Infor WMS Automation Agent Configuration Guide for Components

Version 11.4.x

Important Notices

Trademark Acknowledgements

Publication Information

Release: Infor WMS Automation Agent Release 11.4.x

Publication date: May 19, 2021

Contents

About this guide

This guide provides information for the configuration of Automation Agent components.

Intended audience

This guide is intended for the system administrator or consultant who configures and administers Automation Agent components.

Related documents

You can find the documents in the product documentation section of the Infor Support Portal, as described in "Contacting Infor" on page 9.

- *Infor WMS Automation Agent Installation Guide*
- *Infor WMS Automation Agent Training Guide*
- *Infor WMS Automation Agent Administration Guide*
- *Infor WMS Automation Agent Platform Support Matrix*

Contacting Infor

If you have questions about Infor products, go to Infor Concierge at https://concierge.infor.com/ and create a support incident.

If we update this document after the product release, we will post the new version on the Infor Support Portal. To access documentation, select **Search > Browse Documentation**. We recommend that you check this portal periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1 Overview

Infor WMS Automation Agent directs real-time activities within warehouses and distribution centers (DC) by performing as a data facilitator between applications and material handling equipment (MHE). Multiple connection types, such as Database, API, and Socket, File, can be implemented. Automation Agent builds a library of connection types and feature flows as you continue to use the product. Automation Agent provides a uniform interface to material handling equipment such as Automated Guided Vehicles (AGV), AS/RS, carousels, conveyor systems, sorters, and palletizers.

Grafana is used as the current interface to support application monitoring and for issue resolution.

Component Hierarchy

Automation Agent consists of these ordered components and these characteristics:

AgentController

AgentController is the main startup component of Automation Agent and is contained in the `agent.xml` configuration folder. It includes lists of initiators and alerts that are started when the agent starts.

Initiator

The Initiator components start ProcessChains when they receive data or fire. They monitor a data source for new content, either by polling or entering a wait state for a client to push information. These are configured in the `initiators.xml` configuration file.

Alerts

The alert components notify users of important events. These may be exceptions or successes, and push type notifications are configurable per user. These are configured in the `alerts.xml` configuration file.

ProcessChain

The ProcessChain components are lists of processors that fire sequentially when triggered by an initiator. These are configured in the `processchains.xml` configuration file.

Processors

Each entry in a ProcessChain refers to a Processor. Processors carry out specific tasks such as mapping data, validating conditions, changing the ProcessChain to redirect flows based on data, or connecting to other data sources or systems. These are configured in the `processors.xml` configuration file.

DataMaps

DataMaps provide mapping instructions to be carried out by mapping type Processors. They do not directly perform any action. They are separated into a different file to reduce clutter because they become very large. These are configured in the `datamaps.xml` configuration file.

Lookups and Validations

These components do not directly perform an action. They are used by Lookup or Validation type processors to provide rulesets for that processor to apply. These are configured in the `lookupvalidations.xml` configuration file.

Chapter 2 AgentController

The AgentController component starts and controls all processes managing top level thread pools for pollers and alerts. It provides a common place for components to log alerts, load other components, to create new processes. The configuration consists of a list of Initiators and a list of Alerts.

Note: If the allowed values for a property are defined as **`any of type`**, the property will accept any value that is of the **Type** specified for the property. For example, if the Type is String, then any string is valid.

Sample configuration

This is a sample configuration for the AgentController:

```
<bean id="controller" class="com.infor.cswms.automationagent.controller.AgentController"
primary="true">
<property name="initiators">
<list>
<ref bean="exportPoller" />
</list>
</property>
<property name="alerts">
<list>
<ref bean="emailAlert" />
<ref bean="eventLogAlert"/>
</list>
</property>
<property name = "namedResources">
<list>
<ref bean = "senderSocketClientProcessor"/>
</list>
</property>
```

```
<property name="globalProperties">
<map>
<entry key="restHostName" value="mylocalsce.infor.com "/>
<entry key="port" value="8080"/>
<entry key="tenant" value="INFOR"/>
<entry key="schema" value="INFOR_AA114_wmwhse1"/>
<entry key="username" value="admin1"/>
<entry key="password" value="Pmm_7386"/>
<entry key="jndi" value="java:/jdbc/INFOR_AA114_WMWHSE1"/>
</map>
</property>
</bean>
```

Properties

initiators

- Required
- Description – A list of Initiators to be started when the Agent is started.
- Type – list of reference to type Initiator
- Default – none
- Valid Values – any of type
- Supports replacement - no

alerts

- Optional
- Description – A list of AlertManagers to be called when a component publishes an alert.
- Type – list of reference to type AlertManager
- Default – none
- Valid Values – any of type
- Supports replacement - no

namedResources

- Optional
- Description – A list of processors that are to be initialized
- Type – list of reference to type Processor
- Default – none
- Valid Values – any of type
- Supports replacement - no

globalProperties

- Optional
- Description – A map of key-value pairs that will be added to the message context. This is available in all Processors through the framework
- Type – map
- Default – none
- Valid Values – any of type
- Supports replacement – no

Chapter 3 FTPClientInitiator

This component starts a local directory file monitoring process which will FTP interface files to a host machine.

Sample configuration

This is a sample configuration in the `initiators.xml` file.

```
<bean id="ftpClientInitiator" class="com.infor.cswms.automationagent.initiator.FTPClientInitiator">

<property name="pickupPath" value="localdirectory"/>

<property name="putPath" value=""/>

<property name="filePattern" value=".xml"/>

<property name="sortBy" value="D"/>

<property name="sortType" value="A"/>

<property name="dateFormat" value="dd-MM-yyyyhh:mm:ss"/>

<property name="sortPosition" value="1"/>

<property name="sortDelimiter" value="_"/>

<property name="pollInterval" value="10"/>

<property name="host" value="ftp.infor.com"/>

<property name="Password" value="F2URWL6M"/>
```

```
<property name="userName" value="cwilliams1"/>
</bean>
```

Properties

pickupPath

- Required
- Description – local file directory to be used for file polling
- Type – text
- Default – none
- Valid Values – a valid path
- Supports replacement - no

putPath

- Required
- Description – directory on host machine to put files
- Type – path
- Default – none
- Valid Values – a valid path
- Supports replacement – no

filePattern

- Required
- Description – type of file to pickup
- Type – text
- Default – none
- Valid Values – file types
    - Supports replacement - no

sortBy

- Required
- Description – order to sort files off of the directory by ascending or descending order
- Type – character
- Default – “A”
- Valid Values – “A” or “D”
- Supports replacement - no

sortType

- Required
- Description – order to sort files off the directory by Numeric or Date or Alpha
- Type – character
- Default – “A”
- Valid Values – “A”, “N” or “D”
- Supports replacement - no

dateFormat

- Required
- Description – used only for sorting files by date.
- Type – Datetime
- Default – none
- Valid Values – “dd-MM-yyyyhh:mm:ss”
- Supports replacement - no

sortPosition

- Required
- Description – position of the file name does the sort begin with
- Type – numeric
- Default – 1
- Valid Values – 1 - 100
- Supports replacement - no

pollInterval

- Required
- Description – Directory polling in seconds
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement - no

host

- Required
- Description – name or IP address of the host machine
- Type – Alphanumeric
- Default – none
- Valid Values – host or IP address
- Supports replacement - no

userName

- Required
- Description – username to use to log into the host machine
- Type – alphanumeric
- Default – none
- Valid Values – valid username
- Supports replacement - no

Password

- Required
- Description – Password of usename
- Type – alphanumberic
- Default – none
- Valid Values – the password of username
- Supports replacement - no

Chapter 4 FTPRetrieveInitiator

Sample configuration

This is a sample configuration in the `initiators.xml` file.<bean id="ftpRetrieveInitiator" class="com.infor.cswms.automationagent.initiator.FTPRetrieveInitiator">

<property name="remotePath" value="outbound"/>

<property name="localPath" value="C:\agentDemo\drop\"/>

<property name="filePattern" value=".xml"/>

<property name="sortBy" value="D"/>

<property name="sortType" value="A"/>

<property name="dateFormat" value="dd-MM-yyyyhh:mm:ss"/>

<property name="sortPosition" value="1"/>

<property name="sortDelimiter" value="_"/>

<property name="pollInterval" value="10"/>

<property name="host" value="ftp.infor.com"/>

<property name="Password" value="F2URWL6M"/>

<property name="userName" value="cwilliams1"/>
</bean>

Properties

remotePath

- Required
- Description – Remote file directory to be used for file polling
- Type – directory
- Default – none
- Valid Values – a valid path
- Supports replacement - no

localPath

- Required
- Description – directory on local machine to put files
- Type – path
- Default – none
- Valid Values – a valid path
- Supports replacement – no

filePattern

- Required
- Description – type of file to pickup
- Type – text
- Default – none
- Valid Values – file types
- Supports replacement - no

sortBy

- Required
- Description – order to sort files off of the directory by ascending or descending order
- Type – character

- Default – “A”
- Valid Values – “A” or “D”
- Supports replacement - no

sortType

- Required
- Description – order to sort files off the directory by Numeric or Date or Alpha
- Type – character
- Default – “A”
- Valid Values – “A”, “N” or “D”
- Supports replacement - no

dateFormat

- Required
- Description – used only for sorting files by date.
- Type – Datetime
- Default – none
- Valid Values – “dd-MM-yyyyhh:mm:ss”
- Supports replacement - no

sortPosition

- Required
- Description – position of the file name does the sort begin with
- Type – numeric
- Default – 1
- Valid Values – 1 - 100
- Supports replacement - no

sortDelimiter

- Required
- Description – delimiter used if sorting by segment

- Type – Alphanumeric
- Default – none
- Valid Values – any of type
- Supports replacement – no

pollInterval

- Required
- Description – Directory polling in seconds
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement - no

host

- Required
- Description – name or IP address of the host machine
- Type – Alphanumeric
- Default – none
- Valid Values – host or IP address
- Supports replacement - no

userName

- Required
- Description – username to use to log into the host machine
- Type – alphanumeric
- Default – none
- Valid Values – valid username
- Supports replacement - no

Password

- Required

- Description – Password of username
- Type – alphanumeric
- Default – none
- Valid Values – the password of username
- Supports replacement - no

Chapter 5 FileInitiator

This component polls a directory for a file, reads the file, and passes its contents along to the first processor in the process chain.

Sample configuration

This is a sample configuration in the `initiators.xml` file.<bean id="fileInitiator" class="com.infor.cswms.automationagent.initiator.FileInitiator">

<property name="errorPath" value="C:\demo\error"/>

<property name="pickupPath" value="C:\demo"/>

<property name="useInProcessPath" value="true"/>

<property name="processingPath" value="C:\demo\inprocess"/>

<property name="filePattern" value=".*\.xml"/>

<property name="pollInterval" value="3"/>

<property name="filesPerPoll" value="3"/>

<property name="maxFileSize" value="2"/>

<property name="dateFormat" value="dd-MM-yyyyhhmmss"/>

<property name="sortType" value="D"/>

<property name="sortBy" value="A"/>

```
<property name="sortPosition" value="3"/>

<property name="sortDelimiter" value="_"/>

<!-- using a string here because we want to get a new bean loaded every time, don't re-use the same object sine it may be modified -->
<property name="processChain" value="fileChain"/>
</bean>
```

Properties

errorPath

- Required
- Description – error directory to be used for files that could not be read or written
- Type – directory
- Default – none
- Valid Values – a valid directory
- Supports replacement - no

pickupPath

- Required
- Description – directory on local machine to select files
- Type – path
- Default – none
- Valid Values – a valid path
- Supports replacement – no

useInProcessPath

- Required

- Description – File will be moved to the processing path for processing
- Type – Boolean
- Default – none
- Valid Values – True or False
- Supports replacement – no

processingPath

- Required
- Description – path in which the files will be moved for processing
- Type – path
- Default – none
- Valid Values – a valid path
- Supports replacement – no

filePattern

- Required
- Description – type of file to pickup
- Type – text
- Default – none
- Valid Values – file types
- Supports replacement - no

pollInterval

- Required
- Description – Directory polling in seconds
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement - no

filesPerPoll

- Required
- Description – maximum of file to process at per poll
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement - no

maxFileSize

- Required
- Description – Maximum size in megabytes a file that can be for processing
- Type – numeric
- Default – 1
- Valid Values – 1-100
- Supports replacement - no

sortBy

- Required
- Description – order to sort files off of the directory by ascending or descending order
- Type – character
- Default – “A”
- Valid Values – “A” or “D”
- Supports replacement - no

sortType

- Required
- Description – order to sort files off the directory by Numeric or Date or Alpha
- Type – character
- Default – “A”
- Valid Values – “A”, “N” or “D”
- Supports replacement - no

dateFormat

- Required
- Description – used only for sorting files by date.
- Type – Datetime
- Default – none
- Valid Values – “dd-MM-yyyyhh:mm:ss”
- Supports replacement - no

sortPosition

- Required
- Description – position of the file name does the sort begin with
- Type – numeric
- Default – 1
- Valid Values – 1 - 100
- Supports replacement - no

sortDelimiter

- Required
- Description – delimiter used if sorting by segment
- Type – Alphanumeric
- Default – none
- Valid Values – any of type
- Supports replacement – no

Chapter 6 WebServerInitiator

This component starts a web server to listen to a port and pass it to process chain

Sample configuration

This is a sample configuration in the `initiators.xml` file.<bean id="webServerInitiator" class="com.infor.cswms.automationagent.initiator.WebServerInitiator">

<property name="port" value="8081"/>

<property name="maxThreads" value="10"/>

<property name="processChain" value="webserverChain"/>

</bean>

Properties

port

- Required
- Description – a port number for the socket to listen on
- Type – port
- Default – none
- Valid Values – a valid port number
- Supports replacement - no

maxThreads

- Required
- Description – maximum number of threads that can be open
- Type – numeric
- Default – none
- Valid Values – 1 - 9999
- Supports replacement – no

processChain

- Required
- Description – The name of the process chain that the initiator will follow after receiving the message.
- Type – alphanumeric
- Default – none
- Valid Values – valid process chain
- Supports replacement - no

Chapter 7 SocketInitiatorImpl

This component listens for a socket connection and passes the request off to a processor.

Sample configuration

This is a sample configuration in the `initiators.xml` file.<bean id=”socketInitiator" class="com.infor.cswms.automationagent.initiator.SocketInitiatorImpl">

<property name="*port*" value="*8005*"/>

<property name="*maxThreads*" value="*10*"/>

<property name="*messageTypeStart*" value="*10*"/>

<property name="*messageTypeEnd*" value="*26*"/>

<property name="*paddingChar*" value="*"/>

<property name="processChain" value="spsSocketChain"/>

<property name="messageTypeInboundPositionMap" ref="inboundPositionMap" />

<property name="messageTypeInboundValueMap" ref="inboundValueMap" />

<property name="messageTypeOutboundPositionMap" ref="outboundPositionMap" />

<property name="messageTypeOutboundValueMap" ref="outboundValueMap" />

<property name="*sendAck*" value="*true*"/>

```
<property name="ackMessage" value="QQ0000ACKN"/>

<property name="messageFormat" value="Fixed"/>

</bean>
```

Properties

port

- Required
- Description – port number socket will listen on
- Type – numeric
- Default – none
- Valid Values – port number
- Supports replacement - no

maxThreads

- Required
- Description – maximum number of threads spawned
- Type – numeric
- Default – 1000
- Valid Values – 1-1000
- Supports replacement – no

messageTypeStart

- Required
- Description – position to start parsing message type
- Type – numeric
- Default – none
- Valid Values – 1-999

- Supports replacement – no

messageTypeEnd

- Required
- Description – position to end parsing message type
- Type – path
- Default – none
- Valid Values – a valid path
- Supports replacement – no

paddingChar

- Required
- Description – filler character to be used for fixed length processing
- Type – character
- Default – "_"
- Valid Values – character
- Supports replacement - no

processChain

- Required
- Description – The name of the process chain the initiator will follow after initial parse
- Type – alphanumeric
- Default – none
- Valid Values – valid process chain
- Supports replacement - no

messageTypeInboundPositionMap

- Required
- Description – a hashmap containing the positions for a particular message type
- Type – map
- Default – none

- Valid Values – hashmap
- Supports replacement – no

messageTypeInboundValueMap

- Required
- Description – a hashmap containing the names to be assigned to data for a particular message type
- Type – map
- Default – none
- Valid Values – hashmap
- Supports replacement - no

messageTypeOutboundPositionMap

- Required
- Description – a hashmap containing the positions for a particular message type
- Type – character
- Default – “A”
- Valid Values – “A” or “D”
- Supports replacement - no

messageTypeOutboundvalueMap

- Required
- Description – a hashmap containing the names to be assigned to data for a particular message type
- Type – map
- Default – none
- Valid Values – hashmap
- Supports replacement - no

sendAck

- Required

- Description – used only for sorting files by date.
- Type – boolean
- Default – false
- Valid Values – true or false
- Supports replacement - no

ackMessage

- Description – Will append to the end of the original message as response
- Type – character
- Default – none
- Valid Values – any value
- Supports replacement - no

messageFormat

- Required
- Description – Fixed, Delimited, or Tab only fixed implemented for now
- Type – character
- Default – Fixed
- Valid Values – Fixed, Delimited or Tab
- Supports replacement – no

Chapter 8 SocketClientInitiator

This component acts as a client socket and listens for a socket connection, reads the socket messages, and parses it. It then processes them one by one by passing to a processor.

