

Files Checkers Rule Package & Custom Agents

Telefonica O2

Installation and configuration document

Version 2.0

Status: Issued

Document History

Version	Date	Description	Written by	Verified by
1.0	01/07/2015	First write	G. Viola (Axway PSO)	
2.0	30/05/2016	Improved description of CSVFileChecker parameters	G. Viola (Axway PSO)	

Summary

1. Introduction	4
2. System compliance	5
2.1 ST System Library	5
2.2 Additional Libraries	6
3. Configure the Rule Package	7
3.1 XmlFilesChecker.xml rules package	7
3.2 CsvFilesChecker.xml rules package	11
3.2 DatFilesChecker.xml rules package	16
4. Custom Agent and Rule Package Installation	21
4.1 Install Custom Agent	21
4.2 Install Rule Package	23
Appendix A – CheckedFiles Tracked Object	26

1. Introduction

Secure Transport (ST) is an event based file transfer server.

ST provides a web-based rule package editor in order to define and manage rules able to intercept specific events and then trigger some action.

At run time, Transaction Manager (TM) a component of the ST Server, is in charge to verify if the rule conditions have been met and then triggers actions.

Actions are implemented in two way:

- 1) External Agent (typically based on scripting language)
- 2) In-process Agent (created using java code)

This document describes:

- In-process Agent installation and setup procedures
 - Rule package configuration and installation
- regarding the custom "Files Checker" agent and rule package developed for Telefonica O2.

The goal of the custom agents is to perform a file format validation if compared to their own extension. Each custom agent is designed in order to manage one filename extension (i.e. .xml).

The general behavior of the Files Checker rules/agents is described in the follow:

- TM execute the Rule package in order to intercept file upload events, it performs a check on file extension and others parameters, and, if applicable, triggers the specific agent. Furthermore, rules provide some input parameters to agents.
- The agent performs a file format validation on uploaded file, if the outcome is Invalid File, the agent copy the uploaded file in a quarantine folder and delete it from original target folder. Furthermore, the Agent send a Sentinel Event via Universal Agent, to a Sentinel Tracked Object (described in appendix A) with the final outcome status of the file (Accepted | Refused)

2. System compliance

The In-process Agents are developed in JAVA using some specific ST Software Development Kit (SDK), API and external library.

In order to insure compliance with the following product release:

- SecureTransport Release: 5.2.1 SP4 with AV and CrossGeo Accelerators
- Sentinel release: 4.0.1
- Universal Agent release: 2.3.3 (as per the Sentinel 4.0.1 Installation Prerequisites Guide)

Following SDK bundle it was received from Axway Support (ticket n. 770566):

SecureTransport_5.2.1-SP2_SDK_allOS_BN1187

and then used to develop the custom agents;

Coherently the Java Development kit used to build the agent is **JDK 1.7.0_75**

All listed component are running on OS SLES 11 SP2 (Axway Appliance) where the following underlying libraries must be installed;

2.1 ST System Library

The following library should be already installed on ST directory tree:

Folder: ...**SecureTransport\lib\jars**

- o st-server-api.jar

Folder: ...**SecureTransport\lib\jars\external**

- o axway-common-circularbuffer-3.0.48.jar
- o axway-common-core-3.0.48.jar
- o axway-commons-libs
- o commons-beanutils-1.7.0.jar
- o commons-lang-2.3.jar
- o commons-logging-1.1.1.jar
- o log4j-1.2.16.jar
- o xercesImpl-2.9.0.jar
- o axway-sentinel-ua-3.0.48.jar

Note: these libraries are usually installed without release info embedded in the library name;
So, i.e. you must check that is already installed log4j.jar rather than log4j-1.2.16.jar

2.2 Additional Libraries

In addition, the following external libraries must be installed on Secure Transport folder tree:

Folder: ...\\SecureTransport\\lib\\jars\\external

- xmlParserAPIs-2.6.2.jar
- xml-apis-2.0.2.jar
- opencsv-3.3.jar

So, please copy these libraries in the expected folder.

3. Configure the Rule Package

Each rule package contains at least one rule.

For each rule it is mandatory at list a valid condition and one linked action to execute.

3.1 XmlFilesChecker.xml rules package

For each incoming event, Transaction manager is in charge to verify the *"XmlFilesChecker"* rule package condition and, if applicable, execute the related actions.

XmlFileChecker.xml rule package, contains the following rule:

Rule	check_xml_file
Precedence	9
Condition	EventType <equal> Incoming end AND DXAGENT_TARGET <match> *.xml
Action	id=1 streaming(none) In-process> com.axway.st.xmlfileschecker.XmlFilesCheckerAgent (debug="true", QFolder="/opt/test/Quarantena/", STNLTarget="VMST52", STNLPort="1305", STNLTimeout= "1000", UABuffer="/opt/test/UAoverflows/xmlfilechecker_buffer.dat", ovfSize="2", ovfSleep="3000")
Action	id=2 executeafter(1) streaming(none) In-process> com.tumbleweed.st.server.tm.agents.NextPrecedence (Impersonation="true")

This rule intercept the end of uploading file process (EventType <equal> Incoming end) for each file that match the following filename pattern : *.xml (DXAGENT_TARGET <match> *.xml).

