

AWD

AXA NoMatch Application

Implementation Guide

04/02/2020

Contents

[1. About This Guide 4](#_Toc30063571)

[2. Business Overview 5](#_Toc30063572)

[2.1 Business Requirements 5](#_Toc30063573)

[2.2 Solution Summary 5](#_Toc30063574)

[3. Design Diagrams/Sample XML 6](#_Toc30063575)

[3.1 axaPASNoMatch POJO Software Components Diagram 6](#_Toc30063576)

[3.2 Sample axaPASNoMatch.xml File 7](#_Toc30063577)

[4. Components 8](#_Toc30063578)

[4.1 Software Components 8](#_Toc30063579)

[4.2 Custom 8](#_Toc30063580)

[Appendix A. Revision History 11](#_Toc30063581)

# About This Guide

The following formatting conventions are used throughout this guide:

* Bold—Keyboard keys, selected/displayed values, and interface elements, such as buttons, labels, boxes, tables/columns, menus, and options.
* Code—Code words, phrases, and syntax you must type exactly as shown.

1. Helpful information for the current topic/procedure.
2. Necessary information crucial for the current topic/procedure.
3. Information to increase your speed and efficiency while using the application.
4. Your actions could cause a loss of data or fatal system error.

# Business Overview

## Business Requirements

Within the Phase 1 scope of the AXA AWD and Systems Program, AXA's AWD version will be upgraded to the latest version of AWD 7.4.x. Within the scope of Phase 2, AXA's infrastructure will be migrated to the DPC infrastructure and will use the latest applicable version of AWD at that time. The legacy axaPASNoMatch AWD Process Automation Service (PAS) job will be replaced with a Plain Old Java Object (POJO) application that can be scheduled using CRON or the methodology supported by the AWD infrastructure.

## Solution Summary

The current axaPASNoMatch PAS job will be replaced with a POJO application named axaPASNoMatch that will be scheduled to run as a CRON job (or something similar).

# Design Diagrams/Sample XML

## axaPASNoMatch POJO Software Components Diagram



## Sample configuration xml File

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

<axaPASNoMatchWhiteboard>

<AWDInformation>

<userID>DSTSETUP</userID>

<businessArea>EVSRV</businessArea>

<dbHostAddr>170.40.103.208</dbHostAddr>

<dbHostPort></dbHostPort>

<dbUserID>PS174DB</dbUserID>

<workingDirectory>/opt/NoMatch/dist</workingDirectory>

<propertiesFile>/apps/awd/jboss-axadev/batchJobs/LETPUB.properties</propertiesFile>

<type>

<workType>FUNDSWOAPP</workType>

<workType>B7FNDWOAPP</workType>

<workType>EPACNOCSH</workType>

<workType>EPACWCSH</workType>

<workType>NJFDSWOAPP</workType>

<!--IMPORTANT insert additional work types above. 'DUMMYLAST' should be the last work type -->

<workType>DUMMYLAST</workType>

</type>

<queue>RENDEV</queue>

<age5DaysStatus>NOMATCH05</age5DaysStatus>

<age10DaysStatus>NOMATCH10</age10DaysStatus>

<age15DaysStatus>NOMATCH15</age15DaysStatus>

<age17DaysStatus>NOMATCH17</age17DaysStatus>

<age20DaysStatus>NOMATCH20</age20DaysStatus>

<startTime>0600</startTime>

</AWDInformation>

</axaPASNoMatchWhiteboard>

# Components

## Software Components

Software components in the solution:

* axaPASNoMatch Plain Old Java Object (POJO) Application - Will be created to replace the existing axaPASNoMatch AWD Process Automation Service (PAS) job.
* zspNOMATCH Stored Procedure - Calls AWD to look for existing work objects—matching on the passed in business area, work type, queue, and CRDATTIM values.
* Java KeyStore (JKS) - Used for storing credentials. The AWD user password will be removed from the axaPASNoMatch.xml configuration file. JKS location and password system properties will be passed to the axaPASNoMatch POJO when it is started. The application will use this information to retrieve the AWD user password, hostAddress, hostPort, and workingDirectory values from the JKS.