Sample configuration

This is a sample configuration in the `initiators.xml` file.

```
<bean id="PLC10SocketListener"
class="com.infor.cswms.automationagent.initiator.SocketClientInitiator">
<!-- client side hostName -->
<property name="hostName" value="127.0.0.1"/>
<!-- client side port -->
<property name="port" value="4010"/>
<!-- position to start parsing message type -->
<property name="messageTypeStart" value="18"/>
<!-- position to stop parsing message type -->
<property name="messageTypeEnd" value="21"/>
<!-- filler character to be used for fixed length processing -->
<property name="paddingChar" value="*"/>
<!-- The name of the process chain the initiator will follow after initial parse -->
<property name="processChain" value="PLC10Chain"/>
<!-- a hashmap containing the positions for a particular messagetype -->
<property name="messageTypeInboundPositionMap" ref="PLC10InboundPositionMap" />
<!-- a hashmap containing the names to be assigned to data for a particular messagetype -->
<property name="messageTypeInboundValueMap" ref="PLC10InboundValueMap" />
<!-- Fixed, Delimited, or Tab only fixed implemented for now -->
<property name="messageFormat" value="Fixed"/>
<!-- Enable to send custom comment character -->
<property name="enableCommentChar" value="true"/>
```

```
<!-- Specify comment character if message starts with #, to parse it else the message will not be parsed because the default comment char is # -->
<property name="commentChar" value=" "/>
<!-- Specify header length to parse the header part from the message-->
<property name="headerLength" value="18"/>
<!-- position to start parsing message data-->
<property name="messageLengthStart" value="2"/>
<!-- position to stop parsing message data-->
<property name="messageLengthEnd" value="6"/>
<property name="outputPart" value="PLC10output"/>
<property name="reconnectSleepTime" value="2000"/>
<property name="maxRetry" value="1000"/>
<property name="properties">
<map>
<entry key="sourceputawayzone" value="FG"/>
</map>
</property>

</bean>
```

Properties

hostName

- Required
- Description – hostname of the socket
- Type – String
- Default – none
- Valid Values – hostname or IP address
- Supports replacement - no

port

- Required
- Description – port number socket will listen on
- Type – numeric
- Default – none
- Valid Values – port number
- Supports replacement - no

maxThreads

- Required
- Description – maximum number of threads spawned
- Type – numeric
- Default – 1000
- Valid Values – 1-1000
- Supports replacement – no

messageTypeStart

- Required
- Description – position to start parsing message type
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement – no

messageTypeEnd

- Required
- Description – position to end parsing message type
- Type – path
- Default – none
- Valid Values – a valid path
- Supports replacement – no

paddingChar

- Required
- Description – filler character to be used for fixed length processing
- Type – character
- Default – "_"
- Valid Values – character
- Supports replacement - no

processChain

- Required
- Description – The name of the process chain that the initiator will follow after initial parse
- Type – alphanumeric
- Default – none
- Valid Values – valid process chain
- Supports replacement - no

messageTypeInboundPositionMap

- Required
- Description – a hashmap containing the positions for a particular message type
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

messageTypeInboundValueMap

- Required
- Description – a hashmap containing the names to be assigned to data for a particular message type
- Type – map
- Default – none
- Valid Values – hashmap
- Supports replacement - no

enableCommentChar

- Required
- Description – used to enable comment character while parsing socket messages.
- Type – boolean
- Default – false
- Valid Values – true or false
- Supports replacement - no

commentChar

- Required
- Description – comment character to be used to ignore while reading the socket messages
- Type – character
- Default – “ ”
- Valid Values – character
- Supports replacement - no

messageFormat

- Required
- Description – Fixed, Delimited, or Tab only fixed implemented for now
- Type – character
- Default – Fixed
- Valid Values – Fixed, Delimited or Tab
- Supports replacement – no

headerLength

- Required
- Description – specifies the header length of a socket header message
- Type – numeric
- Default – none
- Valid Values – 1-1000
- Supports replacement – no

messageLengthStart

- Required
- Description – position to start parsing message length after socket header
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement – no

messageLengthEnd

- Required
- Description – position to end parsing message after socket header
- Type – path
- Default – none
- Valid Values – a valid path
- Supports replacement – no

properties

- Optional
- Description – keys with pairing values,
- Type – map
- Default – none
- Valid Values – none
- Supports replacement – no

reconnectSleepTime

- Optional
- Description – time interval to retry socket connection if connection is dropped
- Type – Integer
- Default – “2000”
- Valid Values – any integer value
- Supports replacement – no

maxRetry

- Optional
- Description – Maximum number of retry attempts to connect to a socket server
- Type – Integer
- Default – “1000”
- Valid Values – any integer value
- Supports replacement – no

Chapter 9 PollerInitiator

This component is used to set polling intervals.

Sample configuration

This is a sample configuration in the `initiators.xml` file<bean id="PollerInitiator" class="com.infor.cswms.automationagent.initiator.PollerInitiator">

```
<property name="pollInterval" value="10"/>
<property name="properties">
<map>
        <entry key="restHostName" value="mylocalsce.infor.com"/>
        <entry key="port" value="8080"/>
        <entry key="tenant" value="INFOR"/>
        <entry key="endpoint" value="exportInterfaceEndpoint"/>
        <entry key="schema" value="INFOR_SCP11_wmwhse1"/>
    </map>
</property>

<property name="processChain" value="cswmsExportChain"/>
</bean>
```

Properties

pollInterval

- Required

- Description – component polling in seconds
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement – no

content

- Optional
- Description – position to end parsing message type
- Type – object
- Default – none
- Valid Values – a valid path
- Supports replacement – no

properties

- Optional
- Description – A map of key-value pairs that will be added to the message context. This is available in all initiators through the framework.
- Type – map
- Default – none
- Valid Values – restHostName, port, tenant, endpoint, schema
- Supports replacement – no

processChain

- Required
- Description – The name of the process chain that the initiator will follow after initial parse
- Type – alphanumeric
- Default – none
- Valid Values – valid process chain

Chapter 10 DBInitiator

This component is used to poll the database based on SQL queries segregated into polling, header, and detail queries.

Sample configuration

This is a sample configuration in the `initiators.xml` file

```
<bean id="dbPoller" class="com.infor.cswms.automationagent.initiator.DBInitiator">
<property name="properties">
<map>
        <entry key="pickStatus" value="9"/>
        <entry key="newStatus" value="0"/>
        <entry key="transmittingStatus" value="5"/>
        <entry key="inProcessStatus" value="5"/>
        <entry key="successStatus" value="9"/>
        <entry key="errorStatus" value="E"/>
        <entry key="statusColumn" value="TRANSMITFLAG"/>
        <entry key="jmsdestinationname" value="jms/MessageQueue"/>

    </map>
</property>
<property name="pollInterval" value="10"/>
<property name="jndiname" value="java:/jdbc/SCPRD_WMWHSE1"/>
<property name="transactionISOLevel" value="8"/>
<property name="detailName" value="cancelworkassignmentdetails"/>
<property name="primaryKeyColumn" value="TRANSMITLOGKEY"/>
<property name="processChain" value="cswmsDBOrderPickChain"/>

<!-- Set the jndiResource and jndiPropertiesMap if we want to get a connection using local JNDI -->
<!--
```

```
<property name="jndiResource" value="localjndi"/>
<property name="jndiPropertiesMap" ref="jndiPropertiesValueMap"/>
-->
<property name="pollSQL" value="SELECT * FROM TRANSMITLOG WHERE TABLENAME = 'ASNCREATED'
AND TRANSMITFLAG = '0'"/>
<property name="headerSQL" value="SELECT * FROM TRANSMITLOG WHERE
TABLENAME='ORDERPICKED' AND TRANSMITFLAG='0'"/>
<property name="detailSQL" value="SELECT PICKDETAILKEY  WorkAssignmentID, DROPID ContainerID,
ORDERKEY  HostOrderID,ROUTE  RouteID, '0' ExceptionContainer, 'AutoVas'  Flags, 'SingleLane'
PackBenchRule FROM PICKDETAIL WHERE ORDERKEY=:KEY1"/>

<property name="transmittingSQL" value="UPDATE TRANSMITLOG SET
TRANSMITFLAG=:transmittingStatus WHERE TABLENAME='ORDERPICKED' AND
TRANSMITFLAG=:newStatus AND TRANSMITLOGKEY=:TRANSMITLOGKEY"/>
<property name="inprocessSQL" value="UPDATE TRANSMITLOG SET TRANSMITFLAG=:inProcessStatus
WHERE TABLENAME='ORDERPICKED' AND TRANSMITFLAG=:newStatus AND
TRANSMITLOGKEY=:TRANSMITLOGKEY"/>
<property name="successSQL" value="UPDATE TRANSMITLOG SET TRANSMITFLAG=:successStatus
WHERE TABLENAME='ORDERPICKED' AND TRANSMITFLAG=:inProcessStatus AND
TRANSMITLOGKEY=:TRANSMITLOGKEY"/>
<property name="exceptionSQL" value="UPDATE TRANSMITLOG SET TRANSMITFLAG=:errorStatus
WHERE TABLENAME='ORDERPICKED' AND TRANSMITFLAG=:inProcessStatus AND
TRANSMITLOGKEY=:TRANSMITLOGKEY"/>

</bean>
<!--
<util:map id="jndiPropertiesValueMap" map-class="java.util.HashMap">
<entry key="Driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
   <entry key="Username" value="wmwhse1"/>
   <entry key="Password" value="WMwhSqlWMwhSql1"/>
   <entry key="URL" value="jdbc:sqlserver://inhynckambhamp2:1433;databaseName=S1103"/>
</util:map>
-->
```

Properties

properties

- Optional
- Description – keys with pairing values,
- Type – map
- Default – none
- Valid Values – none
- Supports replacement – no

pollInterval

- Required
- Description – component polling in seconds
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement – no

jndiname

- Required
- Description – path of JNDI
- Type – path
- Default – none
- Valid Values – none
- Supports replacement - yes

transactionISOLevel

- Optional
- Description – set transaction isolation level index
- Type – String

- Default – 8
- Valid Values – 0,1,2,4,8
- Supports replacement – yes

detailName

- Optional
- Description – Specifies the name of the detail records, which can be used to identify the detail records
- Type – String
- Default – details
- Valid Values – Any valid detail name in String format
- Supports replacement – no

primaryKeyColumn

- Optional
- Description – Specifies the Primary Key Column of the table
- Type – String
- Default – none
- Valid Values – none
- Supports replacement – no

processChain

- Required
- Description – The name of the process chain that the initiator will follow after the initial parse
- Type – alphanumeric
- Default – none
- Valid Values – valid process chain

jndiResource

- Optional
- Description – Determines whether to use local JNDI to get the Connection

- Type – String
- Default – none
- Valid Values – localjndi
- Supports replacement - no

jndiPropertiesMap

- Optional
- Description – Refers a hashmap and use those to establish the Connection for the database.
- Type – hash map
- Default – none
- Valid Values – localjndi
- Supports replacement - no

jndiPropertiesValueMap

- Optional
- Description – A key value pairs of values determines the properties of the Database.
- Type – hash map
- Default – none
- Valid Values – key value pairs
- Supports replacement - no

pollSQL

- Optional
- Description – SQL statement that will select polling data
- Type – SQL Statement
- Default – none
- Valid Values – a valid SQL for the selected datasource
- Supports replacement - no

headerSQL

- Required

- Description – SQL statement that will select header information
- Type – SQL Statement
- Default – none
- Valid Values – a valid sql for the selected datasource
- Supports replacement - no

detailSQL

- Required
- Description – SQL statement that will select detail information for the headers selected
- Type – SQL statement
- Default – none
- Valid Values – a valid sql for the selected datasource
- Supports replacement - no

transmittingSQL

- Required
- Description – SQL to update transmitlog to transmitting status
- Type – SQL
- Default – none
- Valid Values – a valid update sql for the selected datasource
- Supports replacement - no

inprocessSQL

- Required
- Description – SQL to update transmitlog to inprocess status
- Type – SQL
- Default – none
- Valid Values – a valid update sql for the selected datasource
- Supports replacement - no

successfulSQL

- Required
- Description – SQL to update transmitlog to successful status
- Type – numeric
- Default – none
- Valid Values – a valid update sql for the selected datasource
- Supports replacement - no

exceptionSQL

- Required
- Description – SQL to update transmitlog to error status
- Type – SQL
- Default – none
- Valid Values – a valid update sql for the selected datasource
- Supports replacement – no

Chapter 11 JMSListener

This component acts as message receiver that consumes messages from the message destination that could be either queue or topic.

Sample configuration

This is a sample configuration.<bean id="jmsListener"
class="com.infor.cswms.automationagent.initiator.JMSListener">

<property name="pollInterval" value="1"/>
<property name="defaultConnectionFactory"
value="java:jboss/exported/jms/RemoteConnectionFactory"/>
<property name="initialContextFactory" value="org.jboss.naming.remote.client.InitialContextFactory"/>
<property name="messageDestination" value="java:jboss/exported/jms/MessageQueue"/>
<property name="providerURL" value="http-remoting://localhost:9480"/>
<property name="username" value="mquser"/>
<property name="password" value="mqpassword"/>
<property name="processChain" value="orderShipChain"/>
<property name="properties">
<map>

<entry key="restHostName" value="inhynckambhamp2.infor.com"/>
<entry key="port" value="3001"/>
<entry key="tenant" value="INFOR"/>
<entry key="endpoint" value="manualInductEndPoint"/>
<entry key="schema" value="INFOR_S1103_wmwhse1"/>
</map>
</property>
</bean>

Properties

pollInterval

- Required
- Description – component polling in seconds
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement – no

defaultConnectionFactory

- Required
- Description – JMS connection factory that comes with the wildfly
- Type – path
- Default – none
- Valid Values – JNDI Names
- Supports replacement - no

initialContextFactory

- Optional
- Description – a factory that creates an initial context
- Type – file
- Default – none
- Valid Values – any of type
- Supports replacement – no

messageDestination

- Required
- Description – JMS destination name either Queue/Topic
- Type – string of the destination

- Default – none
- Valid Values – any of type
- Supports replacement – yes

providerURL

- Required
- Description – URL to the remote system
- Type – string
- Default – none
- Valid Values – any of type
- Supports replacement – no

username

- Required
- Description – username for the remote system
- Type – string
- Default – none
- Valid Values – any of type
- Supports replacement – no

password

- Required
- Description – password of username
- Type – string
- Default – none
- Valid Values – any of type
- Supports replacement – no

processChain

- Required
- Description – The name of the process chain that the initiator will follow after the initial parse

- Type – alphanumeric
- Default – none
- Valid Values – valid process chain
- Supports replacement - no

properties

- Optional
- Description – A map of key-value pairs that will be added to the message context. This is available in all Processors through the framework.
- Type – map
- Default – none
- Valid Values – any of type
- Supports replacement - no

Chapter 12 RoundRobinInitiator

This component is used to execute the list of initiators defined in a round robin manner.

Sample configuration

This is a sample configuration.<bean id="RRInitiator"
class="com.infor.cswms.automationagent.initiator.RoundRobinInitiator">

<property name="maxThreads" value="20"/>

<property name="pollInterval" value="5"/>

<property name="initiators">

<list>

<ref bean="S01DbPoller"/>

<ref bean="RBG3DbPoller"/>

<ref bean="RBG4DbPoller"/>

<ref bean="RBG5DbPoller"/>

<ref bean="RBG6DbPoller"/>

<ref bean="QADbPoller"/>

</list>

</property>

</bean>

Properties

maxThreads

- Optional
- Description – maximum number of threads that can be open

- Type – numeric
- Default – none
- Valid Values – 1 - 9999
- Supports replacement – no

pollInterval

- Required
- Description – component polling in seconds
- Type – numeric
- Default – none
- Valid Values – 1-999
- Supports replacement – no

initiators

- Required
- Description – Specifies list of initiators to be executed in round robin basis
- Type – list
- Default – none
- Valid Values – List
- Supports replacement – no

Chapter 13 XPathParserProcessor

This component maps one XML document to another using a DocumentPathMap

Sample configuration

This is a sample configuration.<bean id="xpathParser"
class="com.infor.cswms.automationagent.processor.XPathParserProcessor" scope="prototype">

<property name="properties">

<map>

<entry key="schema" value="INFOR_ENTERPRISE"/>

<entry key="tenant" value="INFOR"/>

</map>

</property>

<!-- XPath properties - set a property from the XML. Must evaluate to a single entry -->

<property name="xpathProperties">

<map>

<entry key="mapName" value="/*/ApplicationArea/Sender/MessageInfo/Name[1]"/>

</map>

</property>

<!-- Determines what map to load, supports parameters so you can set the name using an xpath lookup -
->

<property name="mapBeanName" value="${mapName}"/>

</bean>

Properties

properties

- Optional
- Description – A map of key-value pairs that will be added to the message context. This is available in all Processors through the framework.
- Type – map
- Default – none
- Valid Values – any of type
- Supports replacement - no

xpathProperties

- Optional
- Description – A map of key-xpath pairs. The xpath will be evaluated against the message and the value added to the message context with the supplied key.
- Type – map
- Default – none
- Valid Values – any of type
- Supports replacement – no

mapBeanName

- Required
- Description – The string name of the DocumentPathMap bean to be loaded
- Type – string pointing to DocumentPathMap bean
- Default – none
- Valid Values – any of type
- Supports replacement – yes

Sample configuration: DocumentPathMap details

This is a sample configuration.<bean id="ASN"
class="com.infor.cswms.automationagent.content.DocumentPathMap">

```
<property name="properties">
<map>
<entry key="resource" value="/advancedshipnotice"/>
<entry key="headerValidationName" value="asnHeaderValidations"/>
<entry key="detailValidationName" value="asnDetailValidations"/>
<entry key="propertyLookupName" value="sharedPropertyLookups"/>
</map>
</property>
<!-- XPath properties - set a property from the document. Must evaluate to a single entry -->
<property name="pathProperties">
<map>
<entry key="messageId" value="/ShippingAdvice/ApplicationArea/MessageID[1]"/>
<entry key="messageType" value="/*/ApplicationArea[1]/Sender[1]/MessageInfo[1]/Name[1]"/>
</map>
</property>

<property name="headerPath" value = "//Header"/>
<property name="headers">
<map>
<entry key="externreceiptkey"
value="References/Reference[ReferenceID/@qualifier='EXTERNRECEIPTKEY']/Description"/>
<entry key="storerkey"
value="PartnerDetails/PartnerDetail/PartnerIDs/PartnerID[@agencyID='Schenker']"/>
</map>
</property>
<property name="detailName" value = "receiptdetails"/>
<property name="detailPath" value = "../Lines/ShippingLine"/> <!-- relative to header path -->
<property name="details">
<map>
<entry key="storerkey"
value="../../Header/PartnerDetails/PartnerDetail/PartnerIDs/PartnerID[@agencyID='Schenker']"/>
<entry key="sku" value="Products/Product/ProductIds/ProductId"/>
<entry key="externlineno"
value="References/Reference[ReferenceID/@qualifier='EXTERNLINENO']/Description"/>
<entry key="qtyexpected" value="Products/Product/Quantities/Quantity[@type='QTYEXPECTED']"/>
<entry key="susr5"
value="Products/Product/References/Reference[ReferenceID/@qualifier='SUSR8']/Description"/>
</map>
</property>
```

```
<property name="headerConstants">
<map>
<!-- <entry key="accountingEntity" value="AE1"/> -->
</map>
</property>
<property name="detailConstants">
<map>
<entry key="toloc" value="QC"/>
</map>
</property>
</bean>
```

Chapter 14 HashParserProcessor

This component maps one hashmap document to another using a DocumentPathMap

Sample configuration

This is a sample configuration.

```
<bean id="hashParser" class="com.infor.cswms.automationagent.processor.HashParserProcessor">
<property name="mapBeanName" value="${mapName}"/>
</bean>
```

Properties

properties

- Optional
- Description – A map of key-value pairs that will be added to the message context. This is available in all Processors through the framework.
- Type – map
- Default – none
- Valid Values – any of type
- Supports replacement - no

mapBeanName

- Required

- Description – The string name of the hash bean to be loaded
- Type – string pointing to DocumentPathMap bean
- Default – none
- Valid Values – any of type
- Supports replacement – yes

Sample configuration: DocumentPathMap Details

```
This is a sample configuration.</property>
<bean id="hashParser" class="com.infor.cswms.automationagent.processor.HashParserProcessor">
<property name="mapBeanName" value = "PICKING"/>
</bean>

<bean id="PICKING" class="com.infor.cswms.automationagent.content.HashParserProcessor">

<property name="headers">
<map>
        <entry key="orderid" value="refid"/>
        <entry key="pallet" value="case"/>
     </map>
</property>
<property name="detailName" value = "headerDetails"/>
<property name="detailPath" value = "newHeaderDetails"/>
<property name="details">
<map>
        <entry key="storerkey" value="PartnerID"/>
        <entry key="sku" value="SKU"/>
        <entry key="externlineno" value="LineItemID"/>
        <entry key="openqty" value="OPENQTY"/>
        <entry key="susr5" value="SUSR5'"/>
     </map>
</property>

</bean>

<!-- Determines what map to load, supports parameters so you can set the name using an xpath lookup -
->
<property name="mapBeanName" value="PICKING"/>
```

```
</bean>
```

Chapter 15 JMSProcessor

This component acts as a message sender or producer that sends messages to the message destination that could be either a queue or a topic.