The precedence rank of execution is setted to 9, so it means that it will be executed:

- after rules with precence <9
- before rule with precedence >9.

If file transfer match the condition the rule will execute 2 action in the following order:

- 1) Call the In-process custom agent: **com.axway.st.xmlfileschecker.XmlFilesCheckerAgent**

TM send a list of invocation parameters to the agent, e.g:

(debug="true", QFolder="/opt/test/Quarantena/", STNLTarget="VMST52", STNLPort="1305",
UABuffer="/opt/test/UAoverflows/xmlfilechecker_buffer.dat", ovfSize="2", ovfSleep="3000")

For more details about these parameters, please see in the follow.

- 2) Call the In-process agent: **com.tumbleweed.st.server.tm.agents.NextPrecedence**

TM send the following list of invocation parameters to the agent:

(Impersonation="true")

In-process agent: **com.axway.st.xmlfileschecker.XmlFilesCheckerAgent**

Invocation parameters details:

Parameter	Description	Admitted Value	Expected Behaviour
debug	Activate Custom Agent verbose log level	true	Custom Agent write full detailed log during execution (info, warn, error)
		false	Custom Agent write just significative msgs on ST log file (warn, error)
Qfolder	Quarantine folder complete path	Existing Folder	In case of invalid File, the Agent put in this folder the upload file
STNLTarget	Sentinel Acquisition server or Event Router host	Resolved hostname	Custom Agent send to this host the Sentinel events
STNLPort	Sentinel Acquisition server or Event Router Port	Integer	Custom Agent send to this port the Sentinel Event
UABuffer	Complete Pathname for Sentinel UA Events Buffer File	Existing Folder + BufferName	In case of unavailability of receiver service custom agent saves, on this buffer, the outgoing sentinel events.
STNLTimeout	Timeout for send Sentinel event attempt (in milliseconds)	Integer	This is the specific timeout for the task in charge to send sentinel events via UA invocation
ovfSize	UA Event Buffer size	Integer	Pre-allocated maximum size (in Mbyte) of the Sentinel UA Events Buffer File
ovfSleep	Time interval to empty the overflow file (in millisecond)	Integer	In case of unavailability of the receiver Sentinel service this is the loop time interval in which the Agent try to send again all the Sentinel Events present in the buffer file

If you need to modify the value of any of this parameters you can act in two way:

- Import the rule package via ST administration web based gui and then modify the imported package;
- Modify the .xml file of the package rule and then import the modified .xml version via ST administration web based gui

Note: after each import or modification you must perform a Transaction Manager service stop/start in order to get the updated rules running.

Please see in the follow the complete xml example of theXMLFileChecker rule package:


```

<?xml version="1.0" encoding="UTF-8"?>
<brpackage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0"
xsi:noNamespaceSchemaLocation="businessrules.xsd" name="xmlfileschecker.xml">
  <rule name="check_xml_file" type="static" precedence="9">
    <condition>
      <expression>
        <item>
          <attribute>EventType</attribute>
          <comparator name="equal" />
          <value>Incoming end</value>
        </item>
        <operator name="and" />
        <item>
          <attribute>DXAGENT_TARGET</attribute>
          <comparator name="match" />
          <value>*.xml</value>
        </item>
      </expression>
    </condition>
    <action Order="yes">
      <inprocess-agent id="1" executeafter="" streamaccess="none"
class="com.axway.st.xmlfileschecker.XmlFilesCheckerAgent">
        <params>
          <param>
            <param-name>debug</param-name>
            <param-value>"true"</param-value>
          </param>
          <param>
            <param-name>QFolder</param-name>
            <param-value>"/opt/test/Quarantena/"</param-value>
          </param>
          <param>
            <param-name>STNLTarget</param-name>
            <param-value>"VMST52"</param-value>
          </param>
          <param>
            <param-name>STNLPort</param-name>
            <param-value>"1305"</param-value>
          </param>
          <param>
            <param-name>STNLTimeout</param-name>
            <param-value>"1000"</param-value>
          </param>
          <param>
            <param-name>UABuffer</param-name>
            <param-value>"/opt/test/UAoverflows/xmlfilechecker_buffer.dat"</param-value>
          </param>
        </params>
      </inprocess-agent>
    </action>
  </rule>
</brpackage>

```

```
<param>
  <param-name>ovfSize</param-name>
  <param-value>"2"</param-value>
</param>
<param>
  <param-name>ovfSleep</param-name>
  <param-value>"3000"</param-value>
</param>
</params>
</inprocess-agent>
<inprocess-agent id="2" executeafter="1" streamaccess="none"
class="com.tumbleweed.st.server.tm.agents.NextPrecedence">
  <params>
    <param>
      <param-name>Impersonation</param-name>
      <param-value>"true"</param-value>
    </param>
  </params>
</inprocess-agent>
</action>
</rule>
</brpackage>
```

3.2 CsvFilesChecker.xml rules package

For each incoming event, Transaction manager is in charge to verify the “CsvFilesChecker” rule package condition and, if applicable, execute the related actions.

