoldType user exit is invoked with the list of hold types to be processed. The processOrderHoldType user exit returns the list of hold types that can be removed from the order.

The transaction then modifies the order and updates the order hold type list based on the output returned by the processOrderHoldType user exit. If now no hold types can be processed, the YFS\_TASK\_Q record is deleted. If some hold types can still be processed, YFS\_TASK\_Q is updated with the next available date.

You can also call the processOrderHoldType user exit to add new hold types or change the status of a hold type that is already applied to an order. For more information about the processOrderHoldType user exit, see the *Selling and Fulfillment Foundation: Javadocs*.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–88** *Process Order Hold Type Attributes*

Attribute	Value
Base Transaction ID	PROCESS_ORDER_HOLD_TYPE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	changeOrder

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–89** *Process Order Hold Type Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

**Table A–89 Process Order Hold Type Parameters**

Parameter	Description
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where the YFS_TASK_Q table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None.

**Pending Job Count**

None

**Events Raised**

The following events are raised by this time-triggered transaction:

**Table A–90 Events Raised by Process Order Hold Type Transaction**

Transaction/Event	Raised when...	Key Data	Data Published	Template Support?
ON_SUCCESS	On success	modifyOrder_dbd.txt	YFS_ORDER_CHANGE.ON_SUCCESS.xml	Yes *
ON_HOLD_TYPE_STATUS_CHANGE	The status of a hold type is changed.	modifyOrder_dbd.txt	YFS_ON_HOLD_TYPE_STATUS_CHANGE.xml	Yes

**Table A–90 Events Raised by Process Order Hold Type Transaction**

Transaction/Event	Raised when...	Key Data	Data Published	Template Support?
ON_ORDER_LINE_HOLD_TYPE_STATUS_CHANGE	The status of a hold type is changed.	modifyOrder_dbd.txt	YFS_ON_ORDER_LINE_HOLD_TYPE_STATUS_CHANGE.xml	Yes
<p><b>* Note:</b> Some of the elements and attributes are not template-driven. Refer to the xml for element level details.</p>				

### A.3.27 Process Work Order Hold Type

This time-triggered transaction is identical to the [Process Order Hold Type](#) transaction, but it is used for work orders instead.

#### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–91 Process Work Order Hold Type Attributes**

Attribute	Value
Base Transaction ID	PROCESS_WO_ORDER_HOLD_TYPE
Base Document Type	Work Order
Base Process Type	VAS Process
Abstract Transaction	Yes
APIs Called	modifyWorkOrder

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table A-92 Process Work Order Hold Type Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

None.

### Pending Job Count

None

### Events Raised

The following events are raised by this time-triggered transaction:

**Table A-93 Events Raised by Process Work Order Hold Type Transaction**

Transaction/Event	Raised when...	Key Data	Data Published	Template Support?
ON_SUCCESS	On success	workOrder_dbd.txt	VAS_MODIFY_WORK_ORDER.ON_SUCCESS.xml	Yes *
ON_HOLD_TYPE_STATUS_CHANGE	The status of a hold type is changed.	workOrder_dbd.txt	VAS_ON_HOLD_TYPE_STATUS_CHANGE.xml	Yes

**\* Note:** Some of the elements and attributes are not template driven. Refer to the xml for elements level details.

### A.3.28 Publish Negotiation Results

This transaction publishes the negotiated terms to the order.

Use this transaction in environments where an order must go through a negotiation phase.

**Note:** This transaction needs to be run after negotiation is completed.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-94 Publish Negotiation Results Attributes**

Attribute	Value
Base Transaction ID	PUBLISH_ORD_NEGOTIATION
Base Document Type	Order
Base Process Type	Order Negotiation

**Table A–94 Publish Negotiation Results Attributes**

Attribute	Value
Abstract Transaction	No
APIs Called	None

**Criteria Parameters**

The following are the criteria parameters for this transaction:

**Table A–95 Publish Negotiation Results Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

The following statistics are tracked for this transaction:

**Pending Job Count**

**Table A–96 Publish Negotiation Results Statistics**

Statistic Name	Description
NumNegotiationsProcessed	Number of negotiations processed.
NumNegotiationsPublished	Number of negotiations published.

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE

value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

### Events Raised

The following events are raised by this time-triggered transaction:

**Table A-97 Events Raised by Publish Negotiation Results Transaction**

Base Transaction	Raised when...	Key Data	Data Published	Template Support?
PUBLISH_ORD_NEGOTIATION/ON_SUCCESS	On success	Negotiation_dbd.txt	YCP_getNegotiationDetails_output.xml	Yes *
RECEIVE_ORD_NEGOTIATION/ON_SUCCESS	On success, when DocumentType is 0001, EntityType is ORDER.	Number of concurrent time-triggered transactions running.	receiveOrderNegotiation_dbd.txt	No
<p><b>* Note:</b> Template used for this event is the same template used by the <code>getNegotiationDetails()</code> API to form the output XML.</p>				

### A.3.29 Release

This transaction releases orders to specific ship nodes, making sure that the scheduled ship nodes have enough inventory to process the order.

This transaction should be invoked after the scheduling process.

For more details, see the information provided under the `releaseOrder()` API in the *Selling and Fulfillment Foundation: Javadocs*.

**Important:** Sterling Commerce recommends that if you run the combined 'Schedule and Release' agent, you do not also run the individual Schedule or the individual Release agents.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-98 Release Attributes**

Attribute	Value
Base Transaction ID	RELEASE
Base Document Type	Order
Base Process Type	Order Fulfillment
APIs Called	releaseOrder ( )

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A-99 Release Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
IgnoreReleaseDate	Optional. Determines whether the schedule process should ignore line release date criteria. Valid values are: <ul style="list-style-type: none"> <li>• Y - Releases line quantities regardless of release date criteria</li> <li>• N - Default value. Releases line quantities only after release date criteria have been met.</li> </ul>
CheckInventory	Optional. Determine whether inventory should be checked. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Inventory needs to be checked.</li> <li>• N - Inventory does not need to be checked.</li> </ul>

**Table A–99 Release Criteria Parameters**

Parameter	Description
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

The following statistics are tracked for this transaction:

**Table A–100 Release Criteria Statistics**

Statistic Name	Description
NumFutureDateFailures	Number of orders did not attempt to release because of future date failures.
NumOrdersAttempted	Number of orders attempted to release.
NumOrdersCannotBeProcessedFailures	Number of orders did not attempt to release because of cannot be processed failures.
NumOrdersProcessed	Number of orders processed.
NumOrdersReleased	Number of orders released.
NumOrdersBackordered	Number of orders backordered.
NumOrderLinesReleased	Number of order lines released.
NumOrderLinesBackordered	Number of order lines backordered.
NumReleasesCreated	Number of order releases created.
NumOrdersCannotBeProcessedFailures	Number of orders that were not released due to process failure.

**Note:** If the release process results in splitting of an order line, NumOrderLinesReleased, NumOrderLinesBackordered, and NumOfReleasesCreated may result in more than one count.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the `AVAILABLE_DATE` value less than or equal to (`<=`) the current date value in the `YFS_Task_Q` table, if tasks on hold are not ready to be processed.

### Events Raised

This transaction raises events as specified under the `releaseOrder()` API in the *Selling and Fulfillment Foundation: Javadocs*.

## A.3.30 Route Shipment

This time-triggered transaction is used to route shipments and belongs to the Outbound Shipment pipeline. It assigns the Carrier and Carrier Service codes for the shipment based on the Routing Guide configured.

The Route Shipment transaction either includes shipments in an existing load or creates a new load and includes the shipments in it.

Shipments can be consolidated to a load, only if the following conditions are met:

- Expected Ship Date - The expected ship date of the shipments must be less than or equal to the must ship before date of the load.
- Expected Load Departure Date - The expected load departure date must be less than or equal to the must ship before date of the shipments in the load.

The must ship before date is a date computed for the load, based on all shipments present in the load. For example, if a load has three shipments with their must ship before dates as 12.22.2005, 12.12.2005, and 12.19.2005 respectively, then the must ship before date of the load is computed as 12.12.2005, as it is the earliest of the three dates.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–101** *Route Shipment*

Attribute	Value
Base Transaction ID	ROUTE_SHIPMENT.0001
Base Document Type	Order
Base Process Type	ORDER_DELIVERY
Abstract Transaction	No
APIs Called	None
User Exits Called	com.yantra.ydm.japi.ue.YDMOverrideDetermineRoutingUE com.yantra.ydm.japi.ue.YDMBeforeDetermineRoutingUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–102** *Route Shipment Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.

**Table A–102 Route Shipment Criteria Parameters**

Parameter	Description
ColonyID	Required in a multischema deployment where YFS_SHIPMENT table may exist in multiple schemas. Runs the agent for the colony.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.

**Statistics Tracked**

The following statistics are tracked for this transaction:

**Table A–103 Route Shipment Statistics**

Statistic Name	Description
NumRouted	Number of shipments routed.

**Pending Job Count**

For this transaction the pending job count is the number of records representing the unheld orders that are available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

**Events Raised**

The following events are raised by this time-triggered transaction:

**Table A–104 Events Raised by the Route Shipment Transaction**

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	shipment_dbd.txt	YDM_ROUTE_SHIPMENT.ON_SUCCESS.xml	Yes
ON_FAILURE	shipment_dbd.txt	YDM_ROUTE_SHIPMENT.ON_FAILURE.xml	Yes

However, note that the template name would read <TransactionId>.ON\_SUCCESS.xml.

### A.3.31 Schedule

This transaction schedules orders to specific ship nodes making sure that the scheduled ship nodes have enough inventory to process the order.

Run this transaction after order creation.

**Important:** Sterling Commerce recommends that if you run the combined 'Schedule and Release' agent, you do not also run the individual Schedule or the individual Release agents.

#### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–105 Schedule Attributes**

Attribute	Value
Base Transaction ID	SCHEDULE
Base Document Type	Order
Base Process Type	Order Fulfillment
APIs Called	scheduleOrder ( )

#### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–106 Schedule Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

**Table A–106 Schedule Criteria Parameters**

Parameter	Description
OptimizationType	Optional. Determines the optimization rules to apply to the scheduling process. Valid values are: <ul style="list-style-type: none"> <li>• 01 - Optimize on date (Default)</li> <li>• 02 - Optimize on ship node priority</li> <li>• 03 - Optimize on number of shipments</li> </ul>
OrderFilter	Optional. Determines the types of orders to filter. Possible values are: <ul style="list-style-type: none"> <li>• A - All orders (Default)</li> <li>• B - Backorders only</li> <li>• N - New orders only</li> </ul>
ScheduleAndRelease	Optional. Notify the schedule process to release all releasable line quantities. Valid values are: <ul style="list-style-type: none"> <li>• Y - Releases successfully scheduled line quantities.</li> <li>• N - Default value. Only schedules line quantities.</li> </ul> <p><b>Note:</b> Enabling this parameter does not validate hold types configured for the release transaction.</p>
IgnoreReleaseDate	Optional. Determines whether the schedule process should ignore line release date criteria. Valid values are: <ul style="list-style-type: none"> <li>• Y - Releases line quantities regardless of release date criteria.</li> <li>• N - Releases lines quantities only after release date criteria have been met. Default.</li> </ul>
Next Task Queue Interval	Not used. This agent updates a failed task so that it is suspended for the back order retry interval setup in the appropriately scheduled rule.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A-107 Schedule Statistics**

Statistic Name	Description
NumFutureDateFailures	<p>Number of orders that Selling and Fulfillment Foundation did not attempt to schedule because of future date failures.</p> <p>Failures can be caused by any of the following:</p> <ul style="list-style-type: none"> <li>• If the OrderFilter is "B" (Backorders Only) and there are no backordered or unscheduled lines.</li> <li>• If the OrderFilter is "N" (New orders Only) and there are some backordered or unscheduled lines.</li> <li>• If order has order lines within only backordered or unscheduled status and the status modify timestamp is after the current time - the back order wait period specified in the scheduling rule.</li> </ul>
NumOrdersAttempted	<p>Number of orders attempted to schedule. This statistic does not include the values for NumFutureDateFailures and NumOrdersCannotBeProcessedFailures statistics.</p>
NumOrderLinesReleased	<p>Number of order lines that have been released.</p>

Table A-107 Schedule Statistics

Statistic Name	Description
NumOrdersCannotBeProcessedFailures	<p>Number of orders that Selling and Fulfillment Foundation did not attempt to schedule because of cannot be processed failures.</p> <p>Failures can be caused by any of the following:</p> <ul style="list-style-type: none"> <li>• The result of the YFSCheckOrderBeforeProcessingUE user exit returns as false.</li> <li>• The Order has the HoldFlag attribute set to 'Y'.</li> <li>• The Order has the SaleVoided attribute set to 'Y'.</li> <li>• The Order does not have PaymentStatus as AUTHORIZED, INVOICED, PAID, nor NOT_APPLICABLE.</li> </ul>
NumOrdersCreated	Number of orders created. This also includes the number of procurement orders created.
NumOrderLinesCreated	Number of order lines created.
NumOrdersProcessed	Number of orders processed.
NumOrdersScheduled	<p>Number of orders that have at least one line that was scheduled.</p> <p><b>Note:</b> This includes scheduled lines in any status except BACKORDER.</p>
NumOrdersProcOrdersCreated	Number of procurement orders created.
NumWorkOrdersCreated	Number of work orders created.
NumOrdersBackordered	Number of orders backordered.
NumOrderLinesScheduled	Number of order lines scheduled.
NumOrderLinesBackordered	Number of order lines backordered.
NumReleasesCreated	Number of order releases created.

### Pending Job Count

For this transaction the pending job count is the number of records representing the unheld orders that are available to be processed by the transaction with the `AVAILABLE_DATE` value less than or equal to (`<=`) the current date value in the `YFS_Task_Q` table, if tasks on hold are not ready to be processed.

### Events Raised

This transaction raises events as specified under the `scheduleOrder()` API in the *Selling and Fulfillment Foundation: Javadocs*.

### Providing Oracle Hints

You can provide Oracle Hints to increase the performance of the `scheduleOrder` agent. The two hints that can be provided for each criteria ID of the `scheduleOrder` agent are the Outer Hint and the Inner Hint. The Outer Hint is always used for the `YFS_TASK_Q` table. The Inner Hint is used for the `YFS_ORDER_HEADER` table only if the earlier hold functionality is used; otherwise, the Inner Hint is used for the `YFS_ORDER_RELEASE_STATUS` table.

Insert the following entries in the `yfs.properties` file in order to enable Oracle Hints:

1. Edit the `<YANTRA_HOME>/Applications/Foundation/resources/yfs.properties` file.
2. Insert `yfs.<agent_criteria_id>.getjobs.hint.outer=/*+ parallel(YFS_TASK_Q 8) full(yfs_task_q) */`  
  
Insert `yfs.<agent_criteria_id>.getjobs.hint.inner=/*+ NL_SJ */`

## A.3.32 Send Invoice

This transaction publishes invoice data that can be directed to an external accounts receivable system.

In environments that require an interface with accounts receivable systems, this transaction needs to be scheduled. This transaction raises

an event for an invoice based on the following configuration at the following times in the order lifecycle:

- Publish invoice at shipment creation - This implies that your accounts payable system takes care of payment collection. Invoices can be published as soon as they are created.
- Publish invoice after payment collection - This implies that the Console take care of the payment collection. When payment is in the AT\_COLLECT status and the payment is not from an external system, an invoice is published only if the entire payment amount is collected. If the payment is in the AT\_CREATE status or the payment is from an external system, the invoice is published unconditionally.

**Note:** Many of this transaction's elements and attributes are template driven. Refer to the XML for element level details.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-108** *Send Invoice Attributes*

Attribute	Value
Base Transaction ID	SEND_INVOICE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	getOrderInvoiceDetails()

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–109 Send Invoice Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–110 Send Invoice Statistics**

Statistic Name	Description
NumInvoicesSent	Number of invoices sent.

### Pending Job Count

For this transaction the pending job count is the number of order invoices in created (“00”) status.

### Events Raised

The following events are raised by this time-triggered transaction:

**Table A–111 Events Raised by the Send Invoice Transaction**

Transaction/Event	Key Data	Data Published	Template Support?
PUBLISH_INVOICE_DETAIL	modifyOrder_dbd.txt and sendInvoice_dbd.txt	YFS_getOrderInvoiceDetails_output.xml	Yes

Additional events may be raised by the `getOrderInvoiceDetails()` API. For detailed information about the events, see the details provided under this API in the *Selling and Fulfillment Foundation: Javadocs*.

### A.3.33 Send Item Changes

In integrated environments, this transaction publishes item data changes that are directed to an external system.

When item changes occur in Selling and Fulfillment Foundation, they need to be communicated to the external system.

The business process may require the synchronization of items all at once in a batch. For example, at the end of each business day, the `sendItemChanges` agent can be configured to synchronize items based on the synchronization logic. This transaction retrieves all items that are not logical kit or dynamic physical kit items and whose `SyncTS` is null or `MaxModifyTS` is greater than the `SyncTS`.

**Note:** The `MaxModifyTS` of an item is updated with the current timestamp whenever an item is modified. The transaction then retrieves detailed information about those items and raises the `ON_SUCCESS` event. This event should be configured to invoke the Send Item Changes action.

For more information about how this integration is implemented, see the *Selling and Fulfillment Foundation: Integration Guide*.

#### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-112** *Send Item Changes Attributes*

Attribute	Value
Base Transaction ID	SEND_ITEM_CHANGES
Base Document Type	None
Base Process Type	General

*Table A–112 Send Item Changes Attributes*

Attribute	Value
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table A–113 Send Item Changes Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Organization Code	Optional. The organization from which items are synchronized. This field is blank by default.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

None.

### Pending Job Count

For this transaction the pending job count is the number of items requiring synchronization. This is determined for product items that are not logical kit or dynamic physical kit items and whose SyncTS is null or MaxModifyTS is greater than the SyncTS.

## Events Raised

The following events are raised by this time-triggered transaction:

**Table A–114** Events Raised by the Send Item Changes Transaction

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	None	YCM_SEND_ITEM_CHANGES_ON_SUCCESS.XML	Yes

### A.3.34 Send Customer Changes

In integrated environments, this transaction publishes customer data changes that are directed to an external system.

When customer changes occur in Selling and Fulfillment Foundation, they need to be communicated to the external system.

The business process may require the synchronization of customers all at once in a batch. For example, at the end of each business day, the `sendItemChanges` agent can be configured to synchronize items based on the synchronization logic. This transaction retrieves all customers that are consumers, have a user ID present, and are required to synchronize. This transaction can also be used to complete the initial synchronization of users between the two systems. For example, if an external system is already in place, and Selling and Fulfillment Foundation is then added, the `SendCustomerChanges` agent synchronizes the users from the external system.

The `sendCustomerChanges` agent also serves as a backup mechanism. If a customer synchronization event fails, the agent automatically retries the synchronization after a specified amount of time.

---

**Note:** The `MaxModifyTS` of an customer is updated with the current timestamp whenever an customer is modified, whenever `syncTS` is less than `MaxModifyTS`, or when `syncTS` is null. The transaction then retrieves detailed information about those customers and raises the `ON_SUCCESS` event. This event should be configured to invoke the `Send Customer Changes` action.

---

For more information about how this integration is implemented, see the *Selling and Fulfillment Foundation: Integration Guide*.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–115** *Send Customer Changes Attributes*

Attribute	Value
Base Transaction ID	SEND_CUSTOMER_CHANGES
Base Document Type	None
Base Process Type	General
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–116** *Send Customer Changes Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Organization Code	Optional. The organization from which customers are synchronized. This field is blank by default.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

None.

### Pending Job Count

For this transaction the pending job count is the number of customers requiring synchronization. This is determined for customers that are consumers, have a user ID present, and are required to synchronize.

### Events Raised

The following events are raised by this time-triggered transaction:

**Table A-117** *Events Raised by the Send Customer Changes Transaction*

Transaction/Event	Key Data	Data Published	Template Support?
SEND_CUSTOMER_CHANGES.ON_SUCCESS	None	YSC_SEND_CUSTOMER_CHANGES.ON_SUCCESS.XML	Yes

### A.3.35 Send Order

This transaction tries to raise the ON\_SUCCESS event for an order whose OrderHeaderKey is stored in the task queue object. The event is raised only if all of the order lines of the order reach particular status(es) completely. That is, the entire ORDERED\_QTY of each line must be in the particular status(es). In addition to raising the event, the line statuses are also changed to the drop statuses, corresponding to the pickup statuses. The SendOrder transaction, derived from the abstract transaction SEND\_ORDER, should have the event, pickup, and drop statuses configured. For more information, see the details provided under the `sendOrder()` API in the *Selling and Fulfillment Foundation: Javadocs*.

If an order needs to be communicated to a third party, use this transaction.

---



---

**Note:** The TransactionKey posted in the task object must be an instance of the Abstract Transaction SEND\_ORDER for the ProcessType associated with the Order. Otherwise, an exception is thrown.

---



---

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-118 Send Order Attributes**

Attribute	Value
Base Transaction ID	SEND_ORDER
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	sendOrder ( )

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A-119 Send Order Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

None.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE

value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

### Events Raised

This transaction raises events as specified under the `sendOrder()` API in the *Selling and Fulfillment Foundation: Javadocs*.

## A.3.36 Send Release

The Send Release Agent dispatches releases to ship nodes.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–120 Send Release Attributes**

Attribute	Value
Transaction Name	Send Release
Transaction ID	SHIP_ADVICE
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	com.yantra.yfs.agent.YFSWMSShipAdviceAgent

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–121 Send Release Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

**Table A–121 Send Release Criteria Parameters**

Parameter	Description
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–122 Send Release Statistics**

Statistic Name	Description
NumReleasesProcessed	Number of order releases processed.
NumReleasesSent	Number of order releases sent.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

### Events Raised

The following events are raised by this time-triggered transaction:

**Table A–123 Events Raised by the Send Release Transaction**

Transaction/Event	Data Published
PUBLISH_SHIP_ADVICE	YFS_publishShipAdvice_output.xml

### A.3.37 Start Order Negotiation

This transaction creates the negotiations for orders that are configured to go through the negotiation process.

Use this transaction in environments where an Order needs to go through a Negotiation phase before it is released.

#### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–124 Start Order Negotiation Attributes**

Attribute	Value
Base Transaction ID	START_ORD_NEGOTIATION
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	createNegotiation()
User Exits Called	YCPBeforeCreateNegotiationUE, YCPGetNegotiationNoUE

#### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–125 Start Order Negotiation Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.

**Table A–125 Start Order Negotiation Criteria Parameters**

Parameter	Description
Node	Required. The warehouse management ship node for which records are being processed.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–126 Start Order Negotiation Statistics**

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumNegotiationsCreated	Number of negotiations created.

### Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

### Events Raised

This transaction raises events as specified under the `createNegotiation()` API in the *Selling and Fulfillment Foundation: Javadocs*.

## A.3.38 Synchronize Colony Map

The Colony Map Synchronizer agent inserts or updates colony mappings of organizations and users in the PLT\_COLONY\_MAP table. When you run the agent for the first time, it populates this table, which is a necessary step in upgrading to multischema mode after installing or upgrading Selling and Fulfillment Foundation.

For more information about upgrading to multischema mode, see the *Selling and Fulfillment Foundation: Multitenant Enterprise Guide*.