Sample configuration

This is a sample configuration in the `processors.xml` file.

```
<bean id="jmsProcessor" class="com.infor.cswms.automationagent.processor.JMSProcessor">
<property name="defaultConnectionFactory" value=" java:jboss/exported/jms/RemoteConnectionFactory
"/>
<property name="initialContextFactory" value="org.jboss.naming.remote.client.InitialContextFactory"/>
<property name="messageDestination" value="${jmsdestinationname}"/>
<property name="providerURL" value="http-remoting://localhost:9480"/>
<property name="username" value="mquser"/>
<property name="password" value="mqpassword"/>
</bean>
```

Properties

defaultConnectionFactory

- Required
- Description – JMS connection factory that comes with the wildfly
- Type – path
- Default – none
- Valid Values – JNDI Names
- Supports replacement - no

initialContextFactory

- Optional
- Description – a factory that creates an initial context
- Type – file
- Default – none
- Valid Values – any of type
- Supports replacement – no

messageDestination

- Required
- Description – JMS destination name either Queue/Topic
- Type – string of the destination
- Default – none
- Valid Values – any of type
- Supports replacement – yes

providerURL

- Required
- Description – URL to the remote system
- Type – string
- Default – none
- Valid Values – any of type
- Supports replacement – no

username

- Required
- Description – username for the remote system
- Type – string
- Default – none
- Valid Values – any of type
- Supports replacement – no

password

- Required
- Description – password of username
- Type – string
- Default – none
- Valid Values – any of type
- Supports replacement – no

Sample configuration: DocumentPathMap details

This is a sample configuration.

```
<bean id="jmsProcessor" class="com.infor.cswms.automationagent.processor.JMSProcessor">
<property name="defaultConnectionFactory" value="jms/RemoteConnectionFactory"/>
<property name="initialContextFactory" value="org.jboss.naming.remote.client.InitialContextFactory"/>
<property name="messageDestination" value="${jmsdestinationname}"/>
<property name="providerURL" value="http-remoting://localhost:9480"/>
<property name="username" value="ususer"/>
<property name="password" value="uspassword"/>
</bean>
```

Chapter 16 DBProcessor

This component connects to a database and can be used to select, insert, or update tables based on the configuration. SQL queries can be segregated into header and detail queries.

Sample configuration

This is a sample configuration in the `processors.xml` file.

```
<bean id="dbPersist" class="com.infor.cswms.automationagent.processor.DBProcessor">
<property name="jndiname" value="java:/jdbc/S1103_WMWHSE1"/>
<property name="transactionISOLevel" value="8"/>

<property name="generateKeyValue" value="true"/>
<property name="headerSQL" value="SELECT S.PACKKEY,LA.LOT FROM SKU S JOIN LOTATTRIBUTE LA
ON S.STORERKEY=LA.STORERKEY AND S.SKU=LA.SKU
WHERE S.STORERKEY=:OwnerID AND S.SKU=:SKUID AND LA.LOTTABLE02=:BatchID"/>
<property name="detailSQL" value=""/>
<property name="detailName" value="detail"/>
<!-- Set the jndiResource and jndiPropertiesMap if we want to get a connection using local JNDI -->
<!--
<property name="jndiResource" value="localjndi"/>
<property name="jndiPropertiesMap" ref="jndiPropertiesValueMap"/>
-->
</bean>
<!--
 <util:map id="jndiPropertiesValueMap" map-class="java.util.HashMap">
<entry key="Driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
   <entry key="Username" value="wmwhse1"/>
   <entry key="Password" value="WMwhSqlWMwhSql1"/>
   <entry key="URL" value="jdbc:sqlserver://inhynckambhamp2:1433;databaseName=S1103"/>
</util:map>
```

--> |

Properties

jndiname

- Required
- Description – path of JNDI
- Type – path
- Default – none
- Valid Values – JNDI Names
- Supports replacement - yes

transactionISOLevel

- Optional
- Description – set transaction isolation level index
- Type – String
- Default – 8
- Valid Values – 0,1,2,4,8
- Supports replacement - yes

headerSQL

- Required
- Description – SQL statement that will select header information
- Type – SQL Statement
- Default – none
- Valid Values – a valid sql for the selected datasource
- Supports replacement - no

detailSQL

- Required
- Description – SQL statement that will select detail information for the headers selected
- Type – SQL statement
- Default – none
- Valid Values – a valid sql for the selected datasource
- Supports replacement - no

detailName

- Optional
- Description – Detail name to which detail-data is added
- Type – String
- Default – “details”
- Valid Values – any of type
- Supports replacement - no

jndiResource

- Optional
- Description – Determines whether to use local JNDI to get the Connection, local jndi will be used if this parameter is set
- Type – String
- Default – none
- Valid Values – localjndi
- Supports replacement - no

jndiPropertiesMap

- Optional
- Description – Refers a hashmap and use those to establish the Connection for the database.
- Type – hash map
- Default – none
- Valid Values – localjndi
- Supports replacement - no

jndiPropertiesValueMap

- Optional
- Description – A key value pairs of values determines the properties of the Database.
- Type – hash map
- Default – none
- Valid Values – key value pairs
- Supports replacement – no

generateKeyValue

- Optional
- Description – if this is set to true, a GUID will be generated and can be used to insert that GUID as a key value into the table values. Used for insert statements.
- Type – boolean
- Default – false
- Valid Values – true/false
- Supports replacement - no

Chapter 17 SocketClientProcessor

This component connects socket server for processing

This is a sample configuration in the `processors.xml` file.

```
<bean id="spsSocketClient" class="com.infor.cswms.automationagent.processor.SocketClientProcessor">
 <property name="hostName" value="localhost"/>
<property name="port" value="8005"/>
<property name="messageTypeInboundPositionMap" ref="inboundPositionMap" />
<!-- a hashmap containing the names to be assigned to data for a particular messagetype -->
<property name="messageTypeInboundValueMap" ref="inboundValueMap" />
<!-- a hashmap containing the positions for a particular messagetype -->
<property name="messageTypeOutboundPositionMap" ref="outboundPositionMap" />
<!-- a hashmap containing the names to be assigned to data for a particular messagetype -->
<property name="messageTypeOutboundValueMap" ref="outboundValueMap" />
<!-- filler character to be used for fixed length processing -->
<property name="paddingChar" value="*"/>
<!-- Fixed, Delimited, or Tab only fixed implemented for now -->
<property name="messageFormat" value="Delmited" />
<!-- if this string is found in response no error occured -->
<property name="successString" value="ERRCODE|000000000|" />
<!-- if this string is found in response error occured -->
<property name="errorString" value="ERROR1|000000000|" />
<!-- if response was not successfull an error enpoint will be set -->
<property name="errorEndpoint" value="postSpsErrorEndpoint" />
<!-- delmiter to be used when message format is delmited -->
<property name="delimiter" value="|" />
<!-- Is message communcation via strings or bytes -->
<property name="isBinary" value="Y" />
<!-- Message size to prepend to beggining of message -->
<property name="messageSizeBytes" value="8" />
<!-- Message to send to socket before main message -->
<property name="connectMessage"
```

```
value="CONNECT|CLIENTTYPE|10|CLIENTID|admin1|LANGUAGE|en_us|AUDIT_USER|admin1|" />
<!-- Message to send to socket after main message -->
<property name="disconnectMessage"
value="DISCONNECT|CLIENTTYPE|10|CLIENTID|admin1|LANGUAGE|en_us|AUDIT_USER|admin1|" />
<!-- prepend message size to main message -->
<property name="prePendMessageSizetoMessage" value="N" />
<!-- prepend message size to pre connect message -->
<property name="prePendMessageSizetoConnectMessage" value="Y" />
<!-- prepend message size to main message -->
<property name="prePendMessageSizetoDisconnectMessage" value="Y" />
</bean>
<util:map id="inboundPositionMap" map-class="java.util.HashMap">
   <entry key="01" value="4,4,4,2,2,2,12,12,2,12,2,4,2,8,4,2,36,2,2"/>
   <entry key="SPSRATEREQUESTED" value="10,16,7,10"/>
</util:map>
<util:map id="inboundValueMap" map-class="java.util.HashMap">
   <entry key="01"
value="Sender,schema,MessageLength,Sequence,TransmissionIndicator,resource,fromloc,toloc,Indicator,f
romid,Status,Device,End-KZ,PalletType,Contour,Height,Reserve,MessageTrailerType,Use"/>
   <entry key="SPSRATEREQUESTED" value="tlogkey,messageType,scac,SPSCaseKey"/>
</util:map>
<util:map id="outboundPositionMap" map-class="java.util.HashMap">
   <entry key="01" value="4,5,40,40,8"/>
   <entry key="SPSRATEREQUESTED" value="15,25,20"/>
</util:map>
<util:map id="outboundValueMap" map-class="java.util.HashMap">
   <entry key="01" value="Year,fromloc,fromid,Description,toloc"/>
   <entry key="SPSRATEREQUESTED" value="rate,tracknumber,spscasekey"/>
</util:map>
```

Properties

hostname

- Required

- Description – hostname of the server
- Type – string
- Default – none
- Valid Values – server name or IP address
- Supports replacement - no

port

- Required
- Description – port number
- Type – numeric
- Default – none
- Valid Values – valid port number
- Supports replacement - no

messageTypeInboundPositionmap

- Required
- Description – reference to utility map id for inbound positions
- Type – string
- Default – none
- Valid Values – any
- Supports replacement - no

messageTypeInboundValueMap

- Required
- Description – reference to a utility map id for inbound values
- Type – string
- Default – none
- Valid Values – any
- Supports replacement - no

messageTypeOutboundPositionMap

- Required
- Description – reference to utility map id for outbound positions
- Type – string
- Default – none
- Valid Values – any
- Supports replacement – no

messageTypeOutboundValueMap

- Required
- Description – reference to utility map id for outbound values
- Type – string
- Default – none
- Valid Values – any
- Supports replacement – no

paddingChar

- Required
- Description – padding character used with fixed position strings
- Type – string
- Default – “_”
- Valid Values – any
- Supports replacement - no

messageFormat

- Required
- Description – Fixed, Delimited, or Tab only fixed implemented for now
- Type – String
- Default – Fixed
- Valid Values – Fixed, Delimited or Tab
- Supports replacement – no

successString

- Optional
- Description – If this string is found in response no error occured
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement – no

errorString

- Optional
- Description – If this string is found in response error occurred
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement – no

errorEndpoint

- Optional
- Description – If response was not successful an error end point will be set
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement – no

delimiter

- Optional
- Description – delimited string value to be used when message format is delimited
- Type – Character
- Default – “|”
- Valid Values – any of type
- Supports replacement – no

isBinary

- Optional
- Description – This property is used to determine whether message communication is via strings or bytes
- Type – Character
- Default – ‘N’
- Valid Values – any of type
- Supports replacement – no

messageSizeBytes

- Optional
- Description – Message size to prepend to the beginning of message.
- Type – Integer
- Default – ‘0’
- Valid Values – any of type
- Supports replacement – no

connectMessage

- Optional
- Description – Message to send to socket before main message
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement – no

disconnectMessage

- Optional
- Description – Message to send to socket after main message
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement – no

prePendMessageSizetoMessage

- Optional
- Description – prepend message size of the main message
- Type – Character
- Default – ‘N’
- Valid Values – any of type
- Supports replacement – no

prePendMessageSizetoConnectMessage

- Optional
- Description – prepend message size of the pre connect message
- Type – Character
- Default – ‘N’
- Valid Values – any of type
- Supports replacement – no

prePendMessageSizetoDisconnectMessage

- Optional
- Description – prepend message size of the main message
- Type – Character
- Default – ‘N’
- Valid Values – any of type
- Supports replacement – no

Utility Map

inboundPositionMap

- Required
- Description – Key value pair for positions of fields in a fixed length record
- Type – string

- Default – none
- Valid Values – any
- Supports replacement - no

inboundValueMap

- Required
- Description – Key value pair for values fields in a fixed length record
- Type – string
- Default – none
- Valid Values – any
- Supports replacement – no

outboundPositionMap

- Required
- Description – Key value pairs for positions of fields in a fixed length record
- Type – string
- Default – none
- Valid Values – any
- Supports replacement – no

outboundValueMap

- Required
- Description – Key value pair for values fields in a fixed length record
- Type – string
- Default – none
- Valid Values – any
- Supports replacement – no

Chapter 18 SocketProcessor

This component acts as a client socket and listens for a socket connection and writes the messages by appending different fields onto a socket.

Sample configurationThis is a sample configuration in the `processors.xml` file.

```
<bean id="SenderSocketClientProcessor"
class="com.infor.cswms.automationagent.processor.SocketProcessor">
<property name="hostName" value="127.0.0.1"/>
<property name="port" value="5009"/>
<property name="paddingChar" value="0"/>
<property name="messageTypeOutboundPositionMap" ref="OutboundPositionMap" />
<property name="messageTypeOutboundValueMap" ref="OutboundValueMap" />
<property name="messageFormat" value="Fixed"/>
<property name="setAlignment" value="true"/>
<property name="fieldAlignment" value="right"/>
<property name="fieldAlignmentPositions" ref="FieldAlignmentPositionList"/>
<property name="inputPart" value="output"/>
<property name="reconnectSleepTime" value="2000"/>
<property name="maxRetry" value="1000"/>
</bean>
```

Properties

hostName

- Required
- Description – hostname of the socket
- Type – String
- Default – none

- Valid Values – hostname or IP address
- Supports replacement - no

port

- Required
- Description – port number socket will listen on
- Type – numeric
- Default – none
- Valid Values – port number
- Supports replacement - no

paddingChar

- Required
- Description – filler character to be used for fixed length processing
- Type – character
- Default – “_”
- Valid Values – character
- Supports replacement - no

messageFormat

- Required
- Description – Fixed, Delimited, or Tab only fixed implemented for now
- Type – character
- Default – Fixed
- Valid Values – Fixed, Delimited or Tab
- Supports replacement – no

messageTypeOutboundPositionMap

- Required
- Description – a hashmap containing the positions for a particular message type
- Type – map

- Default – none
- Valid Values – Hashmap
- Supports replacement – no

messageTypeOutboundValueMap

- Required
- Description – a hashmap containing the names to be assigned to data for a particular message type
- Type – map
- Default – none
- Valid Values – hashmap
- Supports replacement – no

setAlignment

- Optional
- Description – used to set alignment of the fields to get appended while writing to the socket.
- Type – boolean
- Default – false
- Valid Values – true or false
- Supports replacement - no

fieldAlignment

- Optional
- Description – used to specify the alignment of the different fields of a socket message
- Type – String
- Default – none
- Valid Values – left/right/center
- Supports replacement - no

fieldAlignmentPositions

- Optional

- Description – Specifies what positions of fields to be aligned in a message
- Type – list
- Default – none
- Valid Values – List
- Supports replacement – no

reconnectSleepTime

- Optional
- Description – Time interval to retry socket connection if connection is dropped
- Type – Integer
- Default – “2000”
- Valid Values – any integer value
- Supports replacement – no

maxRetry

- Optional
- Description – Maximum number of retry attempts to connect to a socket server
- Type – Integer
- Default – “1000”
- Valid Values – any integer value
- Supports replacement – no

Chapter 19 FileWriterProcessor

This component can write data directly to a file or it can convert the data to XML, JSON, or Delimited format and then write to a file. This component accepts property for file-size in MB. If the file size exceeds the maximum size then a new file will be created and the data will be written into that file. If the append property is set to true, then the file is opened in append mode. The new content is written at the end of the file. If the **useInProcessPath** property is set to true and if the file already exists, then it moves that file to **inProcessPath** path before processing. After processing is done it moves the file back to original location.