CsvFileChecker.xml rule package, contains the following rule:

Rule	check_csv_file
Precedence	9
Condition	EventType <equal> Incoming end AND DXAGENT_TARGET <match> *.csv AND DXAGENT_TARGET <contain> SmartM2M
Action	id=1 streaming(none) In-process> com.axway.st.csvfileschecker.CsvFilesCheckerAgent (debug="true", QFolder="/opt/test/Quarantena/", STNLTarget="VMST52", STNLPort="1305", STNLTimeout= "1000", UABuffer="/opt/test/UAoverflows/csvfilechecker_buffer.dat", ovfSize="2", ovfSleep="3000"), Separator=";",ElementNumber="12"
Action	id=2 executeafter(1) streaming(none) In-process> com.tumbleweed.st.server.tm.agents.NextPrecedence (Impersonation="true")

This rule intercept the end of uploading file process (EventType <equal> Incoming end) for each file that match the following filename pattern : *.csv (DXAGENT_TARGET <match> *.csv).

Furthermore the filename must contain the string “SmartM2M” (DXAGENT_TARGET <contain> SmartM2M)

Note: if you want that the rule execute action for all .csv file, without additionl filter on filename, you can replace the value “SmartM2M” with “.csv”

The precedence rank of execution is setted to 9, so it means that it will be executed:

- after rules with precence <9
- before rule with precedence >9.

If file transfer match the condition the rule will execute 2 action in the following order:

- 1) Call the In-process custom agent: **com.axway.st.csvfileschecker.CsvFilesCheckerAgent**
TM send a list of invocation parameters to the agent, e.g.:
(debug="true", QFolder="/opt/test/Quarantena/", STNLTarget="VMST52", STNLPort="1305", UABuffer="/opt/test/UAoverflows/csvfilechecker_buffer.dat", ovfSize="2", ovfSleep="3000" , Separator=";",ElementNumber="12")
For more details about these parameters, please see in the follow.
- 2) Call the In-process agent: **com.tumbleweed.st.server.tm.agents.NextPrecedence**
TM send the following list of invocation parameters to the agent:
(Impersonation="true")

In-process agent: **com.axway.st.csvfileschecker.CsvFilesCheckerAgent**

Invocation parameters details:

Parameter	Description	Admitted Value	Expected Behaviour
debug	Activate Custom Agent verbose log level	true	Custom Agent write full detailed log during execution (info, warn, error)
		false	Custom Agent write just significative msgs on ST log file (warn, error)
Qfolder	Quarantine folder complete path	Existing Folder	In case of invalid File, the Agent put in this folder the upload file
STNLTarget	Sentinel Acquisition server or Event Router host	Resolved hostname	Custom Agent send to this host the Sentinel events
STNLPort	Sentinel Acquisition server or Event Router Port	Integer	Custom Agent send to this port the Sentinel Event
UABuffer	Complete Pathname for Sentinel UA Events Buffer File	Existing Folder + BufferName	In case of unavailability of receiver service custom agent saves, on this buffer, the outgoing sentinel events.
STNLTimeout	Timeout for send Sentinel event attempt (in milliseconds)	Integer	This is the specific timeout for the task in charge to send sentinel events via UA invocation
ovfSize	UA Event Buffer size	Integer	Pre-allocated maximum size (in Mbyte) of the Sentinel UA Events Buffer File
ovfSleep	Time interval to empty the overflow file (in millisecond)	Integer	In case of unavailability of the receiver Sentinel service this is the loop time interval in which the Agent try to send again all the Sentinel Events present in the buffer file
Separator	The chosen separator for .csv uploaded file	Char	Note: if you want use a comma: ",", like Separator, please take care that from RulePackages editor point of view, the comma value it's a special character used in order to make parameters separation; So in order to use this value please write the right escape sequence using a backslash before the comma, so write: Separator= "\", instead of Separator=","
ElementNumber	Number of expected element in the	NA	Custom Agent extract the expected element number from the first record of the

	uploaded .csv file		<p>uploaded .csv file; (i.e. if the first record is: A;2;home;K and Separator it's setted to ";" the value of ElementNumber it's setted automatically to 4)</p> <p>After that, the Agent will check that any records in the file contains the same number of element (4), separated by Separator (;). If all the records in the file contains the n. <ElementNumber> of elements the flow pass the check with OK status, otherwise the file will be put in quarantine folder.</p>
		Integer	<p>Custom Agent will check that each record of the uploaded .csv file contain the assigned value of elements. In case of outcome invalid elements number, the file will be put in quarantine folder.</p>

If you need to modify the value of any of these parameters you can act in two way:

- Import the rule package via ST administration web based gui and then modify the imported package;
- Modify the .xml file of the package rule and then import the modified .xml version via ST administration web based gui