### Attributes

The following are attributes for this time-triggered transaction:

**Table A-127 Colony Map Synchronizer Attributes**

Attribute	Value
Base Transaction ID	COLONY_MAP_SYNC
Base Process Type	General
Abstract Transaction	No

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A-128 Colony Map Synchronizer Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records to Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	The colony to be synchronized.  Initially, you must run the agent on the DEFAULT colony provided by the Selling and Fulfillment Foundation installation so that it populates the PLT_COLONY_MAP table. After this, you can run the agent on another ColonyID.
InsertDefaultMappings	If set to Y, users for which the colony cannot be determined will be mapped to the colony for which the Colony Map Synchronizer agent is run.

### Statistics Tracked

None.

### Pending Job Count

None.

**Events Raised**

None.

**Tables Purged**

None.

**A.3.39 Update Best Match Region**

The Update Best Match Region transaction manages the YFS\_REGION\_BEST\_MATCH table, which is used by Data Warehouse Analytics to report best match region data. The best match region is defined by the following five address attributes in person info records:

- ADDRESS\_LINE6
- CITY
- STATE
- SHORT\_ZIP\_CODE
- COUNTRY

The agent for the Update Best Match Region transaction runs in two modes that allow you to set up and update the YFS\_REGION\_BEST\_MATCH table.

**Attributes**

The following are the attributes for this time-triggered transaction:

*Table A-129 Update Best Match Region Attributes*

Attribute	Value
Base Transaction ID	UPDATE_BEST_MATCH_REGION
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YSCGetShortZipCode UE

**Criteria Parameters**

The following are the criteria parameters for this transaction:

**Table A–130 Update Best Match Region Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If UpdateOnly = N, only distinct records are returned per agent call. If left blank, it defaults to 1000.
TableType	<p>Required in a multischema deployment when YFS_Person_Info table may exist in multiple schemas.</p> <p>Valid Values: CONFIGURATION, TRANSACTION, MASTER.</p> <p>If set to CONFIGURATION, the agent runs for the YFS_Person_Info records associated with tables that have TableType as CONFIGURATION; for example, YFS_Organization, YFS_Ship_Node, and so forth.</p> <p>If set to TRANSACTION, the agent runs for the YFS_Person_Info records associated with tables that have TableType as TRANSACTION; for example, YFS_Order_Header, YFS_Shipment, and so forth.</p> <p>Note that the agent would run for all TableTypes that exist in the same schema as the one passed. For example, if set to TRANSACTION, the agent would also run for YFS_Person_Info records associated with tables that have TableType as MASTER, since they reside in the same schema.</p>
ColonyID	Required in a multi schema deployment where the YFS_PERSON_INFO table may exist in multiple schemas. Runs the agent for the colony.

**Table A-130 Update Best Match Region Criteria Parameters**

Parameter	Description
UpdateOnly	<p>Mode in which to run. Valid values are:</p> <ul style="list-style-type: none"> <li>• N - Default value. Adds records from the YFS_PERSON_INFO table to the YFS_REGION_BEST_MATCH table and populates the region key in the YFS_BEST_MATCH table. To perform the initial setup of Best Match Region for Analytics, set UpdateOnly to N.</li> <li>• Y - Update mode. Updates region keys based on addresses in YFS_REGION_BEST_MATCH. After performing the initial setup of Best Match Region for Analytics, set this value to Y to specify update mode.</li> </ul>
LastPersonInfoKey	<p>Optional. If UpdateOnly is set to N, LastPersonInfoKey determines the first person info record to populate. If no key is specified, the value defaults to Null.</p>
LastRegionBestMatchKey	<p>Optional. If UpdateOnly is set to Y, LastRegionBestMatchKey determines the first region best match key to update. If no key is specified, the value defaults to Null.</p>

**Statistics Tracked**

None.

**Pending Job Count**

None.

**Events Raised**

None.

**Tables Purged**

None.

### A.3.40 PopulateOwnershipTransferSummary

This method updates the YFS\_OWNERSHIP\_TRANSFER\_SUMMARY table.

This transaction updates the YFS\_OWNERSHIP\_TRANSFER\_SUMMARY table by checking the records in YFS\_INV\_OWN\_TRANSFER\_RCD table.

It also updates the IS\_STATISTICS\_UPDATED to 'Y' in YFS\_INV\_OWN\_TRANSFER\_RCD table after the record has been used by the transaction.

#### Attributes

Following are the attributes for this time-triggered transaction:

**Table A-131** YFSPopulateOwnershipTransfer Attributes

Attribute	Value
Base Transaction ID	POPULATE_OWN_TRANS_SUMM
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

#### Criteria Parameters

Following are the criteria parameters for this transaction:

**Table A-132** YFSPopulateOwnershipTransfer Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, which is the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where the YFS_OWNERSHIP_TRANSFER_SUMMARY and YFS_INV_OWN_TRANSFER_RCD tables may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

None

### Pending Job Count

None

### Events Raised

None

## A.4 Time-Triggered Purge Transactions

There are several transactions that you can use to purge your database tables at specific time intervals.

Purge transactions determine when a table should be purged by determining the current date and subtracting the retention days specified by the purge. If the timestamp on the table is less than or equal to (current day - retention days) the table is purged.

---

---

**Note:** In some cases, a purge may look at another field other than the table's timestamp. These are pointed out in the documentation.

---

---

---

---

**Note:** When an entity is being purged, the related or dependent information that is present in other tables should be taken into consideration for purging along with it. For example, if a sales order with live shipments is being purged, any cross reference to that order is not accurate in the Order Shipment Console.

---

---

---

---

**Note:** Some of the statistics collected and tracked in Release 8.5 for time-triggered transactions, monitors, and integration and application servers may change with the next release of Selling and Fulfillment Foundation.

---

---

---

---

**Note:** All Time-Triggered Purge Transactions have a `CollectPendingJobs` criteria parameter. If this parameter is set to `N`, the agent does not collect information on the pending jobs for that time-triggered transaction. This pending job information is used for monitoring the monitor in the System Management Console.

By default, `CollectPendingJobs` is set to `Y`. It can be helpful to set it to `N` if one particular time-triggered transaction is performing a significant amount of `getPendingJobs` queries, and the overhead cost is too high.

---

---

### A.4.1 Purge Strategy

The following recommendations should be taken into consideration when planning a purge strategy for each purge transaction:

- Test purges by setting `Live` to 'N'.
- Turn on logging to test what is purged.
- Set up purge traces in the System Management Console and analyze the information.

### A.4.2 Configuring Purge Transaction Log Files

You can configure purges to write log files to a directory you specify. Each time you run a particular purge, new data is appended to this file. If no file exists, one is created.

To specify a purge log file directory:

1. Configure the `yfs.purge.path` property in the `<INSTALL_DIR>/properties/customer_overrides.properties` file. For example, on UNIX you might specify the log files to be written to the `/app/yfs/logs/purges` directory.

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

2. Run the `<INSTALL_DIR>/bin/setupfiles.sh` script on UNIX, or the `<INSTALL_DIR>/bin/setupfiles.cmd` script on Windows.

### A.4.3 Available Purges

This section contains details of all purge transactions in alphabetical order. The time-triggered purge transactions are:

- [Alert Purge](#)
- [Capacity Purge](#)
- [Draft Order History Purge](#)
- [Draft Order Purge](#)
- [Delivery Plan Purge](#)
- [Export Table Purge](#)
- [Import Table Purge](#)
- [Inventory Audit Purge](#)
- [Inventory Purge](#)
- [Inventory Supply Temp Purge](#)
- [Item Audit Purge](#)
- [Load History Purge](#)
- [Load Purge](#)
- [Negotiation History Purge](#)
- [Negotiation Purge](#)
- [Order History Purge](#)
- [Order Purge](#)
- [Order Release Status Purge](#)
- [Order Status Audit Purge](#)
- [Organization Audit Purge](#)
- [Person Info Purge](#)
- [Person Info History Purge](#)
- [Picklist Purge](#)
- [Price List Purge](#)
- [Purge Catalog Mass Audits](#)

- Receipt History Purge
- Receipt Purge
- Reprocess Error Purge
- Reservation Purge
- Shipment History Purge
- Shipment Purge
- Shipment Statistics Purge
- Statistics Purge
- User Activity Purge
- User Activity Audit Purge
- Work Order History Purge
- Work Order Purge
- YFS Audit Purge
- YFSInventoryOwnershipAudit Purge
- Password Reset Request Purge
- User Login Failure Purge

#### **A.4.3.1 Alert Purge**

This purge removes alert data from the system. This reduces the load on frequently accessed tables. The alert should be marked as CLOSED.

Any enterprise that uses the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, an alert is picked up for purge:

- The alert should be in "Closed" status.
- An inactive alert should have the resolution date earlier than or equal to the current date minus the purge criteria's retention days.
- If the alert is in "Open" status, the number of expiration days should be greater than 0, and the modified timestamp should be less than the current date minus the number of expiration days.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–133 Alert Console Purge Attributes**

Attribute	Value
Base Transaction ID	INBOXPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–134 Alert Console Purge Criteria Parameters**

Criteria Parameters	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. The organization for which the Alert Purge needs to be run. If not passed, then all enterprises are monitored.
ExceptionsWithBlank EnterpriseOnly	Optional. If the parameter is set to Y, the agent purges only those exceptions that has blank enterprise code. In this case, the value set for the EnterpriseCode criteria parameter is ignored.

**Table A–134 Alert Console Purge Criteria Parameters**

Criteria Parameters	Description
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–135 Alert Console Purge Statistics**

Statistic Name	Description
NumInboxPurged	Number of inbox records purged.

### Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the YFS\_INBOX table.

### Events Raised

None.

### Tables Purged

YFS\_INBOX

YFS\_INBOX\_NOTES

YFS\_INBOX\_AUDIT

YFS\_INBOX\_REFERENCES

### A.4.3.2 Capacity Purge

This purge removes capacity data from the system. This reduces the load on frequently accessed tables.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a capacity data gets picked up for purge:

- All resource pool standard capacity periods with effective to date earlier than or equal to the current date minus the purge criteria's retention days.
- All resource pool overridden capacity with the capacity date earlier than or equal to the current date minus the purge criteria's retention days.
- All resource pool capacity consumption with consumption date less than or equal to the current date minus the purge criteria's retention days.
- All resource pool capacity consumption details where appointment date is earlier than the system date minus the purge criteria's retention days (or ManualReservationPurgeLeadDays for manually created reservations).
- All resource pool capacity consumption details where expiration date has passed and reservation Id is not blank.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–136 Capacity Purge Attributes**

Attribute	Value
Base Transaction ID	CAPACITYPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–137 Capacity Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>

**Table A–137 Capacity Purge Criteria Parameters**

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–138 Capacity Purge Statistics**

Statistic Name	Description
NumStdCapacityPeriodsPurged	Number of standard capacity periods purged.
NumCapacityOverridesPurged	Number of capacity overrides purged.
NumCapacityConsumptionsPurged	Number of capacity consumptions purged.

### Pending Job Count

For this transaction the pending job count is the total number of records that can be purged from the YFS\_RES\_POOL\_STD\_CAPCTY\_PERD, YFS\_RES\_POOL\_CAPCTY\_OVERRIDE, YFS\_RES\_POOL\_CONSMPTN\_DTLS and YFS\_RES\_POOL\_CAPCTY\_CONSMPTN tables.

### Events Raised

None.

### Tables Purged

The YFS\_RES\_POOL\_STD\_CAPCTY\_PERD table is purged when  $EFFECTIVE\_TO\_DATE \leq (CurrentDate - LeadDays)$

The YFS\_RES\_POOL\_CAPCTY\_OVERRIDE table is purged when  $CAPACITY\_DATE \leq (CurrentDate - LeadDays)$

The YFS\_RES\_POOL\_CAPCTY\_CONSMPTN table is purged when CONSUMPTION\_DATE  $\leq$  (CurrentDate - LeadDays), or if a manual reservation is taken, when CONSUMPTION\_DATE  $\leq$  (CurrentDate - Manual Reservation Retention Days). When this table is purged, YFS\_RES\_POOL\_CONSMPTN\_DTLS is also purged.

The YFS\_RES\_POOL\_CONSMPTN\_DTLS table is purged when RESERVATION\_EXPIRATION\_DATE  $\leq$  (CurrentDate - LeadDays)

#### A.4.3.3 Draft Order History Purge

This purge deletes data from history tables after a specified interval, which in turn, reduces the load on frequently accessed tables.

You can use purge codes' pseudo-logic to analyze the purges. If the following condition is met, a draft order is picked up for history purge:

- The last modified date of the draft order exceeds the retention day period.

All the enterprise using the Console must schedule purge transactions.

For more information about Additional Purge Criteria Based on Line Type, see the *Sterling Distributed Order Management: Configuration Guide*.

---



---

**Note:** The draft order must be purged and moved to the history tables before you purge the draft order history tables. See [Section A.4.3.4, "Draft Order Purge"](#).

---



---



---



---

**Note:** Selling and Fulfillment Foundation does not provide a transaction for draft order history purges. If you are defining a transaction that purges draft order history tables, refer to ["Criteria Parameters"](#) on page 452 for information about the transaction criteria.

If you do not want to define your own transaction to purge draft order history tables, you can use the Order Purge transaction and specify DRAFTORDERHISTPRG for the PurgeCode. To configure the Order Purge transaction for draft order history table purges, refer to ["Order Purge"](#) on page 490 for more information.

---



---

### Criteria Parameters

The following are the criteria parameters for defining a draft order history transaction:

**Table A–139 Draft Order History Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Required. Enterprise for which the Draft Order History Purge has to be run. If not passed, all the enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Removes qualifying records from the history tables that are listed in <a href="#">Tables Purged</a>.</li> <li>N - Test mode. Determines the rows that are removed without actually removing them.</li> </ul>
PurgeCode	Required. Set to DRAFTORDERHISTPRG. Used for internal calculations, such as determining retention days. Corresponds to the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

None.

### Events Raised

None.

### Tables Purged

YFS\_ANSWER\_SET\_TRAN\_H

YFS\_ANSWER\_TRAN\_H  
YFS\_CHARGE\_TRAN\_DIST\_H  
YFS\_CHARGE\_TRANSACTION\_H  
YFS\_CREDIT\_CARD\_TRANSACTION\_H  
YFS\_ENTITY\_ADDRESS\_H  
YFS\_HEADER\_CHARGES\_H  
YFS\_INSTRUCTION\_DETAIL\_H  
YFS\_INVOICE\_COLLECTION\_H  
YFS\_LINE\_CHARGES\_H  
YFS\_NOTES\_H  
YFS\_ORDER\_AUDIT\_DETAIL\_H  
YFS\_ORDER\_AUDIT\_H  
YFS\_ORDER\_AUDIT\_LEVEL\_H  
YFS\_ORDER\_DATE\_H  
YFS\_ORDER\_HEADER\_H  
YFS\_ORDER\_HOLD\_TYPE\_H  
YFS\_ORDER\_HOLD\_TYPE\_LOG\_H  
YFS\_ORDER\_INVOICE\_DETAIL\_H  
YFS\_ORDER\_INVOICE\_H  
YFS\_ORDER\_KIT\_LINE\_H  
YFS\_ORDER\_KIT\_LINE\_SCHEDULE\_H  
YFS\_ORDER\_LINE\_H  
YFS\_ORDER\_LINE\_OPTION\_H  
YFS\_ORDER\_LINE\_REQ\_TAG\_H  
YFS\_ORDER\_LINE\_SCHEDULE\_H  
YFS\_ORDER\_PROD\_SER\_ASSOC\_H  
YFS\_ORDER\_RELEASE\_H  
YFS\_ORDER\_RELEASE\_STATUS\_H

YFS\_ORDER\_SER\_PROD\_ITEM\_H  
YFS\_PAYMENT\_H  
YFS\_PROMOTION\_AWARD\_H  
YFS\_PROMOTION\_H  
YFS\_RECEIVING\_DISCREPANCY\_DTL\_H  
YFS\_RECEIVING\_DISCREPANCY\_H  
YFS\_REFERENCE\_TABLE\_H  
YFS\_TAX\_BREAKUP\_H

### A.4.3.4 Draft Order Purge

This purge archives data into history tables after a specified interval, which in turn, reduces the load on frequently accessed tables. For information about purging draft orders from history tables, see [Section A.4.3.3, "Draft Order History Purge"](#).

---

---

**NOTE:** Selling and Fulfillment Foundation does not provide a transaction for draft order purges. If you are defining a transaction that purges draft orders, refer to [Criteria Parameters on page 456](#) for details about the transaction criteria.

If you do not want to define your own transaction to purge draft orders, you can use the Order Purge transaction and specify DRAFTORDERPRG for the PurgeCode. To configure the Order Purge transaction for draft order purges, refer to ["Order Purge"](#) on page 490 for more information.

---

---

All the enterprise using the Console must schedule purge transactions.

Draft orders are picked up by the agent for validation when the following conditions are met:

- Draft order flag is set to Y.
- Modifyts is set for the retention date.

After the draft orders are picked up, each draft order is validated for purging based on the following conditions:

- No eligible order release status records (records with a status larger than zero) exist for the order.
- All the open child orders (derived, chained, return, exchange, or refund fulfillment) for the order are already purged.

If a draft order meets the set of conditions for validation listed earlier, the agent continues to verify the draft orders against the following criteria:

- Contains the Draft Created (1000) status, and all the extended Draft Created statuses.
- Does not have an order release status record that does not meet the retention days.
- The order's last modification should be before the lead time (in days) setup.
- In the case when an exchange order is part of a return order, the exchange order should be purged from history tables before the return order is purged.
- In the case of an order line reservation, the draft order cannot be purged.
- If the Draft Order Payment Processing flag is set to N, the draft orders are purged.
- If the Draft Order Payment Processing flag is set to Y and a charge exists on a draft order, the draft order is not purged. However, authorizations are not considered when validating draft orders for purge.
- For order lines, except service order lines:
  - If the Seller inventory update is required, the Status Inventory Type has the Update Seller Supply option turned on, and the Seller Supply Type is Onhand, or blank. (The Seller Supply Type can also be a custom seller supply type, with the Onhand Supply check box enabled.)
  - If the Seller Demand Type is blank.
  - If the Buyer inventory update is required, and the Buyer Supply Type is Onhand, or blank.

### Criteria Parameters

The following are the criteria parameters for defining a draft order purge transaction:

**Table A–140 Draft Order Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies (in hours) how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
EnterpriseCode	Required. Enterprise for which the Draft Order Purge has to be run. If not passed, all the enterprises are monitored.  <b>Note:</b> When the EnterpriseCode is blank, the purge criteria configured for the DEFAULT enterprise is used, and not the purge criteria configured for the draft order's enterprise.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a>, to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Set to DRAFTORDERPRG. Used for internal calculations, such as determining retention days. Corresponds to the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None.

**Events Raised**

None.

**Tables Purged**

YFS\_ACTIVITY\_DEMAND  
YFS\_ANSWER\_SET\_TRAN  
YFS\_ANSWER\_TRAN  
YFS\_CHARGE\_TRANSACTION  
YFS\_CHARGE\_TRAN\_DIST  
YFS\_CREDIT\_CARD\_TRANSACTION  
YFS\_ENTITY\_ADDRESS  
YFS\_HEADER\_CHARGES  
YFS\_INSTRUCTION\_DETAIL  
YFS\_INVOICE\_COLLECTION  
YFS\_LINE\_CHARGES  
YFS\_MONITOR\_ALERT  
YFS\_NOTES  
YFS\_ORDER\_AUDIT  
YFS\_ORDER\_AUDIT\_DETAIL  
YFS\_ORDER\_AUDIT\_LEVEL  
YFS\_ORDER\_HEADER  
YFS\_ORDER\_HOLD\_TYPE  
YFS\_ORDER\_HOLD\_TYPE\_LOG  
YFS\_ORDER\_INVOICE  
YFS\_ORDER\_INVOICE\_DETAIL  
YFS\_ORDER\_KIT\_LINE

YFS\_ORDER\_KIT\_LINE\_SCHEDULE  
YFS\_ORDER\_LINE  
YFS\_ORDER\_LINE\_OPTION  
YFS\_ORDER\_LINE\_REQ\_TAG  
YFS\_ORDER\_LINE\_RESERVATION  
YFS\_ORDER\_LINE\_SCHEDULE  
YFS\_ORDER\_LINE\_SRC\_CNTRL  
YFS\_ORDER\_PROD\_SER\_ASSOC  
YFS\_ORDER\_RELEASE  
YFS\_ORDER\_RELEASE\_STATUS  
YFS\_ORDER\_SER\_PROD\_ITEM  
YFS\_ORDER\_DATE  
YFS\_PAYMENT  
YFS\_PMNT\_TRANS\_ERROR  
YFS\_PROMOTION  
YFS\_PROMOTION\_AWARD  
YFS\_RECEIVING\_DISCREPANCY  
YFS\_RECEIVING\_DISCREPANCY\_DTL  
YFS\_REFERENCE\_TABLE  
YFS\_TAX\_BREAKUP

### **A.4.3.5 Delivery Plan Purge**

This purge deletes delivery plans after they have completed their typical life-cycle. It purges all the delivery plans that have been marked as 'Closed' for a period greater than the retention days specified in the criteria parameters and those that do not have any shipments or loads. The order should have been moved to history before the lead time (in days) setup.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a delivery plan is picked up for purge:

- The delivery plan should be in the "Closed" status.
- The delivery plan should not be associated with any load or shipment.
- All orders associated with the delivery plan should be purged.
- The last modification performed on the delivery plan should fall before the lead time (in days) setup.

**Note:** All the loads and shipments that are associated with the delivery plans should have been purged before running this purge agent.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–141** *Delivery Plan Purge Attributes*

Attribute	Value
Base Transaction ID	DELIVERYPLANPRG
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–142 Delivery Plan Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Delivery Plan Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
BatchDelete	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Records are deleted in batches.</li> <li>• N - Records are deleted one by one.</li> </ul>
ColonyID	Required in a multi schema deployment where the YFS_DELIVERY_PLAN table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table A–143 Delivery Plan Purge Statistics*

Statistic Name	Description
NumDeliveryPlansPurged	Number of delivery plans purged.

## Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the YFS\_DELIVERY\_PLAN table.

## Events Raised

None.

## Tables Purged

YFS\_DELIVERY\_PLAN

### A.4.3.6 Export Table Purge

This purge removes export table data from the system. This reduces the load on frequently accessed tables.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, the YFS\_EXPORT table is picked up for purge:

- YFS\_EXPORT records should be marked as processed (Status = 10).
- The last modified time should fall before the lead time (in days) setup.

---



---

**Note:** This purge only reads the rules defined by the hub. Enterprise overridden rules are not considered. This purge should be single threaded when you run it in batch delete mode(BatchDelete=Y).

---



---

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–144** *Export Table Purge Attributes*

Attribute	Value
Base Transaction ID	EXPORTTBLPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–145** *Export Table Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
BatchDelete	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>Y - Records are deleted in batches.</li> <li>N - Default value. Records are deleted one by one.</li> </ul>

**Table A–145 Export Table Purge Criteria Parameters**

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where the YFS_EXPORT table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–146 Export Table Purge Statistics**

Statistic Name	Description
NumExportsPurged	Number of exports purged.

### Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the YFS\_Export table.