Sample configuration

This is a sample configuration.

```
<bean id="FileWriterProcessor" class="com.infor.cswms.automationagent.processor.FileWriterProcessor">
<property name="filePath" value="C:\demo\${timestamp}_data.txt "/>
<property name="inputMapName" value="maptoDelimited" />
<property name="append" value="true" />
<property name="maxsize" value=”5” />
<property name="useInProcessPath" value="true" />
<property name="inProcessPath" value="C:\demo\inprocess" />
<property name="timeStampFormat" value="dd-MM-yyyy HH-mm-ss" />
</bean>

<bean id="maptoDelimited"
class="com.infor.cswms.automationagent.content.MapMessageToDelimitedImpl">
<property name="messageTypeinboundboundValueMap" ref="inboundValueMap" />
<property name="delimiter" value="," />
<property name="defaultValue" value="" />
</bean>

<util:map id="inboundValueMap" map-class="java.util.HashMap">
<entry key="header" value="serialkey,WaveKey,WaveType,EditWho" />
```

```
</util:map>

    |
```

FileWriterProcessor -Properties

filePath

- Required
- Description – path of writing directory
- Type – path
- Default – none
- Valid Values – none
- Supports replacement – yes

inputMapName

- Optional
- Description – A map to be applied to the data before writing. If empty, the data will be written as is.
- Type – Reference to a MessageMapper
- Default – none
- Valid Values – Any of type
- Supports replacement – no

append

- Optional
- Description – If this is set to true it opens the file in append mode
- Type – Boolean
- Default – “false”
- Valid Values – “true” or “false”
- Supports replacement – no

maxsize

- Optional
- Description – It is the maximum size up to which a file can be written. If it exceeds the specified file size, A new file will be created
- Type – numeric
- Default – 10
- Valid Values – 1-100
- Supports replacement – no

useInProcessPath

- Optional
- Description – File will be moved to the processing path for processing
- Type – Boolean
- Default – “false”
- Valid Values – “true” or “false”
- Supports replacement – no

inProcessPath

- Required
- Description – path in which the files will be moved for processing
- Type – path
- Default – none
- Valid Values – a valid path
- Supports replacement – yes

timeStampFormat

- Optional
- Description –The time stamp format with which the file will be created
- Type – String
- Default – “MM-dd-yyyy HH-mm-ss”
- Valid Values – valid timestamp format
- Supports replacement – no

Chapter 20 SoapProcessor

This component allows the Automation Agent to make a call to an API using Simple Object Access Protocol (SOAP). It is intended for making calls to legacy Infor WMS applications and will build a proprietary XML payload that will not be suitable for other applications. You should use the REST or WebService processors for those applications, or for current versions of Infor WMS where REST is supported.

Sample configuration

This is a sample configuration.

```
<bean id="dbSchenkerSoapPost"
class="com.infor.cswms.automationagent.processor.WMSoapProcessor">

<property name="hostName" value="inhynckambhamp2" />
<property name="port" value="8080" />
<property name="endpointName" value="soapEndpoint" />
<!-- soapclient settings below default to wm settings if not set -->
<property name="action" value="executeAPI" />
<property name="nameSpace" value="http://mobility.sce.webservices.infor.com/" />
<property name="nameSpacePrefix" value="mob" />
<property name="portName" value="MobilityRemotePort" />
<property name="url" value="http://SERVERNAME:PORT/wmwebservice_ejb/MobilityBean?wsdl" />

</bean>

<bean id="soapEndpoint " class="com.infor.cswms.automationagent.util.WMSoapEndpoint">
  <property name="messageType" value="Location" />
  <property name="action" value="store" />
  <property name="userName" value="${username}" />
  <property name="password" value="${password}" />
  <property name="tenant" value="${tenant}" />
```

```
    <property name="recipient" value="${schema}" />
    <property name="objectLevelTag" value="Location" />
    <property name="systemID" value="${schema}" />
</bean>
```

Properties

hostName

- Required
- Description – host name or IP address
- Type – string
- Default – none
- Valid Values – none
- Supports replacement – yes

port

- Required
- Description – port number
- Type – numeric
- Default – 8080
- Valid Values – none
- Supports replacement – yes

endpointName

- Required
- Description – API name
- Type – string
- Default – none
- Valid Values – none
- Supports replacement – no

action

- Optional
- Description – action
- Type – string
- Default – “executeAPI”
- Valid Values – none
- Supports replacement – no

nameSpace

- Optional
- Description – Name space
- Type – string
- Default – http://mobility.sce.webservices.infor.com/
- Valid Values – none
- Supports replacement – no

nameSpacePrefix

- Optional
- Description – prefix of namespace
- Type – path
- Default – “mob”
- Valid Values – none
- Supports replacement – no

url

- Optional
- Description – Uniform Resource Locator
- Type – URL
- Default – http://SERVERNAME:PORT/wmwebservice_ejb/MobilityBean?wsdl
- Valid Values – none
- Supports replacement – yes

Properties- WMSoapEndpoint

messageType

- Required
- Description – The API name (ShipmentOrderAPI, PackAPI, etc.) for the message
- Type – String
- Default – none
- Valid Values – any existing
- Supports replacement – no

action

- Required
- Description – The method to be executed on the API
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

userName

- Required
- Description – username to authenticate the soap call
- Type – String
- Default – none
- Valid Values – valid username
- Supports replacement – yes

password

- Required
- Description – password for the username
- Type – String

- Default – none
- Valid Values – valid password encoding supported
- Supports replacement – yes

tenant

- Required
- Description – Tenant name
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – yes

recipient

- Required
- Description – The recipient system corresponds to the datasource name for the target warehouse
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – yes

objectLevelTag

- Required
- Description – Defines the top-level repeating element containing multiple headers
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

systemID

- Required

- Description – name of the sending system. This will have no functional impact but will be used for logging and troubleshooting.
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – yes

Chapter 21 REST Processor

This component manages calls to REST endpoint services. The configuration consists of the **RestProcessor** bean, as well as one or more **RestEndpoint** beans and possibly an optional **OAuthClientConfig** bean.

Sample configuration

This is a sample configuration.<bean id="restPost" class="com.infor.cswms.automationagent.processor.RestProcessor">

<property name="hostName" value="${restHostName}"/>

<property name="port" value="${port}"/>

<property name="contextRoot" value="/wmwebservice_rest/"/>

<property name="headers">

<map>

<entry key="x-infor-identity2" value="automation"/>

<entry key="x-infor-tenantID" value="${tenant}"/>

<entry key="Content-Type" value="application/json"/>

</map>

</property>

<property name="oauth">

<ref bean="oauth1_mylocalsce" />

</property>

<property name="endpointName" value="${endpoint}"/>

</bean>

<bean id="oauth1_mylocalsce" class="com.infor.cswms.automationagent.util.OAuthClientConfig">

<property name="oauthVersion" value="1.0"/>

<property name="clientId" value="ci_here"/>

<property name="clientSecret" value="cs_here"/>

```
</bean>

<bean id="oauth2_SCEQA9CUST1_AX1"
class="com.infor.cswms.automationagent.util.OAuthClientConfig">
<property name="oauthVersion" value="2.0"/>
<property name="authorizationEndpoint" value="https://qac-
sso.qac.awsdev.infor.com:443/SCEQA9CUST1_AX1/as/token.oauth2"/>
 <property name="accountName" value="accountName"/>
 <property name="accountPassword" value="accountPass"/>
 <property name="clientId" value="ci"/>
 <property name="clientSecret" value="cs"/>
</bean>

<bean id="exportInterfaceEndpoint" class="com.infor.cswms.automationagent.util.RestEndpoint">
 <property name="verb" value="GET"/>
 <property name="resource" value="${schema}/exports"/>
 <!--<property name="sendMap" value="..."/>-->
<property name="responseMap" value="exportSplitter"/>
 <property name="parameters">
<map>
      <entry key="type" value="DEPOSIT|CUSTOMERORDERCREATED"/>
<entry key="updatestatus" value="5"/>
      <entry key="pollersearchstatus" value="0"/>
         <entry key="transmitflagtouse" value="TRANSMITFLAG2"/>
         <entry key="eventcategory" value="M"/>
         <entry key="asjson" value="true"/>
      </map>
</property>
</bean>

<bean id="PickWaveRestPost" class="com.infor.cswms.automationagent.processor.RestProcessor">
<property name="hostName" value="purkan.6river.org"/>
<property name="contextRoot" value="/cfs/v2/"/>
<property name="headers">
<map>
<entry key="Content-Type" value="application/json+vnd+pur"/>
</map>
</property>
<property name="authorizationHeaders">
```

```
<map>
<entry key="type" value="basic"/>
<entry key="username" value="PURKAN"/>
<entry key="password" value="PjKpSSXThXGn"/>
</map>
</property>
<property name="endpointName" value="PickWaveEndPoint"/>
<property name="inputPart" value="pickwavejoltoutput"/>
<property name="outputPart" value="restoutput"/>
</bean>
```

Properties: RestProcessor

hostName

- Required
- Description – The server name that will be called to make the actual REST call. This value is used in building the URL.
- Type – String
- Default – none
- Valid Values – Must be a server name accessible from the machine where Automation Agent is running
- Supports replacement - yes

port

- Required
- Description – The port used when making the actual REST call. This value is used in building the URL.
- Type – Datatype is String to support replacement, but must evaluate to a port (int)
- Default – 443
- Valid Values – Must be an open port listening for REST calls on the server specified
- Supports replacement - yes

contextRoot

- Optional
- Description – The prefix for all calls made from this client. This value is used in building the URL.
- Type – String
- Default – none
- Valid Values – Must point to a valid context root on the server. Can be omitted if each endpoint includes the context root as part of its URL.
- Supports replacement - yes

Headers

- Optional
- Description – A map of HTTP headers to be added to the message when making the REST call
- Type – Map
- Default – none
- Valid Values – Any
- Supports replacement - yes

Oauth

- Optional
- Description – A reference to an OAuthClientConfig bean if OAuth is being used.
- Type – Reference to OAuthClientConfig
- Default – none
- Valid Values – Any of type
- Supports replacement - no

endpointName

- Required
- Description – A reference to a RestEndpoint giving information about the call to be made to the server. In almost every instance you will want this to be a replacement variable so that the RestProcess can be used for more than just one call.
- Type – Reference to RestEndpoint
- Default – none

- Valid Values – Any of type
- Supports replacement - yes

authorizationHeaders

- Optional
- Description – A map of authorization headers to be added to the message when making the REST call
- Type – Map
- Default – none
- Valid Values – Any
- Supports replacement - yes

Properties: OAuthClinetConfig

oauthVersion

- Required
- Description – Determines if OAuth 1.0a or 2 will be used.
- Type – String
- Default – none
- Valid Values – “1.0” or “2.0”
- Supports replacement - no

clientId

- Required
- Description – The client id (commonly called ci) for the OAuth validation.
- Type – String
- Default – none
- Valid Values – Must be the client id specified by the server for validation
- Supports replacement - no

clientSecret

- Required
- Description – The client secret (commonly called cs) for the OAuth validation
- Type – String
- Default – none
- Valid Values – Must be the client secret specified by the server for the give client id
- Supports replacement - no

authorizationEndpoint

- Required if OAuth 2
- Description – The authorization URL used to get the bearer token for OAuth 2.0. This will must be provided by the REST service provider.
- Type – String
- Default – none
- Valid Values – A valid URL pointing to an authorization service hosted by the provider of the REST service
- Supports replacement - no

accountName

- Required if OAuth 2
- Description – OAuth 2 uses a password grant, this will be an abstraction of the account associated with that grant. Commonly called “saak”.
- Type – String
- Default – none
- Valid Values – Must be a key for a user with permission to access the requested REST API
- Supports replacement - no

accountPassword

- Required if OAuth 2
- Description – OAuth 2 uses a password grant, this will be an abstraction of the password for the account associated with that grant. Commonly called “sask”.
- Type – String
- Default – none

- Valid Values – Must be a key for the password for a user with permission to access the requested REST API
- Supports replacement - yes

Properties: RestEndpoint

verb

- Required
- Description – The HTTP verb to use for the call
- Type – String
- Default – none
- Valid Values – GET, POST, DELETE, PUT
- Supports replacement - yes

resource

- Required
- Description – The URL end for the call to be made. This will be appended after the host, port, and context root to form the URL.
- Type – String
- Default – none
- Valid Values – A published URL endpoint on the server being called
- Supports replacement - yes

sendMap

- Optional
- Description – A map to be applied to the data before sending. If empty, the data will be sent as is.
- Type – Reference to a Mapper
- Default – none
- Valid Values – Any of type
- Supports replacement - yes

responseMap

- Optional
- Description – A map to be applied to the response received. If empty, the data will be saved as is.
- Type – Reference to a Mapper
- Default – none
- Valid Values – Any of type
- Supports replacement - yes

parameters

- Optional
- Description – A Map of key value pairs to be sent with the request as query parameters
- Type – Map
- Default – none
- Valid Values – Any supported by the call being made
- Supports replacement - yes

Chapter 22 Validation Lookup Processor

This component performs validations and lookups. If a validation fails, then an error alert is sent. Lookups map a value from the message properties or message header to create new entries or replace existing values.

Sample configuration

This is a sample configuration.<bean id="validationProcessor"
class="com.infor.cswms.automationagent.processor.LookupValidationProcessor" scope="prototype">

<property name='headerValidationName' value='${headerValidationName}'/>

<property name='detailValidationName' value='${detailValidationName}'/>

<property name='propertyLookupName' value='${propertyLookupName}'/>

<property name='headerLookupName' value='headerCoordinatesLookups'/>

<property name='detailLookupName' value='detailCoordinatesLookups'/>

</bean>

<util:list id="asnHeaderValidations" list-class="java.util.ArrayList">

<ref bean="storerkeyValidation" />

</util:list>

<util:list id="asnDetailValidations" list-class="java.util.ArrayList">

<ref bean="storerkeyValidation" />

<ref bean="quantityValidation" />

</util:list>

<util:list id="shipmentHeaderValidations" list-class="java.util.ArrayList">

<ref bean="storerkeyValidation" />

</util:list>

<util:list id="shipmentDetailValidations" list-class="java.util.ArrayList">

<ref bean="storerkeyValidation" />

```
</util:list>


<util:list id="sharedPropertyLookups" list-class="java.util.ArrayList">
<ref bean="lookupFacility" />
</util:list>

<bean id="storerkeyValidation" class="com.infor.cswms.automationagent.content.Validation">
<property name="name" value="storerkey"/>
<property name="required" value="true"/>
<property name="dataType" value="String"/>
<property name="min" value="0"/>
<property name="max" value="50"/>
<property name="defaultValue" value="AE1"/>
<property name="lookupMap" ref="storerKeyLookupMap"/>
<property name="lookupRequired" value="false"/>
</bean>

<util:map id="storerKeyLookupMap" map-class="java.util.HashMap">
<entry key="CUSTOMER_223" value="AE1"/>
<entry key="CUSTOMER_786" value="AE2"/>
</util:map>

<bean id="lookupFacility" class="com.infor.cswms.automationagent.content.Lookup">
<property name="property" value="schema"/>
<property name="sourceName" value="storerkey"/>
<property name="sourceType" value="header"/>
<property name="defaultValue" value="INFOR_ENTERPRISE"/>
<property name="lookupMap" ref="facilityLookupMap"/>
<property name="lookupRequired" value="true"/>
</bean>

<util:map id="facilityLookupMap" map-class="java.util.HashMap">
<entry key="AE1" value="INFOR_SCP11_wmwhse1"/>
<entry key="AE2" value="INFOR_SCP11_wmwhse2"/>
</util:map>

<util:list id="headerCoordinatesLookups" list-class="java.util.ArrayList">
```

```
<ref bean="XLookup" />
</util:list>

<bean id="XLookup" class="com.infor.cswms.automationagent.content.Lookup">
<property name="property" value="action"/>
<property name="sourceName" value="Xaxis"/>
<property name="sourceType" value="header"/>
<property name="defaultValue" value="none"/>
<property name="lookupMap" ref="Xmap"/>
<property name="lookupRequired" value="false"/>
<property name="destinationName" value="abc"/>
</bean>
<util:map id="Xmap" map-class="java.util.HashMap">
<entry key="1" value="0001"/>
<entry key="2" value="0002"/>
</util:map>
```

Properties: LookupValidationProcessor

headerValidationName

- Optional
- Description – A reference to a list of validations to be run against header level elements in the message
- Type – List of Validation
- Default – none
- Valid Values – Any of type
- Supports replacement – yes

detailValidationName

- Optional
- Description – A reference to a list of validations to be run against detail level elements in the message
- Type – List of Validation

- Default – none
- Valid Values – Any of type
- Supports replacement – yes

propertyLookupName

- Optional
- Description – A reference to a list of lookups to be performed to set or update properties
- Type – List of Lookup
- Default – none
- Valid Values – Any of type
- Supports replacement – yes

headerLookupName

- Optional
- Description – A reference to a list of Lookups to performed to update header properties
- Type – List of Lookup
- Default – none
- Valid Values – Any of type
- Supports replacement – yes

detailLookupName

- Optional
- Description – A reference to a list of Lookups to performed to update detail properties
- Type – List of Lookup
- Default – none
- Valid Values – Any of type
- Supports replacement – yes

Properties: Validation

name

- Required
- Description – The name of the element to be validated
- Type – String
- Default – none
- Valid Values – Any
- Supports replacement – no

required

- Required
- Description –Will raise an exception if a required value does not exist
- Type – String
- Default – false
- Valid Values – “true”, “false”
- Supports replacement – no

datatype

- Optional
- Description – If specified, will validate that the value is of the datatype specified
- Type – String
- Default – none
- Valid Values – “Numeric”, “TimeStamp”
- Supports replacement – no

min

- Optional
- Description – Minimum value if datatype is Numeric, minimum string length otherwise.
- Type – Integer

- Default – none
- Valid Values – Any of type
- Supports replacement – no

max

- Optional
- Description – Maximum value if datatype is Numeric, maximum string length otherwise.
- Type – Integer
- Default – none
- Valid Values – Any of type
- Supports replacement – no

defaultValue

- Optional
- Description – A default value to use if the element does not exist
- Type – String
- Default – none
- Valid Values – Any of type
- Supports replacement – no

lookupMap

- Optional
- Description – A map to convert one value to another (i.e. “CUSTOMER_223” to “AE1”)
- Type – Reference to Map
- Default – none
- Valid Values – Any of type
- Supports replacement – no

lookupRequired

- Optional
- Description – If true and the value is not in the specified lookupMap, will raise an alert

- Type – boolean
- Default – false
- Valid Values – true, false
- Supports replacement – no

Properties: Lookup

property

- Required
- Description – The name of the property to be set
- Type – String
- Default – none
- Valid Values – Any of type
- Supports replacement – no

sourceName

- Required
- Description – The name of the header or property to use as a data source
- Type – String
- Default – none
- Valid Values – Any header or property
- Supports replacement – no

destinationName

- Optional
- Description – A new property that will be added to header with value from lookup map
- Type – String
- Default – none
- Valid Values – Any of type
- Supports replacement – no

sourceType

- Required
- Description – Determines if the source to lookup is from the message header or properties
- Type – String
- Default – none
- Valid Values – “header”, “property”
- Supports replacement – no

inputSource

- Optional
- Description – The message part name from where the source value is to be fetched
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement – no

defaultValue

- Required
- Description – The value to use if the sourceName does not exist on the sourceType
- Type – String
- Default – none
- Valid Values – Any of type
- Supports replacement – no

lookupMap

- Optional
- Description – A map to convert one value to another (i.e. “AE1” to “INFOR_SCP11_wmwhse1”)
- Type – Reference to Map
- Default – none
- Valid Values – Any of type
- Supports replacement – no

lookupRequired

- Optional
- Description – If true and the value is not in the specified lookupMap, will raise an alert
- Type – boolean
- Default – false
- Valid Values – true, false
- Supports replacement – no

Chapter 23 ProcessChainRouter

This component uses instructions to manipulate the process chain based on data. ProcessChains using this must be scoped as prototype to avoid unpredictable behavior.