Note: after each import or modification you must perform a Transaction Manager service stop/start in order to get the updated rules running.

```
<?xml version="1.0" encoding="UTF-8"?>
<brpackage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0"
xsi:noNamespaceSchemaLocation="businessrules.xsd" name="csvfileschecker.xml">
  <rule name="check_csv_file" type="static" precedence="9">
    <condition>
      <expression>
        <item>
          <attribute>EventType</attribute>
          <comparator name="equal" />
          <value>Incoming end</value>
        </item>
        <operator name="and" />
        <item>
          <attribute>DXAGENT_TARGET</attribute>
          <comparator name="match" />

```

```

        <value>*.csv</value>
    </item>
    <operator name="and" />
    <item>
        <attribute>DXAGENT_TARGET</attribute>
        <comparator name="contain" />
        <value>SmartM2M</value>
    </item>
</expression>
</condition>
<action Order="yes">
    <inprocess-agent id="1" executeafter="" streamaccess="none"
class="com.axway.st.csvfileschecker.CsvFilesCheckerAgent">
    <params>
        <param>
            <param-name>debug</param-name>
            <param-value>"true"</param-value>
        </param>
        <param>
            <param-name>QFolder</param-name>
            <param-value>"/opt/test/Quarantena/"</param-value>
        </param>
        <param>
            <param-name>STNLTarget</param-name>
            <param-value>"VMST52"</param-value>
        </param>
        <param>
            <param-name>STNLPort</param-name>
            <param-value>"1305"</param-value>
        </param>
        <param>
            <param-name>STNLTimeout</param-name>
            <param-value>"1000"</param-value>
        </param>
        <param>
            <param-name>UABuffer</param-name>
            <param-value>"/opt/test/UAoverflows/csvfilechecker_buffer.dat"</param-value>
        </param>
        <param>
            <param-name>ovfSize</param-name>
            <param-value>"2"</param-value>
        </param>
        <param>
            <param-name>ovfSleep</param-name>
            <param-value>"3000"</param-value>
        </param>
        <param>

```

```
        <param-name>Separator</param-name>
        <param-value>"</param-value>";"
    </param>
    <param>
        <param-name>ElementNumber</param-name>
        <param-value>"12"</param-value>
    </param>
</params>
</inprocess-agent>
<inprocess-agent id="2" executeafter="" streamaccess="all"
class="com.tumbleweed.st.server.tm.agents.NextPrecedence">
    <params>
        <param>
            <param-name>Impersonation</param-name>
            <param-value>"true"</param-value>
        </param>
    </params>
</inprocess-agent>
</action>
</rule>
</brpackage>
```

3.2 DatFilesChecker.xml rules package

For each incoming event, Transaction manager is in charge to verify "*datFilesChecker*" rule package condition and, if applicable, execute the related actions.

DatFileChecker.xml rule package, contains the following rule:

Rule	check_csv_file
Precedence	9
Condition	EventType <equal> Incoming end AND DXAGENT_TARGET <match> *.dat AND DXAGENT_TARGET <contain> SIM_SECURITY
Action	id=1 streaming(none) In-process > com.axway.st.csvfileschecker.CsvFilesCheckerAgent (debug="true", QFolder="/opt/test/Quarantena/", STNLTarget="VMST52", STNLPort="1305", STNLTimeout= "1000", UABuffer="/opt/test/UABuffer/datfilechecker_buffer.dat", ovfSize="2", ovfSleep="3000"), fixedLenght="100", RecordHeaderCheck="Y", CheckString="001", fChars="";!<>@#&?")
Action	id=2 executeafter(1) streaming(none) In-process > com.tumbleweed.st.server.tm.agents.NextPrecedence (Impersonation="true")

This rule intercept the end of uploading file process (EventType <equal> Incoming end) for each file that match the following filename pattern : *.dat (DXAGENT_TARGET <match> *.dat).

Furthermore the filename must contain the string "SIM_SECURITY" DXAGENT_TARGET <contain>SIM_SECURITY

Note: if you want that the rule execute action for all .csv file, without additional filter on filename, you can replace the value "SIM_SECURITY" with ".dat"

The precedence rank of execution is set to 9, so it means that it will be executed:

- after rules with precedence <9
- before rule with precedence >9.

If file transfer match the condition the rule will execute 2 action in the following order:

- 1) Call the In-process custom agent: **com.axway.st.datfileschecker.DatFilesCheckerAgent**
TM send a list of invocation parameters to the agent, e.g.:
(debug="true", QFolder="/opt/test/Quarantena/", STNLTarget="VMST52", STNLPort="1305",
UABuffer="/opt/test/UABuffer/datfilechecker_buffer.dat", ovfSize="2", ovfSleep="3000" , fixedLenght="100",

RecordHeaderCheck="Y", CheckString="001", fChars=";<>@#&?")

For more details about these parameters, please see in the follow.