### Events Raised

None.

### Tables Purged

YFS\_EXPORT

#### A.4.3.7 Import Table Purge

This purge removes import table data from the system. This reduces the load on frequently accessed tables.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, the YFS\_IMPORT table is picked up for purge:

- YFS\_IMPORT records should be marked as processed (Status = "10").
- The "last modified time" should fall before the lead time (in days) setup.

**Note:** This purge only reads the rules defined by the hub. Enterprise overridden rules are not considered. This purge should be single threaded when you run it in batch delete mode(BatchDelete=Y).

Any enterprise using the Console must schedule purge transactions.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-147 Import Table Purge Attributes**

Attribute	Value
Base Transaction ID	IMPORTTBLPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–148 Import Table Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
BatchDelete	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>• Y - Records are deleted in batches.</li> <li>• N - Default value. Records are deleted one by one.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where the YFS_IMPORT table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

*Table A-149 Import Table Purge Statistics*

Statistic Name	Description
NumImportsPurged	Number of import tables purged.

### Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the YFS\_Import table.

### Events Raised

None.

### Tables Purged

YFS\_IMPORT

#### A.4.3.8 Inventory Audit Purge

This purge removes inventory audit data from the system. This reduces the load on frequently accessed tables.

Any enterprise using the Console must schedule purge transactions.

All inventory audits of the provided organization with modify timestamp earlier than the current date minus the purge criteria's retention days can be configured to be picked up by the Inventory Audit Purge.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, an inventory audit record is picked up for purge:

- The inventory audit record's last modification is earlier than the current timestamp minus the retention days.

**Note:** Number of threads for this purge's agent criteria details must be set to 1. For more information about agent criteria, see the *Selling and Fulfillment Foundation: Application Platform Configuration Guide*.

**Important:** The Inventory Audit purge does not purge any records under 60 days old, even if configured to do so.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–150 Inventory Audit Purge Attributes**

Attribute	Value
Base Transaction ID	INVENTORYAUDITPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–151 Inventory Audit Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. The inventory organization for which the Inventory Audit Purge needs to be run. If not passed, then all enterprises are monitored.

**Table A–151 Inventory Audit Purge Criteria Parameters**

Parameter	Description
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Table Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–152 Inventory Audit Statistics**

Statistic Name	Description
NumInventoryAuditsPurged	Number of inventory audits purged.

### Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the YFS\_Inventory\_Audit table.

### Events Raised

None.

### Table Purged

YFS\_INVENTORY\_AUDIT

### A.4.3.9 Inventory Purge

This purge removes inventory data from the system. This reduces the load on frequently accessed tables. This purge does not take retention days into account when purging.

You can use purge codes pseudo-logic to analyze purges.

For YFS\_INVENTORY\_SUPPLY, if the following conditions are met, an inventory supply is picked up for purge:

- Supply record has the same availability type as the node. For example, TRACK or INFINITE.
- Supply record has 0 quantity.
- Supply record does not contain the supply type "INFO".

For YFS\_INVENTORY\_DEMAND, if the following conditions are met, an inventory demand is picked up for purge:

- Demand record has 0 quantity or lesser.
- Demand record does not have demand details as well as matching demand record in YFS\_INVENTORY\_DEMAND\_ADDNL tables.

For YFS\_INVENTORY\_TAG, it is purged if the INVENTORY\_TAG\_KEY is not used by any of the existing supply and demand.

For YFS\_INVENTORY\_RESERVATION, an inventory reservation is picked up for purge if it meets the following conditions:

- Inventory reservation record has 0 quantity or ship date is earlier than the system date minus the purge criteria's retention days.

For YFS\_INVENTORY\_NODE\_CONTROL, it is purged if the INV\_PIC\_INCORRECT\_TILL\_DATE is earlier than the current time stamp minus the purge criteria's retention days.

For YFS\_IBA\_TRIGGER, it is purged if IBA\_REQUIRED = 'N', IBA\_RUN\_REQUIRED = 'N', and LAST\_IBA\_PROCESSED\_TS is earlier than the current time stamp minus the purge criteria's retention days.

Any enterprise using the Console must schedule purge transactions.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–153 Inventory Purge Attributes**

Attribute	Value
Base Transaction ID	INVENTORYPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–154 Inventory Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>

**Table A–154 Inventory Purge Criteria Parameters**

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–155 Inventory Purge Statistics**

Statistic Name	Description
NumInventoryDemandsPurged	Number of inventory demands purged.
NumInventoryNodeControlsPurged	Number of inventory node controls purged.
NumInventoryReservationsPurged	Number of inventory reservations purged.
NumInventoryTagsPurged	Number of inventory tags purged.
NumItemBasedAllocationTriggersPurged	Number of item based allocation triggers purged.

### Pending Job Count

For this transaction, the pending job count is the total number of records that can be purged from the YFS\_Inventory\_Supply, YFS\_Inventory\_Demand, YFS\_Inventory\_Tag, YFS\_Inventory\_Reservation, YFS\_IBA\_Trigger, and YFS\_Inventory\_Node\_Control tables.

### Events Raised

None.

### Tables Purged

YFS\_IBA\_TRIGGER

YFS\_INVENTORY\_DEMAND

YFS\_INVENTORY\_TAG

YFS\_INVENTORY\_RESERVATION

YFS\_INVENTORY\_SUPPLY

YFS\_INVENTORY\_NODE\_CONTROL

### A.4.3.10 Inventory Supply Temp Purge

The Inventory Supply Temp purge agent cleans up the contents in the temporary inventory tables generated by the process of synchronizing the Selling and Fulfillment Foundation inventory picture with the actual inventory picture at the nodes.

The node inventory picture is stored during the loading process into the YFS\_INVENTORY\_SUPPLY\_TEMP table. Once the synchronization phase is complete and the YFS\_INVENTORY\_SUPPLY table has been updated, the YFS\_INVENTORY\_SUPPLY\_TEMP table needs to be purged, which is done through this agent.

For more information about configuring the synchronization with node inventory, see the *Sterling Global Inventory Visibility: Configuration Guide*.

The Inventory Supply Temp purge agent is used to purge all records in the YFS\_INVENTORY\_SUPPLY\_TEMP table whose modify timestamp is less than current time minus the purge criteria's retention days for a group of YantraMessageGroupID.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–156 Inventory Supply Temp Purge Attributes**

Attribute	Value
Base Transaction ID	SUPPLYTEMPPRG
Base Document Type	General
Base Process Type	General

**Table A–156 Inventory Supply Temp Purge Attributes**

Attribute	Value
Abstract Transaction	No
APIs Called	None
User Exits Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–157 Inventory Supply Temp Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
EnterpriseCode	Optional. The inventory organization for which the Inventory Supply Temp Purge needs to be run. If not passed, then all enterprises are monitored.organization.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_INVENTORY_SUPPLY_TEMP table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

*Table A–158 Inventory Supply Temp Purge Statistics*

Statistic Name	Description
NumInventorySupplyTempsPurged	Number of entries in the YFS_INVENTORY_SUPPLY_TEMP table purged.

### Pending Job Count

Number of unique YantraMessageGroupIDs from YFS\_INVENTORY\_SUPPLY\_TEMP table whose maximum modify timestamp is less than current timestamp minus purge criteria's lead day.

### Events Raised

None.

### Tables Purged

YFS\_INVENTORY\_SUPPLY\_TEMP

#### A.4.3.11 Item Audit Purge

This purge removes the YFS\_AUDIT table data from the system, which reduces the load on frequently accessed tables. It purges records in the YFS\_AUDIT and the YFS\_AUDIT\_HEADER tables that meet the following conditions:

- YFS\_AUDIT records that have 'modifyts' greater than the retention days specified and the records have the table name as 'YFS\_ITEM'.
- The last modified time is before the lead time (in days) setup.

When the enterprise modifies records in the YFS\_ITEM table through the Applications Manager, the YFS\_ITEM is audited and the audit records are inserted in the YFS\_AUDIT table. In order to clean up the audit records, this purge transaction can be used.

Any enterprise using the Console must schedule purge transactions accordingly.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–159** *Item Audit Purge Attributes*

Attribute	Value
Base Transaction ID	YFS_ITEM_AUDIT_PURGE
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–160** *Item Audit Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, the value defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), this value defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Production mode. Deletes records from the regular tables.</li> <li>N - Test mode.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_AUDIT and YFS_AUDIT_HEADER tables may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A-161** *Item Audit Purge Statistics*

Statistic Name	Description
NumItemAuditRecordsPurged	Number of item audit records purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_AUDIT table that match the criteria values.

### Events Raised

None.

### Tables Purged

YFS\_AUDIT, YFS\_AUDIT\_HEADER

### A.4.3.12 Load History Purge

This purge deletes the load data from history tables after it completes its typical lifecycle. This reduces the load on frequently accessed tables.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a load is picked up for purge:

- The last modification made to the load is before the lead time (in days) setup.

**Note:** Before you run this transaction, ensure to purge loads and move them to history tables. For more information about purging loads, see [Section A.4.3.13, "Load Purge"](#).

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–162 Load History Purge Attributes**

Attribute	Value
Base Transaction ID	LOADHISTPRG
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–163 Load History Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Load Purge needs to be run. If not passed, all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>

**Table A–163 Load History Purge Criteria Parameters**

Parameter	Description
Purge Code	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–164 Load History Purge Statistics**

Statistic Name	Description
NumLoadHistoriesPurged	Number of load histories purged.
NumLoadShipmentHistoriesPurged	Number of load shipment histories purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Load\_H table.

### Events Raised

None.

### Tables Purged

YFS\_LOAD\_H

YFS\_LOAD\_STOP\_H

YFS\_LOAD\_SHIPMENT\_CHARGE\_H

YFS\_LOAD\_STATUS\_AUDIT\_H

YFS\_SHIPMENT\_CONTAINER\_H

YFS\_CONTAINER\_ACTIVITY\_H

YFS\_LOADED\_CONTAINER\_H  
YFS\_LOAD\_SHIPMENT\_H  
YFS\_ADDITIONAL\_DATE\_H  
YFS\_LOAD\_HOLD\_TYPE\_H  
YFS\_LOAD\_HOLD\_TYPE\_LOG\_H

#### **A.4.3.13 Load Purge**

This purge removes load data from the system. It picks up all loads that have been marked as 'Closed' and purges them. Empty Loads (for example, loads with no shipments) are not considered for purge. As a part of this purge, the associated child tables are also purged.

This is not a pipeline transaction. It also does not work from the task queue.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a load is picked up for purge:

- The Load's last modification should fall before the lead time (in days) setup.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–165 Load Purge Attributes**

Attribute	Value
Base Transaction ID	LOADPRG
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–166 Load Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Load Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>

**Table A-166 Load Purge Criteria Parameters**

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A-167 Load Purge Statistics**

Statistic Name	Description
NumLoadShipmentsPurged	Number of load shipments purged.
NumLoadsPurged	Number of loads purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Load table.

### Events Raised

None.

### Tables Purged

YFS\_ADDITIONAL\_DATE

YFS\_LOAD

YFS\_LOAD\_HOLD\_TYPE

YFS\_LOAD\_HOLD\_TYPE\_LOG

YFS\_LOAD\_STOP

YFS\_LOAD\_SHIPMENT

YFS\_LOAD\_SHIPMENT\_CHARGES (charges that pertain to this load)

YFS\_LOAD\_STATUS\_AUDIT  
YFS\_LOADED\_CONTAINER  
YFS\_SHIPMENT\_CONTAINER  
YFS\_CONTAINER\_ACTIVITY

### A.4.3.14 Negotiation History Purge

This purge deletes negotiation history data from the system. This reduces the load on frequently accessed tables. It purges data from the order negotiation history tables.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a negotiation is picked up for history purge:

- The last modified date of the negotiation exceeds the retention day period.

Any enterprise using the Console must schedule purge transactions.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table A-168 Negotiation History Purge Attributes*

Attribute	Value
Base Transaction ID	NEGOTIATIONHISTPRG
Base Document Type	Order
Base Process Type	Order Negotiation
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–169** *Negotiation History Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Negotiation History Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–170** *Negotiation History Purge Statistics*

Statistic Name	Description
NumNegotiationHistoriesPurged	Number of negotiation histories purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Negotiation\_Hdr\_H table.

### Events Raised

None.

### Tables Purged

YFS\_AUDIT

YFS\_NEGOTIATION\_HDR\_H

YFS\_NEGOTIATION\_LINE\_H

YFS\_RESPONSE\_H

YFS\_RESPONSE\_HDR\_H

YFS\_RESPONSE\_LINE\_H

YFS\_RESPONSE\_LINE\_DTL\_H

### A.4.3.15 Negotiation Purge

This purge archives data into history tables after it completes its typical lifecycle. This reduces the load on frequently accessed tables. It works from the task queue (YFS\_TASK\_Q) table.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a negotiation is picked up for purge:

- The last modification performed on the negotiation falls before the lead time (in days) setup.
- The negotiation is in pickable status.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–171 Negotiation Purge Attributes**

Attribute	Value
Base Transaction ID	ORD_NEGOTIATION_PURGE
Base Document Type	Order
Base Process Type	Order Negotiation
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–172 Negotiation Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Negotiation Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>

**Table A–172 Negotiation Purge Criteria Parameters**

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–173 Negotiation Purge Statistics**

Statistic Name	Description
NumOrderNegotiationsPurged	Number of order negotiations purged.

### Pending Job Count

For this transaction, the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

### Events Raised

None

### Tables Purged

YFS\_AUDIT

YFS\_NEGOTIATION\_HDR

YFS\_NEGOTIATION\_LINE

YFS\_RESPONSE

YFS\_RESPONSE\_HDR  
 YFS\_RESPONSE\_LINE  
 YFS\_RESPONSE\_LINE\_DTL

#### A.4.3.16 Order History Purge

This purge deletes data from history tables after it completes its typical lifecycle. This reduces the load on frequently accessed tables.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, an order is picked up for history purge:

- The last modified date of the order exceeds the retention day period.

Any enterprise using the Console must schedule purge transactions.

For more information about Additional Purge Criteria Based on Line Type, see the *Sterling Distributed Order Management: Configuration Guide*.

---



---

**Note:** The order should have been purged and moved into the history tables before you can run this transaction. See [Section A.4.3.17, "Order Purge"](#).

---



---

#### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–174 Order History Purge Attributes**

Attribute	Value
Base Transaction ID	ORDERHISTPRG
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

#### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–175 Order History Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Order History Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Removes qualifying records from the history tables listed under <a href="#">Tables Purged</a>.</li> <li>• N- Test mode. Determines the rows that are removed without actually removing them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–176 Order History Purge Statistics**

Statistic Name	Description
NumOrderHistoriesPurged	Number of order histories purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Order\_HEADER\_H table.

**Events Raised**

None.

**Tables Purged**

YFS\_ANSWER\_SET\_TRAN\_H  
YFS\_ANSWER\_TRAN\_H  
YFS\_CHARGE\_TRAN\_DIST\_H  
YFS\_CHARGE\_TRAN\_REQUEST\_H  
YFS\_CHARGE\_TRAN\_RO\_MAP\_H  
YFS\_CHARGE\_TRANSACTION\_H  
YFS\_CREDIT\_CARD\_TRANSACTION\_H  
YFS\_ENTITY\_ADDRESS\_H  
YFS\_HEADER\_CHARGES\_H  
YFS\_INSTRUCTION\_DETAIL\_H  
YFS\_INVOICE\_COLLECTION\_H  
YFS\_LINE\_CHARGES\_H  
YFS\_NOTES\_H  
YFS\_ORDER\_AUDIT\_DETAIL\_H  
YFS\_ORDER\_AUDIT\_H  
YFS\_ORDER\_AUDIT\_LEVEL\_H  
YFS\_ORDER\_DATE\_H  
YFS\_ORDER\_HEADER\_H  
YFS\_ORDER\_HOLD\_TYPE\_H  
YFS\_ORDER\_HOLD\_TYPE\_LOG\_H  
YFS\_ORDER\_INVOICE\_DETAIL\_H  
YFS\_ORDER\_INVOICE\_H  
YFS\_ORDER\_KIT\_LINE\_H  
YFS\_ORDER\_KIT\_LINE\_SCHEDULE\_H  
YFS\_ORDER\_LINE\_H

YFS\_ORDER\_LINE\_OPTION\_H  
YFS\_ORDER\_LINE\_REQ\_TAG\_H  
YFS\_ORDER\_LINE\_SCHEDULE\_H  
YFS\_ORDER\_PROD\_SER\_ASSOC\_H  
YFS\_ORDER\_RELEASE\_H  
YFS\_ORDER\_RELEASE\_STATUS\_H  
YFS\_ORDER\_SER\_PROD\_ITEM\_H  
YFS\_PAYMENT\_H  
YFS\_PROMOTION\_AWARD\_H  
YFS\_PROMOTION\_H  
YFS\_RECEIVING\_DISCREPANCY\_DTL\_H  
YFS\_RECEIVING\_DISCREPANCY\_H  
YFS\_REFERENCE\_TABLE\_H  
YFS\_TAX\_BREAKUP\_H

### A.4.3.17 Order Purge

This purge archives data into history tables after it completes its typical lifecycle. To purge orders from history tables, see [Section A.4.3.16, "Order History Purge"](#). This reduces the load on frequently accessed tables. It works on a task queue. It picks up the orders from YFS\_TASK\_Q table that are available for the transaction PURGE.

---

---

**Note:** This transaction depends on all lines of an order being in a status pickable by the Purge transaction.

---

---

---

---

**Note:** If purge criteria are not met, AVAILABLE\_DATE is calculated based on the modify time stamp of the order in YFS\_ORDER\_HEADER table as well as the YFS\_TASK\_Q table, whichever is maximum. To this value, retention days is added to the new AVAILABLE\_DATE.

---

---

The following statuses are available for configuration to be picked up by Order Purge:

- Draft Created (1000) and all extended Draft Created Statuses.
- Created (1100) and all extended Created statuses. These statuses are available only for document types Sales Order, Purchase Order and Transfer Order.
- Released (3200) and all extended Released statuses.
- Shipped (3700) and all extended Shipped statuses.
- Completed (3700) and all extended Completed statuses. These statuses are available only for the document type Master Order.
- Received (3900) and all extended Received statuses.
- Cancelled (9000) and all extended Cancelled statuses.
- Shorted (9020) and all extended Shorted statuses.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, an order is picked up for purge:

- All open child orders (derived, chained, return, exchange, procurement, or refund fulfillment) for the order must already be purged.
- No pending transfer-out charges to another order exceeding the transfer-ins.
- No pending adjustment invoices.

An order is purged immediately if it meets the above three criteria and is completely cancelled with payment collection complete.

---

---

**Note:** In order for the purge agent to pick up a cancelled order, the payment status of the order must be one of the following:

- Paid
  - Not Applicable
- 
- 

If an order does not meet any of the above criteria, continue checking for the criteria given below:

- No order release status record that does not meet the retention days.
- It should be in the correct status for purge. For example,
  - All service requests for the order should have Shipped or extended Shipped status.
  - The payment status for the order should be Paid or Not Applicable.
  - It must not have any unpurged negotiations.
- For all order lines other than service request lines:
  - If the Seller inventory update is required, the Status Inventory Type has the "Update Seller Supply" option turned on, and the Seller Supply Type is "Onhand", or blank. (The Seller Supply Type can also be a custom seller supply type with the "Onhand Supply" checkbox enabled.)
  - If the Seller Demand Type is blank.
  - If the Buyer inventory update is required and the Buyer Supply Type is "Onhand", or blank.
- The order's last modification should fall before the lead time (in days) setup.
- Any enterprise using the Console must schedule purge transactions.
- The order must not have a undelivered service line.

- In the case of an exchange order for processing a return order, the exchange order should be purged from history before the return order can be purged.

**Note:** With no change to status inventory type, a sales order in Shipped (3700) status or its extended status is purged if the Buyer is not passed.

An order in Shipped status or extended Shipped status in the default pipeline is not purged if the Buyer passed on the sales order is tracking inventory. This prevents the purging of the order relating to the pending supply for the Buyer tracking inventory.

To purge such orders, the status inventory type for the Shipped or extended Shipped status should be configured such that the Buyer Supply Type is ONHAND for the status inventory type.

When the purge agent is run, the draft order without lines are purged to the order history table. Once the purge history agent is run, the draft orders without lines gets deleted permanently.

**Attributes**

The following are the attributes for this time-triggered transaction:

*Table A-177 Order Purge Attributes*

Attribute	Value
Base Transaction ID	PURGE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

**Criteria Parameters**

The following are the criteria parameters for this transaction:

Table A-178 Order Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
EnterpriseCode	Optional. Enterprise for which the Order Purge needs to be run. If not passed, then all enterprises are monitored. <b>Note:</b> When the EnterpriseCode is blank, the purge criteria configured for the DEFAULT enterprise is used; not the purge criteria configured for the order's enterprise.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>

**Table A–178 Order Purge Criteria Parameters**

Parameter	Description
PurgeCode	<p>Required. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria. You can set this parameter to the following values:</p> <ul style="list-style-type: none"> <li>• DRAFTORDERHISTPRG to purge draft order information from the order history tables.</li> <li>• DRAFTORDERNOLINEHISTPRG to purge draft orders without order lines from the order history tables.</li> <li>• DRAFTORDERNOLINEPRG to purge draft orders that have no order lines.</li> <li>• DRAFTORDERPRG to purge draft order information and archive it in the order history tables.</li> </ul> <p>PurgeCode cannot be set to the value ORDER_RELEASE_STATUS_PURGE.</p>
AdditionalPurgeCode	<p>Optional. To purge order release status records, set this parameter to ORDER_RELEASE_STATUS_PURGE.</p> <p>For more information, see <a href="#">Section A.4.3.18, "Order Release Status Purge"</a>.</p>
ColonyID	<p>Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.</p>

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–179 Order Purge Statistics**

Statistic Name	Description
NumOrdersProcessed	Number of order processed.
NumOrdersPurged	Number of orders purged.

### Pending Job Count

For this transaction, the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

### Events Raised

None.

### Tables Purged

YFS\_ACTIVITY\_DEMAND  
YFS\_ANSWER\_SET\_TRAN  
YFS\_ANSWER\_TRAN  
YFS\_CHARGE\_TRANSACTION  
YFS\_CHARGE\_TRAN\_DIST  
YFS\_CHARGE\_TRAN\_REQUEST  
YFS\_CHARGE\_TRAN\_RQ\_MAP  
YFS\_CREDIT\_CARD\_TRANSACTION  
YFS\_ENTITY\_ADDRESS  
YFS\_HEADER\_CHARGES  
YFS\_INSTRUCTION\_DETAIL  
YFS\_INVOICE\_COLLECTION  
YFS\_LINE\_CHARGES  
YFS\_MONITOR\_ALERT  
YFS\_NOTES  
YFS\_ORDER\_AUDIT  
YFS\_ORDER\_AUDIT\_DETAIL  
YFS\_ORDER\_AUDIT\_LEVEL  
YFS\_ORDER\_HEADER  
YFS\_ORDER\_HOLD\_TYPE

YFS\_ORDER\_HOLD\_TYPE\_LOG  
YFS\_ORDER\_INVOICE  
YFS\_ORDER\_INVOICE\_DETAIL  
YFS\_ORDER\_KIT\_LINE  
YFS\_ORDER\_KIT\_LINE\_SCHEDULE  
YFS\_ORDER\_LINE  
YFS\_ORDER\_LINE\_OPTION  
YFS\_ORDER\_LINE\_REQ\_TAG  
YFS\_ORDER\_LINE\_RESERVATION  
YFS\_ORDER\_LINE\_SCHEDULE  
YFS\_ORDER\_LINE\_SRC\_CNTRL  
YFS\_ORDER\_PROD\_SER\_ASSOC  
YFS\_ORDER\_RELEASE  
YFS\_ORDER\_RELEASE\_STATUS  
YFS\_ORDER\_SER\_PROD\_ITEM  
YFS\_ORDER\_DATE  
YFS\_PAYMENT  
YFS\_PMNT\_TRANS\_ERROR  
YFS\_PROMOTION  
YFS\_PROMOTION\_AWARD  
YFS\_RECEIVING\_DISCREPANCY  
YFS\_RECEIVING\_DISCREPANCY\_DTL  
YFS\_REFERENCE\_TABLE  
YFS\_TAX\_BREAKUP

#### **A.4.3.18 Order Release Status Purge**

The Order Release Status Purge agent extends the Order Purge agent's capabilities by purging order release status records before the Order Purge agent completely purges data to history tables.