To read an instruction, use this format: **`if {sourceName} in {sourceType} {sourceCondition} {sourceValue} then {changeType} {changeReference}`**.

Sample configuration

This is a sample configuration.

For example, if the lot in the header equals 0000000007 then replaceChain with erroredEventChain. This makes greater or less than comparisons more natural. The sourceName is always on the left and the sourceValue on the right, so lot is greaterThan 0000000007 in this scenario if we changed the sourceCondition.

```
<bean id="inboundProcessRouter"
class="com.infor.cswms.automationagent.processor.ProcessChainRouter">
<property name='processListUntil' value='firstMatch'/>
<property name="instructions">
<list>
<ref bean="lot7RI" />
 <ref bean="storerRouteAE2" />
</list>
</property>
</bean>

<bean id="lot7RI" class="com.infor.cswms.automationagent.content.RoutingInstruction">
<property name="changeType" value="replaceChain"/>
<property name="changeReference" value="erroredEventChain"/>
<property name="sourceType" value="header"/>
<property name="sourceName" value="lot"/>
<property name="sourceCondition" value="equals"/>
```

```
<property name="sourceValue" value="0000000007"/>
</bean>
```

Properties: ProcessChainRouter

processListUntil

- Required
- Description – Determines if all instructions should be evaluated, if the router should stop on the first success, or if all instructions must pass. When using “and”, only the changes from the last instruction will be applied, the others are ignored.
- Type – String
- Default – none
- Valid Values – firstMatch, all, and
- Supports replacement - no

instructions

- Required
- Description – A list of RoutingInstructions to be processed
- Type – list of reference to type RoutingInstruction
- Default – none
- Valid Values – any of type
- Supports replacement - no

Properties: RoutingInstruction

changeType

- Required

- Description – Determines if the chain or process in changeReference will be inserted before, appended after, or replace the existing chain
- Type – String
- Default – none
- Valid Values – nextStep, appendStep, replaceChain, insertChain, appendChain
- Supports replacement - no

changeReference

- Required
- Description – The bean name for the chain or processor to insert before, append to, or replace the current chain
- Type – String
- Default – none
- Valid Values – The name of a chain if changeType references a chain, or processor if changeType references a step
- Supports replacement - no

sourceType

- Required
- Description – Where should the value to be tested come from
- Type – String
- Default – none
- Valid Values – header, properties
- Supports replacement - no

sourceName

- Required
- Description – The name of the header or property to evaluate
- Type – String
- Default – none
- Valid Values – Any existing header or property name
- Supports replacement - no

sourceCondition

- Required
- Description – Type of evaluation to make from the sourceName against the sourceValue with sourceName on the left, and sourceValue on the right. See the description above for a sample. *NOTE, these values are case sensitive.
- Type – String
- Default – none
- Valid Values – equals, notEquals, contains, notContains, startsWith, notStartsWith, endsWith, notEndsWith, lessThan, lessThanOrEqual, greaterThan, greaterThanOrEqual, isNullOrEmpty, isNotNullOrNotEmpty
- Supports replacement - no

sourceValue

- Required
- Description – The value against which source condition is evaluated
- Type – String
- Default – none
- Valid Values – Any of type
- Supports replacement – no

Chapter 24 FileLoggingProcessor

This component takes named message parts list and writes the message parts, exceptions, and properties data to a text file in the directory with the file name that is specified by the user

Sample configuration

This is a sample configuration in the initiators.xml file.

```
<bean id="fileLoggingProcessor"
class="com.infor.cswms.automationagent.processor.FileLoggingProcessor">
<property name="fileoutputpath" value="c:\demo"/>
<property name="filename" value="demo.xml"/>
<property name="messageparts" ><!-- partnames -->
<list>
 <value>part1</value>
 <value>part2</value>
 <value>splitoutput</value>
 <value>response_4</value>
 <value>lookupoutput</value>
 <!-- <value>all</value> -->
</list>
</property>
<property name="exceptions" value="true"/>
<property name="properties" value="true"/>
</bean>
```

Properties

fileoutputpath

- Required
- Description – directory where the file is to be created
- Type – directory
- Default – none
- Valid Values – a valid directory
- Supports replacement - no

filename

- Required
- Description – name of the file
- Type – text
- Default – none
- Valid Values – file types
- Supports replacement - no

messageparts

- Optional
- Description – It is a list which accepts the message partnames
- Type – String
- Default – none
- Valid Values – Any message part name or “all”
- Supports replacement - no

exceptions

- Optional
- Description – If this is set to true it writes the message exceptions to a file
- Type – Boolean

- Default – “false”
- Valid Values – “true” or “false”
- Supports replacement - no

properties

- Optional
- Description – If this is set to true it writes the message properties to a file
- Type – Boolean
- Default – “false”
- Valid Values – “true” or “false”
- Supports replacement - no

Chapter 25 AppendMessagePartsProcessor

This component takes two named parts of the message stack and appends them to another named part and adds it to message stack.

Sample configuration

This is a sample configuration.

```
<bean id="appendMessagePartsProcessor"
class="com.infor.cswms.automationagent.processor.AppendMessagePartsProcessor">
<property name="messagePart1" value="socketoutput"/>
<property name="messagePart2" value="restoutput"/>
<property name="outputPart" value="appendedoutput"/>
</bean>
```

Properties: AppendMessagePartsProcessor

messagePart1

- Required
- Description – Takes the named part of message stack if set otherwise gets the most recently added data of the message stack.
- Type – String
- Default – none
- Valid Values – String
- Supports replacement - no

messagePart2

- Required
- Description – Takes the named part of message stack if set otherwise gets the most recently added data of the message stack.
- Type – String
- Default – none
- Valid Values – String
- Supports replacement - no

outputPart

- Optional
- Description – The appended contents are added with the name provided in this property and gets added to message stack.
- Type – String
- Default – none
- Valid Values – String
- Supports replacement - no

Chapter 26 InstructionChainProcessor

This component takes a list of manipulation instructions and executes each instruction in order. to support the functions that follow. Functions can affect properties, headers, or details. Where headers or details are referenced, all headers and details are accessed. You must make logical configurations.

For example, if you set your result to a property, meaningful results are not produced if your instructions are to add two detail values. Only the last detail will be produced for the property. You can add a detail to a property storing the result in the property. This will create sum function for a single parameter from all details.

These are the available actions:

- Add, subtract, multiply, divide
  - Performs the function: source1 +-*/ source2 and stores the result in destination
  - Values must be numeric
  - If a value is not set, it is default to effectively no-op (add/subtract default to 0, multiply to 1, dividend to 0, divisor to 1)
- Increment, decrement
  - Increments or decrements the destination
  - Must be an integer already initialized
- Initialize – sets destination to source1 if it has not already been set
- Set – sets destination to source1 regardless of its current state
- Concatenate – sets destination to string concatenation of source1 + source2
- Sleep – sleeps for source1 in seconds or millis per sourceType1
- Date – Gets the source1 dateformat and converts it to source2 dateformat

Sample configuration

This is a sample configuration.

```
<bean id="variantChain"
class="com.infor.cswms.automationagent.processor.InstructionChainProcessor">
<property name="instructions">
```

```
<list>
<ref bean="instruction1" />
<ref bean="instruction2" />
</list>
</property>
</bean>
```

Properties: InstructionChainProcessor

instructions

- Required
- Description – A list of ManipulationInstruction
- Type – list
- Default – none
- Valid Values – Reference to ManipulationInstruction
- Supports replacement - no

Sample configuration

This is a sample configuration.

```
<bean id="instruction1" class="com.infor.cswms.automationagent.content.ManipulationInstruction">
<property name="instructionType" value="add"/>
<property name="destinationName" value="resultDetail"/>
<property name="destinationType" value="detail"/>
<property name="detailName" value="details"/>
<property name="sourceType1" value="detail"/>
<property name="source1" value="num-d2"/>
<property name="sourceType2" value="constant"/>
<property name="source2" value="0.1"/>
</bean>
```

Properties: ManipulationInstruction

instructionType

- Required
- Description – The task to execute for this instruction
- Type – String
- Default – none
- Valid Values – add, subtract, multiply, divide, increment, decrement, initialize, set, concatenate, sleep, date
- Supports replacement - no

destinationName

- Required for all but sleep
- Description – The name of the property, header, or detail to change
- Type – String
- Default – none
- Valid Values – any
- Supports replacement - no

destinationType

- Required for all but sleep
- Description – The location of the destination
- Type – String
- Default – none
- Valid Values – property, header, detail
- Supports replacement – no

detailName

- Required if accessing a detail
- Description – The name of the detail array in the header map
- Type – String

- Default – none
- Valid Values – The name of the detail list
- Supports replacement – no

sourceType1

- Required for all but increment / decrement
- Description – The location of source1
- Type – String
- Default – none
- Valid Values – property, header, detail, constant, millis, seconds
- Supports replacement – no

source1

- Required for all but increment / decrement
- Description – The first value in an operation
- Type – String
- Default – none
- Valid Values – name of the property, header, or detail to use OR the value for constant, seconds, and millis
- Supports replacement – no

sourceType2

- Required for two argument operations (math and concatenate)
- Description – The location of source2
- Type – String
- Default – none
- Valid Values – property, header, detail, constant
- Supports replacement – no

Source2

- Required for two argument operations (math and concatenate)

- Description – The first value in an operation
- Type – String
- Default – none
- Valid Values – name of the property, header, or detail to use OR the value for constant
- Supports replacement – no

Chapter 27 TransformMessageToXMLProcessor

This component takes the message object in array list of hash map objects. It formats and transforms it to an XML structure and adds it to the message stack as string format.

Sample configuration

This is a sample configuration in the `processors.xml` file.

```
 <bean id="transformMessageToXMLProcessor"
class="com.infor.cswms.automationagent.processor.TransformMessageToXMLProcessor">
<property name="useNameSpace" value="true"/>
<property name="nameSpaceElementName"  value="Download"/>
<property name="nameSpaceURI" value="http://Dematic.com.au/WCSXMLSchema/DBSchenkerSG"/>
<property name="nameSpacePrefix" value="p"/>
<!-- look into datamaps.xml file for query column mapping -->
<property name="mapName" value="manualInductPackColumnMap"/>
<property name="omitDeclaration" value="no"/>
</bean>
```

Properties

nameSpaceElementName

- Optional
- Description – Specifies the name of the namespace element
- Type – String
- Default – none

- Valid Values – any string
- Supports replacement – no

nameSpaceURI

- Optional
- Description – Specifies the URI for the namespace element
- Type – String
- Default – none
- Valid Values – URI
- Supports replacement – no

nameSpacePrefix

- Optional
- Description – Specifies the prefix for the namespace element
- Type – String
- Default – none
- Valid Values – any string literal
- Supports replacement – no

useNameSpace

- Optional
- Description – Specifies whether to use any namespace element in the XML
- Type – boolean
- Default – true
- Valid Values – true/false
- Supports replacement – no

omitDeclaration

- Optional
- Description – Specifies whether to have xml version declaration on top of xml string
- Type – String

- Default – no
- Valid Values – yes/no
- Supports replacement – no

mapName

- required
- Description – Specifies Document path map structure which specifies headers, details of a list to be transformed into an XML string
- Type – String
- Default – no
- Valid Values – map bean names
- Supports replacement – no

Chapter 28 TransformMessageToJSONProcessor

This component takes the message object in array list of hash map objects, formats and transforms it to a JSON structure and adds it to the message stack as string format.

Sample configuration

This is a sample configuration in the `processors.xml` file.

```
<bean id="transformMessagetoJson"
class="com.infor.cswms.automationagent.processor.TransformMessageToJSONProcessor">
<!-- look into datamaps.xml file for query column mapping -->
<property name="mapName" value="6RiversPickWaveColumnMap"/>
<property name="returnJsonAsArray" value="true"/>
</bean>
```

Properties

mapName

- required
- Description – Specifies Document path map structure which specifies headers, details of a list to be transformed into a JSON string
- Type – String
- Default – no
- Valid Values – map bean names
- Supports replacement – no

returnJsonAsArray

- Optional
- Description – If this property is true Json data will be returned as an array
- Type – Boolean
- Default – “false”
- Valid Values – “true”, “false”
- Supports replacement – no

Chapter 29 TransformJSONUsingJoltProcessor

This component is used to transform the simple JSON to complex JSON structure using a template called JOLT spec json. The JOLT spec json can be specified or designed by following the JOLT library specifications.

Sample configuration

This is a sample configuration in the `processors.xml` file.

```
<bean id="transformJsonUsingJoltProcessor"
class="com.infor.cswms.automationagent.processor.TransformJSONUsingJoltProcessor">
<property name="inputJSON" value="transformedOutput"/>
<property name="inputJSONReference" value="Message"/> <!-- Message,File,String -->
<property name="transformJSON" value="./configuration/clients/pure-
fishing/consolidatedconfigs/6Rivers/6RiversPickWaveAPI/resources/pickwavejoltspec.json"/>
<property name="transformJSONReference" value="File"/><!-- Message,File,String -->
<property name="outputPart" value="pickwavejoltoutput"/>
</bean>
```

Properties

inputJSON

- Required
- Description – Specifies message part name, path of a json file, or a json string to act as input to the template/spec json.
- Type – String
- Default – no

- Valid Values – input part name or file path or json string
- Supports replacement – no

inputJSONReference

- Required
- Description – Specifies the reference of inputJSON and it could be referred from message/file/json string.
- Type – String
- Default – no
- Valid Values – Message/File/String
- Supports replacement – no

transformJSON

- Required
- Description – Specifies message part name, path of a spec json file, or a spec json string to transform the input JSON to spec JSON structure.
- Type – String
- Default – no
- Valid Values – input part name or file path or json string
- Supports replacement – no

transformJSONReference

- Required
- Description – Specifies the reference of transformJSON and it could be referred from message/file/json string.
- Type – String
- Default – no
- Valid Values – Message/File/String
- Supports replacement – no

Chapter 30 MapMessageToXMLImpl

This component is used to map message elements with different names and convert it to XML structure. It is used in File Writer Processor to write to XML file.

Sample configuration

This is a sample configuration in the `datamaps.xml` file.

```
<bean id="ModulaPutawayTaskCreatedColumnMap"
class="com.infor.cswms.automationagent.content.MapMessageToXMLImpl">
  <property name="nameSpaceElementName" value="MODULA_WMS" />
  <property name="useNameSpace" value="true" />
  <property name="omitDeclaration" value="no" />
  <property name="headerName" value="IMP_ORDINI" />
  <property name="headers">
    <map>
      <entry key="ORD_ORDINE" value="ID" />
      <entry key="ORD_DES" value="TaskType" />
    </map>
  </property>
  <property name="headerConstants">
    <map>
      <entry key="ORD_TIPOOP" value="V" />
    </map>
  </property>
  <property name="detailName" value="IMP_ORDINI_RIGHE" />
  <property name="details">
    <map>
      <entry key="RIG_ORDINE" value="ID" />
      <entry key="RIG_ARTICOLO" value="Sku" />
      <entry key="RIG_QTAR" value="UOMQty" />
```

```
      </map>
    </property>
  <property name="detailConstants">
  <map>
          <entry key="UOM" value="EA"/>
       </map>
  </property>
  <property name="headerFromProperties">
  <map>
          <entry key="TenantID" value="tenant"/>
       </map>
  </property>
  <property name="detailFromProperties">
  <map>
          <entry key="Schema" value="schema"/>
       </map>
  </property>

  </bean>
```

Properties

nameSpaceElementName

- Optional
- Description – Specifies the name of the namespace element
- Type – String
- Default – none
- Valid Values – any string
- Supports replacement – no

nameSpaceURI

- Optional

- Description – Specifies the URI for the namespace element
- Type – String
- Default – none
- Valid Values – URI
- Supports replacement – no

nameSpacePrefix

- Optional
- Description – Specifies the prefix for the namespace element
- Type – String
- Default – none
- Valid Values – any string literal
- Supports replacement – no

useNameSpace

- Optional
- Description – Specifies whether to use any namespace element in the XML
- Type – boolean
- Default – true
- Valid Values – true/false
- Supports replacement – no

omitDeclaration

- Optional
- Description – Specifies whether to have xml version declaration on top of xml string
- Type – String
- Default – no
- Valid Values – yes/no
- Supports replacement – no

headerName

- Required
- Description – Specifies the header element name of the header records in the XML string
- Type – String
- Default – none
- Valid Values – Any valid header name in String format
- Supports replacement – no

detailName

- Optional
- Description – Specifies the detail element name of the detail records in the XML string
- Type – String
- Default – details
- Valid Values – Any valid detail name in String format
- Supports replacement – no

headers

- Required
- Description – a hashmap containing the header elements of an XML string
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

details

- Required
- Description – a hashmap containing the detail elements of an XML string
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

headerConstants

- Optional
- Description – a hashmap containing the header constant elements of an XML string
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

detailConstants

- Optional
- Description – a hashmap containing the detail constant elements of an XML string
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

headerFromProperties

- Optional
- Description – keys with pairing values, used to add values to header part of XML string from properties
- Type – map
- Default – none
- Valid Values – none
- Supports replacement – no

detailFromProperties

- Optional
- Description – keys with pairing values, used to add values to detail part of XML string from properties
- Type – map
- Default – none
- Valid Values – none
- Supports replacement – no

Chapter 31 MapMessageToJSONImpl

This component is used to map message elements with different names and converts it to JSON structure. It is used in File Writer Processor to write to JSON file.