2) Call the In-process agent: **com.tumbleweed.st.server.tm.agents.NextPrecedence**

TM send the following list of invocation parameters to the agent:

(Impersonation="true")

In-process agent: **com.axway.st.datfileschecker.DatFilesCheckerAgent**

Invocation parameters details:

Parameter	Description	Admitted Value	Expected Behaviour
Debug	Activate Custom Agent verbose log level	true	Custom Agent write full detailed log during execution (info, warn, error)
		false	Custom Agent write just significative msgs on ST log file (warn, error)
Qfolder	Quarantine folder complete path	Existing Folder	In case of invalid File, the Agent put in this folder the uploaded file
STNLTarget	Sentinel Acquisition server or Event Router host	Resolved hostname	Custom Agent send to this host the Sentinel events
STNLPort	Sentinel Acquisition server or Event Router Port	Integer	Custom Agent send to this port the Sentinel events
UABuffer	Complete Pathname for Sentinel UA Events Buffer File	Existing Folder + BufferName	In case of unavailability of receiver service custom agent saves, on this buffer, the outgoing sentinel events. .
STNLTimeout	Timeout for send Sentinel event attempt (in milliseconds)	Integer	This is the specific timeout for the task in charge to send sentinel events via UA invocation
ovfSize	UA Event Buffer size	Integer	Pre-allocated maximum size (in Mbyte) of the Sentinel UA Events Buffer File
ovfSleep	Time interval to empty the overflow file (in millisecond)	Integer	In case of unavailability of the receiver Sentinel service this is the loop time interval in which the Agent try to send again all the Sentinel Events present in the buffer file
fixedLenght	Assigned fixed length for records	NA	Custom Agent will not perform any checks on records size
		Integer	Custom Agent will check that each record of the uploaded .dat file contain the

			assigned number of chars. In case of outcome invalid chars number, the file will be put in quarantine folder
RecordHeaderCheck	Activate record head check	true	Custom Agent will perform a check on record head.
		false	Custom Agent will not perform any check on record head.
CheckString	String to check on record head		If RecordHeaderCheck is setted to « true » this string will be checked as record head. In case of outcome invalid record head, the file will be put in quarantine folder
fChars	Forbidden Chars	NA	In case of NA value Custom Agent will perform that just the comma char “,” is not present on the uploaded .dat file. In case of outcome “comma” char it’s present on uploaded file, the file will be put in quarantine folder.
		String	Custom Agent will perform a check for comma “,”, and all the chars present in the parameter fchar. In case of outcome forbidden char it’s present on uploaded file, the file will be put in quarantine folder.

If you need to modify the value of any of these parameters you can act in two way:

- Import the rule package via ST administration web based gui and then modify the imported package;
- Modify the .xml file of the package rule an then import the modified .xml version via ST administration web based gui

Note: after each import or modification you must perform a Transaction Manager service stop/start in order to get the updated rules running.

```
<?xml version="1.0" encoding="UTF-8"?>
<brpackage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0"
xsi:noNamespaceSchemaLocation="businessrules.xsd" name="datfileschecker.xml">
  <rule name="check_dat_files" type="static" precedence="9">
    <condition>
      <expression>
        <item>
```

```

        <attribute>EventType</attribute>
        <comparator name="equal" />
        <value>Incoming end</value>
    </item>
    <operator name="and" />
    <item>
        <attribute>DXAGENT_TARGET</attribute>
        <comparator name="match" />
        <value>*.dat</value>
    </item>
    <operator name="and" />
    <item>
        <attribute>DXAGENT_TARGET</attribute>
        <comparator name="contain" />
        <value>SIM_SECURITY</value>
    </item>
</expression>
</condition>
<action Order="yes">
    <inprocess-agent id="1" executeafter="" streamaccess="none"
class="com.axway.st.datfileschecker.DatFilesCheckerAgent">
    <params>
        <param>
            <param-name>debug</param-name>
            <param-value>"false"</param-value>
        </param>
        <param>
            <param-name>QFolder</param-name>
            <param-value>"/opt/test/Quarantena/"</param-value>
        </param>
        <param>
            <param-name>STNLTarget</param-name>
            <param-value>"VMST52"</param-value>
        </param>
        <param>
            <param-name>STNLPort</param-name>
            <param-value>"1305"</param-value>
        </param>
        <param>
            <param-name>STNLTimeout</param-name>
            <param-value>"1000"</param-value>
        </param>
        <param>
            <param-name>UABuffer</param-name>
            <param-value>"/opt/test/UAoverflows/datfilechecker_buffer.dat"</param-value>
        </param>
        <param>