If an order meets the criteria for purging, the order release status records with quantities of 0 are deleted from the YFS\_ORDER\_RELEASE\_STATUS table and are not put into the history table. When the Order Release Status Purge agent has completed, the task queue's AVAILABLE\_DATE is reset to the date specified by the purge criteria for Order Purge. This enables the Order Purge agent to pick up and process an order as necessary. Order Purge will continue to purge order release status records as usual.

If the following conditions are met, the Order Purge agent purges order release status records:

- All conditions for Order Purge have been met. See [Section A.4.3.17, "Order Purge"](#) for information about conditions for Order Purge.
- Order release records have 0 quantity.
- AdditionalPurgeCode in the Order Purge criteria is set to ORDER\_RELEASE\_STATUS\_PURGE.
- The order has been modified within the Order Purge lead days AdditionalPurgeCode.

### Criteria Parameters

The following are the criteria parameters for Order Release Status Purge:

**Table A–180 Order Release Status Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.

**Table A–180 Order Release Status Purge Criteria Parameters**

Parameter	Description
EnterpriseCode	Optional. Enterprise for which the Order Purge needs to be run. If not passed, then all enterprises are monitored.  <b>Note:</b> When the EnterpriseCode is blank, the purge criteria configured for the DEFAULT enterprise is used; not the purge criteria configured for the order's enterprise.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. To extend the Order Purge agent to purge order release status records, set to ORDERPRG. Used for internal calculations, such as determining retention days. You must also set AdditionalPurgeCode.
AdditionalPurgeCode	Required. To purge order release status records, set this parameter to ORDER_RELEASE_STATUS_PURGE.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None.

**Pending Job Count**

The pending job count is the number of records available to be processed by Order Purge with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

### Events Raised

None.

### Tables Purged

YFS\_ORDER\_RELEASE\_STATUS

#### A.4.3.19 Order Status Audit Purge

This purge removes order status audit data from the system. This reduces the load on frequently accessed tables.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, an order status audit is picked up for history purge:

- The last modified time falls before the lead time (in days) setup.

Any enterprise using the Console must schedule purge transactions.

**Note:** This transaction needs to be run after negotiation is completed.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–181** Order Status Audit Purge Attributes

Attribute	Value
Base Transaction ID	STATUSAUDITPRG
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A-182 Order Status Audit Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Order Status Audit Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_STATUS_AUDIT Table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

### Pending Job Count

*Table A-183 Order Status Audit Purge Statistics*

Statistic Name	Description
NumStatusAuditsPurged	Number of status audits purged.

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Status\_Audit table.

### Events Raised

None.

### Tables Purged

YFS\_STATUS\_AUDIT

### A.4.3.20 Organization Audit Purge

This purge removes the YFS\_AUDIT table data from the system, which reduces the load on frequently accessed tables. It purges records in the YFS\_AUDIT and the YFS\_AUDIT\_HEADER tables that meet the following conditions:

- The YFS\_AUDIT records that have 'modifyts' greater than the retention days specified and the records have the table name as 'YFS\_ORGANIZATION'.
- The last modified time is before the lead time (in days) setup.

When the enterprise modifies records in the YFS\_ORGANIZATION table through the Applications Manager, the YFS\_ORGANIZATION is audited and the audit records are inserted in the YFS\_AUDIT table. In order to clean up the audit records, this purge transaction can be used.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–184 Organization Audit Purge Attributes**

Attribute	Value
Base Transaction ID	YFS_ORGANIZATION_AUDIT_PURGE
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–185 Organization Audit Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, the value defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), this value defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Production mode. Deletes records from the regular tables.</li> <li>N - Test mode.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds to the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_AUDIT and YFS_AUDIT_HEADER tables may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–186 Organization Audit Purge Statistics**

Statistic Name	Description
NumOrganizationAuditRecordsPurged	Number of organization audit records purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_AUDIT table that match the criteria values.

## Events Raised

None.

## Tables Purged

YFS\_AUDIT

YFS\_AUDIT\_HEADER

### A.4.3.21 Person Info Purge

This purge gets a list of dates with the person info record count and sorts them by date in ascending order. Then, based on the specified number of records to buffer and the modify timestamp, it purges the applicable records and places them in the YFS\_PERSON\_INFO\_H table.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–187 PersonInfo Purge Attributes**

Attribute	Value
Base Transaction ID	PERSONINFOPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No

**Table A–187 PersonInfo Purge Attributes**

Attribute	Value
APIs Called	None
User Exits Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–188 PersonInfo Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. <ul style="list-style-type: none"> <li>If left blank or the number specified is less than 10000, it defaults to 5000.</li> <li>If the number specified is greater than 10000, then that value is used.</li> </ul>
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.

**Table A-188 PersonInfo Purge Criteria Parameters**

Parameter	Description
EnterpriseCode	Optional. Enterprise for which the PersonInfo Purge needs to be run. If not passed, then all enterprises are monitored.
TableType	<p>Required in a multi schema deployment when YFS_Person_Info table may exist in multiple schemas.</p> <p>Valid Values: CONFIGURATION, TRANSACTION, MASTER.</p> <p>If set to CONFIGURATION, purge runs for the YFS_Person_Info records associated with tables that have TableType as CONFIGURATION; for example, YFS_Organization, YFS_Ship_Node, and so forth.</p> <p>If set to TRANSACTION, purge runs for the YFS_Person_Info records associated with tables that have TableType as TRANSACTION; for example, YFS_Order_Header, YFS_Shipment, and so forth.</p> <p>Note that purge would run for all TableTypes that exist in the same schema as the one passed. For example, if set to TRANSACTION, purge would also run for YFS_Person_Info records associated with tables that have TableType as MASTER, since they reside in the same schema.</p>
ColonyID	Required in a multi schema deployment where the YFS_PERSON_INFO table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

The following statistics are tracked for this transaction:

If it is left blank or any number less than 10,000 is specified, then it defaults to 10,000. But if any number > 10,000 is specified, then that value would be used.

*Table A–189 PersonInfo Purge Statistics*

Statistic Name	Description
NumPersonInfoPurged	Number of person info records purged.

**Pending Job Count**

For this transaction, the pending job count is the number of records that can be purged from the YFS\_PERSON\_INFO table.

**Events Raised**

None.

**Tables Purged**

YFS\_PERSON\_INFO

**A.4.3.22 Person Info History Purge**

This purge deletes records from the YFS\_PERSON\_INFO\_H table based on the purge criteria.

**Attributes**

The following are the attributes for this time-triggered transaction:

*Table A–190 PersonInfo Purge Attributes*

Attribute	Value
Base Transaction ID	PERSONINFOHISTPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–191 PersonInfo Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. <ul style="list-style-type: none"> <li>If left blank or the number specified is less than 10000, it defaults to 5000.</li> <li>If the number specified is greater than 10000, then that value is used.</li> </ul>
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
EnterpriseCode	Optional. Enterprise for which the PersonInfo Purge needs to be run. If not passed, then all enterprises are monitored.

**Table A–191 PersonInfo Purge Criteria Parameters**

Parameter	Description
TableType	<p>Required in a multi schema deployment when YFS_Person_Info table may exist in multiple schemas.</p> <p>Valid Values: CONFIGURATION, TRANSACTION, MASTER.</p> <p>If set to CONFIGURATION, purge runs for the YFS_Person_Info records associated with tables that have TableType as CONFIGURATION; for example, YFS_Organization, YFS_Ship_Node, and so forth.</p> <p>If set to TRANSACTION, purge runs for the YFS_Person_Info records associated with tables that have TableType as TRANSACTION; for example, YFS_Order_Header, YFS_Shipment, and so forth.</p> <p>Note that purge would run for all TableTypes that exist in the same schema as the one passed. For example, if set to TRANSACTION, purge would also run for YFS_Person_Info records associated with tables that have TableType as MASTER, since they reside in the same schema.</p>
ColonyID	Required in a multi schema deployment where the YFS_PERSON_INFO_H table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–192 PersonInfo Purge Statistics**

Statistic Name	Description
NumPersonInfoHistoryRecordsPurged	Number of person info history records purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_PERSON\_INFO\_H table.

### Events Raised

None.

### Tables Purged

YFS\_PERSON\_INFO\_H

#### A.4.3.23 Picklist Purge

This purge picks up all picklists that have been existing for a period greater than the retention days specified in the criteria parameters and those that do not have any shipments.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a picklist is picked up for purge:

- The picklist should exist for more than the specified retention period.
- The picklist should not be associated with any shipment.

### Attributes

**Note:** All shipments associated with the picklists should have been purged before running this purge agent.

The following are the attributes for this time-triggered transaction:

**Table A–193** *Picklist Purge Attributes*

Attribute	Value
Base Transaction ID	PICKLISTPRG
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–194** *Picklist Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_PICK_LIST table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

*Table A–195 Picklist Purge Statistics*

Statistic Name	Description
NumPickListsPurged	Number of picklists purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_PICK\_LIST table.

### Events Raised

None.

### Tables Purged

YFS\_PICK\_LIST

#### A.4.3.24 Price List Purge

This purge removes price list data from the system. This reduces the load on frequently accessed tables.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a price list is picked up for purge:

- The price list has valid date less than the current date minus the purge criteria's retention days.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table A–196 Price List Purge Attributes*

Attribute	Value
Base Transaction ID	PRICELISTPRG
Base Document Type	General
Base Process Type	General

**Table A–196 Price List Purge Attributes**

Attribute	Value
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–197 Price List Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–198 Price List Purge Statistics**

Statistic Name	Description
NumPriceSetsPurged	Number of price sets purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Price\_Set table.

### Events Raised

None.

### Tables Purged

YFS\_PRICE\_SET table with VALID\_TILL\_DATE less than or equal to (CurrentDate - LeadDays)

YFS\_PRICE\_PROGRAM\_DEFN

YFS\_ITEM\_PRICE\_SET

YFS\_ITEM\_PRICE\_SET\_DTL

### A.4.3.25 Purge Catalog Mass Audits

This purge removes old audit records from the YFS\_CATALOG\_MASS\_AUDIT table. This table contains data about changes to the catalog due to assignment of attributes and attribute values to categories and items. It also contains information about inherited attributes and attribute values. The purge transaction finds mass audit records that have not been modified in a specified number of days and removes those records from the database.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–199 Purge Catalog Mass Audits Attributes**

Attribute	Value
Base Transaction ID	CATALOG_MASS_AUDIT_PURGE
Base Document Type	General

**Table A–199 Purge Catalog Mass Audits Attributes**

Attribute	Value
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–200 Purge Catalog Mass Audits Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_CATALOG_MASS_AUDIT table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–201 Purge Catalog Mass Audits Statistics**

Statistic Name	Description
NumCatalogMassAuditsPurged	Number of mass audit records purged.

### Pending Job Count

For this transaction the pending job count is the total number of records that can be purged from the YFS\_CATALOG\_MASS\_AUDIT table.

### Events Raised

None.

### Tables Purged

The YFS\_CATALOG\_MASS\_AUDIT table is purged when  $\text{MODIFYTS} < (\text{CurrentDate} - \text{LeadDays})$

### A.4.3.26 Receipt History Purge

This transaction deletes receipts previously archived by the Receipt Purge. See [Section A.4.3.27, "Receipt Purge"](#).

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a receipt that is previously purged by the receipt purge agent is picked up for history purge:

- The last modified date of the receipt should exceed the retention day period.
- The shipment associated with the receipt should be purged from the history table.

**Note:** To purge a receipt history, ensure that the Receipts are closed and Shipments are purged.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–202 Receipt History Purge Attributes**

Attribute	Value
Base Transaction ID	RECEIPTHISTPRG
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–203 Receipt History Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Receipt History Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Removes qualifying records from the history tables listed under <a href="#">Tables Purged</a>.</li> <li>N- Test mode. Determines the rows that are removed without actually removing them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–204 Receipt History Purge Statistics**

Statistic Name	Description
NumReceiptLineHistoriesPurged	Number of receipt line histories purged.
NumReceiptHistoriesPurged	Number of receipt histories purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Receipt\_Header\_H table.

### Events Raised

None.

### Tables Purged

YFS\_RECEIPT\_HEADER\_H

YFS\_RECEIPT\_LINE\_H

YFS\_RECEIPT\_STATUS\_AUDIT\_H

YFS\_INSTRUCTION\_DETAIL\_H

### A.4.3.27 Receipt Purge

This purge removes receipt data from the system. This reduces the load on frequently accessed tables. This transaction picks up all receipts that are not open and not pending inspection and archives them into their history tables. See [Section A.4.3.26, "Receipt History Purge"](#). It also archives and purges the receipt's child tables.

This is a pipeline transaction and works from a task queue.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a receipt is picked up for purge:

- The last modified date of the receipt should exceed the retention day period.

- The shipment associated with the receipt should be purged.
- The receipt should be in pickable status for the purge transaction.
- The value of the OpenReceiptFlag field should be set to "N".
- The receipt should not have pending inspections.
- There is no inventory in the warehouse for the receipt.

---

---

**Note:** To purge a receipt, ensure that the receipts are closed and Shipments are purged.

---

---

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–205 Receipt Purge Attributes**

Attribute	Value
Base Transaction ID	RECEIPTPRG
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–206 Receipt Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Receipt Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>

**Table A–206 Receipt Purge Criteria Parameters**

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

The following statistics are tracked for this transaction:

**Table A–207 Receipt Purge Statistics**

Statistic Name	Description
NumReceiptLinesPurged	Number of Receipt Lines purged.
NumReceiptsPurged	Number of receipts purged.

**Pending Job Count**

For this transaction, the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

**Events Raised**

None.

**Tables Purged**

YFS\_RECEIPT\_HEADER

YFS\_RECEIPT\_LINE

YFS\_RECEIPT\_STATUS\_AUDIT

YFS\_INSTRUCTION\_DETAIL

### A.4.3.28 Reprocess Error Purge

This purge deletes reprocess errors from the system. This reduces the load on frequently accessed tables.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a YFS\_REPROCESS\_ERROR table is picked up for purge:

- YFS\_REPROCESS\_ERROR records with State = Fixed or Ignored is processed.
- The last modified time is earlier than the lead time (in days) setup.

**Note:** This purge only reads the rules defined by the hub. Enterprise overridden rules are not considered.

Any enterprise using the Console must schedule purge transactions.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–208** *Reprocess Error Purge Attributes*

Attribute	Value
Base Transaction ID	REPROCESSPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–209** *Reprocess Error Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_REPROCESS_ERROR table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–210** *Reprocess Error Purge Statistics*

Statistic Name	Description
NumReprocessErrorsPurged	Number of reprocess errors purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_REPROCESS\_ERROR table.

### Events Raised

None.

### Tables Purged

YFS\_REPROCESS\_ERROR

### A.4.3.29 Reservation Purge

This purge deletes expired inventory reservations from the system. This reduces the load on frequently accessed tables as well as free up demands that are consumed by expired reservations.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, all records in the YFS\_INVENTORY\_RESERVATION tables are picked up for purge:

- EXPIRATION\_DATE is earlier than the current date or quantity is less than or equal to 0

Any enterprise using the Console must schedule purge transactions.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table A–211 Reservation Purge Attributes*

Attribute	Value
Base Transaction ID	RESERVATIONPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–212 Reservation Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_INVENTORY_RESERVATION table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–213 Reservation Purge Statistics**

Statistic Name	Description
NumReservationsPurged	Number of reservations purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_INVENTORY\_RESERVATION table.

### Events Raised

None.

### Tables Purged

YFS\_INVENTORY\_RESERVATION

### A.4.3.30 Shipment History Purge

This transaction deletes shipments previously archived by the Shipment Purge. See [Section A.4.3.31, "Shipment Purge"](#).

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, all records archived in the history table are picked up for purge:

- The last modification performed on the shipment falls before the lead time (in days) setup.

**Note:** Orders related to the shipments should have been purged by order purge. Shipments should have been closed by the Close Shipment transaction. See [Section A.3.10, "Close Shipment"](#).

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–214 Shipment History Purge Attributes**

Attribute	Value
Base Transaction ID	SHIPMENTHISTPRG
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–215 Shipment History Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Shipment History Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Removes qualifying records from the history tables listed under <a href="#">Tables Purged</a>.</li> <li>N- Test mode. Determines the rows that are removed without actually removing them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–216 Shipment History Purge Statistics**

Statistic Name	Description
NumShipmentHistoriesPurged	Number of shipment histories purged.
NumShipmentLineHistoriesPurged	Number of shipment line histories purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Shipment\_H table.

### Events Raised

None.

### Tables Purged

YFS\_ADDITIONAL\_ATTRIBUTE\_H

YFS\_ADDITIONAL\_DATE\_H

YFS\_AUDIT

YFS\_CONTAINER\_DETAILS\_H

YFS\_CONTAINER\_STS\_AUDIT\_H

YFS\_INSTRUCTION\_DETAIL\_H

YFS\_SHIPMENT\_CONTAINER\_H

YFS\_SHIPMENT\_H

YFS\_SHIPMENT\_LINE\_H

YFS\_SHIPMENT\_LINE\_REQ\_TAG\_H

YFS\_SHIPMENT\_STATUS\_AUDIT\_H

YFS\_SHIPMENT\_TAG\_SERIAL\_H

YFS\_CONTAINER\_ACTIVITY\_H

### A.4.3.31 Shipment Purge

This purge removes shipment data from the system. This reduces the load on frequently accessed tables. This transaction picks up all shipments that have been marked as 'Closed' and archives them into their history tables. See [Section A.4.3.30, "Shipment History Purge"](#). It also archives and purges the shipment's child tables.

This is not a pipeline transaction. It also does not work from the task queue.

Any enterprise using the Console must schedule purge transactions.

**Note:** Orders related to the shipments should have been purged by order purge. Shipments should have been closed by the Close Shipment transaction. See [Section A.3.10, "Close Shipment"](#).

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a shipment is picked up for purge:

- The last modification performed on the shipment should fall before the lead time (in days) setup.
- The value of the ShipmentClosedFlag field should be set to "Y".
- The order record should already be purged for all shipment lines.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-217 Shipment Purge Attributes**

Attribute	Value
Base Transaction ID	SHIPMENTPRG
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–218 Shipment Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Number of Days To Execute	Optional. Maximum number of days before the lead days the agent will look for shipment records to purge.
EnterpriseCode	Optional. Enterprise for which the Shipment Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A-219 Shipment Purge Statistics**

Statistic Name	Description
NumShipmentsPurged	Number of Shipments purged.
NumShipmentLinesPurged	Number of Shipment Lines purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Shipment table.

## Events Raised

None.

## Tables Purged

YFS\_ADDITIONAL\_ATTRIBUTES  
 YFS\_ADDITIONAL\_DATE  
 YFS\_AUDIT  
 YFS\_CONTAINER\_DETAILS  
 YFS\_LOAD\_SHIPMENT\_CHARGE  
 YFS\_MONITOR\_ALERT  
 YFS\_SHIPMENT\_CONTAINER  
 YFS\_SHIPMENT\_STATUS\_AUDIT  
 YFS\_SHIPMENT  
 YFS\_INSTRUCTION\_DETAIL  
 YFS\_SHIPMENT\_MONITOR\_ALERT  
 YFS\_HEADER\_CHARGES  
 YFS\_LINE\_CHARGES  
 YFS\_TAX\_BREAKUP  
 YFS\_SHIPMENT\_HOLD\_TYPE

YFS\_SHIPMENT\_HOLD\_TYPE\_LOG  
YFS\_SHIPMENT\_TAG\_SERIALS  
YFS\_SHIPMENT\_LINE  
YFS\_SHIPMENT\_LINE\_REQ\_TAG  
YFS\_ACTIVITY\_DEMAND  
YFS\_CONTAINER\_STS\_AUDIT  
YFS\_CONTAINER\_ACTIVITY

### A.4.3.32 Shipment Statistics Purge

This transaction deletes the shipment statistics from the table older than the specified retention days.

This agent should be used whenever shipment statistics records need to be removed, such as after application server restart.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, the shipment statistics are picked up for purge:

- The last modification performed on the shipment statistics should fall before the lead time (in days) setup.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-220** *Shipment Statistics Purge Attributes*

Attribute	Value
Base Transaction ID	PRG_SHIP_STATS
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–221** *Shipment Statistics Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Shipment Statistics Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_SHIPMENT_STATISTICS table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Parameters

The following are the statistics parameters for this transaction:

**Table A–222** *Shipment Statistics Purge Statistics*

Parameter	Description
NumShipmentStatisticsPurged	Number of shipment statistics purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_SHIPMENT\_STATISTICS table.

### Events Raised

None.

### Tables Purged

YFS\_SHIPMENT\_STATISTICS

### A.4.3.33 Statistics Purge

This purge removes statistics data from the system. It purges all records older than the specified retention days.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, the statistics detail is picked up for purge:

- The last modification performed on the statistics detail should fall before the lead time (in days) setup.

---

---

**Note:** This purge only reads the rules defined by the hub. Enterprise overridden rules are not considered. This purge should be single threaded when you run it in batch delete mode (BatchDelete=Y).

---

---

---

---

**Note:** Sterling Commerce recommends that this agent be run often. In a production environment, the YFS\_STATISTICS\_DETAIL table can grow very large, very quickly. It does not carry any old data, therefore it is a good practice to purge it aggressively, from once a day to once a week, depending on the table size.

---

---

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–223 Statistics Purge Attributes**

Attribute	Value
Base Transaction ID	STATTLPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–224 Statistics Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
BatchDelete	Required. The mode in which all records get deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Records are deleted in batches.</li> <li>N - Records are deleted one by one.</li> </ul>

**Table A–224 Statistics Purge Criteria Parameters**

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where the YFS_STATISTICS_DETAIL table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–225 Statistics Purge Statistics**

Statistic Name	Description
NumStatisticsPurged	Number of statistics purged

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_STATISTICS\_DETAIL table.

### Events Raised

None.

### Tables Purged

YFS\_STATISTICS\_DETAIL

#### A.4.3.34 User Activity Purge

This purge deletes the user activity data from the system. It purges all records older than the specified retention days, and those records which

have a logged out status. This purge must be single threaded when you run it in batch delete mode (BatchDelete=Y).