Sample configuration

This is a sample configuration in the `datamaps.xml` file.

```
<bean id="ASNCREATEDJSONMAP"
class="com.infor.cswms.automationagent.content.MapMessageToJSONImpl">

<property name="headerName" value = "Receipt"/>
<property name="headers">
<map>
<entry key="ReceiptKey" value="ReceiptKey" />
<entry key="ExternReceiptKey" value="ExternReceiptKey" />
      </map>
</property>
<property name="detailName" value = "ReceiptDetail"/>
<property name="details">
<map>
         <entry key="ReceiptLineNumber" value="ReceiptLineNumber"/>
         <entry key="ToID" value="ToID"/>
      </map>
</property>
<property name="detailConstants">
<map>
         <entry key="UOM" value="EA"/>
      </map>
</property>
<property name="detailFromProperties">
```

```
<map>
        <entry key="Schema" value="schema"/>
     </map>
</property>
<property name="headerFromProperties">
<map>
        <entry key="TenantID" value="tenant"/>
     </map>
</property>
<property name="headerConstants">
<map>
        <entry key="whseid" value="wmwhse1"/>
     </map>
</property>
</bean>
<
```

Properties

headerName

- Required
- Description – Specifies the header element name of the header records in the JSON string
- Type – String
- Default – none
- Valid Values – Any valid header name in String format
- Supports replacement – no

detailName

- Optional
- Description – Specifies the detail element name of the detail records in the JSON string
- Type – String
- Default – details

- Valid Values – Any valid detail name in String format
- Supports replacement – no

headers

- Required
- Description – a hashmap containing the header elements of a JSON string
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

details

- Required
- Description – a hashmap containing the detail elements of a JSON string
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

headerConstants

- Optional
- Description – a hashmap containing the header constant elements of a JSON string
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

detailConstants

- Optional
- Description – a hashmap containing the detail constant elements of a JSON string
- Type – map

- Default – none
- Valid Values – Hashmap
- Supports replacement – no

headerFromProperties

- Optional
- Description – keys with pairing values, used to add values to header part of JSON string from properties
- Type – map
- Default – none
- Valid Values – none
- Supports replacement – no

detailFromProperties

- Optional
- Description – keys with pairing values, used to add values to detail part of JSON string from properties
- Type – map
- Default – none
- Valid Values – none
- Supports replacement – no

Chapter 32 MapMessageToDelimitedImpl

This component is used to map message elements with different names and convert it to delimited structure. It is used in File Writer Processor to write delimited data to a text file.

Sample configuration

This is a sample configuration.

```
<bean id="maptoDelimited"
class="com.infor.cswms.automationagent.content.MapMessageToDelimitedImpl">
<property name="messageTypeinboundboundValueMap" ref="inboundValueMap" />
<property name="delimiter" value="," />
<property name="defaultValue" value="" />
</bean>

<util:map id="inboundValueMap" map-class="java.util.HashMap">
<entry key="header" value="serialkey,WaveKey,WaveType,EditWho" />
<entry key="detail" value="WaveDetailKey,OrderKey,Status" />
</util:map>
```

Properties

messageTypeinboundboundValueMap

- Required
- Description – reference to utility map id for mapping data
- Type – string

- Default – none
- Valid Values – any
- Supports replacement – no

delimiter

- Optional
- Description – delimited string value
- Type – string
- Default – “,”
- Valid Values – any
- Supports replacement – no

defaultValue

- Optional
- Description – Default value for the data fields which are not found in the message
- Type – string
- Default – none
- Valid Values – any
- Supports replacement - no

Chapter 33 AlertProcessor

This component is used to publish specific alerts that are configured by the user.

Sample configuration

This is a sample configuration.

```
<bean id="alertProcessor" class="com.infor.cswms.automationagent.processor.AlertProcessor">
<property name="severity" value="WARNING"/>
<property name="priority" value="HIGH"/>
<property name="area" value="SYSTEM"/>
<property name="summary" value="testing alert processor"/>
<property name="content">
<list>
<value>Sample content</value>
</list>
</property>

</bean>
```

Properties

severity

- Optional
- Description – This property is used to publish only those alerts which match with this severity
- Type – String

- Default – OTHER
- Valid Values – FATAL, EXCEPTION, WARNING, SUCCESS, OTHER, ALL
- Supports replacement – no

priority

- Optional
- Description – This property is used to publish only those alerts which match with this priority
- Type – String
- Default – LOW
- Valid Values – URGENT, HIGH, MEDIUM, LOW, ALL
- Supports replacement – no

area

- Optional
- Description – This property is used to publish only those alerts which match with this area
- Type – String
- Default – OTHER
- Valid Values – SYSTEM, DATA, OTHER, ALL
- Supports replacement – no

summary

- Optional
- Description – Subject of the email
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement - yes

content

- Optional
- Description – Message body

- Type – list
- Default – none
- Valid Values – any of type
- Supports replacement - yes

Chapter 34 EmailAlert

This component is used to send alert emails.

Sample configuration

This is a sample configuration.

```
<bean id="EmailAlert" class="com.infor.cswms.automationagent.alert.EmailAlert">

<property name="recipients">
<list>
<ref bean="user1"/>
</list>
</property>
<property name="smtpHost" value="smtp-relay.infor.com"/>
<property name="smtpPort" value="25"/>

<property name="fromEmail" value="username@infor.com"/>
<property name="userName" value="username@infor.com"/>
<property name="password" value="GoAwayHacker"/>
</bean>

<bean id="user1" class="com.infor.cswms.automationagent.alert.Recipient">

<property name="toEmail" value="usrname@infor.com"/>

<property name="severities">
<util:list id="alertsstuff4" list-class="java.util.ArrayList">
    <value type="com.infor.cswms.automationagent.content.Enums.severity">EXCEPTION</value>
</util:list>
```

```
</property>
<property name="priorities">
<util:list id="alertstuff3" list-class="java.util.ArrayList">
    <value type="com.infor.cswms.automationagent.content.Enums.priority">URGENT</value>
</util:list>
</property>
<property name="areas">
<util:list id="alertAreas3" list-class="java.util.ArrayList">
    <value type="com.infor.cswms.automationagent.content.Enums.functionalArea">DATA</value>
</util:list>
</property>

</bean>
```

Properties- EmailAlert

recipients

- Required
- Description – A list of Recipients that are to be processed
- Type – Recipient
- Default – none
- Valid Values – any of type
- Supports replacement – no

smtpHost

- Required
- Description – Host name of smtp server
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

smtpPort

- Required
- Description – port number of smtp server
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

fromEmail

- Required
- Description – Email address of the sender
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

userName

- Optional
- Description – Username of the email
- Type – String
- Default – none
- Valid Values – any
- Supports replacement - no

password

- Optional
- Description – Password for the username
- Type – String
- Default – none
- Valid Values – any
- Supports replacement - no

Properties: Recipient

toEmail

- Required
- Description – Email address of the receiver
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement – yes

priorities

- Optional
- Description – This property is used to filter the alerts based on priority. So, that the email is sent only on specific events
- Type – list
- Default – none
- Valid Values – FATAL, EXCEPTION, WARNING, SUCCESS, OTHER, ALL
- Supports replacement – no

severities

- Optional
- Description – This property is used to filter the alerts based on severity. So, that the email is sent only on specific events
- Type – list
- Default – none
- Valid Values – URGENT, HIGH, MEDIUM, LOW, ALL
- Supports replacement – no

areas

- Optional
- Description – This property is used to filter the alerts based on areas. So, that the email is sent only on specific events
- Type – list

- Default – none
- Valid Values – SYSTEM, DATA, OTHER, ALL
- Supports replacement – no

Chapter 35 TransmitLogAlert

This component is used to create a transmit log event in Infor WMS.

Sample configuration

This is a sample configuration.

```
<bean id="transmitLogAlert" class="com.infor.cswms.automationagent.alert.TransmitLogAlert">
<property name="hostName" value="mylocalsce.infor.com"/>
<property name="port" value="8080"/>
<property name="contextRoot" value="/wmwebservice_rest/"/>
<property name="user" value="admin1"/>
<property name="password" value="Pmm_7386"/>
<property name="tenant" value="INFOR"/>
<property name="resource" value="${schema}/sproceduremaps/json/NSPRFADDEVENT"/>
<property name="verb" value="POST"/>
<property name="transmitlogKey1" value="AAKey"/>
<property name="transmitlogKey2" value="Summary"/>
<property name="eventName" value="CUSTOMALERT"/>
</bean>
```

Properties

hostName

- Required
- Description – hostname of the server

- Type – String
- Default – none
- Valid Values – hostname
- Supports replacement – no

port

- Required
- Description – The port used when making the actual REST call. This value is used in building the URL.
- Type – numeric
- Default – none
- Valid Values – Must be an open port listening for REST calls on the server specified
- Supports replacement – no

contextRoot

- Required
- Description – The prefix for all calls made from this client. This value is used in building the URL.
- Type – String
- Default – none
- Valid Values – Must point to a valid context root on the server. Can be omitted if each endpoint includes the context root as part of its URL.
- Supports replacement – no

user

- Required
- Description – Valid username to authenticate the REST call
- Type – String
- Default – none
- Valid Values – Valid username
- Supports replacement - no

password

- Required
- Description – Password of username
- Type – String
- Default – none
- Valid Values – any
- Supports replacement - no

tenant

- Required
- Description – Tenant name
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

resource

- Required
- Description – The URL end for the call to be made. This will be appended after the host, port, and context root to form the URL.
- Type – String
- Default – none
- Valid Values – A published URL endpoint on the server being called
- Supports replacement – yes

verb

- Required
- Description – The HTTP verb to use for the call
- Type – String
- Default – POST
- Valid Values – GET, POST, DELETE, PUT
- Supports replacement - no

transmitlogKey1

- Required
- Description – The name of the message property
- Type – String
- Default – none
- Valid Values – Any existing message property name
- Supports replacement – no

transmitlogKey2

- Optional
- Description – The name of the message property
- Type – String
- Default – none
- Valid Values – Any existing message property name
- Supports replacement - no

transmitlogKey3

- Optional
- Description – The name of the message property
- Type – String
- Default – none
- Valid Values – Any existing message property name
- Supports replacement - no

transmitlogKey4

- Optional
- Description – The name of the message property
- Type – String
- Default – none
- Valid Values – Any existing message property name
- Supports replacement - no

transmitlogKey5

- Optional
- Description – The name of the message property
- Type – String
- Default – none
- Valid Values – Any existing message property name
- Supports replacement - no

eventName

- Required
- Description –The event name that is to be created in WMS
- Type – String
- Default – AutomationAgentAlert
- Valid Values – any
- Supports replacement - no

Chapter 36 SplitByHeaderProcessor

This component splits the multiple headers from a message and creates a new thread to process each header separately.

Sample configuration

This is a sample configuration.

```
<bean id="splitByHeader" class="com.infor.cswms.automationagent.processor.SplitByHeaderProcessor"
scope="prototype">
<property name='processChainBeanName' value='cswmsExportContinueChain'/>
</bean>
```

Properties

processChainBeanName

- Optional
- Description – The name of the process chain bean that is to be processed. If this property is specified the process will start in a new chain or else the process will continue in the same chain.
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

Chapter 37 SplitByHeaderDetailProcessor

This component splits the multiple header-details from a message and creates a new thread to process each detail separately.

Sample configuration

This is a sample configuration.

```
<bean id="splitByHeader" class="com.infor.cswms.automationagent.processor.SplitByHeaderProcessor"
scope="prototype">
<property name='processChainBeanName' value='cswmsExportContinueChain'/>
</bean>
```

Properties

processChainBeanName

- Optional
- Description – The name of the process chain bean that is to be processed. If this property is specified the process will start in a new chain or else the process will continue in the same chain.
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

Chapter 38 Mapper

This component is used to map the json data that gets called by REST processor.

Sample configuration: Mapper

This is a sample configuration.

```
<bean id="ExportMapper" class="com.infor.cswms.automationagent.content.Mapper">
<property name="mapBeanName" value = "ExportMap"/>
<property name="inputType" value = "json"/>
<property name="pathToMap" value = "/tablename"/>
</bean>
```

Sample configuration: DocumentPathMap

This is a sample configuration in the `processors.xml` file.

```
<bean id="ExportMap" class="com.infor.cswms.automationagent.content.DocumentPathMap">
  <property name="headerName" value="header" />
  <property name="detailName" value="detail" />
  <property name="headerPath" value="/header" />
  <property name="detailPath" value="/detail" />
  <property name="headers">
    <map>
      <entry key="Serialkey" value="/Serialkey" />
      <entry key="TaskDetailKey" value="/TaskDetailKey" />
      <entry key="FromID" value="/FromID" />
      <entry key="FromLoc" value="/FromLoc" />
      <entry key="TaskType" value="/TaskType" />
      <entry key="StorerKey" value="/StorerKey" />
```

```
        <entry key="Sku" value="/Sku" />
        <entry key="Qty" value="/Qty" />
        <entry key="Status" value="/Status" />
      </map>
    </property>
    <property name="headerConstants">
      <map>
        <entry key="constant" value="2" />
      </map>
    </property>
    <property name="headerFromProperties">
      <map>
        <entry key="headerProperty" value="propertyOne" />
      </map>
    </property>
    <property name="details">
      <map>
        <entry key="PICKDETAILKEY" value="/PICKDETAILKEY" />
        <entry key="ORDERKEY" value="/ORDERKEY" />
        <entry key="QTY" value="/QTY" />
        <entry key="STORERKEY" value="/STORERKEY" />
        <entry key="SKU" value="/SKU" />
        <entry key="LOT" value="/LOT" />
        <entry key="ID" value="/ID" />
        <entry key="STATUS" value="/STATUS" />
      </map>
    </property>
    <property name="detailConstants">
      <map>
        <entry key="TWO" value="2" />
      </map>
    </property>
    <property name="detailFromProperties">
      <map>
        <entry key="propertyOne" value="something" />
      </map>
    </property>
  <property name="pathProperties">
```

```
    <map>
      <entry key="taskdetailkey" value="/taskdetailkey" />
    </map>
  </property>
  <property name="properties">
    <map>
      <entry key="messageType" value="SPSRATEREQUESTED"/>
      <entry key="endpoint" value="postSpsRequestEndpoint"/>
    </map>
  </property>
  </bean>
```

Properties: Mapper

mapBeanName

- Required
- Description – A reference to a DocumentPathMap which maps the properties from the JSON
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

pathProperties

- Optional
- Description – It is used to set message properties from the json message
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

inputType

- Required
- Description – Input type that is to be mapped
- Type – String
- Default – none
- Valid Values – “json”
- Supports replacement – no

pathToMap

- Optional
- Description – If this property is specified then the DocumentPathMap bean gets called dynamically.
- Type – String
- Default – none
- Valid Values – xpath
- Supports replacement – no

Properties: DocumentPathMap

headerName

- Optional
- Description – Name of the header
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

detailName

- Optional
- Description – Name of the detail

- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

headerPath

- Required
- Description – A valid xpath in the message pointing to a set of header elements
- Type – String
- Default – none
- Valid Values – xpath
- Supports replacement – no

detailPath

- Optional
- Description – A valid xpath in the message pointing to a set of detail elements
- Type – String
- Default – none
- Valid Values – xpath
- Supports replacement – no

headerSubPath

- Optional
- Description – This property is used to parse array of headers
- Type – String
- Default – none
- Valid Values – xpath
- Supports replacement – no

headers

- Required

- Description – A valid xpath to parse a particular key value.
- Type – map
- Default – none
- Valid Values – xpath
- Supports replacement – no

details

- Required - This property is only required when “detailPath” property is used
- Description – A valid xpath to parse a particular key value.
- Type – map
- Default – none
- Valid Values – xpath
- Supports replacement – no

headerConstants

- Optional
- Description – A map of key value pairs that are to be added to the header
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

detailConstants

- Optional
- Description – A map of key value pairs that are to be added to the detail
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

headerFromProperties

- Optional
- Description – Properties that are to be added to the header from message properties
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

detailFromProperties

- Optional
- Description – Properties that are to be added to the detail from message properties
- Type – map
- Default – none
- Valid Values – Hashmap
- Supports replacement – no

pathProperties

- Optional
- Description – Values that are to be added to the message properties from the Json
- Type – map
- Default – none
- Valid Values – xpath
- Supports replacement – no

properties

- Optional
- Description – A map of key-value pairs that will be added to the message context. This is available in all Processors through the framework
- Type – map
- Default – none
- Valid Values – any
- Supports replacement – no

Chapter 39 FileDeleteProcessor

This component is used to delete a file in a particular directory. It takes the file directory path fileURL property from the message and deletes the file.

Sample configuration

This is a sample configuration.

```
<bean id="FileDeleteProcessor"
class="com.infor.cswms.automationagent.processor.FileDeleteProcessor">
</bean>
```

Chapter 40 FileMoveProcessor

This component is used to move the file from one path to another. It takes the source file path fileURL property from the message and moves it to a specified filepath.

Sample configuration

This is a sample configuration.

```
<bean id="FilemoveProcessor" class="com.infor.cswms.automationagent.processor.FileMoveProcessor">
<property name="filePath" value = "C:\demo"/>
<property name="fileUseTimeStamp" value = "false"/>
</bean>
```

Properties

filePath

- Required
- Description – Directory to which the file is to be moved
- Type – path
- Default – none
- Valid Values – valid file path
- Supports replacement – no

fileUseTimeStamp

- Optional
- Description – If this property is set to “true” timestamp will be prepended to the filename
- Type – Boolean
- Default – false
- Valid Values – true or false
- Supports replacement – no

Chapter 41 FilterProcessor

This component uses instructions to manipulate the process chain based on data.

To read an instruction, specify **`if {sourceName} in {sourceType} {sourceCondition} {sourceValue} then {changeReference}`**.

Sample configuration

This is a sample configuration.