```

```

        <param-name>ovfSize</param-name>
        <param-value>"1"</param-value>
    </param>
    <param>
        <param-name>ovfSleep</param-name>
        <param-value>"500"</param-value>
    </param>
    <param>
        <param-name>fixedLenght</param-name>
        <param-value>"100"</param-value>
    </param>
    <param>
        <param-name>RecordHeaderCheck</param-name>
        <param-value>"Y"</param-value>
    </param>
    <param>
        <param-name>CheckString</param-name>
        <param-value>"001"</param-value>
    </param>
    <param>
        <param-name>fChars</param-name>
        <param-value>"!&lt;&gt;@#&amp;?"</param-value>
    </param>
</params>
</inprocess-agent>
<inprocess-agent id="2" executeafter="1" streamaccess="none"
class="com.tumbleweed.st.server.tm.agents.NextPrecedence">
    <params>
        <param>
            <param-name>Impersonation</param-name>
            <param-value>"true"</param-value>
        </param>
    </params>
</inprocess-agent>
</action>
</rule>
</brpackage>

```

4. Custom Agent and Rule Package Installation

The general procedure of installation should be:

- 1) Check that the needed ST System libraries are already installed([see chapter 2.1 of this document](#))
- 2) Check that the additional needed libraries are already installed ([see chapter 2.2 of this document](#))
- 3) Install the Custom Agent
- 4) Add a row on agentlist file for each new installed Custom Agent
- 5) Install at least a Rule Package, with at least a rule to execute the agent
- 6) Enable the Rule Package
- 7) Stop and Start the ST Transaction Manager service.

The three Custom Agents are distributed via the following Java Archive file (.jar):

- XmlFilesChecker_agent.jar
- CsvFilesChecker_agent.jar
- DatFilesChecker_agent.jar

The related rows to add on the following agentlist file are:

.../SecureTransport/brules/local/agentlist

- com.axway.st.xmlfileschecker.XmlFilesCheckerAgent
- com.axway.st.csvfileschecker.CsvFilesCheckerAgent
- com.axway.st.datfileschecker.DatFilesCheckerAgent

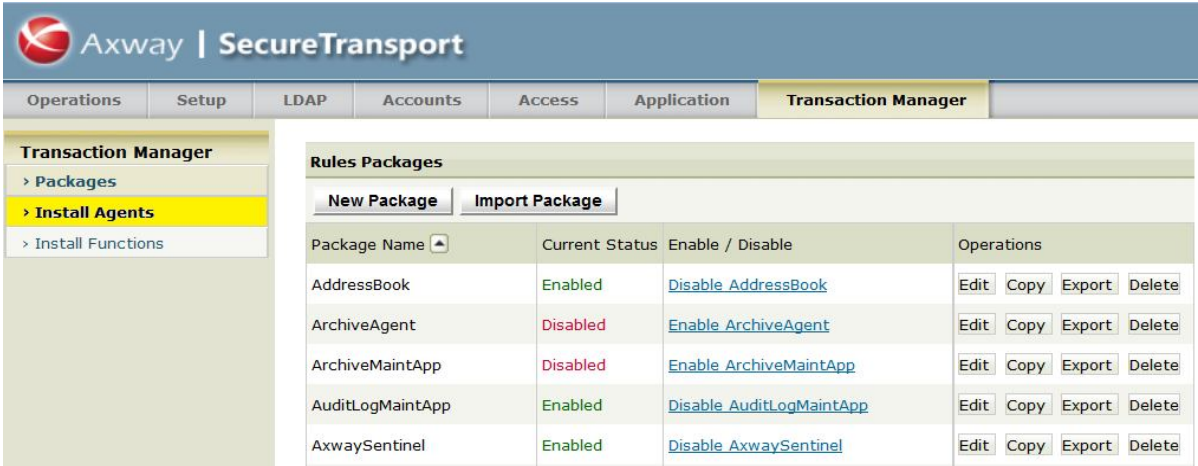
The three default Rule Package are distributed via the following.xml files:

- xmlfileschecker.xml
- csvfileschecker.xml
- datfileschecker.xml

4.1 Install Custom Agent

In order to install the custom agent, please follow these steps:

- a) Copy the custom agent jar file in a local folder accessible via ST web based administration GUI.
- b) Log in, with Administration grants, on ST web based administration GUI.
- c) Navigate on Transaction Manager tab, and click on: Install Agents



The screenshot shows the 'Transaction Manager' section of the Axway | SecureTransport interface. The left sidebar has 'Install Agents' highlighted. The main area displays a table of 'Rules Packages'.

Package Name	Current Status	Enable / Disable	Operations
AddressBook	Enabled	Disable AddressBook	Edit Copy Export Delete
ArchiveAgent	Disabled	Enable ArchiveAgent	Edit Copy Export Delete
ArchiveMaintApp	Disabled	Enable ArchiveMaintApp	Edit Copy Export Delete
AuditLogMaintApp	Enabled	Disable AuditLogMaintApp	Edit Copy Export Delete
AxwaySentinel	Enabled	Disable AxwaySentinel	Edit Copy Export Delete

d) Click on the Browse button to open the file explorer tool:



The screenshot shows the 'Install Agents' section of the Axway | SecureTransport interface. The left sidebar has 'Install Agents' highlighted. The main area contains a form for selecting agents.