The following limitation is assumed when purging records:

This purge do not purge any record if the Application server goes down abruptly because the audit records of users connected to the application server at the time when the server went down cannot be updated. As a result, the last activity time or the logout time is not populated. The purge does not know whether the user has logged out or still logged in. Therefore, you need to manually delete these records.

The following are the attributes for this time-triggered transaction:

**Table A–226 User Activity Purge Attributes**

Attribute	Value
Base Transaction ID	USERACTIVITYPRG
Base Document Type	None
Base Process Type	None
APIs Called	None
User Exits Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–227 User Activity Purge Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.

**Table A–227 User Activity Purge Parameters**

Parameter	Description
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
Number of Records To Buffer	Required. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 100.
BatchDelete	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Records are deleted in batches.</li> <li>N - Records are deleted one by one.</li> </ul>
ColonyID	Required in a multi schema deployment where the YFS_USER_ACTIVITY table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–228 Statistics Purge Statistics**

Statistic Name	Description
NumStatisticsPurged	Number of statistics purged

### Pending Job Count

None.

**Events Raised**

None.

**Tables Purged**

YFS\_USER\_ACTIVITY

**A.4.3.35 User Activity Audit Purge**

This purge removes user activity audit data from the system. It purges all records older than the specified retention days. It purges only those records which have a logged out status (records with a Login\_Type of 'T' or 'N'). This purge should be single threaded when you run it in batch delete mode (BatchDelete=Y).

The following limitation is assumed when purging records:

- This purge does not purge any records if the Application server goes down abruptly because the audit records of users connected to application servers at the time the server went down cannot be updated. As a result, the last activity time or the logout time does not get populated and the purge does not know whether the user was logged out or was still logged in. These records have to be deleted manually.

The following are the attributes for this time-triggered transaction:

**Table A-229 User Activity Audit Purge Attributes**

Attribute	Value
Base Transaction ID	USERACTAUDPURGE
Base Document Type	None
Base Process Type	None
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–230 User Activity Audit Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
Number of Records To Buffer	Required. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 100.
BatchDelete	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Records are deleted in batches.</li> <li>• N - Records are deleted one by one.</li> </ul>
ColonyID	Required in a multi schema deployment where the YFS_USER_ACT_AUDIT table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table A–231 Statistics Purge Statistics*

Statistic Name	Description
NumStatisticsPurged	Number of statistics purged

## Pending Job Count

None.

## Events Raised

None.

## Tables Purged

YFS\_USR\_ACT\_AUDIT

### A.4.3.36 Work Order History Purge

This transaction deletes tasks previously archived by the Work Order Purge. See [Section A.4.3.37, "Work Order Purge"](#).

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a work order that is previously purged by the work order purge agent is picked up for history purge:

- The last modified date of the work order should exceed the retention day period.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table A–232 Work Order History Purge Attributes*

Attribute	Value
Base Transaction ID	WORK_ORDER_HISTORY_PURGE
Base Document Type	Work Order
Base Process Type	VAS
Abstract Transaction	No

**Table A–232 Work Order History Purge Attributes**

Attribute	Value
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–233 Work Order History Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Live	Optional. Mode in which to run. Defaults to N. <ul style="list-style-type: none"> <li>Y - Default value. Removes qualifying records from the history tables listed under <a href="#">Tables Purged</a>.</li> <li>N- Test mode. Determines the rows that are removed without actually removing them.</li> </ul>
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Node	Optional. Node for which the Work Order History Purge needs to be run. If not passed, then all nodes are monitored.

**Table A–233 Work Order History Purge Criteria Parameters**

Parameter	Description
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by WMS time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

The following statistics are tracked for this transaction:

**Table A–234 Work Order History Purge Statistics**

Statistic Name	Description
NumWorkOrderHistoriesPurged	Number of work order histories purged.

**Pending Job Count**

For this transaction, the pending job count is the number of records that can be purged from the YFS\_WORK\_ORDER\_H table.

**Events Raised**

None.

**Tables Purged**

YFS\_AUDIT

YFS\_WO\_APPT\_USER\_H

YFS\_WORK\_ORDER\_H

YFS\_WORK\_ORDER\_APPT\_H

YFS\_WORK\_ORDER\_ACTIVITY\_H

YFS\_WORK\_ORDER\_ACTY\_DTL\_H  
 YFS\_WORK\_ORDER\_AUDT\_DTL\_H  
 YFS\_WORK\_ORDER\_COMPONENT\_H  
 YFS\_WORK\_ORDER\_COMP\_TAG\_H  
 YFS\_WORK\_ORDER\_HOLD\_TYPE\_H  
 YFS\_WORK\_ORDER\_HOLD\_TYPE\_LOG\_H  
 YFS\_WORK\_ORDER\_PROD\_DEL\_H  
 YFS\_WORK\_ORDER\_SERVICE\_LINE\_H  
 YFS\_WORK\_ORDER\_STS\_AUDIT\_H  
 YFS\_WORK\_ORDER\_TAG\_H

### A.4.3.37 Work Order Purge

This time-triggered transaction purges all the work orders for a period greater than the retention days specified in the Work Order Purge criteria and those, which are either in the status of cancelled or completed.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a work order is picked up for purge:

- The last modified date of the work order should exceed the retention day period.
- The order associated with the work order should be purged.
- The work order should be in pickable status by the purge transaction.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table A–235 Work Order Purge Attributes*

Attribute	Value
Base Transaction ID	WORK_ORDER_PURGE
Base Document Type	Work Order
Base Process Type	VAS
Abstract Transaction	No

**Table A–235 Work Order Purge Attributes**

Attribute	Value
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–236 Work Order Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Live	Optional. Mode in which to run. Defaults to Y. <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Node	Optional. Node for which the Work Order Purge needs to be run. If not passed, then all nodes are monitored.

**Table A–236 Work Order Purge Criteria Parameters**

Parameter	Description
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by WMS time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–237 Work Order Purge Statistics**

Statistic Name	Description
NumWorkOrdersPurged	Number of work orders purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_WORK\_ORDER table.

### Events Raised

None.

### Tables Purged

YFS\_AUDIT

YFS\_WO\_APPT\_USER

YFS\_WORK\_ORDER

YFS\_WORK\_ORDER\_ACTIVITY

YFS\_WORK\_ORDER\_ACTY\_DTL

YFS\_WORK\_ORDER\_HOLD\_TYPE  
 YFS\_WORK\_ORDER\_HOLD\_TYPE\_LOG  
 YFS\_WORK\_ORDER\_APPT  
 YFS\_WORK\_ORDER\_AUDT\_DTL  
 YFS\_WORK\_ORDER\_COMPONENT  
 YFS\_WORK\_ORDER\_COMP\_TAG  
 YFS\_WORK\_ORDER\_PROD\_DEL  
 YFS\_WORK\_ORDER\_SERVICE\_LINE  
 YFS\_WORK\_ORDER\_STS\_AUDIT  
 YFS\_WORK\_ORDER\_TAG

#### A.4.3.38 YFS Audit Purge

This purge removes the YFS\_AUDIT table data from the system, which reduces the load on frequently accessed tables. It purges records in the YFS\_AUDIT and the YFS\_AUDIT\_HEADER tables that meet the following conditions:

- YFS\_AUDIT records that have 'modifyts' greater than the retention days specified and the value of table name matches in the YFS\_AUDIT table.
- The last modified time is before the lead time (in days) setup.

---



---

**Note:** The way you configure the YFS Audit Purge may have some effect on the functioning of the Configuration Data Versioning Tool. For more information about configuration of the Data Versioning Tool, see the *Selling and Fulfillment Foundation: Configuration Deployment Tool Guide*.

---



---

When the enterprise extends the entities and sets the extended entities attribute AuditTable="Y", the extended tables are audited and the audit records are inserted in the YFS\_AUDIT table. In order to clean up the audit records, this purge transaction can be used.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–238 YFS Audit Purge Attributes**

Attribute	Value
Base Transaction ID	YFS_AUDIT_PURGE
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–239 YFS Audit Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, this value defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), this value defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Production mode. Deletes records from the regular tables.</li> <li>N - Test mode.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Table Name	Required. The table name for which the audit records need to be purged.

**Table A–239 YFS Audit Purge Criteria Parameters**

Parameter	Description
TableType	<p>Required in a multischema deployment when YFS_AUDIT table may exist in multiple schemas.</p> <p>Valid Values: CONFIGURATION, TRANSACTION, MASTER.</p> <p>If set to CONFIGURATION, the agent runs for the YFS_AUDIT records associated with tables that have TableType as CONFIGURATION; for example, YFS_Organization, YFS_Ship_Node, and so forth.</p> <p>If set to TRANSACTION, the agent runs for the YFS_AUDIT records associated with tables that have TableType as TRANSACTION; for example, YFS_Order_Header, YFS_Shipment, and so forth.</p> <p>Note that the agent would run for all TableTypes that exist in the same schema as the one passed. For example, if set to TRANSACTION, the agent would also run for YFS_AUDIT records associated with tables that have TableType as MASTER, since they reside in the same schema.</p>
ColonyID	<p>Required in a multi schema deployment where the YFS_AUDIT and YFS_AUDIT_HEADER tables may exist in multiple schemas. Runs the agent for the colony.</p>

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–240 YFS Audit Purge Statistics**

Statistic Name	Description
NumAuditRecordsPurged	Number of audit records purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_AUDIT table that match the criteria values.

### Events Raised

None.

### Tables Purged

YFS\_AUDIT, YFS\_AUDIT\_HEADER

### A.4.3.39 YFSInventoryOwnershipAudit Purge

This transaction purges all the records from YFS\_INV\_OWN\_TRANSFER\_RCD prior to the lead days specified in criteria parameters.

### Attributes

Following are the attributes for this time-triggered transaction:

*Table A–241 YFSInventoryOwnership Purge Attributes*

Attribute	Value
Base Transaction ID	PURGE_INV_TRANSFR_RECORD
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

### Criteria Parameters

Following are the criteria parameters for this transaction:

*Table A–242 YFSInventoryOwnership Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, this value defaults to Get, which is the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), this value defaults to 5000.

**Table A–242 YFSInventoryOwnership Purge Criteria Parameters**

Parameter	Description
EnterpriseCode	Optional. The inventory organization for which the YFSInventoryOwnership Audit Purge needs to run. If not passed, all the enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Production mode. Deletes records from the regular tables.</li> <li>• N - Test mode.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds to the PurgeCode used in the Business Rules Purge Criteria.
Lead Days	Number of days before the present date, the agent will purge the records.
ColonyID	Required in a multi schema deployment where the YFS_INV_OWN_TRANSFER_RCD table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None.

**Pending Job Count**

None.

**Tables Purged**

YFS\_INV\_OWN\_TRANSFER\_RCD

**A.4.3.40 Password Reset Request Purge**

This purge deletes password reset request data from the system.

You can use purge codes pseudo-logic to analyze purges.

Any enterprise using the Console must schedule purge transactions.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–243 Password Reset Request Purge Attributes**

Attribute	Value
Base Transaction ID	None
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–244 Password Reset Request Purge Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>

**Table A–244 Password Reset Request Purge Criteria Parameters**

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the PLT_PWD_REQ table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–245 Password Reset Request Purge Statistics**

Statistic Name	Description
NumPasswordRequestPurged	Number of password requests purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the PLT\_PWD\_REQ table.

### Events Raised

None.

### Tables Purged

PLT\_PWD\_REQ

#### A.4.3.41 User Login Failure Purge

This purge deletes data on number of failed login attempts of users from the system.

You can use purge codes pseudo-logic to analyze purges.

Any enterprise using the Console must schedule purge transactions.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–246** *User Login Failure Purge Attributes*

Attribute	Value
Base Transaction ID	None
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–247** *User Login Failure Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under <a href="#">Tables Purged</a> to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>

**Table A–247** *User Login Failure Purge Criteria Parameters*

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the PLT_USER_LOGIN_FAILED table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

The following statistics are tracked for this transaction:

**Table A–248** *User Login Failure Purge Statistics*

Statistic Name	Description
NumUserLoginFailPurged	Number of failed login attempts purged.

**Pending Job Count**

For this transaction, the pending job count is the number of records that can be purged from the PLT\_USER\_LOGIN\_FAILED table.

**Events Raised**

None.

**Tables Purged**

PLT\_USER\_LOGIN\_FAILED

## A.5 Task Queue Syncher Time-Triggered Transactions

Many transactions use the task queue as their work repository. The workflow manager automatically creates tasks for transactions to handle the next processing step, as configured in your pipeline.

In some situations, the task queue repository may become out of date. For example, when reconfiguring the processing pipeline while the

pipeline is active, the queue may go out of synch with the new pipeline configuration.

Alerts that indicate a halt in the lifecycle of a business document may indicate an out-dated task queue repository.

The task queue syncher transactions are designed to update the task queue repository with the latest list of open tasks to be performed by each transaction, based on the latest pipeline configuration.

The available task queue synchroners are:

- [Load Execution Task Queue Syncher](#)
- [Order Delivery Task Queue Syncher](#)
- [Order Fulfillment Task Queue Syncher](#)
- [Order Negotiation Task Queue Syncher](#)

---

---

**Note:** Some of the statistics collected and tracked in Release 8.5 for time-triggered transactions, monitors, and integration and application servers may change with the next release.

---

---

### A.5.1 Load Execution Task Queue Syncher

This transaction synchronizes the task queue for the load execution process type.

You can use the following pseudo-logic to analyze this time-triggered transaction. If the following conditions are met, a task queue for the load execution process type is synchronized:

- `LOAD_CLOSED_FLAG` of Load should not be 'Y'.
- Load should be in a status that is pickable by a transaction in the pipeline.
- There should not be any Task Q record for the load, transaction combination in the Task Q table. In this case, the system inserts one Task Q record for this load, transaction combination with the current database time as the available date.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–249 Load Execution Task Queue Syncher Attributes**

Attribute	Value
Base Transaction ID	TASK_QUEUE_SYNCER_L_D
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–250 Load Execution Task Queue Syncher Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–251 Load Execution Task Queue Syncher Statistics**

Statistic Name	Description
NumTasksCreated	Number of tasks created.

## Pending Job Count

None.

**Events Raised**

None.

**A.5.2 Order Delivery Task Queue Syncher**

This transaction synchronizes the order delivery process type.

**Attributes**

The following are the attributes for this time-triggered transaction:

*Table A–252 Order Delivery Task Queue Syncher Attributes*

Attribute	Value
Base Transaction ID	TASK_QUEUE_SYNCER_O_D
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None

**Criteria Parameters**

The following are the criteria parameters for this transaction:

*Table A–253 Order Delivery Task Queue Syncher Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

The following statistics are tracked for this transaction:

*Table A–254 Order Delivery Task Queue Syncher Statistics*

Statistic Name	Description
NumTasksCreated	Number of tasks created.

**Pending Job Count**

None.

**Events Raised**

None.

**A.5.3 Order Fulfillment Task Queue Syncher**

This transaction synchronizes the order fulfillment process type.

**Attributes**

The following are the attributes for this time-triggered transaction:

*Table A–255 Order Fulfillment Task Queue Syncher Attributes*

Attribute	Value
Base Transaction ID	TASK_QUEUE_SYNCER_O_F
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–256 Order Fulfillment Task Queue Syncher Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–257 Order Fulfillment Task Queue Syncher Statistics**

Statistic Name	Description
NumTasksCreated	Number of tasks created.

### Pending Job Count

None.

### Events Raised

None.

## A.5.4 Order Negotiation Task Queue Syncher

This transaction synchronizes the order negotiation process type.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–258** *Order Negotiation Task Queue Syncher Attributes*

Attribute	Value
Base Transaction ID	TASK_QUEUE_SYNCER_O_N
Base Document Type	Order
Base Process Type	Order Negotiation
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

**Table A–259** *Order Negotiation Task Queue Syncher Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–260** *Order Negotiation Task Queue Syncher Statistics*

Statistic Name	Description
NumTasksCreated	Number of tasks created.

## Pending Job Count

None.

### Events Raised

None.

## A.6 Monitors

Monitors are transactions that watch for processes or circumstances that are out of bounds and then raise alerts.

---

---

**Note:** Some of the statistics collected and tracked in Release 8.5 for time-triggered transactions, monitors, and integration and application servers may change with the next release of Selling and Fulfillment Foundation.

---

---

---

---

**Note:** All Monitors have a `CollectPendingJobs` criteria parameter. If this parameter is set to N, the agent does not collect information on the pending jobs for that monitor. This pending job information is used for monitoring the monitor in the System Management Console. By default, `CollectPendingJobs` is set to Y. It can be helpful to set it to N if one monitor is performing a significant amount of `getPendingJobs` queries and the overhead cost is too high.

---

---

### A.6.1 Availability Monitor

This time-triggered transaction monitors inventory availability. The Availability Monitor raises global alerts when the available inventory falls below the configured quantities on the current day, on subsequent days within the ATP time frame, and on subsequent days outside of the ATP time frame. The quantities for the days outside of the ATP time frame are determined by the maximum monitoring days. Unlike the schedule and release transactions, the Availability Monitor calculates the actual availability beyond the ATP horizon and does not assume infinite inventory.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–261 Availability Monitor Attributes**

Attribute	Value
Base Transaction ID	ATP_MONITOR
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this monitor:

**Table A–262 Availability Monitor Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
MonitorOption	Optional. Specifies how to monitor inventory. Valid values are: <ul style="list-style-type: none"> <li>• 1 - current inventory</li> <li>• 0 - inventory within and outside of the ATP time frame. This is the default value.</li> </ul>
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
InventoryOrganizationCode	Optional. Valid owner inventory organization. Organization to process in this run. If not passed, all inventory organizations are processed.

**Table A–262 Availability Monitor Criteria Parameters**

Parameter	Description
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
Status	The negotiation status you are monitoring.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None.

**Pending Job Count**

None.

**Events Raised**

No events are raised. Individual actions associated with the monitoring rule are run.

Data published to the actions is `AVAILABILITY_MONITOR_dbd.txt`.

**A.6.2 Exception Monitor**

This time-triggered transaction monitors exceptions in your system as noted below. It monitors the exceptions logged in the system and escalates these exceptions:

- If an exception has not been assigned to a user by a certain time
- If an exception has not been resolved by a certain time
- If the active size of the queue is more than a certain maximum size

In order to prevent re-alerts on exceptions during every run of the Exception Monitor, specify a re-alert interval through Alert Management

in the Applications Manager. This attribute is associated with a queue and can be configured for each queue.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–263 Exception Monitor Attributes**

Attribute	Value
Base Transaction ID	EXCEPTION_MONITOR
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this monitor:

**Table A–264 Exception Monitor Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
QueueID	Optional. Defines the Alert Queue into which exceptions from this monitor are stored.
OrganizationCode	Optional. Organization to process in this run. If not passed, all inventory organizations are processed.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.

**Table A–264 Exception Monitor Criteria Parameters**

Parameter	Description
QueueGroup	Optional. Defines the set of Queues for which the exceptions will be monitored. If both QueueId and QueueGroup are supplied, QueueId is ignored.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–265 Exception Monitor Statistics**

Statistic Name	Description
NumInboxProcessed	Number of alerts processed.
NumExceededQueueSizeAlerts	Number of actions raised when the number of unresolved alerts exceeds the queue's maximum active size.
NumUnResolvedAlerts	Number of actions raised when the unresolved alert time of an alert exceeds the queue's resolution time.
NumUnAssignedAlerts	Number of actions raised when the unassigned alert time of an alert exceeds the queue's assignment time.

### Pending Job Count

None.

### Events Raised

No events are raised. Individual actions associated with the monitoring rule are run.

## A.6.3 Inventory Monitor

This time-triggered transaction monitors inventory availability at ship node level. It raises alerts at the ship node level when the available inventory exceeds or drops below the configured quantities.

This monitor uses the OPEN\_ORDER demand type to calculate available inventory at a given node. All supplies assigned to a supply type that is considered by the OPEN\_ORDER demand type are considered. For more information about configuring inventory supply and demand considerations, refer to the *Sterling Global Inventory Visibility: Configuration Guide*.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–266** *Inventory Monitor Attributes*

Attribute	Value
Base Transaction ID	INVENTORY_MONITOR
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	checkAvailability()

### Criteria Parameters

The following are the criteria parameters for this monitor:

**Table A–267** *Inventory Monitor Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

**Table A–267 Inventory Monitor Criteria Parameters**

Parameter	Description
InventoryOrganizationCode	Optional. Valid inventory owner organization. Organization to process in this run. If not passed, all inventory organizations are processed.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
AllowedOverriddenCriteria	If this parameter is set to Y, the overriding value for the agent criteria parameters can be provided in the command line in the following format when triggering the agent:  <AgentCriteriaAttribute> <OverriddenValue>  For more information about passing these attributes, see the <i>Selling and Fulfillment Foundation: Installation Guide</i>
ShipNodes	Optional. Comma-separated list of valid ship nodes that should be processed in this run. If not passed, all the ship nodes are processed.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None.

**Pending Job Count**

None.

### Events Raised

No events are raised. Individual actions associated with the monitoring rule are run.

Data published to the actions is `<INSTALL_DIR>/xapidocs/api_javadocs/dbd/INVENTORY_MONITOR_dbd.txt`.

## A.6.4 Negotiation Monitor

This time-triggered transaction alerts the Enterprise when a negotiation remains in a particular status for a specific amount of time. This also monitors the negotiation expiration date. This time-triggered transaction invokes the actions configured against the negotiation statuses. Configure status Expired (2000) to monitor negotiation expiration date.

Use this monitor in environments where Order or order release has to go through a negotiation phase and you want to monitor the negotiation.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A-268** *Negotiation Monitor Attributes*

Attribute	Value
Base Transaction ID	ORD_NEGOTIATION_MONITOR
Base Document Type	Order
Base Process Type	Order Negotiation
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this monitor:

**Table A–269 Negotiation Monitor Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Negotiation Monitor needs to be run. If not passed, then all enterprises are monitored.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
Status	The negotiation status you are monitoring.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–270 Negotiation Monitor Statistics**

Statistic Name	Description
NumNegotiationsProcessed	Number of negotiations processed.
NumNegotiationsRequiringAlert	Number of negotiations which have at least one alert raised.

## Pending Job Count

None.

### Events Raised

This invokes the actions configured against the negotiation statuses.

Key Data - Not Applicable.

Data Published - `YCP_getNegotiationDetails_output.xml`

## A.6.5 Enhanced Order Monitor

The enhanced order monitor enables you to monitor the following situations:

- Milestone x has not been reached y hours before a given date type.
- Milestone x has not been reached within y hours of a given date type.
- Milestone x has not been reached within y hours of milestone z.
- Milestone x has been reached y hours before a given date type.
- Milestone x has been reached within y hours of a given date type.
- Milestone x has been reached within y hours after milestone z.
- The order has been in status x for y hours.
- Date type x is y hours before date type z.
- Date type x is y hours after date type z.
- The order has been in hold type x for y hours.
- The order has been in hold type x for y hours before date type z.