For example, it FromPAZone in the header equals SFP1 or CFP1 or PFP1 or VLM1 then a new chain is started for ExternalChain. This makes greater or less than, contains or notContains , startsWith , notStartsWith , endsWith or notEndsWith comparisons more natural.

```
<bean id="FilterProcessor" class="com.infor.cswms.automationagent.processor.FilterProcessor">
<property name="inputPart" value="0"/>
<property name="instructions">
<list>
<ref bean="FilterIns1"/>
<ref bean="FilterIns2"/>
</list>
</property>
</bean>

<bean id="FilterIns1" class="com.infor.cswms.automationagent.content.FilterInstruction">
<property name="chainName" value="ExternalChain"/>
<property name="sourceType" value="header"/> <!-- header, property -->
<property name="sourceName" value="FromPAZone"/>
<property name="sourceCondition" value="equals"/> <!-- equals, notEquals, contains, notContains,
startsWith, notStartsWith, endsWith, notEndsWith -->
<property name="sourceValues">
<list>
```

```
<value>SFP1</value>
<value>CFP1</value>
<value>PFP1</value>
<value>VLM1</value>

</list>
</property>

</bean>

<bean id="FilterIns2" class="com.infor.cswms.automationagent.content.FilterInstruction">
<property name="chainName" value="ContinueChain"/>
<property name="sourceType" value="header"/> <!-- header, property -->
<property name="sourceName" value="FromPAZone"/>
<property name="sourceCondition" value="notEquals"/> <!-- equals, notEquals, contains, notContains,
startsWith, notStartsWith, endsWith, notEndsWith -->
<property name="sourceValues">
<list>
<value>SFP1</value>
</list>
</property>

</bean>
```

Properties: FilterProcessor

instructions

- Required
- Description – A list of FilterInstruction to be processed
- Type – list of reference to type FilterInstruction
- Default – none
- Valid Values – any of type
- Supports replacement - no

Properties: FilterInstruction

chainName

- Required
- Description – The bean name for the chain that is to be called if the sourceCondition condition is satisfied
- Type – String
- Default – none
- Valid Values – The name of a process chain
- Supports replacement - no

sourceType

- Required
- Description – Where should the value to be tested come from
- Type – String
- Default – none
- Valid Values – header, properties
- Supports replacement - no

sourceName

- Required
- Description – The name of the header or property to evaluate
- Type – String
- Default – none
- Valid Values – Any existing header or property name
- Supports replacement - no

sourceCondition

- Required

- Description – Type of evaluation to make from the sourceName against the sourceValue with sourceName on the left, and sourceValue on the right. See the description above for a sample. *NOTE, these values are case sensitive.
- Type – String
- Default – none
- Valid Values – equals, notEquals, contains, notContains, startsWith, notStartsWith, endsWith, notEndsWith, lessThan, lessThanOrEqual, greaterThan, greaterThanOrEqual
- Supports replacement – no

sourceValues

- Required
- Description – The list of values against which source condition is evaluated
- Type – list
- Default – none
- Valid Values – Any of type
- Supports replacement - no

Chapter 42 SplitByRuleHeaderProcessor

This component splits the multiple headers from a message based on the rule and creates a new thread to process them separately.

Sample configuration

This is a sample configuration.

```
<bean id="SplitByRuleHeaderDetailProcessor" class="com.infor.cswms.automationagent.processor.
SplitByRuleHeaderProcessor">
<property name="splittingRule" value="dataelement"/><!--nrecords,dataelement-->
<property name="dataName" value="GroupID"/>
<property name="inputPart" value="0"/>
<property name="processChainBeanName" value="NewFilterChain"/>
</bean>

<bean id="SplitByRulenrecordsProcessor" class="com.infor.cswms.automationagent.processor.
SplitByRuleHeaderProcessor">
<property name="splittingRule" value="nrecords"/><!--nrecords,dataelement-->
<property name="recordsToSplit" value="5"/>
<property name="inputPart" value="0"/>
<property name="processChainBeanName" value="NewFilterChain"/>
</bean>
```

Properties

splittingRule

- Required
- Description – The condition on which splitting is done
- Type – String
- Default – none
- Valid Values – nrecords, dataelement
- Supports replacement – no

dataName

- This property is required only when splittingRule is “dataelement”
- Description – The name of the header element
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement – no

recordsToSplit

- This property is required only when splittingRule is “nrecords”
- Description – The number of records that are to be combined and processed together
- Type – Integer
- Default – “0”
- Valid Values – any integer value
- Supports replacement – no

processChainBeanName

- Optional
- Description – The name of the processchain bean that is to be processed, If this property is specified process will start in a new chain else process will continue in the same chain.
- Type – String

- Default – none
- Valid Values – any
- Supports replacement – no

Chapter 43 SplitByRuleHeaderDetailProcessor

This component splits the multiple header-details from a message based on the rule and creates a new thread to process them separately.

Sample configuration

This is a sample configuration.

```
<bean id="SplitByRuleHeaderDetailProcessor"
class="com.infor.cswms.automationagent.processor.SplitByRuleHeaderDetailProcessor">
<property name="splittingRule" value="dataelement"/><!--nrecords,dataelement-->
<property name="dataName" value="GroupID"/>
<property name="inputPart" value="0"/>
<property name="processChainBeanName" value="NewFilterChain"/>
</bean>

<bean id="SplitByRulenrecordsProcessor"
class="com.infor.cswms.automationagent.processor.SplitByRuleHeaderDetailProcessor">
<property name="splittingRule" value="nrecords"/><!--nrecords,dataelement-->
<property name="recordsToSplit" value="5"/>
<property name="inputPart" value="0"/>
<property name="processChainBeanName" value="NewFilterChain"/>
</bean>
```

Properties

splittingRule

- Required
- Description – The condition on which splitting is done
- Type – String
- Default – none
- Valid Values – nrecords, dataelement
- Supports replacement – no

dataName

- This property is required only when splittingRule is “dataelement”
- Description – The name of the header element
- Type – String
- Default – none
- Valid Values – any of type
- Supports replacement – no

recordsToSplit

- Required - This property is required only when **splittingRule** is **`nrecords`**
- Description – The number of records that are to be combined and processed together
- Type – Integer
- Default – “0”
- Valid Values – any integer value
- Supports replacement – no

processChainBeanName

- Optional
- Description – The name of the process chain bean that is to be processed, If this property is specified process will start in a new chain else process will continue in the same chain.
- Type – String

- Default – none
- Valid Values – any
- Supports replacement – no

Chapter 44 JSONParserProcessor

This component takes JSON string read from file and maps it to string fields. It is internally referred to Automation Agent Mapper class using **mapName** property where all the mappings are handled in header and detail manner and gets added to the message stack.

Sample configuration

This is a sample configuration in processors.xml:

```
<bean id="jsonParserProcessor"
class="com.infor.cswms.automationagent.processor.JSONParserProcessor" scope="prototype">
<!-- look into datamaps.xml file for parsing json message -->
<property name="mapName" value="externalResponseMapper"/>
</bean>
```

This is a sample configuration in datamaps.xml:

```
<bean id="externalResponseMapper" class="com.infor.cswms.automationagent.content.Mapper">
<property name="mapBeanName" value = "externalResponseMap"/>
<property name="inputType" value = "json"/>
</bean>

<bean id="externalResponseMap" class="com.infor.cswms.automationagent.content.DocumentPathMap">

<property name="headerPath" value = "/ManualInduct"/>
<property name="headers">
<map>
        <entry key="HostOrderID" value="/HostOrderID"/>
<entry key="ContainerID" value="/ContainerID"/>
<entry key="Quantity" value="/Quantity"/>
    </map>
</property>
```

```
<property name="detailName" value = "ManualDetails"/>
<property name="detailPath" value = "/details"/>
<property name="details">
<map>
        <entry key="detailA" value="/detailA"/>
<entry key="detailB" value="/detailB"/>
      </map>
</property>
</bean>
```

Properties

mapName

- Required
- Description – A reference to a Mapper bean from datamaps.xml where the mappings specified in header and detail manner for the JSON String properties.
- Type – String
- Default – none
- Valid Values – any
- Supports replacement – no

Chapter 45 ResetProcessor

This component is designed to take a list of Automation Agent components to reset based on reset type such as start/stop/reload. This is called as part of processors list in process chain and when this is executed it will reset the initiators and some processors that have start/stop/reload functionality.

Sample configuration

```
<bean id="resetProcessor" class="com.infor.cswms.automationagent.processor.ResetProcessor">
<property name="resetList" ref="socketResetList"/>
<property name="resetType" value="stop"/><!-- start, stop, reload -->
</bean>

<util:list id="socketResetList">
  <value>socketListener</value>
  <value>senderSocketClientProcessor</value>
</util:list>
```

Properties

resetList

- Required
- Description – This property refers to a list of components to reset to using list utility class.
- Type – List
- Default – none
- Valid Values – List containing bean id’s
- Supports replacement – no

resetType

- Required
- Description – This property specifies the reset type i.e. start/stop/reload.
- Type – String
- Default – none
- Valid Values – start/stop/reload
- Supports replacement – no

Appendix A Configuring WMS Automation Agent Cloud Edition for SPS Communication

Infor WMS Cloud Edition has an existing on-premises integration with BluJay's Parcel Application. BluJay Parcel was formerly known as Kewill Flagship. BluJay Parcel supports major parcel carrier systems, such as FedEx, UPS, and USPS, and provides shipment rate and tracking information. Through the integration, WMS sends data that pertains to package shipments scheduled to be shipped from the warehouse. BluJay Parcel sends back rate and tracking information for the packages on the shipment and the data is stored in the WMS database.

Infor WMS Cloud Edition provides a Small Parcel Service (SPS) API and application events that allow for universal integration with SPS applications. The solution provides the flexibility to connect Infor WMS Cloud Edition to SPS Cloud, and Infor WMS Cloud Edition to SPS on-premises. Infor WMS Cloud Edition requires WMS application setup to start using SPS. If you are familiar with using SPS actions in prior versions of WMS, you will find little difference in using WMS Cloud Edition when connected to BluJay Parcel.

With the release of Automation Agent, a standard BluJay Parcel configuration is available for routing SPS Event message data from Infor WMS Cloud Edition to a BluJay Parcel application on-premises. This section provides the guidelines for configuring Automation Agent to pole Infor WMS MT SPS Events and transform the Message XML to data that an on-premises BluJay Parcel Application can consume.

This configuration is advanced and requires an administrator with the proper permissions to complete the tasks for configuration. In addition, the administrator should have the BluJay Parcel guides available to validate message requirements for BluJay Parcel.

For more information review the *Infor WMS Automation Agent Administration Guide.*

Note: This configuration defines the basic integration required. You can extend any of these features but that is not addressed here.

Prerequisites

These are prerequisites for configuring Infor WMS Cloud Edition and SPS communication:

- Infor WMS Cloud Edition tenant access
- Infor ION API Authorized App created for Automation Agent

- Downloaded credentials from Infor ION API Authorized App
- Automation Agent installation
- SPS Samples files. You can find the sample files in the in the subdirectory of the Automation Agent installation.

Configuring Automation Agent initial setup

1 Install Infor WMS Automation Agent using the Automation Agent installer.

Note: These steps assume that only one node, **`Agent1`**, is installed.

2 Backup the files found in `\aa\AAPRD\agent1\conf` folder to a backup directory. This allows you to restore configuration back to a basic installation if needed.

Note: Do not delete the files in this directory as several files will be re-used.

3 Navigate to the installation training folder: `\aa\AAPRD\agent1\sample`

4 Open the SPS folder. Note the number of files and file names in the folder.

5 Copy the contents of the SPS folder and paste to `\aa\AAPRD\agent1\conf.` Replace the files in the `conf` folder.

These files represent a standard communication configuration with BluJay Parcel on premises.

Automation Agent in Infor ION API

Before you continue, ensure that Automation Agent has been configured as an Authorized App in Infor ION API. Make note of the service account that is used to download the credentials. You should download a set of credentials and place in a secure place for access during the configuration.

For more information, see Infor ION API documentation.

The keys in this file are sensitive and must be handled as sensitive data. Follow IT security requirements concerning this data. For easier reading of the credential file, make a copy of the credential file. Using a tool like NotePad++, copy and replace `,` to **`,/r`** . This will align the keys into a line format for easier reading.

Configuration Files:

These are the core configuration files for this setup.

- agent.xml

- alerts.xml
- datamaps.xml
- initiators.xml
- lookupvalidations.xml
- processchains.xml
- processors.xml

Two files require configuration:

- initiators.xml: configuration required
- processors.xml: configuration required

See Sample files on page 188.

Configuring initiators.xml

1 Open `initiators.xml` with an xml editing tool.

2 Locate `bean id="exportPollerSps".` This is referred to as bean id.

3 Within the bean id locate `property name="properties″`.

4 Within `property name="properties″` configure these parameters:

   a `entry key="restHostName"`

   Specify the value from the downloaded credentials file using the "`iu″` key. Remove the `https://` and format the value like this: `value="qac-ionapi.qac.awsdev.infor.com"`.

   b `entry key="port"`

   Specify the secure port "`443″`, typically used or open in the firewall for this use.

   c `entry key="tenant"`

   Specify the value taken from the downloaded credentials file using the "`ti″` key. This is the same tenant for which Infor ION API Authorized App was configured and credentials downloaded.

   d `entry key="schema"`

   Taken from Infor WMS Cloud Edition 'dbName'. To locate the dbName, navigate to **Enterprise > Administration > Facility Management > Facility Activation**. The dbName is in this format: **`WMSM3INTCUST1_TST_WMSM3INTCUST1_TST_WMS_PRD_3_wmwhse1`**

Configuring processors.xml

1 Open `processors.xml` with an xml editing tool.

2 Identify these bean ids in the file for familiarity before you begin the configuration:

- `bean id="spsSocketClient"`
- `bean id="spsPoller"`
- `bean id="oauth2_WMSM3INTCUST1_TST"`
- `bean id="exportInterfaceSps"`
- `bean id="postSpsResults"`
- `bean id="updateTransmitlog"`
- `bean id="exportUpdateStatus"`

3 Locate `bean id="spsSocketClient"`. This is the configuration for the SPS Server and application. Within `bean id="spsSocketClient"`, locate and configure these properties:

a `property name="hostName"`

Specify the computer name of the SPS Server used for the Socket Connection to the server.

b `property name="port"`

Specify the SPS Server Socket connection port.

c `property name="connectMessage"`

Complete the '`connectMessage`' content for the applicable SPS Application.

d `property name="disconnectMessage"`

Complete the '`disconnectMessage`' content for the applicable SPS Application.

4 Locate `bean id="spsPoller"`. This is the API configuration for WMS.

5 Within `bean id="spsPoller"`, locate and configure these properties:

a `property name="hostName"`

Specify the value from the downloaded credentials file using the `"iu"` key. Remove the `https://` and format the value like this: `value="qac-ionapi.qac.awsdev.infor.com"`.

b `property name="port"`

Value is the secure port "`443`", typically used or open in the firewall for this use.

c `property name="contextRoot"`

Specify `/Tenant` **+** `/WM/wmwebservice_rest/`. It will appear like this when strung together correctly: `/WMSM3INTCUST1_TST/WM/wmwebservice_rest/`

d `property name="headers"`

These headers are used in the API call to WMS via ION API Gateway.

e `entry key="x-infor-identity2"`

Specify the **userID** of the service account used to create the downloaded credentials, for example, **`Bob Smith`**, **`AA Admin`**, or **`API User.`**

f `entry key="x-infor-tenantID"`

You can substitute (`${tenant}`) here to use the **tenant** identified elsewhere. Optionally, you can specify the actual tenant ID.

g `property name="oauth"`

The value is displayed in this format: `bean="oauth2_WMSM3INTCUST1_TST"`

Note: This is the same value that was used as the bean id in the oath section where `bean id="oauth2_WMSM3INTCUST1_TST`.

6 Locate `bean id="oauth2_WMSM3INTCUST1_TST`. This is the configuration regarding the secure connection to the ION API Gateway using the credentials previously downloaded.

7 Within `bean id="oauth2_WMSM3INTCUST1_TST"`, locate and configure the following properties:

a `property name="authorizationEndpoint"` – from the downloaded credentials files this is `pu + ot` and will appear similar to `https://qac-sso.qac.awsdev.infor.com:443/SCEM3INTCUST1_TST/as/token.oauth2`

b `property name="accountName"` – from the downloaded credentials files this is `saak`

c `property name="accountPassword"` – from the downloaded credentials files this is `sask`

d `property name="clientId"` – from the downloaded credentials files this is `ci`

e `property name="clientSecret"` - from the downloaded credentials files this is `cs`

8 Locate `bean id="exportInterfaceSps"`. This is the configuration for Get API call to WMS Cloud Edition.

9 Within `bean id="exportInterfaceSps"`, locate and configure these properties:

a `property name="parameters"`

Configure these properties:

- `entry key="type"`

  Specify the pipe delimited WMS event names for SPS events.
- `entry key="transmitflagtouse"`

  Specify the numbered WMS transmitflag of choice. Do not use ‘transmitflag’ because that column is used for BOD integrations.

10 Locate `bean id="postSpsResults"`. Specify the configuration regarding the secure connection to the ION API Gateway using the credentials previously downloaded to POST the SPS response.

11 Within `bean id="postSpsResults"`, locate and configure these properties:

a `property name="hostName"`

Specify the value taken from the downloaded credentials file using the `"iu"` key. Remove the `https://` and use this format: **`value="qac-ionapi.qac.awsdev.infor.com".`**

b `property name="port"`

Specify the secure port “`443`”. This value is used or open in the firewall for this use.

c `property name="contextRoot"`

Specify the value `/Tenant` + `/WM/wmwebservice_rest/`. The value will display in this format when strung together: `/WMSM3INTCUST1_TST/WM/wmwebservice_rest/`

d `property name="headers"`

Specify these headers that are used in the API call to WMS via ION API Gateway

e `entry key="x-infor-identity2"`

Specify the userID of the service account used to create the downloaded credentials, for example, `Bob Smith`, `AA Admin`, or `API User`.

f `entry key="x-infor-tenantID"`

You can substitute (`${tenant}`) here to use the 'tenant' identified elsewhere. You can also specify the tenant ID.

g `property name="oauth"`

The value is displayed in this format: `bean="oauth2_WMSM3INTCUST1_TST"`. This is the same value that is used as the same 'bean id' used in the oath section where `bean id="oauth2_WMSM3INTCUST1_TST`.

12 Locate `bean id="updateTransmitlog"`. This is the configuration for the secure connection to the ION API Gateway using the credentials previously downloaded to update the transmitlog.

13 Within `bean id="updateTransmitlog"`, locate and configure these properties:

a `property name="hostName"`

Specify the value taken from the downloaded credentials file using the `"iu"` key. Remove the `https://` and use this format: `value="qac-ionapi.qac.awsdev.infor.com"`.

b `property name="port"`

Specify the secure port "`443`". This value is typically used or open in the firewall for this use.

c `property name="contextRoot"`

Specify the value `/Tenant` + `/WM/wmwebservice_rest/`. The value is displayed in this format when strung together: `/WMSM3INTCUST1_TST/WM/wmwebservice_rest/`

d `property name="headers"`

These headers are used in the API call to WMS via ION API Gateway

e `entry key="x-infor-identity2"`

Specify the userID of the service account used to create the downloaded credentials, for example, `Bob Smith`, `AA Admin`, or `API User`.

f `entry key="x-infor-tenantID"`

You can substitute (`${tenant}`) here to use the 'tenant' identified elsewhere. You can also specify the tenant.

g `property name="oauth"`

The value is displayed in this format: `bean="oauth2_WMSM3INTCUST1_TST".` This is the same value used as the 'bean id' in the oath section where `bean id="oauth2_WMSM3INTCUST1_TST`.