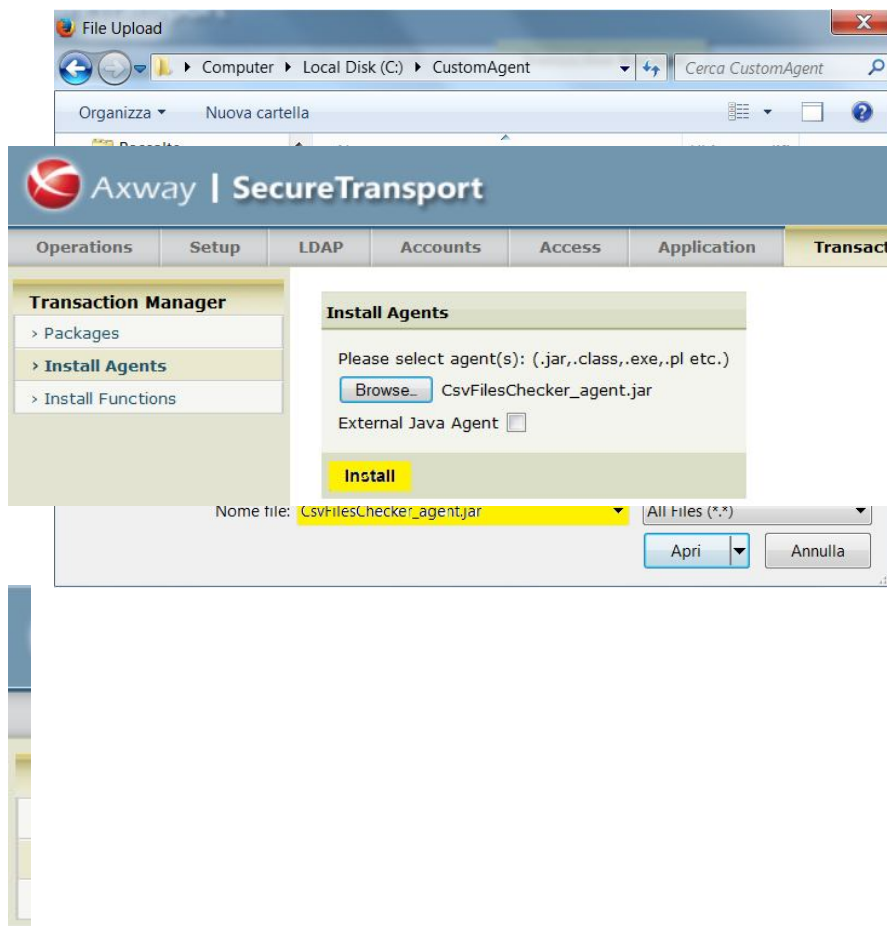
Please select agent(s): (.jar,.class,.exe,.pl etc.)

Browse... No file selected.

External Java Agent ☐

Install

e) Navigate the local folder to select the appropriate .jar file:



f) Verify that you have chosen the right jar and then click on Install button:

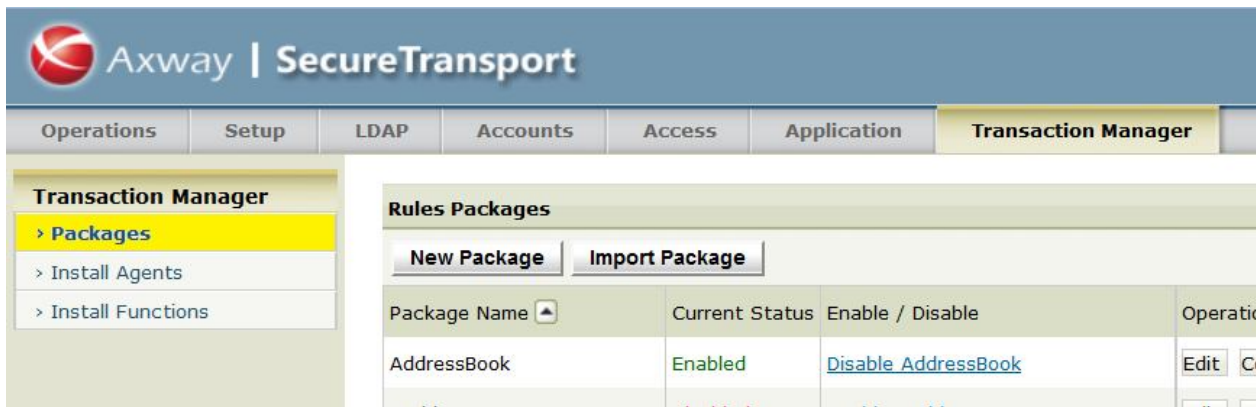
If everything is gone fine you will get the following

confirmation message:

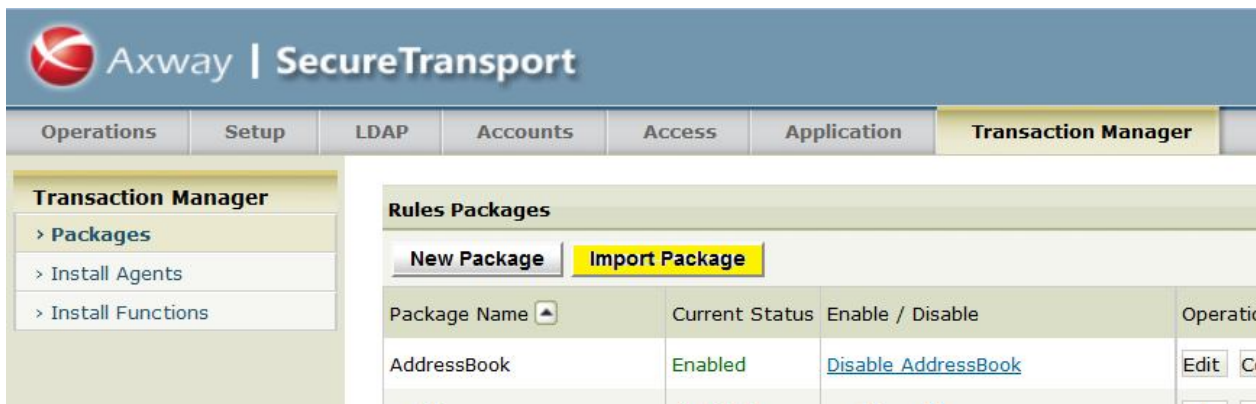
4.2 Install Rule Package

In order to install the Rule Package, please follow these steps:

- Copy the Rule Package files in a local folder accessible via ST web based administration GUI.
- Log in, with Administration grants, on ST web based administration GUI.
- Navigate on Transaction Manager tab, and click on: Packages




d) Click on Import Package



- Click on the Browse button:
- Select the appropriate Rule Package file:



- Verify that you have choose the right .xml and then click on Submit button:



Operations	Setup	LDAP	Accounts	Access	Application	Transaction Manager
------------	-------	------	----------	--------	-------------	----------------------------

Transaction Manager

- > Packages
- > Install Agents
- > Install Functions

Import Rules Package

Please select the import file: csvfileschecker.xml

If you would like to save the imported rule package with other name, please specify:

If everything it is gone fine you will get the update list of installed Rule Package with the new one.

Streaming	Enabled	Disable Streaming	Edit	Copy	Export	Delete
TransferLogMaintApp	Enabled	Disable TransferLogMaintApp	Edit	Copy	Export	Delete
UnlicensedAccountMaintApp	Enabled	Disable UnlicensedAccountMaintApp	Edit	Copy	Export	Delete
WebServicesAPI	Enabled	Disable WebServicesAPI	Edit	Copy	Export	Delete
csvfileschecker	Disabled	Enable csvfileschecker	Edit	Copy	Export	Delete
datfileschecker	Enabled	Disable datfileschecker	Edit	Copy	Export	Delete
fileschecker_rules	Enabled	Disable fileschecker rules	Edit	Copy	Export	Delete

e) At this point please note that the Rule Package status is: Disabled;
So please click on Enable csvfileschecker in order to Enable the Rule Package:

WebServicesAPI	Enabled	Disable WebServicesAPI	Edit	Copy	Export	Delete
csvfileschecker	Disabled	Enable csvfileschecker	Edit	Copy	Export	Delete
datfileschecker	Enabled	Disable datfileschecker	Edit	Copy	Export	Delete

So at this point the Rule Package should be Enabled.

Streaming	Enabled	Disable Streaming	Edit	Copy	Export	Delete
TransferLogMaintApp	Enabled	Disable TransferLogMaintApp	Edit	Copy	Export	Delete
UnlicensedAccountMaintApp	Enabled	Disable UnlicensedAccountMaintApp	Edit	Copy	Export	Delete
WebServicesAPI	Enabled	Disable WebServicesAPI	Edit	Copy	Export	Delete
csvfileschecker	Enabled	Disable csvfileschecker	Edit	Copy	Export	Delete
datfileschecker	Enabled	Disable datfileschecker	Edit	Copy	Export	Delete
fileschecker_rules	Enabled	Disable fileschecker rules	Edit	Copy	Export	Delete

Appendix A – CheckedFiles Tracked Object

In order to collect the Sentinel events incoming from the three custom agents a dedicated Sentinel Tracked Object (TO) has been developed.

The public name of this TO is CheckedFile. It is delivered via the “**checkedfile.xml**” file.

In addition to all the technical default fields included in TO, it implements the following additional fields:

Name	Data Type	Length	Description
Filename	Variable string	100	Name of the uploaded file
Type	Variable string	10	Expected data type as incoming file uploaded. Possible values are: (xml csv dat)
Status	Variable string	15	Status of the uploaded file. Possible value are: (Accepted Refused)
Error	Variable string	4000	Error description for refused files. “NA” for Accepted files.
QFile	Variable string	100	Quarantine Folder for Refused uploaded File. In case of Accepted file this column it's filled with “NA” value.

In case of expected Filename or Quarantine Folder with length that exceed the default value, please adjust the column value before deploying the TO.