The order monitor can be configured to monitor the following system date types for Sales Order and Purchase Order document types:

- Actual Order Date - Read from the ORDER\_DATE column of the YFS\_ORDER\_HEADER table.
- Actual Next Iteration Date - Read from the NEXT\_ITER\_DATE column of the YFS\_ORDER\_HEADER table.
- Requested Ship Date - If there is an order release, read from the REQ\_SHIP\_DATE column of the YFS\_ORDER\_RELEASE table. Otherwise, read from the REQ\_SHIP\_DATE of the YFS\_ORDER\_LINE table.

- Expected Ship Date - Read from the EXPECTED\_SHIPMENT\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If it is null, uses the same logic as Requested Ship Date.
- Actual Ship Date - If the date is before 01/01/2500, read from the EXPECTED\_SHIPMENT\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If the date is on or after 01/01/2500, this date type is returned as null.
- Requested Delivery Date - If there is a release, read from the REQ\_DELIVERY\_DATE column of the YFS\_ORDER\_RELEASE table.
- Expected Delivery Date - Read from the EXPECTED\_DELIVERY\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If it is null, uses the same logic as Requested Delivery Date.
- Actual Delivery Date - If the date is before 01/01/2500, read from the EXPECTED\_DELIVERY\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If the date is on or after 01/01/2500, this date type is returned as null.

---

---

**Note:** For Order Fulfillment, Planned Order Execution, Reverse Logistics, and Purchase Order Execution pipelines, the system defined dates such as Shipment and Delivery are stored without a time component. Therefore when you configure a rule using these dates, all time computations are carried out assuming they are always 12:00:00 AM.

---

---

For more information about milestones, date types, and monitoring rules, refer to the appropriate section in this guide, the *Sterling Distributed Order Management: Configuration Guide*, and the *Sterling Reverse Logistics: Configuration Guide*.

---

---

**Important:** If you run the Enhanced Order Monitor, you must configure and run the Close Order time-triggered transaction in all applicable pipelines. For more information about the Close Order time-triggered transaction, see [Section A.3.8, "Close Order"](#).

---

---

---



---

**Note:** The same relog interval is used for all document types.

---



---

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–271** *Enhanced Order Monitor Attributes*

Attribute	Value
Base Transaction ID	ORDER_MONITOR_EX
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this monitor:

**Table A–272** *Enhanced Order Monitor Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Order Monitor needs to be run. If not passed, then all enterprises are monitored.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this monitor:

*Table A–273 Enhanced Order Monitor Statistics*

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumAlertsRaised	Number of alerts raised.

## Pending Job Count

For this transaction the pending job count is the number of open orders with the value of NEXT\_ALERT\_TS less than or equal to (<=) the current date.

## Events Raised

*Table A–274 Events Raised by the Enhanced Order Monitor Transaction*

Transaction/Event	Key Data	Data Published*	Template Support?
ON_AUTO_CANCEL	ORDER_MONITOR_dbd.txt	YFS_ORDER_MONITOR_EX.ON_AUTO_CANCEL.html	Yes
<p>* These files are located in the following directory: &lt;INSTALL_DIR&gt;/xapidocs/api_javadocs/XSD/HTML</p>			

**Note:** The Enhance Order Monitor transaction raises the ON\_AUTO\_CANCEL event, but does not cancel the order. A service on this event should be configured to cancel the order.

## Monitor Rule's Condition Template

If a monitor rule contains a condition, the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX.CONDITION.xml template file is used to obtain both the order details and the evaluating monitor rule details. See the provided <INSTALL\_

DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX\_CONDITION.xml.sample file for more details.

If the <INSTALL\_

DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX\_CONDITION.xml template file does not exist, the MonitorConsolidation->Order element of the default monitor template, the <INSTALL\_

DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX.xml file, is used.

---



---

**Note:** Note: If the default monitor template is used, the MonitorConsolidation-> Order->OrderStatuses-> OrderStatus->MonitorRule element is ignored and is not passed into the condition.

---



---

## A.6.6 Enhanced Return Monitor

The enhanced return monitor allows you to monitor the following situations:

- Milestone x has not been reached y hours before a given date type.
- Milestone x has not been reached within y hours of a given date type.
- Milestone x has not been reached within y hours of milestone z.
- Milestone x has been reached y hours before a given date type.
- Milestone x has been reached within y hours of a given date type.
- Milestone x has been reached within y hours after milestone z.
- The order has been in status x for y hours.
- Date type x is y hours before date type z.
- Date type x is y hours after date type z.

The enhanced return monitor can be configured to monitor the following system date types:

- Actual Order Date - Read from the ORDER\_DATE column of the YFS\_ORDER\_HEADER table
- Requested Ship Date - If there is an order release, read from the REQ\_SHIP\_DATE column of the YFS\_ORDER\_RELEASE table.

Otherwise, read from the REQ\_SHIP\_DATE of the YFS\_ORDER\_LINE table.

- Expected Ship Date - Read from the EXPECTED\_SHIPMENT\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If it is null, uses the same logic as Requested Ship Date.
- Actual Ship Date - If the date is before 01/01/2500, read from the EXPECTED\_SHIPMENT\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If the date is on or after 01/01/2500, this date type is returned as null.
- Requested Delivery Date - If there is a release, read from the REQ\_DELIVERY\_DATE column of the YFS\_ORDER\_RELEASE table. Otherwise, read from the REQ\_DELIVERY\_DATE of the YFS\_ORDER\_LINE table.
- Expected Delivery Date - Read from the EXPECTED\_DELIVERY\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If it is null, uses the same logic as Requested Delivery Date.
- Actual Delivery Date - If the date is before 01/01/2500, read from the EXPECTED\_DELIVERY\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If the date is on or after 01/01/2500, this date type is returned as null.

---

---

**Note:** For Order Fulfillment, Planned Order Execution, Reverse Logistics, and Purchase Order Execution pipelines, the system defined dates such as Shipment and Delivery are stored without a time component. Therefore when you configure a rule using these dates, all time computations are carried out assuming they are always 12:00:00 AM.

---

---

For more information about milestones, date types, and monitoring rules, refer to the appropriate section in this guide, the *Sterling Distributed*

*Order Management: Configuration Guide, and the Sterling Reverse Logistics: Configuration Guide.*

**Important:** If you run the Enhanced Return Monitor, you must configure and run the Close Order time-triggered transaction in all applicable pipelines. For more information about the Close Order time-triggered transaction, see [Section A.3.8, "Close Order"](#).

**Note:** The same relog interval is used for all document types.

## Attributes

The following are the attributes for this time-triggered transaction:

**Table A–275** *Enhanced Order Monitor Attributes*

Attribute	Value
Base Transaction ID	RETURN_MONITOR_EX
Base Document Type	Return Order
Base Process Type	Reverse Logistics
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this monitor:

**Table A–276** *Enhanced Order Monitor Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

**Table A–276 Enhanced Order Monitor Criteria Parameters**

Parameter	Description
EnterpriseCode	Optional. Enterprise for which the Order Monitor needs to be run. If not passed, then all enterprises are monitored.
FromStatus	Optional. Statuses to monitor that are greater than or equal to the passed status.
ToStatus	Optional. Statuses to monitor that are less than or equal to the passed status.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this monitor:

**Table A–277 Enhanced Order Monitor Statistics**

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumAlertsRaised	Number of alerts raised.

### Pending Job Count

For this transaction the pending job count is the number of open orders with the value of NEXT\_ALERT\_TS less than or equal to (<=) the current date.

### Events Raised

No events are raised. Individual actions associated with the monitoring rule are run.

The data published is RETURN\_MONITOR\_EX.xml.

## Monitor Rule's Condition Template

If a monitor rule contains a condition, the `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER_MONITOR_EX_CONDITION.xml` template file is used to obtain both the order details and the evaluating monitor rule details. See the provided `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER_MONITOR_EX_CONDITION.xml.sample` file for more details.

If the `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER_MONITOR_EX_CONDITION.xml` template file does not exist, the `MonitorConsolidation->Order` element of the default monitor template, the `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER_MONITOR_EX.xml` file, is used.

---

**Note:** Note: If the default monitor template is used, the `MonitorConsolidation-> Order->OrderStatuses-> OrderStatus->MonitorRule` element is ignored and is not passed into the condition.

---

## A.6.7 Real-time Availability Monitor

The Real-time Availability Monitor time-triggered transaction monitors the inventory availability of inventory items. It can be configured to raise the `REALTIME_AVAILABILITY_CHANGE` event when the inventory level for a given item changes between the thresholds defined in the Applications Manager in the Global Inventory Visibility module.

It can be run in three modes:

- **Activity Based:** Raises the event in real time every time an item goes above or below one of the thresholds.
- **Quick Sync:** Re-sends the most recently published inventory availability information.
- **Full Sync:** Monitors all of the items regardless of activity and publishes the inventory information for all of the items.

In all cases, the percentage of future inventory availability is used for considering inventory availability at retrieval time. For more information

about future inventory availability, see the *Sterling Global Inventory Visibility: Configuration Guide*.

Inventory available at the current date is considered as on-hand. The processing time in the ATP rules must be set to at least 1 day, or else past due supply is included as part of on-hand inventory. For more information about configuring ATP Rules, see the *Sterling Global Inventory Visibility: Configuration Guide*.

Demand of type OPEN\_ORDER is used in getting the inventory availability picture.

If sourcing is maintained, the Real-time Availability Monitor can either monitor the total availability across nodes or the availability at individual nodes.

When monitoring the total availability across nodes, the Real-time Availability Monitor monitors all nodes in the default distribution group of the inventory organization.

When monitoring the availability at individual nodes, the Real-time Availability Monitor monitors all nodes in a specified distribution group. For more information about configuring distribution groups and node-level inventory monitoring, see the *Sterling Global Inventory Visibility: Configuration Guide*.

Inventory items without an Availability Monitor rule, or with a rule that is disabled, is unable to be processed by this time-triggered transaction.

If configured, the Real-time Availability Monitor also considers the onhand and future inventory availability safety factor during monitoring. For more information about the inventory availability safety factors and the `findInventory()` API, see the *Sterling Global Inventory Visibility: Configuration Guide* and the *Selling and Fulfillment Foundation: Javadocs*.

When the onhand quantity is greater than the configured low threshold, the `REALTIME_ONHAND` alert type is raised, and the alert level is based on the onhand quantity.

When the onhand quantity falls below the configured low threshold, the `REALTIME_FUTURE_MAX` alert type is raised, and the alert level is based on the total future supply (`FutureAvailableQuantity`) with `FirstFutureAvailableDate` set to the date on which the first future

supply is available, and `FutureAvailableDate` set to the date on which the maximum future supply is available.

**Note:** When the Real-time Availability Monitor is run in activity based mode, changing one of the thresholds of an inventory item does not cause the agent to monitor it unless there is a change in activity. For example, if item I with available quantity 700 is being monitored with a low threshold of 600, and the low threshold is then changed to 1000, no event is published unless there is change in I's activity. In order to ensure that in such a scenario I is not left unmonitored, call the `createInventoryActivity` API when changing a monitoring rule for an item.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–278** *Real-time Availability Monitor Attributes*

Attribute	Value
Base Transaction ID	REALTIME_ATP_MONITOR
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	FindInventory

## Criteria Parameters

The following are the criteria parameters for this monitor:

**Table A–279 Real-time Availability Monitor Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
InventoryOrganizationCode	Inventory organization code to use when MonitorOption is passed as 3. The inventory organization has to be an enterprise.  If this is not passed, the monitor runs for all inventory organizations.
MonitorOption	1 - Activity Based (Monitor based on distinct inventory items in YFS_INVENTORY_ACTIVITY table).  2 – Quick Sync (Re-raise event to publish information from the YFS_INVENTORY_ALERT table).  3 – Full Sync (Monitor based on all inventory items maintained by the inventory organization provided. If no InventoryOrganizationCode is provided, all inventory item is monitored).  If not provided, default value is 1.
ItemStatuses	List of valid statuses of items to be processed. Statuses must be separated by a , for example 3000,2000. This is only used when MonitorOption is passed as 2 or 3. If provided, only items with the matching statuses is monitored.

**Table A–279 Real-time Availability Monitor Criteria Parameters**

Parameter	Description
FromAlertTimestamp	<p>This is only used when MonitorOption is passed as 2. If provided, the agent raises the REALTIME_AVAILABILITY_CHANGE event to re-publish inventory availability information which was published between the time that the agent started and FromAlertTimestamp.</p> <p>If not provided, all inventory availability information published before the time that the agent started is re-published.</p>
AllowedOverriddenCriteria	<p>If set to Y, the overridden value for the agent criteria parameters can be provided at the command line while triggering the agent in the following format:</p> <pre data-bbox="768 786 1146 838">&lt;AgentCriteriaAttribute&gt; &lt;OverriddenValue&gt;</pre> <p>For more information about passing these attributes, see the <i>Selling and Fulfillment Foundation: Installation Guide</i>.</p>
FromLastNumberOfHours	<p>This is only used when MonitorOption is passed as 2 to calculate the FromAlertTimestamp parameter, if necessary.</p> <p>If the FromAlertTimestamp parameter is not provided, it is calculated as current timestamp minus FromLastNumberOfHours.</p>
CollectPendingJobs	<p>If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.</p>

**Table A–279 Real-time Availability Monitor Criteria Parameters**

Parameter	Description
RaiseEventsOnAllAvailability Changes	When set to Y, REALTIME_AVAILABILITY_CHANGE event is raised on all availability changes regardless of whether availability exceeds or falls below specified thresholds. This is only used when MonitorOption is passed as 1. Valid values: Y or N. Default value: N.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Statistics Tracked**

None.

**Pending Job Count**

None.

**Events Raised**

The following events are raised by this time-triggered transaction:

**Table A–280 Events Raised by the Realtime Availability Monitor Transaction**

Transaction/Event	Key Data	Data Published*	Template Support?
REALTIME_AVAILABILITY_CHANGE	None	YFS_REALTIME_ATP_MONITOR.REALTIME_AVAILABILITY_CHANGE.html	Yes
<p>* These files are located in the following directory:            &lt;INSTALL_DIR&gt;/xapidocs/api_javadocs/XSD/HTML</p>			

---



---

**Note:** Although described as 'real-time', availability changes may not be triggered immediately as inventory changes occur if the agent has a backlog of messages to process. Furthermore, this monitor exists as a time-triggered transaction, and thus monitors availability of inventory items only when the monitor is triggered based on the configured runtime properties.

---



---

### A.6.8 Shipment Monitor

This time-triggered transaction reports the states of a shipment, based on rules in the YFS\_MONITOR\_RULE table. This transaction enables you to monitor the following situations:

- If the Shipment has been in a status for more than a specified amount of time.
- If a specified date that is associated with the shipment is:
  - n hours before another specified date
  - n hours after another specified date
  - n hours not before another specified date
  - n hours not after another specified date
- If the Shipment has been in a hold type for a specified amount of time.
- If the Shipment has been in a hold type for n hours before a specified date.

Monitoring rules can be configured for shipment's origin and destination points.

Monitoring rules cannot be configured for a shipment's intermediate pickup and drop off points. A shipment has intermediate pickup or drop off only if it has multiple pickup or drop off points. For example, a shipment has more than one loads carrying it. The shipment status on first load deposit, second load deposit, and so forth cannot be monitored. Once the last load deposits the shipment at its destination, then the shipment status can be marked and monitored.

This is not a pipeline transaction. It also does not work from the task queue.

For more information about milestones, date types, and monitoring rules, see the appropriate section in this guide, the *Sterling Distributed Order Management: Configuration Guide*, and the *Sterling Reverse Logistics: Configuration Guide*.

### Attributes

The following are the attributes for this time-triggered transaction:

**Table A–281** *Shipment Monitor Attributes*

Attribute	Value
Base Transaction ID	SHIPMENT_MONITOR
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this monitor:

**Table A–282** *Shipment Monitor Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Shipment Monitor needs to be run. If not passed, then all enterprises are monitored.

**Table A–282 Shipment Monitor Criteria Parameters**

Parameter	Description
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–283 Shipment Monitor Statistics**

Statistic Name	Description
NumShipmentsMonitored	Number of shipments monitored.

### Pending Job Count

For this transaction the pending job count is the number of open shipments with the value of NEXT\_ALERT\_TS less than or equal to (<=) the current date.

### Events Raised

This invokes the actions configured against shipment statuses.

Key Data - Not Applicable.

Data Published - SHIPMENT\_MONITOR.xml

### Monitor Rule's Condition Template

If a monitor rule contains a condition, the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/SHIPMENT\_MONITOR\_CONDITION.xml template file is used to obtain the shipment details and the evaluating monitor rule details. See the provided <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/SHIPMENT\_MONITOR\_CONDITION.xml.sample file for more details.

If the <INSTALL\_  
DIR>/repository/xapi/template/source/smcfs/monitor/SHIPMENT\_  
MONITOR\_CONDITION.xml template file does not exist, the  
MonitorConsolidation->Shipment element of the default monitor  
template, the <INSTALL\_  
DIR>/repository/xapi/template/source/smcfs/monitor/SHIPMENT\_  
MONITOR.xml file, is used.

**Note:** Note: If the default monitor template is used, the  
MonitorConsolidation->Shipment->MonitorRule element is  
ignored and is not passed into the condition.

## A.6.9 Work Order Monitor

This time-triggered transaction alerts the enterprise when a work order remains in a particular state or hold type for a specific amount of time.

Use this monitor to track how long work orders stay in a particular state or hold type.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table A–284 Work Order Monitor Attributes*

Attribute	Value
Base Transaction ID	WORK_ORDER_MONITOR
Base Document Type	Work Order
Base Process Type	VAS Process
Abstract Transaction	No

## Criteria Parameters

The following are the criteria parameters for this monitor:

**Table A–285 Work Order Monitor Criteria Parameters**

Parameter	Description
Action	Required. Triggers the transaction. If left blank it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Work Order Monitor needs to be run. If not passed, then all enterprises are monitored.
Node	Optional. Node for which the Work Order Monitor needs to be run. If not passed, then all nodes are monitored.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

**Table A–286 Work Order Monitor Statistics**

Statistic Name	Description
NumWorkOrdersMonitored	Number of work orders monitored.

## Pending Job Count

For this transaction the pending job count is the number of Work Orders that are monitored, where NEXT\_ALERT\_TS less than or equal to ( $\leq$ ) current date.

### Events Raised

No events are raised. Individual actions associated with the monitoring rule are run. Data published to the actions is workOrder\_dbd.txt.

### Monitor Rule's Condition Template

If a monitor rule contains a condition, the `<INSTALL_  
DIR>/repository/xapi/template/source/smcfs/monitor/monitor/WOR  
K_ORDER_MONITOR_CONDITION.xml` template file is used to obtain the  
work order details and the evaluating monitor rule details. See the  
provided `<INSTALL_  
DIR>/repository/xapi/template/source/smcfs/monitor/WORK_ORDER_  
MONITOR_CONDITION.xml.sample` file for more details.

If the `<INSTALL_  
DIR>/repository/xapi/template/source/smcfs/monitor/WORK_ORDER_  
MONITOR_CONDITION.xml` template file does not exist, the  
MonitorConsolidation->WorkOrder element of the default monitor  
template, the `<INSTALL_  
DIR>/repository/xapi/template/source/smcfs/monitor/WORK_ORDER_  
MONITOR.xml` file, is used.

**Note:** If the default monitor template is used, the MonitorConsolidation->WorkOrder->->MonitorRule element is ignored and is not passed into the condition.

# B

## Order Modification Types

---

The following are the default order modification types and their associated modification levels:

**Table B–1 Order Document Modification Types**

Modification Types	Description	Modification Levels
Ad	An instruction can be added to an order document's header, line, or shipment. For example, you may want to add an instruction stating that a line item needs to be gift wrapped.	Header Line Shipment Receipt
Add Line	A line can be added to an order document's header, release, negotiation, or shipment. <b>Important:</b> When adding a line to an order, the Add Line modification type does not get audited, if the prices are not configured.	Header Release Negotiation Shipment
Add Note	A note can be added to an order document's header or release.	Header Release
Add Option	An option can be added to a provided service or delivery service order line.	Line

**Table B-1 Order Document Modification Types**

Modification Types	Description	Modification Levels
Add Quantity	Additional quantity can be added to an order document's line or release line.	Line Release Line
Add/Remove Additional Date	<p>A date type used for shipment monitoring (such as, Ship Date) can either be added to or removed from an order document's shipment.</p> <p>For example, you may want to add an additional delivery date used by your organization to monitor shipments.</p>	Shipment
Add/Remove Charge	<p>A charge can either be added to or removed from an order document's shipment.</p> <p>For example, if a shipment contains hazardous materials and your organization has an extra shipping charge for shipment of hazardous materials, you can add an extra charge to the shipment.</p>	Shipment
Add/Remove Container	A container can either be added to or removed from an order document's shipment.	Shipment
Associate Delivery Line With Product Line	When the delivery method of a product order line is delivery, the product line can be associated to a delivery line to indicate how the product line is delivered.	Line

**Table B–1 Order Document Modification Types**

Modification Types	Description	Modification Levels
Associate Product Line With Delivery Line	When the delivery method of a product order line is delivery, the product line can be associated to a delivery line to indicate how the product line is delivered.	Line
Associate Product Line With Service Line	A provided service can be associated to a product line to indicate that the service is somehow dependent on the product line.	Line
Associate Service Line With Product Line	A provided service can be associated to a product line to indicate that the service is somehow dependent on the product line.	Line
Attribute Modification	A receipts attributes can be modified. For a list of attributes that can be modified, see the <i>changeReceipt</i> API in the <i>Selling and Fulfillment Foundation: Javadocs</i> .	Receipt
Backorder	An order document's line, release, or release line can be backordered.  For example, if an order is released to a node and the node does not have enough quantity to fulfill the order, they can backorder the release.	Line Release Release Line
Cancel	An order document's header, line, release, or release line can be manually cancelled from the Application Consoles.	Header Line Release Release Line

*Table B-1 Order Document Modification Types*

<b>Modification Types</b>	<b>Description</b>	<b>Modification Levels</b>
Change Additional Address	A modification can be made to the fields of any additional addresses that may have been configured for an order document's header or line.	Header Line
Change Appointment	Appointments can be taken and changed for delivery and provided service order lines.	Line
Change Bill To	A modification can be made to any bill to address field associated with an order document's header or release.	Header Release

**Table B-1 Order Document Modification Types**

Modification Types	Description	Modification Levels
Change Bundle Definition	<p>The existing bundle definition can be replaced with the new bundle definition.</p> <p>For example, you can change an existing bundle definition by passing the 'REPLACE_BUNDLE' action to the bundle parent. All the components passed remain with order and as well as with bundle. All remaining components are deleted.</p> <p><b>Important:</b> In addition to this, the modification type DELETE is executed on all the components getting removed and modification type ADD_LINE is executed on components getting added. This modification is applied to bundle parent's immediate components.</p>	Line
Change Buyer Organization	<p>The buyer organization associated with an order document's header can be changed. This modification can only be made in the Order Detail screen.</p>	Header