14 Locate `bean id="exportUpdateStatus"`. This is the configuration for updating the transmitflag when the SPS response is posted successfully.

15 Within `bean id="exportUpdateStatus"`, locate and configure these properties:

a `property name="parameters"` – Configure this property:

- `entry key="transmitflagtouse"`

  Specify the numbered WMS `transmitflag` of your choice. Do not use 'transmitflag' as that column is used for BOD integrations.

Confirming the configuration

Review these steps to confirm your configuration of Infor WMS Cloud Edition and Automation Agent.

Note When using Infor WMS Cloud Edition with Automation Agent and APIs, this setup is required:

1 At the Warehouse Level, configure these SPS Events for use with the WMS Cloud Edition APIs. These events are available to be configured at the warehouse level. Select the **Message Enabled** check box for the events used.

Caution: Do not select **Publish BODs** unless instructed to do so by the integration team

- SPSRATEREQUESTED
- SPSVOIDREQUESTED
- SPSDELETEINTREQUESTED
- SPSEXTRACTINTREQUESTED

2 Navigate to **WMS > Administration > System Settings**. At the Warehouse Level, select the **SPS_INSTALLED** checkbox parameter to turn on that feature.

3 Configure a report printer in WMS.

4 Configure a label printer with SPS settings in WMS.

5 Configure **Owner Processing** tab for Small Parcel setup in WMS. Include a valid address and a contact phone number.

6 Create small parcel carriers by selecting **Small Parcel System Carrier** and configuring **Carrier SPS Parameters** tab in WMS.

7 Configure **Facility** with a valid address and a contact phone number.

For more detailed information on SPS features in WMS, see *Infor WMS Small Parcel Shipping User Guide.*

Sample files

These are samples of the core configuration files.

agent.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-
3.0.xsd http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-
2.0.xsd" xmlns:util="http://www.springframework.org/schema/util"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans"><import resource="initiators.xml"/><import
resource="processchains.xml"/><import resource="processors.xml"/><import
resource="datamaps.xml"/><import resource="lookupvalidations.xml"/><import
resource="alerts.xml"/><bean primary="true"
class="com.infor.cswms.automationagent.controller.AgentController" id="controller"><property
name="initiators"><list><ref bean="exportPollerSps"/>
<!-- Polls ExportInterface for SPS events -->
</list></property></bean></beans>
```

alerts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-
3.0.xsd http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-
2.0.xsd" xmlns:util="http://www.springframework.org/schema/util"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans"><import resource="initiators.xml"/><import
resource="processchains.xml"/><import resource="processors.xml"/><import
resource="datamaps.xml"/><import resource="lookupvalidations.xml"/><import
```

```
resource="alerts.xml"/><bean primary="true"
class="com.infor.cswms.automationagent.controller.AgentController" id="controller"><property
name="initiators"><list><ref bean="exportPollerSps"/>
<!-- Polls ExportInterface for SPS events -->
</list></property></bean></beans>
```

datamaps.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-
3.0.xsd http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans"><bean
class="com.infor.cswms.automationagent.content.Mapper" id="spsExportSplitter"><property value=""
name="mapBeanName"/><property value="json" name="inputType"/><property value="/tablename"
name="pathToMap"/></bean><bean
class="com.infor.cswms.automationagent.content.DocumentPathMap"
id="SPSRATEREQUESTED"><property name="properties"><map><entry value="SPSRATEREQUESTED"
key="messageType"/><entry value="postSpsRequestEndpoint" key="endpoint"/></map></property>
<!-- XPath properties - set a property from the document. Must evaluate to a single entry -->
<property name="pathProperties"><map>
<!-- <entry key="SPSCaseKey" value="/SPSRateRequested/SPSCaseKey"/> -->
</map></property><property value="/jsonMessage" name="headerPath"/><property
name="headers"><map><entry value="/SPSRateRequested/SPSRateRequested/SPSMessage"
key="SPSMessage"/><entry value="/SPSRateRequested/SPSRateRequested/SPSCaseKey"
key="SPSCaseKey"/><entry value="/SPSRateRequested/SerialKey"
key="SerialKey"/></map></property><property name="headerConstants"><map>
</map></property></bean><bean
class="com.infor.cswms.automationagent.content.DocumentPathMap"
id="SPSVOIDREQUESTED"><property name="properties"><map><entry value="SPSVOIDREQUESTED"
key="messageType"/><entry value="postSpsVoidEndpoint" key="endpoint"/></map></property>
<!-- XPath properties - set a property from the document. Must evaluate to a single entry -->
<property name="pathProperties"><map>
<!-- <entry key="SPSCaseKey" value="/SPSVoidRequested/SPSCaseKey"/> -->
</map></property><property value="/jsonMessage" name="headerPath"/><property
name="headers"><map><entry value="/SPSVoidRequested/SPSVoidRequested/SPSMessage"
key="SPSMessage"/><entry value="/SPSVoidRequested/SPSVoidRequested/SPSCaseKey"
key="SPSCaseKey"/><entry value="/SPSVoidRequested/SerialKey"
```

key="SerialKey"/></map></property><property name="headerConstants"><map>
</map></property></bean></beans>

initiators.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-2.0.xsd" xmlns:util="http://www.springframework.org/schema/util"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans">

<!-- Bean to poll for sps messages -->

<bean class="com.infor.cswms.automationagent.initiator.PollerInitiator" id="exportPollerSps"><property value="10" name="pollInterval"/><property value="Initiator Starting SPS Kewill Process Chain" name="logFriendlyName"/><property value="false" name="instrumentationOn"/><property name="content"><ref bean="exportPollerContentList"/></property><property name="properties"><map><entry value="qac-ionapi.qac.awsdev.infor.com" key="restHostName"/><entry value="443" key="port"/><entry value="WMSM3INTCUST1_TST" key="tenant"/><entry value="exportInterfaceSps" key="endpoint"/><entry value="WMS3INTCUST1_TST_WMSM3INTCUST1_TST_WMS_PRD_3_wmwhse1" key="schema"/></map></property><property value="cswmsExportSpsChain" name="processChain"/></bean>

<!-- Content ArrayList<HashMap> for poller -->

<util:list id="exportPollerContentList" list-class="java.util.ArrayList"><ref bean="exportPollerContent"/></util:list><util:map id="exportPollerContent" map-class="java.util.HashMap"><entry value="0" key="updatestatus"/></util:map></beans>

lookupvalidations.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-

```
2.0.xsd" xmlns:util="http://www.springframework.org/schema/util"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans"><util:list list-class="java.util.ArrayList"
id="kewillPropertyLookups"><ref bean="kewillSPSKeyPropertyLookup"/><ref
bean="kewillSerialKeyPropertyLookup"/></util:list><bean id="kewillSPSKeyPropertyLookup"
class="com.infor.cswms.automationagent.content.Lookup"><property value="SPSCaseKey"
name="property"/><property value="SPSCaseKey" name="sourceName"/><property value="header"
name="sourceType"/><property value="none" name="defaultValue"/></bean><bean
id="kewillSerialKeyPropertyLookup"
class="com.infor.cswms.automationagent.content.Lookup"><property value="SerialKey"
name="property"/><property value="SerialKey" name="sourceName"/><property value="header"
name="sourceType"/><property value="none" name="defaultValue"/></bean></beans>
```

processchains.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-
3.0.xsd http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans">

<!-- Use scope="prototype" to get new instance on each call, otherwise you get the same shared instance
every time -->

<bean scope="prototype" class="com.infor.cswms.automationagent.processor.ProcessChain"
id="cswmsExportSpsChain"><property name="processors"><list><ref bean="spsPoller"/><ref
bean="splitHeaders"/><ref bean="kewillLookupProcessor"/><ref bean="spsSocketClient"/><ref
bean="postSpsResults"/><ref bean="updateTransmitlog"/></list></property></bean></beans>
```

processors.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-
3.0.xsd http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-
```

```
2.0.xsd" xmlns:util="http://www.springframework.org/schema/util"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans"><bean
class="com.infor.cswms.automationagent.processor.ConsoleLoggingProcessor" id="loggingProcessor">
</bean><bean class="com.infor.cswms.automationagent.processor.SplitByHeaderProcessor"
id="splitHeaders"><property value="Splitting multiple messages from poll result"
name="logFriendlyName"/></bean><bean
class="com.infor.cswms.automationagent.processor.SocketClientProcessor"
id="spsSocketClient"><property value="Calling the Kewill Socket" name="logFriendlyName"/><property
value="spsserver.infor.com" name="hostName"/><property value="1200" name="port"/><property
name="messageTypeInboundValueMap" ref="inboundValueMap"/>
<!-- Fixed, Delimited, or Tab only fixed implemented for now -->
<property value="Delmited" name="messageFormat"/>
<!-- delmiter to be used when message format is delmited -->
<property value="|" name="delimiter"/>
<!-- Is message communcation via strings or bytes -->
<property value="Y" name="isBinary"/>
<!-- Message size to prepend to beggining of message -->
<property value="8" name="messageSizeBytes"/>
<!-- Message to send to socket before main message -->
<property value="CONNECT|CLIENTTYPE|10|CLIENTID|admin1|LANGUAGE|en_us|AUDIT_USER|admin1|"
name="connectMessage"/>
<!-- Message to send to socket after main message -->
<property
value="DISCONNECT|CLIENTTYPE|10|CLIENTID|admin1|LANGUAGE|en_us|AUDIT_USER|admin1|"
name="disconnectMessage"/>
<!-- prepend message size to main message -->
<property value="N" name="prePendMessageSizetoMessage"/>
<!-- prepend message size to pre connect message -->
<property value="Y" name="prePendMessageSizetoConnectMessage"/>
<!-- prepend message size to main message -->
<property value="Y" name="prePendMessageSizetoDisconnectMessage"/>
<!-- if this string is found in response no error occured -->
<property value="ERRCODE|000000000|" name="successString"/>
<!-- if response was not successfull an error enpoint will be set -->
<property value="postSpsErrorEndpoint" name="errorEndpoint"/></bean><util:map
id="inboundValueMap" map-class="java.util.HashMap"><entry
value="Sender,schema,MessageLength,Sequence,TransmissionIndicator,resource,fromloc,toloc,Indicator,f
romid,Status,Device,End-KZ,PalletType,Contour,Height,Reserve,MessageTrailerType,Use"
key="01"/><entry value="SPSMessage" key="SPSRATEREQUESTED"/><entry value="SPSMessage"
```

key="SPSVOIDREQUESTED"/></util:map><bean
class="com.infor.cswms.automationagent.processor.RestProcessor" id="spsPoller"><property
value="Polling wms for SPS records" name="logFriendlyName"/><property value="qac-
ionapi.qac.awsdev.infor.com" name="hostName"/><property value="443" name="port"/>

<!-- <property name="contextRoot" value="/wmwebservice_rest/"/> -->

<property value="/WMSM3INTCUST1_TST/WM/wmwebservice_rest/" name="contextRoot"/><property
name="headers"><map><entry value="John Smith" key="x-infor-identity2"/><entry value="${tenant}"
key="x-infor-tenantID"/><entry value="application/json" key="Content-
Type"/></map></property><property name="oauth">

<!-- <ref bean="oauth1_usspvwscmwm1" /> -->

<ref bean="oauth2_WMSM3INTCUST1_TST"/></property><property value="exportInterfaceSps"
name="endpointName"/></bean><bean
class="com.infor.cswms.automationagent.util.OAuthClientConfig"
id="oauth2_WMSM3INTCUST1_TST"><property value="2.0" name="oauthVersion"/><property
value="https://qac-sso.qac.awsdev.infor.com:443/WMSM3INTCUST1_TST/as/token.oauth2"
name="authorizationEndpoint"/><property value="SAAK_KEY____fJGo7123BeNT4b_123Up-
zcPfXHU8123bwvmxNDRH7eJy-OqnsfXRcY7RvjNVFR123" name="accountName"/><property
value="SASK_KEY____T123K9KiXw4bL19Bk1fCa123gxGhTkuTgGswK038fyWwpb0H123"
name="accountPassword"/><property value="CI_KEY____123FGRh8KZNmOEiu1234zQ8Ap1xsXoLloe123"
name="clientId"/><property value="CS_KEY____123Bsclx7JkH-tx123-HmTav33oIGZ1tmi0C123"
name="clientSecret"/></bean><bean class="com.infor.cswms.automationagent.util.RestEndpoint"
id="exportInterfaceSps"><property value="json" name="inputType"/><property value="GET"
name="verb"/><property value="${schema}/exports" name="resource"/>

<!-- <property name="sendMap" value="..."/> -->

<property value="spsExportSplitter" name="responseMapName"/><property
name="parameters"><map><entry value="SPSRATEREQUESTED|SPSVOIDREQUESTED"
key="type"/><entry value="5" key="updatestatus"/><entry value="0"
key="pollersearchstatus"/><entry value="TRANSMITFLAG3" key="transmitflagtouse"/><entry value="M"
key="eventcategory"/><entry value="true" key="asjson"/></map></property></bean><bean
class="com.infor.cswms.automationagent.processor.RestProcessor" id="postSpsResults"><property
value="Posting results of kewill call to WM" name="logFriendlyName"/><property value="qac-
ionapi.qac.awsdev.infor.com" name="hostName"/><property value="443" name="port"/><property
value="/WMSM3INTCUST1_TST/WM/wmwebservice_rest/" name="contextRoot"/><property
name="headers"><map><entry value="John Smith" key="x-infor-identity2"/><entry value="${tenant}"
key="x-infor-tenantID"/><entry value="application/json" key="Content-
Type"/></map></property><property name="oauth">

<!-- <ref bean="oauth1_usspvwscmwm1" /> -->

<ref bean="oauth2_WMSM3INTCUST1_TST"/></property><property value="${endpoint}"
name="endpointName"/></bean><bean
class="com.infor.cswms.automationagent.processor.RestProcessor" id="updateTransmitlog"><property
value="Updating Transmitlog status" name="logFriendlyName"/><property value="qac-
ionapi.qac.awsdev.infor.com" name="hostName"/><property value="443" name="port"/><property
value="/WMSM3INTCUST1_TST/WM/wmwebservice_rest/" name="contextRoot"/><property
name="headers"><map><entry value="John Smith" key="x-infor-identity2"/><entry value="${tenant}"
key="x-infor-tenantID"/><entry value="application/json" key="Content-
Type"/></map></property><property name="oauth">

```
<!-- <ref bean="oauth1_usspvwscmwm1" /> -->
<ref bean="oauth2_WMSM3INTCUST1_TST"/></property><property value="exportUpdateStatus"
name="endpointName"/></bean><bean class="com.infor.cswms.automationagent.util.RestEndpoint"
id="postSpsRequestEndpoint"><property value="json" name="inputType"/><property value="POST"
name="verb"/><property value="1" name="inputSource"/><property value="RequestMap"
name="sendMapName"/><property value="${schema}/spscases/shipresults"
name="resource"/></bean><bean class="com.infor.cswms.automationagent.content.Mapper"
id="RequestMap"><property value="RequestMapSPS" name="mapBeanName"/><property value="json"
name="inputType"/></bean><bean
class="com.infor.cswms.automationagent.content.DocumentPathMap" id="RequestMapSPS"><property
value="" name="headerPath"/><property name="headers"><map><entry value="/TRACKNUM-1"
key="tracknumber"/><entry value="/FREIGHT-1" key="rate"/></map></property><property
name="headerConstants"><map> </map></property><property
name="headerFromProperties"><map><entry value="SPSCaseKey"
key="spscasekey"/></map></property></bean><bean
class="com.infor.cswms.automationagent.util.RestEndpoint" id="postSpsVoidEndpoint"><property
value="json" name="inputType"/><property value="POST" name="verb"/><property value="1"
name="inputSource"/><property value="VoidMap" name="sendMapName"/><property
value="${schema}/spscases/voidresults" name="resource"/></bean><bean
class="com.infor.cswms.automationagent.content.Mapper" id="VoidMap"><property value="VoidMapSPS"
name="mapBeanName"/><property value="json" name="inputType"/></bean><bean
class="com.infor.cswms.automationagent.content.DocumentPathMap" id="VoidMapSPS"><property
value="" name="headerPath"/><property name="headers"><map> </map></property><property
name="headerConstants"><map> </map></property><property
name="headerFromProperties"><map><entry value="SPSCaseKey"
key="spscasekey"/></map></property></bean><bean
class="com.infor.cswms.automationagent.util.RestEndpoint" id="postSpsErrorEndpoint"><property
value="json" name="inputType"/><property value="POST" name="verb"/><property value="1"
name="inputSource"/><property value="ErrorMap" name="sendMapName"/><property
value="${schema}/spscases/shipresults" name="resource"/></bean><bean
class="com.infor.cswms.automationagent.content.Mapper" id="ErrorMap"><property
value="ErrorMapSPS" name="mapBeanName"/><property value="json"
name="inputType"/></bean><bean
class="com.infor.cswms.automationagent.content.DocumentPathMap" id="ErrorMapSPS"><property
value="" name="headerPath"/><property name="headers"><map><entry value="/ERRMSG"
key="error"/></map></property><property name="headerConstants"><map>
</map></property><property name="headerFromProperties"><map><entry value="SPSCaseKey"
key="spscasekey"/></map></property></bean><bean
class="com.infor.cswms.automationagent.processor.LookupValidationProcessor"
id="kewillLookupProcessor" scope="prototype"><property value="Mapping SPS case key and serialkey to
properties" name="logFriendlyName"/>
<!-- <property name='headerLookupName' value='accellLookups'/> -->
<property value="kewillPropertyLookups" name="propertyLookupName"/></bean>
<!-- Export update status endpoint -->
<bean class="com.infor.cswms.automationagent.util.RestEndpoint" id="exportUpdateStatus"><property
value="PUT" name="verb"/><property value="${schema}/exports/${SerialKey}" name="resource"/>
```

```
<!--<property name="sendMap" value="..."/>-->
<!-- <property name="responseMapName" value="restExportMap"/> -->
<property name="parameters"><map><entry value="9" key="status"/>
<!-- <entry key="status" value="${status}"/> -->
<entry value="TRANSMITFLAG3" key="transmitflagtouse"/></map></property></bean></beans>
```