*Table B-1 Order Document Modification Types*

Modification Types	Description	Modification Levels
Change Carrier	<p>A modification can be made to the Carrier/Service or Carrier field associated with an order document's header, line, or release.</p> <p>For example, you can change the carrier and service from UPS Next Day Air to FedEx Express Saver Pack.</p> <p><b>Important:</b> If you want this modification type to be allowed, Change Carrier Service Code must also be allowed.</p>	Header Line Release
Change Carrier Account No	<p>A modification can be made to the Carrier Account # field associated with an order document's header, line, or release.</p>	Header Line Release
Change Carrier Service Code	<p>A modification can be made to the Carrier/Service field associated with an order document's header, line, or release.</p> <p>For example, you can change the carrier and service from UPS Next Day Air to FedEx Express Saver Pack.</p> <p><b>Important:</b> If you want this modification type to be allowed, Change Carrier must also be allowed.</p>	Header Line Release
Change Contact Info	<p>A modification can be made the fields for the Buyer/Seller contact information associated with an order document's header.</p>	Header

**Table B–1 Order Document Modification Types**

Modification Types	Description	Modification Levels
Change Cost	A adjustment can be made to the Unit Cost field associated with an order document's release or release line.	Release Release Line
Change Currency	The currency associated with an order document's header can be changed. Upon a change to the currency, Selling and Fulfillment Foundation automatically re-prices the order. However, pre-existing charges and taxes have to be converted manually.	Header
Change Custom Date	<p>A modification can be made to the date type fields used for order monitoring associated with an order document's header, line, or release.</p> <p>The following custom date fields can be modified when this modification type is allowed:</p> <ul style="list-style-type: none"> <li>• Requested</li> <li>• Expected</li> <li>• Actual</li> </ul> <p>For example, if there is a delay in a release's processing, you can change the expected delivery date.</p>	Header Line Release

**Table B-1 Order Document Modification Types**

Modification Types	Description	Modification Levels
Change Delivery Code	<p>A modification can be made to the Delivery Code field associated with an order document's header, line, or release.</p> <p>For example, if you want to indicate that an order's freight charges are paid by the Enterprise, you can choose the ENTERPRISE delivery code.</p>	Header Line Release
Change Delivery Method	<p>A product order line indicates how the product is sent to its final destination. It can be changed to SHIP, DELIVER, or PICKUP.</p>	Line
Change Expiration Date	<p>A modification can be made to the expiration date associated with an order document's negotiation.</p>	Negotiation
Change Freight Terms	<p>A modification can be made to the Freight Terms field associated with an order document's header, line, or release.</p> <p>For example, you can change an order line's freight term from CIF (Cost Insurance and Freight) to CFR (Cost and Freight).</p>	Header Line Release

**Table B–1 Order Document Modification Types**

<b>Modification Types</b>	<b>Description</b>	<b>Modification Levels</b>
Change Instruction	<p>A modification can be made to an instruction associated with an order document's header, line, or shipment.</p> <p>The following instruction fields can be modified when this modification type is allowed:</p> <ul style="list-style-type: none"> <li>• Instruction Type</li> <li>• Text</li> <li>• URL</li> </ul>	Header Line Shipment Receipt
Change Item Description	A modification can be made to the Description field of an item associated with an order document's line.	Line
Change Iteration	<p>A modification can be made to the iteration fields associated with an order document's header or line.</p> <p>For example, you can change the next iteration date of a master order to a time in the future.</p>	Header Line
Change Mark For	A modification can be made to the fields of the mark for address associated with an order document's header, line, or release.	Header Line Release
Change Order Name	A modification can be made to the Order Name field associated with an order document's header.	Header

**Table B-1 Order Document Modification Types**

Modification Types	Description	Modification Levels
Change Other Attributes	A modification can be made to fields that do not have system or user-defined modification types associated with them.	Header Line Release Negotiation Negotiation Line Shipment
Change Other Relationships	Not used in this version.	Shipment
Change Payment Method	A modification can be made to the Payment Type field associated with an order document's header or release.  For example, you can change an order's payment type from Check to Credit Card.	Header Release
Change Payment Rule ID	The Payment Rule field associated with an order document's header can be changed.  For example, you can change the payment rule from the default rule to a custom rule that pertains to the order.	Header
Change Payment Status	The Payment Status field associated with an order document's header can be changed.  For example, you can change an order's payment status from Await Authorization to Authorized.	Header
Change Price	Charges can be added to an order document's header or line.	Header Line

**Table B–1 Order Document Modification Types**

<b>Modification Types</b>	<b>Description</b>	<b>Modification Levels</b>
Change Receiving Node	<p>The Receiving Node field associated with an order document's line can be changed.</p> <p>For example, if for some reason it has been determined that an order line's original receiving node cannot receive the line, you can change it to another receiving node.</p>	Line
Change References	A modification can be made to the name/value pair in the YFS_REFERENCE_TABLE using APIs.	Header Line
Change Requested Ship Date	<p>A modification can be made to the Requested Ship Date associated with an order document's header, line, or release.</p> <p>For example, if the customer decides they want an order to be shipped on a date later than what they originally requested, you can change the requested shipment date.</p>	Header Line Release
Change Schedule	A modification can be made to schedule attributes, such as expected ship date, expected delivery date, and lot number, associated with an order document's header, line, release, or release line.	Header Line Release Release Line

*Table B-1 Order Document Modification Types*

Modification Types	Description	Modification Levels
Change Schedule Rule ID	A modification can be made to the schedule rule associated with an order document's header. This allows the user to select the scheduling rule they want to use for the order from the Scheduling Rule drop-down list on the Schedule Order popup window.	Header
Change Ship Node	The Ship Node field associated with an order document's header or line can be changed.  For example, if for some reason it has been determined that an order line's original ship node cannot handle the order line, you can change it to another node.	Header Line
Change Ship To	A modification can be made to the fields of a ship to address associated with an order document's header, line, or release.	Header Line Release
Change Status	The order status (such as, Created) associated with an order document's header, line, release, release line, or negotiation can be changed.  <b>Note:</b> Only order statuses existing in process type repositories are affected by this modification type. Actions performed against order documents, such as putting an order on hold or canceling an order, are not impacted.	Header Line Release Release Line Negotiation

**Table B-1 Order Document Modification Types**

<b>Modification Types</b>	<b>Description</b>	<b>Modification Levels</b>
Change Tax	A modification can be made to the Tax Amount associated with an order document's header or line.	Header Line
Delete Shipment	An order document's shipment can be deleted.	Shipment
Hold	An order document's header or release can be manually put on hold.  For example, you may want to perform a security check on a particular Buyer, you can then place the order on hold until you clear the necessary information before the order is scheduled.	Header Release
Include In Load	An order document's shipment can be included in a load document.	Shipment
Include Shipment in Delivery Plan	An order document's shipment can be included in a delivery plan.	Shipment
Pack Shipment	An order document's shipment can be packed.	Shipment
Price Program	The price program associated with an order document's header can be changed.	Header
Receipt Complete	An order document's receipt can be marked as complete.	Receipt
Release from Hold	An order document's header can be released from hold.	Header
Remove Delivery Line From Product Line Association	Delivery lines can be removed from product order lines.	Line

**Table B-1 Order Document Modification Types**

<b>Modification Types</b>	<b>Description</b>	<b>Modification Levels</b>
Remove Line	A line can be removed from an order document's header, line, and shipment.	Header Line Shipment
Remove Option	Options can be removed from delivery and provided services.	Line
Remove Product Line From Delivery Line Association	Product lines can be removed from delivery lines.	Line
Remove Product Line From Service Line Association	Product lines can be removed from provided service order lines.	Line
Remove Service Line From Product Line Association	Provided service lines can be removed from product order lines.	Line
Remove Shipment From Delivery Plan	An order document's shipment can be removed from a delivery plan.	Shipment
Short	An order document's header, line, release, release line, and receipt can be shorted. This occurs when there is a shortage in the expected quantity.	Header Line Release Release Line Receipt
Split Line	An order document's line or release line can be split into multiple lines.	Line Release Line
Unpack Shipment	An order document's shipment can be unpacked.	Shipment

**Table B-1 Order Document Modification Types**

<b>Modification Types</b>	<b>Description</b>	<b>Modification Levels</b>
Unreceive	An order document's receipt can be fully or partially unreceived. This moves the quantity you are identifying as unreceived back to Shipped status.	Receipt
Unschedule	An order document's header or line can be unscheduled from a scheduled node. This cancels any inventory that has been reserved for the order at the scheduled node.	Header Line



# Condition Builder Attributes

---

Statements in the condition builder are built using attributes that are defined throughout the Applications Manager. This appendix describes all of those attributes for each process type.

Click one of the links below to be taken to the appropriate condition builder attributes description.

## **Sales Order**

- [Order Fulfillment](#)
- [Order Negotiation](#)
- [Outbound Shipment](#)
- [Sales Order Receipt](#)

## **Planned Order**

- [Planned Order Execution](#)
- [Planned Order Negotiation](#)

## **Return Order**

- [Reverse Logistics](#)
- [Return Shipment](#)
- [Return Receipt](#)

## **Template Order**

- [Template Order](#)

## **Purchase Order**

- Purchase Order Execution
- Purchase Order Negotiation
- Inbound Shipment
- Purchase Order Receipt

## **Transfer Order**

- Transfer Order Execution
- Transfer Order Delivery
- Transfer Order Receipt

## **Master Order**

- Master Order Fulfillment

## **Load**

- Load Execution

## **General**

- General
- WMS Putaway
- WMS Layout Definition
- WMS Inventory
- Trailer Loading
- Task Execution
- Move Request Execution
- Manifesting
- Over Pack Build

## **Count**

- Count Execution

**Container**

- [Pack Process](#)

**Wave**

- [Outbound Picking](#)

**Work Order**

- [VAS Process](#)

**Item-Based Allocation (IBA)**

- [Item-Based Allocation \(IBA\) Order](#)

## F.1 Sales Order

### F.1.1 Order Fulfillment

*Table F-1 Order Fulfillment Condition Builder Attributes*

Attribute	Description
<b>Order Attributes</b>	
Condition Variable 1	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
Condition Variable 2	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
Delivery Method	The delivery method of the order (shipment, pickup or delivery).
Disposition Code	The disposition code of the item. This field is only applicable for Reverse Logistics and Supply Collaboration.

**Table F–1 Order Fulfillment Condition Builder Attributes**

<b>Attribute</b>	<b>Description</b>
Line Type	The type of the order line. Selling and Fulfillment Foundation has no application logic associated with the order line type. This field can be set up as per your business practices.
Order Type	The type of the order. Selling and Fulfillment Foundation has no application logic associated with the order type. This field can be set up as per your business practices.
Payment Status	The payment status of the order.
Sale Voided	The flag indicating whether the order is voided.
Transaction ID	The ID of the last transaction that was run on the order.
<b>Participant Attributes</b>	
Bill To ID	The ID of the bill to address for the order.
Buyer Organization Code	The code of the organization that is buying the goods or services.
Enterprise Code	The code of the enterprise on the order.
Receiving Node	The node that receives the shipment for the order.
Seller Organization Code	The code of the organization that is selling the goods or services.
Ship Node	The node that ships the shipment for the order.
Ship Node Interface Type	The interface type of the ship node on the order (External Application, Console, Sterling WMS, or WMS 6.2).
Ship To ID	The ID of the ship to address for the order.
Supplier Code	The code of the supplier for the order.
<b>Item Attributes</b>	
Item ID	The ID of the item on the order line.

**Table F–1 Order Fulfillment Condition Builder Attributes**

Attribute	Description
Item Group Code	The group code of the service item. For example, if the service is a provided service item, then the item group code is PS.
Product Line	The product line of the item on the order line.
<b>Sourcing Attributes</b>	
Fulfillment Type	The fulfillment type of the order.
Intentional Backorder	The flag indicating whether the order was intentionally dropped into backordered status at order creation.
Is Firm Predefined Node	The flag indicating whether the node on the order is a firm predefined node.
Order Sourcing Classification	The order sourcing classification of the order.
Reservation Mandatory	The flag indicating whether the reservation is mandatory.
<b>Related Order Attributes</b>	
Chain Type	The chain type of the order.
Is Chained Line	The flag indicating whether the order line is chained with another order line.
Is Derived Line	The flag indicating whether the order line is derived from another order line.
Order Purpose	The purpose of the order. If this is an exchange order, this field is set to EXCHANGE.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.1.2 Order Negotiation

*Table F–2 Order Negotiation Condition Builder Attributes*

Attribute	Description
Enterprise Code	The code of the enterprise on the order.
Initiator Organization Code	The code of the organization that initiates the negotiation.
Negotiator Organization Code	The code of the organization that can accept, counter-offer, or reject the initiator's offer.
Negotiation Pipeline Key	The key of the negotiation pipeline this order is going through.
Negotiation Number	The negotiation number of this order.
Negotiation Rule Key	The key of the negotiation rule for this order.
Header Entity	The entity for which the negotiation was initiated. Currently, the only applicable entity is Order.
Negotiation Status	The status of the negotiation for this order.
Document Type	The document type for this order. Typical value is Sales Order.
Freight Terms	The freight terms for this order.
Payment Terms	The payment terms for this order.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.1.3 Outbound Shipment

**Table F-3 Outbound Shipment Condition Builder Attributes**

Attribute	Description
Enterprise Code	The code of the enterprise on the outbound shipment.
Buyer Organization Code	The code of the organization that is buying the goods or services.
Seller Organization Code	The code of the organization that is selling the goods or services.
Ship Node	The node that ships this shipment.
Ship Node Interface Type	The interface type of the ship node on the order (External Application, Console, Sterling WMS, or WMS 6.2).
Receiving Node	The node that receives this shipment.
Ship Mode	The shipment mode that is used for the shipment. For example, Parcel, Truck Load, Less-Than Truck Load.
Freight Terms	The freight terms for this shipment.
Carrier Type	The shipment's carrier type for this shipment.
Hazardous Materials Flag	The flag indicating whether these materials are hazardous.
ESP Check Required	The flag indicating whether an Economic Shipping Parameters check is required at shipment consolidation time.
Is Appointment Required	The flag indicating whether an appointment is required for a service execution.
Routing Guide Maintained	The flag indicating whether a routing guide is maintained for this shipment.
Carrier	The carrier for the shipment.

Table F–3 Outbound Shipment Condition Builder Attributes

Attribute	Description
Real-time Integration with WMS 6.2	The flag indicating whether the node this shipment is shipping from is integrating with the Sterling WMS. Setting this field to N means that you are integrating with WMS 6.2, or any other warehouse management system.
Manually Entered	The flag indicating whether or not the shipment was entered through the Console.
Delivery Code	The code of the entity that pays for the transportation costs.
Country	The country that the shipment is being shipped to.
Delivery Method	The delivery method of the shipment (shipment, pickup or delivery).
Is Serial Requested	The flag indicating whether the shipment has any line with a specific serial number passed. If that is the case, a different outbound shipment process can be selected in the pipeline.
Is Provided Service	The flag indicating whether the shipment has an associated provided service item.
Shipment Type	Indicates a set of shipments that are of the same nature.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i>.</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

### F.1.4 Sales Order Receipt

The Sales Order Receipt condition builder attributes are identical to the [Return Receipt](#) attributes.

## F.2 Planned Order

### F.2.1 Planned Order Execution

The Planned Order Execution condition builder attributes are identical to the [Order Fulfillment](#) attributes.

### F.2.2 Planned Order Negotiation

The Planned Order Negotiation condition builder attributes are identical to the [Order Negotiation](#) attributes.

## F.3 Return Order

### F.3.1 Reverse Logistics

*Table F–4 Return Fulfillment Condition Builder Attributes*

Attribute	Description
<b>Order Attributes</b>	
Condition Variable 1	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
Condition Variable 2	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
Delivery Method	The delivery method of the return (shipment, pickup or delivery).
Disposition Code	The disposition code of the item.
Line Type	The type of the return line. Selling and Fulfillment Foundation has no application logic associated with the return line type. This field can be set up as per your business practices.

**Table F-4 Return Fulfillment Condition Builder Attributes**

<b>Attribute</b>	<b>Description</b>
Order Type	The type of the return. Selling and Fulfillment Foundation has no application logic associated with the return type. This field can be set up as per your business practices.
Payment Status	The payment status of the return.
Sale Voided	The flag indicating whether the return is voided.
Transaction ID	The ID of the last transaction that was run on the return.
<b>Participant Attributes</b>	
Bill To ID	The ID of the bill to address for the return.
Buyer Organization Code	The code of the organization that is buying the goods or services.
Enterprise Code	The code of the enterprise on the return.
Receiving Node	The node that receives the shipment for the return.
Seller Organization Code	The code of the organization that is selling the goods or services.
Ship Node	The node that be ships the shipment for the return.
Ship Node Interface Type	The interface type of the ship node on the return (External Application, Console, Sterling WMS, or WMS 6.2).
Ship To ID	The ID of the ship to address for the return.
Supplier Code	The code of the supplier for the return.
<b>Item Attributes</b>	
Item ID	The ID of the item on the return line.
Item Group Code	The group code of the service item. For example, if the service is a provided service item, then the item group code is PS.
Product Line	The product line of the item on the return line.

**Table F–4 Return Fulfillment Condition Builder Attributes**

Attribute	Description
<b>Sourcing Attributes</b>	
Fulfillment Type	The fulfillment type of the return.
Intentional Backorder	The flag indicating whether the return was intentionally dropped into backordered status at return creation.
Is Firm Predefined Node	The flag indicating whether the node on the return is a firm predefined node.
Order Sourcing Classification	The order sourcing classification of the return.
Reservation Mandatory	The flag indicating whether the reservation is mandatory.
<b>Related Order Attributes</b>	
Chain Type	The chain type of the return.
Is Chained Line	The flag indicating whether the return line is chained with another return line.
Is Derived Line	The flag indicating whether the return line is derived from another return line.
Order Purpose	This field is only applicable to sales orders.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

### F.3.2 Return Shipment

The Return Shipment condition builder attributes are identical to the [Outbound Shipment](#) attributes.

### F.3.3 Return Receipt

*Table F-5 Return Receipt Condition Builder Attributes*

Attribute	Description
Document Type	The document type on the receipt. Typical value is Return Order.
Enterprise Code	The code of the enterprise that owns the receipt.
Seller Organization Code	The code of the organization that is selling the goods or services.
Ship Node	The node where the shipment was shipped out of.
Buyer Organization Code	The code of the organization that is buying the goods or services.
Receiving Node	The node where the shipment was received.
Receiving Node Interface Type	The interface type of the receiving node on the order (External Application, Console, Sterling WMS, or WMS 6.2).
Ship Mode	The shipment mode that is used for the shipment. For example, Parcel, Truck Load, Less-Than Truck Load.
Freight Terms	The freight terms on the receipt.
Carrier Type	The carrier type on the receipt.
Is Hazardous Material	The flag indicating whether there are hazardous materials that are being received.
Is Inspection Pending	The flag indicating whether there is an inspection pending on this return.

**Table F–5 Return Receipt Condition Builder Attributes**

Attribute	Description
Is Receiving Node Integrated Real Time	The flag indicating whether the receiving node is integrating with WMS 6.2, or with another WMS system.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.4 Template Order

The Template Order condition builder attributes are identical to the [Order Fulfillment](#) attributes.

## F.5 Purchase Order

### F.5.1 Purchase Order Execution

**Table F–6 Purchase Order Execution Condition Builder Attributes**

Attribute	Description
<b>Order Attributes</b>	
Condition Variable 1	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
Condition Variable 2	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.

**Table F–6 Purchase Order Execution Condition Builder Attributes**

<b>Attribute</b>	<b>Description</b>
Delivery Method	The delivery method of the inbound order (shipment, pickup or delivery).
Disposition Code	The disposition code of the item.
Line Type	The type of the inbound order line. Selling and Fulfillment Foundation has no application logic associated with the inbound order line type. This field can be set up as per your business practices.
Order Type	The type of the inbound order. Selling and Fulfillment Foundation has no application logic associated with the inbound order type. This field can be set up as per your business practices.
Payment Status	The payment status of the inbound order.
Sale Voided	The flag indicating whether the inbound order is voided.
Transaction ID	The ID of the last transaction that was run on the inbound order.
<b>Participant Attributes</b>	
Bill To ID	The ID of the bill to address for the inbound order.
Buyer Organization Code	The code of the organization that is buying the goods or services.
Enterprise Code	The code of the enterprise on the inbound order.
Receiving Node	The node that receives the shipment for the inbound order.
Seller Organization Code	The code of the organization that is selling the goods or services.
Ship Node	The node that ships the shipment for the inbound order.
Ship Node Interface Type	The interface type of the ship node on the inbound order (External Application, Console, Sterling WMS, or WMS 6.2).

**Table F-6 Purchase Order Execution Condition Builder Attributes**

<b>Attribute</b>	<b>Description</b>
Ship To ID	The ID of the ship to address for the inbound order.
Supplier Code	The code of the supplier for the inbound order.
<b>Item Attributes</b>	
Item ID	The ID of the item on the inbound order line.
Item Group Code	The group code of the service item. For example, if the service is a provided service item, then the item group code is PS.
Product Line	The product line of the item on the inbound order line.
<b>Sourcing Attributes</b>	
Fulfillment Type	The fulfillment type of the inbound order.
Intentional Backorder	The flag indicating whether the inbound order was intentionally dropped into backordered status at inbound order creation.
Is Firm Predefined Node	The flag indicating whether the node on the inbound order is a firm predefined node.
Order Sourcing Classification	The order sourcing classification of the inbound order.
Reservation Mandatory	The flag indicating whether the reservation is mandatory.
<b>Related Order Attributes</b>	
Chain Type	The chain type of the inbound order.
Is Chained Line	The flag indicating whether the inbound order line is chained with another inbound order line.
Is Derived Line	The flag indicating whether the inbound order line is derived from another inbound order line.

**Table F–6 Purchase Order Execution Condition Builder Attributes**

Attribute	Description
Order Purpose	This field is only applicable to sales orders.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i>.</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.5.2 Purchase Order Negotiation

**Table F–7 Purchase Order Negotiation Condition Builder Attributes**

Attribute	Description
Enterprise Code	The code of the enterprise on the inbound order.
Initiator Organization Code	The code of the organization that initiates the negotiation.
Negotiator Organization Code	The code of the organization that can accept, counter-offer, or reject the initiator's offer.
Negotiation Pipeline Key	The key of the negotiation pipeline this inbound order is going through.
Negotiation Number	The negotiation number of this inbound order.
Negotiation Rule Key	The key of the negotiation rule for this inbound order.
Header Entity	The entity for which the negotiation was initiated. Currently, the only applicable entity is Order.
Negotiation Status	The status of the negotiation for this inbound order.
Document Type	The document type for this inbound order. Typical value is Purchase Order.
Freight Terms	The freight terms for this inbound order.

**Table F–7 Purchase Order Negotiation Condition Builder Attributes**

Attribute	Description
Payment Terms	The payment terms for this inbound order.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

### F.5.3 Inbound Shipment

The Inbound Shipment condition builder attributes are identical to the [Outbound Shipment](#) attributes.

### F.5.4 Purchase Order Receipt

The Purchase Order Receipt condition builder attributes are identical to the [Return Receipt](#) attributes.

## F.6 Transfer Order

### F.6.1 Transfer Order Execution

The Transfer Order Execution condition builder attributes are identical to the [Order Fulfillment](#) attributes.

### F.6.2 Transfer Order Delivery

The Transfer Order Delivery condition builder attributes are identical to the [Outbound Shipment](#) attributes.

### F.6.3 Transfer Order Receipt

The Transfer Order Receipt condition builder attributes are identical to the [Return Receipt](#) attributes.

## F.7 Master Order

### F.7.1 Master Order Fulfillment

*Table F–8 Master Order Fulfillment Condition Builder Attributes*

Attribute	Description
<b>Master Order Attributes</b>	
Condition Variable 1	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
Condition Variable 2	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
Delivery Method	The delivery method of the order (shipment, pickup or delivery).
Disposition Code	The disposition code of the item. This field is only applicable for Reverse Logistics and Supply Collaboration.
Line Type	The type of the order line. Selling and Fulfillment Foundation has no application logic associated with the order line type. This field can be set up as per your business practices.
Order Type	The type of the order. Selling and Fulfillment Foundation has no application logic associated with the order type. This field can be set up as per your business practices.
Payment Status	The payment status of the order.
Sale Voided	The flag indicating whether the order is voided.
Transaction ID	The ID of the last transaction that was run on the order.

**Table F–8 Master Order Fulfillment Condition Builder Attributes**

Attribute	Description
<b>Participant Attributes</b>	
Bill To ID	The ID of the bill to address for the order.
Buyer Organization Code	The code of the organization that is buying the goods or services.
Enterprise Code	The code of the enterprise on the order.
Receiving Node	The node that receives the shipment for the order.
Seller Organization Code	The code of the organization that is selling the goods or services.
Ship Node	The node that ships the shipment for the order.
Ship Node Interface Type	The interface type of the ship node on the order (External Application, Console, Sterling WMS, or WMS 6.2).
Ship To ID	The ID of the ship to address for the order.
Supplier Code	The code of the supplier for the order.
<b>Item Attributes</b>	
Item ID	The ID of the item on the order line.
Item Group Code	The group code of the service item. For example, if the service is a provided service item, then the item group code is PS.
Product Line	The product line of the item on the order line.
<b>Sourcing Attributes</b>	
Fulfillment Type	The fulfillment type of the order.
Intentional Backorder	The flag indicating whether the order was intentionally dropped into backordered status at order creation.
Is Firm Predefined Node	The flag indicating whether the node on the order is a firm predefined node.
Order Sourcing Classification	The order sourcing classification of the order.

**Table F–8 Master Order Fulfillment Condition Builder Attributes**

Attribute	Description
Reservation Mandatory	The flag indicating whether the reservation is mandatory.
<b>Related Master Order Attributes</b>	
Chain Type	The chain type of the order.
Is Chained Line	The flag indicating whether the order line is chained with another order line.
Is Derived Line	The flag indicating whether the order line is derived from another order line.
Order Purpose	The purpose of the order. If this is an exchange order, this field is set to EXCHANGE.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.8 Load Execution

**Table F–9 Load Execution Condition Builder Attributes**

Attribute	Description
Load Type	The type of the load document.
Enterprise Code	The code of the enterprise on the load document.
Owner Organization Code	The code of the organization that owns the load document.
Carrier	The carrier used to carry the load.

**Table F–9 Load Execution Condition Builder Attributes**

Attribute	Description
Carrier Service Code	The code of the carrier service used to carry the load.
Ship Mode	The shipment mode that is used for the shipment. For example, Parcel, Truck Load, Less-Than Truck Load.
Hazardous Material	The flag indicating whether hazardous materials are being carried in this load.
Origin Node	The node where the load originated from.
Destination Node	The node where the load is being shipped to.
Multiple Load Stop	The flag indicating whether or not a shipment goes through multiple stops to load or unload additional shipments.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.9 General

**Table F–10 General Condition Builder Attributes**

Attribute	Description
Enterprise Code	The code of the enterprise.
Organization Code	The code of the organization.
Provider Organization Code	The code of the organization that provides the service.
Ship Node	The node that ships this shipment.
Supply Type	The supply type associated with the inventory status. Typical values are Onhand, Held, etc.

**Table F-10 General Condition Builder Attributes**

<b>Attribute</b>	<b>Description</b>
Item ID	The ID of the item on the order line.
Unit Of Measure	The unit of measure of the item.
Product Class	The inventory classification of an item based on the product's characteristics. Typical values are FQ - First Quality, SQ - Second Quality, etc.
Inventory Status	The inventory sub classification of the product, based on the results of the inventory control processes within the warehouse. Typical values are Good - Good Inventory, Damaged - Damaged inventory, Qlty-Hold - Quality Hold, etc.
Adjustment Type	The type of inventory adjustment. Typical values are Cycle Count, Receipt, Picking, Packing, Shipping, etc.
Alert Type	The type of alert raised when an exception occurs.
Carrier	The carrier used to carry the shipment.
Task Type	The Task Type applicable to a task. Typical values are Receipt, QC, Count, Replenishment, Retrieval, Putaway, VAS, Pack, Shipping, and Picking.
Assigned To User ID	The ID of the user to whom the task is assigned.
Task Status	The Task Status within the pipeline that the task travels through. Typical values are Open, Suggested, In Progress, Held, Completed, Canceled, etc.
Document Type	The document type for this order. Typical values are Sales Order, Purchase Order, Transfer Order, and Return Order.
SC UI Client Version	The Rich Client Platform application version number.

**Table F–10 General Condition Builder Attributes**

Attribute	Description
Activity Group ID	The identifier for the activity group.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.10 WMS Putaway

The WMS Putaway condition builder attributes are identical to the [General](#) attributes.

## F.11 WMS Layout Definition

The WMS Layout Definition condition builder attributes are identical to the [General](#) attributes.

## F.12 WMS Inventory

The WMS Layout Inventory condition builder attributes are identical to the [General](#) attributes.

## F.13 Trailer Loading

The Trailer Loading condition builder attributes are identical to the [General](#) attributes.

## F.14 Task Execution

The Task Execution condition builder attributes are identical to the [General](#) attributes.

## F.15 Move Request Execution

The Move Request Execution condition builder attributes are identical to the [General](#) attributes.

## F.16 Manifesting

The Manifesting condition builder attributes are identical to the [General](#) attributes.

## F.17 Over Pack Build

The Over Pack Build condition builder attributes are identical to the [General](#) attributes.

## F.18 Count Execution

*Table F–11 Count Execution Condition Builder Attributes*

Attribute	Description
Enterprise Code	The code of the enterprise for which the count request is created.
Request Type	The type of count requested.
Count Program Name	The name of the count program for which the count request is created.
Node Key	The node where the count request is processed.
Zone ID	The zone where the count must be performed.
Location Size Code	The capacity of the location where the count must be performed.
Is LPN Level	The flag indicating whether the count tasks are performed at the LPN level.
Is Case Level	The flag indicating whether the count tasks are performed at the case level.
Is Pallet Level	The flag indicating whether the count tasks are performed at the pallet level.

**Table F–11 Count Execution Condition Builder Attributes**

<b>Attribute</b>	<b>Description</b>
Is Item Level	The flag indicating whether the count tasks are performed at the item level.
Is Resolvable	The flag indicating whether variance can be resolved for this count result.
Product Class	The inventory classification of an item based on the product's characteristics. Typical values are FQ - First Quality, SQ - Second Quality, etc.
Unit Of Measure	The unit of measure of the item that was counted.
Item Classification 1	The first item classification attribute for determining the Count Strategy.
Item Classification 2	The second item classification attribute for determining the Count Strategy.
Item Classification 3	The third item classification attribute for determining the Count Strategy.
Has Variance	The flag indicating whether the count request has a variance.
Has Absolute Variance	The flag indicating whether the count request has an absolute variance.
Variance Quantity	The difference in quantity (+/-) between the count result and system quantity.
Absolute Variance Quantity	The absolute difference between the count result and system quantity.
Variance Value	The difference in cost/value (+/-) between the count result and system quantity.
Absolute Variance Value	The absolute difference in cost/value between the count result and system quantity.

Table F–11 Count Execution Condition Builder Attributes

Attribute	Description
Has Variance With Previous Count	The flag indicating whether the variance between the current count result and previous count results displays.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i>.</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.19 Pack Process

Table F–12 Pack Process Condition Builder Attributes

Attribute	Description
<b>Node Attributes</b>	
Ship Node	The node that ships this shipment.
Receiving Node	The node that receives this shipment.
Ship from Ship Node Interface Type	The interface type of the ship node from which the shipment is shipped (External Application, Console, Sterling WMS, or WMS 6.2).
Ship from Supplier Code	The code of the supplier that is shipping the shipment.
Ship from DCM Integration Real Time	The flag indicating whether the node from which the shipment is shipped uses WMS 6.2.
Ship from Country	The code of the country from which the shipment is being shipped.
Ship to Ship Node Interface Type	The interface type of the ship node to which the shipment is shipped (External Application, Console, Sterling WMS, or WMS 6.2).

**Table F–12 Pack Process Condition Builder Attributes**

<b>Attribute</b>	<b>Description</b>
Ship to Supplier Code	The code of the supplier to whom the shipment is being shipped.
Ship to DCM Integration Real Time	The flag indicating whether the node to which the shipment is shipped uses WMS 6.2.
Ship to Country	The code of the country to which the shipment is being shipped.
<b>Organization Attributes</b>	
Enterprise Code	The code of the enterprise that owns the shipment.
Buyer Organization Code	The code of the organization that is buying the goods or services.
Seller Organization Code	The code of the organization that is selling the goods or services.
<b>Shipment Attributes</b>	
Ship Mode	The shipment mode that is used for the shipment. For example, Parcel, Truck Load, Less-Than Truck Load.
Carrier	The carrier used to carry the shipment.
Freight Terms	The freight terms of the shipment.
Delivery Code	The code of the entity that pays for the transportation costs.
Pack And Hold	The flag indicating whether the shipment needs to be packed and put away for retrieval at a later date.
Shipment Container Count	The number of containers in the shipment.
Shipment Containerized Flag	The flag indicating the containerization state of the shipment. The values are: 01 - not containerized, 02 - containerization in progress and 03 - containerization completed.

Table F–12 Pack Process Condition Builder Attributes

Attribute	Description
<b>Container Attributes</b>	
Is Shipment Container	The flag indicating whether the container belongs to a shipment.
Is Load Container	The flag indicating whether the container is part of a load.
Is Inventory Pallet	The flag indicating whether the container is an inventory pallet.
Is Converted From LPN	The flag indicating whether the inventory container has been converted to a shipment container.
Is Serial Capture Pending	The flag indicating whether the serial capture is pending for the container.
Is Pack Process Complete	The flag indicating whether any more pack activities are pending for the container.
Is Product Placing Complete	The flag indicating whether placing the product into the container according to the system's suggestion has been completed.
Requires VAS	The flag indicating whether the container requires value added services.
Has Child Containers	The flag indicating whether a container is a parent container having other containers.
Number of Items	The number of items contained in the container.
Container Type	The attribute that specifies whether a shipment container is a case or pallet.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i>.</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.20 Outbound Picking

*Table F–13 Outbound Picking Condition Builder Attributes*

Attribute	Description
Activity Group ID	The identifier for the activity group.
Shipment Group ID	The identifier for the shipment group.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.21 VAS Process

*Table F–14 VAS Process Condition Builder Attributes*

Attribute	Description
Enterprise Code	The code of the enterprise that owns the item or license plate.
Provider Organization Code	The code of the organization that provides the service.
Node Key	The node, where the work orders are run.
Purpose	The purpose for the work order (ORDER / STOCK / SHIP)
Service Item Group Code	The code of the service item group (KIT/DKIT/COMPL/INVC/PS)
Service Item ID	The identifier for the service Item.
Segment Type	The type of segment. This may be MTO (made to order) or MTC (made to customer).
Segment	The segment to which the inventory involved in the work order belongs.

**Table F–14 VAS Process Condition Builder Attributes**

Attribute	Description
Has Components	The flag indicating whether the work order has component items.
Status	The status of the work order.
Pre Call Status	The flag indicating the status of the pre-call process.
Appt Status	The status of the appointment. This is in sync with the service order line. The appointment status is used in case of provided service work order.
Number Of Attempts	The number of attempts made to run the work order.
Number Of Hours until Appointment	The number of hours left before the appointment for the service item.
Number Of Hours After Appointment	The number of hours after the last appointment for the service item.
Number Of Hours After Last Execution	The number of hours after the last attempt to run the service.
Last Execution Success	The flag indicating whether the last attempt to run the service was successful or not.
Open Work Order Flag	The flag indicating whether the execution of the work order has ended or not.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i> .</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## F.22 Item-Based Allocation (IBA) Order

*Table F-15 IBA Attributes*

Attribute	Description
<b>Order Attributes</b>	
Exchange Type	The exchange type of the order.
Priority Code	Customizable priority code of the order.
Priority Number	The numeric priority code of the order.
Document Type	The document type for this order. Typical value is 0001 (Sales Order).
Order Type	The order classification attribute. This field can be used for reporting purposes or to build conditions for modeling your business process. Selling and Fulfillment Foundation has no default logic based on this field.
Entry Type	The channel through which this order was created.
Department Code	Department code to which the order was placed.
Search Criteria 1	Customizable field for allowing searches.
Search Criteria 2	Customizable field for allowing searches.
<b>Order Line</b>	
Line Type	The line type can be used in process modeling for pipeline determination or conditional processing.
Condition Variable 1	A user-defined variable that can be used for condition building in process modeling.
Condition Variable 2	A user-defined variable that can be used for condition building in process modeling.
<b>Shipping Attributes</b>	
Level of Service	The order or the line's level of service.

Table F-15 IBA Attributes

Attribute	Description
Ship To ID	The ship-to identifier. If a customer definition representing the buyer organization exists within Selling and Fulfillment Foundation, the ship-to ID can represent the Customer ID. Otherwise, the ship-to ID can represent the PersonID of the ship-to address or the receiving node of the order.
Carrier Service Code	The code of the carrier service used to carry the load.
<b>Participant Attributes</b>	
Enterprise Code	The code of the enterprise on the order.
Buyer Organization Code	The code of the organization that is buying the goods or services.
Seller Organization Code	The code of the organization that is selling the goods or services.
Bill To ID	The identifier of the customer to whom the order is being billed.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Selling and Fulfillment Foundation: Extending the Condition Builder Guide</i>.</p> <p><b>Note:</b> This field is limited only to unexposed key attributes that are pre-defined by Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## A

---

Account Number With Hub field, 71  
actions, 142, 155, 177, 186, 202, 211, 224, 233  
    creating, 155, 187, 211, 234  
Actual field, 166, 168, 188, 191  
additional logistic rules  
    defining, 43  
application rules side panel, 12  
Applications Manager  
    actions, 28  
        document types, 29  
        entering dates/times, 34  
        lists, 31  
        lookup functionality, 28  
        on-line help, 34  
        special characters, 35  
        troubleshooting, 34  
        users, 31  
    layout, 10  
    starting, 9  
    work area, 23  
Automatically Resolve Alerts field, 172, 195

## B

---

backorder reasons, 6, 101, 102, 103  
base repositories, 143, 177, 202, 225  
Break Bulk Node field, 58  
building  
    catalog index, 290  
business analysts, 1  
business models, 2

business rules, 3  
Buyer Organization field, 172, 173, 198, 213  
Buyer Supply Type field, 162, 163  
buyers, 2, 3

## C

---

carrier modification reasons  
    creating, 41  
    defining, 41  
    deleting, 42  
    modifying, 42  
Carrier Service Code field, 58  
Carrier/Service field, 58  
Case Content Entry Required field, 214  
catalog  
    index building, 290  
charge categories, 238, 242  
charge definitions, 237  
charge names, 240, 241  
Commitment Time field, 71  
common codes, 93  
condition builder, 523 to 551  
conditions, 142, 153, 177, 185, 202, 209, 224, 232  
    creating, 154, 185, 210, 232  
configuration screens  
    accessing, 13  
Consider Buyer's Routing Guide field, 39  
Consolidator field, 57, 59  
consumers, 2  
Contact Type field, 73  
contact types  
    creating, 72

- deleting, 74
- modifying, 73
- Country field, 57
- Create A New Organization field, 71
- Customer Classification field, 198

## D

---

- Damaged field, 216
- Date Type field, 165, 168, 188, 191
- date types, 165, 187
  - creating, 165, 188
  - deleting, 166, 189
  - modifying, 166, 189
- dealers, 2
- delivery codes
  - creating, 45
  - defining, 45
  - deleting, 46
  - modifying, 46
- Description field, 165, 168, 171, 188, 191, 194
- Do not mix in Shipment flag, 50
- Do Not Recompute Expected Dates When Requested Dates On The Order Are Changed field, 159
- document types, 249
- drop statuses, 150, 183, 207, 230
- DUNS Number field, 71

## E

---

- enterprise administrators, 1
- enterprises, 2, 3, 4
- environment variable
  - INSTALL\_DIR, xxxiv
  - INSTALL\_DIR\_OLD, xxxv
- Event ID field, 171, 194
- events, 155, 186, 211, 233
- Expected field, 166, 168, 188, 191

## F

---

- Final Disposition field, 217
- financial components, 235

- freight terms
  - creating, 38
  - defining, 37
  - deleting, 40
  - modifying, 40
- From field, 58
- fulfillment, 3, 142, 176, 202, 224
- Functional Acknowledgment Time field, 71

## H

---

- Hub, 3
- Hub administrators, 1

## I

---

- index
  - catalog search, 290
- inheritance
  - determining, 14
- INSTALL\_DIR, xxxiv
- INSTALL\_DIR\_OLD, xxxv
- instruction types, 93 to 95
  - creating, 93
  - deleting, 95
  - modifying, 94
- Item Classification field, 198, 213
- Item ID field, 198, 213

## L

---

- Level field, 169, 191
- Line Type field, 198, 213
- Locale field, 71
- logistics
  - configuring components, 37
  - defining attributes, 37

## M

---

- manufacturers, 2
- marketplaces, 2, 3
- milestones, 167, 189
  - creating, 167, 190

- deleting, 170, 192
- modifying, 169, 192
- modification components, 105 to 113
  - rules, 105 to 109
  - types, 110 to 113
- modification reasons, 6, 97 to 100
  - creating, 97
  - deleting, 100
  - modifying, 99
- monitoring components, 164, 187
  - date types, 165, 187
    - creating, 165, 188
    - deleting, 166, 189
    - modifying, 166, 189
  - events, 170, 192
    - creating, 170, 193
    - deleting, 174, 196
    - modifying, 174, 195
  - milestones, 167, 189
    - creating, 167, 190
    - deleting, 170, 192
    - modifying, 169, 192
- monitoring events, 170, 192
  - creating, 170, 193
  - deleting, 174, 196
  - modifying, 174, 195
- multi-divisional corporations, 2

## N

---

- negotiation, 142, 176, 202, 224
- Node field, 59
- note reasons, 117 to 119

## O

---

- order attributes
  - external references
    - order level, 79, 80, 81
    - order line level, 81, 82, 83
  - line types, 85, 86
  - order address types, 83, 84, 85
  - other attributes, 87
    - generating prime line numbers, 87
- Order Available On System field, 200

- Order field, 172, 173
- Order Line field, 172, 173
- Order Release field, 172, 173
- order sources, 77, 78, 79
- order types, 75, 76, 77, 121, 122, 123
- order validations, 89 to 92
  - buyer validation, 92
  - seller validation, 91
- Organization Code field, 71
- organization levels, 14
  - rules, 17
- Organization Name field, 71
- organization rules, 17
  - loading another organization's rules, 21
  - overriding, 18
- original equipment manufacturers, 2
- outbound constraints
  - defining, 48
- over receipt tolerance definitions, 211
- over receipts, 211
- Over Ship Percentage field, 198, 213
- over shipment, 196
- over shipment tolerance definitions, 196
- Override Freight Terms field, 59
- Override Ship To field, 59

## P

---

- payment rules, 245
- payment terms, 235, 236, 237
- pickup statuses, 150, 183, 207, 230
- pipeline determination, 143, 177, 203, 225
- pipelines, 142, 177, 202, 224
  - creating, 145, 179, 205, 227
  - purchase order receipt
    - viewing action details, 155, 187, 211, 234
    - viewing condition details, 154, 185, 209, 232
    - viewing details, 144, 178, 204, 226
    - viewing status details, 150, 183, 208, 231
    - viewing transaction details, 147, 181, 206, 229
- Pricing organization, 14, 17
- Priority field, 58
- process type pipelines, 142, 176, 202, 224
- process types, 6, 125, 175, 201, 223
- Product Class field, 216

Purge Code field, 249  
purge criteria, 247 to 250  
rules, 247

## Q

---

QC Required field, 214  
Quantity Type field, 169, 191  
queue management, 63

## R

---

Realert Interval field, 172, 195  
receipt preferences, 211  
    creating, 212  
    deleting, 214  
    modifying, 214  
receipts, 142, 176, 202, 224  
receiving discrepancy reason  
    deleting, 221  
    modifying, 220  
receiving discrepancy reasons  
    creating, 218  
    defining, 218  
Receiving Disposition field, 216  
receiving dispositions  
    creating, 215  
    deleting, 217  
    modifying, 217  
rejection reasons, 64, 65, 66  
    creating, 65  
    deleting, 66  
    modifying, 65  
repositories, 142, 176, 202, 224  
Requested field, 165, 168, 188, 191  
Requires Realert field, 171, 194  
response actions, 63  
    modifying, 64  
retailers, 2  
Retention Days field, 250  
Rollback Segment field, 249  
routing guide lines  
    creating, 54  
    definition, 54  
    deleting, 62

    modifying, 62  
routing guides, 51  
    creating, 51  
    deleting, 62  
    modifying, 53

## S

---

Select An Existing Organization field, 70  
Seller Demand Type field, 162, 163  
Seller Organization field, 172, 173, 198, 213  
Seller Supply Type field, 162, 163  
sellers, 2, 3  
Send ASN field, 71  
Sends Commitment field, 71  
Sends Functional Acknowledgement field, 71  
services, 142, 177, 202, 224  
Ship Node field, 172, 173  
shipment modes  
    creating, 47  
    defining, 46  
    deleting, 48  
    modifying, 48  
shipment preferences, 196  
    creating, 197  
    deleting, 199  
    modifying, 199  
shipments, 142, 176, 202, 224  
Short Description field, 216  
State field, 57  
Status field, 162, 163, 168, 191  
status inventory types, 160, 161  
    creating, 162  
    deleting, 164  
    modifying, 164  
statuses, 142, 150, 177, 183, 202, 207, 224, 230  
    creating, 150, 183, 208, 231  
Sterling Warehouse Management System, 152  
Store# field, 58, 59  
suppliers, 2  
supply chains, 3  
Synchronize Dates Between Master Order Dates  
    And Dates On Order Line And Schedules  
    field, 159

## T

---

tax names, 242, 243, 244  
third-party logistics models, 2, 3  
This Organization Is Also A Ship Node field, 71  
To field, 58  
transaction processing, 3  
transactions, 142, 146, 177, 180, 202, 205, 206, 224, 228, 229  
    creating, 147, 181, 206, 229  
transactions rules  
    defining, 155, 199  
transportation optimization, 50

## U

---

Update Seller Supply field, 162, 163  
Use Advanced Transit Time Calculation flag, 43

## V

---

Validate Customer ID flag, 92  
Validate Item flag, 92  
Validate Vendor ID flag, 92  
Vendor Classification Code field, 68  
Vendor Classification field, 70  
Vendor ID field, 70

## W

---

web stores, 3  
workflows, 3

## Z

---

Zip Code field, 57