## Custom

The axaNoMatch application doesn’t has loop like PAS job, to keep job consistent with others, the job will be scheduled to start every 70 seconds

1. Obtain the AWDPassword password from the JKS.

The whiteboard XML data that was passed to the axaPASNoMatch AWD Job Designer job will now need to be made available to the rewritten axaPASNoMatch Java application using a configuration file. The values that were pulled from the AWD Job Designer macros will now be passed to the rewritten Java application by placing them in a JKS. These include the following keys: hostAddress, hostPort, and workingDirectory. The AWD password will be removed from the configuration file for security reasons. The JKS location and password will be passed to the axaPASNoMatch application through the system properties when the application is started.

1. Read in the axaAWDcalendar.xml file information as an XML document. Set the workingDirectory value stored in the JKS to the directory where the axaAWDcalendar.xml file resides.
2. Set the currentTime variable to the current date/time.
3. Determine if today is a business day based on the contents of the axaAWDcalendar.xml file:
   1. If not, shut down the axaPASNoMatch application and the application will start again by the cron job.
   2. Else continue.
4. Check whether the current time equals the configuration file’s AWDInformation/startTime:
   1. If Yes, continue.
   2. Else, if the current time is greater than the AWDInformation/startTime, determine if the process ran earlier by checking for the NoMatch\_RunReport\_{currentDate in yyyy-MM-dd format}.xml file.
5. If the file exists, shut down the axaPASNoMatch application and the application will start again by the cron job.
6. If the process did not run earlier, continue.
   1. Else, if the current time is not greater than the AWDInformation/startTime, shut down the axaPASNoMatch application and the application will start again by the cron job.
7. Calculate the dates for objects that need to be updated:

* age5days (date that is 5 business days older than the current date)
* age10Days (date that is 10 business days older than the current date)
* age15Days (date that is 15 business days older than the current date)
* age17Days (date that is 17 business days older than the current date)
* age20Days (date that is 20 business days older than the current date)

1. Call the zspNOMATCH stored procedure:

* Pass in the following information: the configuration file’s businessArea, workType list, queue, and the calculated age5days value, age10Days value, age15Days value, age17Days value, and age20Days value.
* The zspNOMATCH stored procedure returns the AWD key for items where the UNITCD equals the passed in businessArea, the WRKTYPE is included in the passed in list of work types, the QUEUECD equals the passed in queue, and the CRDATTIM is included in the passed in list of age5days, age10Days, age15Days, age17Days, and age20Days values.
* If the zspNOMATCH stored procedure call is successful, continue.
* Else, return the error code and description.

1. Create a list of objects to update:

* Build a list of transaction and case items including the item key and updateStatus based on the item's CRDATTIM—matching the age5days, age10Days, age15Days, age17Days, or age20Days value. Map the age in days to the appropriate configuration status value. For example, if the item's CRDATTIM equals the age5days variable value, then the updateStatus should be set to the configuration age5DaysStatus value.

1. Iterate through the list of objects to update, and do the following for each:
   1. Update the status of the work item to the updateStatus value using the updateObjects SOAP service: <routing route="Status">[updateStatus]</routing>.
   2. If an error occurs updating the status, do the following:
2. Unlock and unassign the work item using the updateObjects SOAP service.
3. Write the error information to an error file in the JKS workingDirectory with the following naming convention: userID value, plus "axaPASNoMatch", plus current date/time, plus "Error.xml" (containing the error code and description).
4. Shut down the axaPASNoMatch application.
5. After processing the items in the objects to update list, do the following:
   1. Write out the NoMatch\_RunReport\_{currentDate}.xml file, containing the updated items.
   2. Shut down the axaPASNoMatch application and the application will start again by the cron job.

# Design Flowchart

Config file

The job will be scheduled to process every 70 seconds

Database

Calendar XML file

JKS file

No

No

No

Yes

Yes

Yes

End

End

End

End

Create Report

Call Update Object SOAP

Read XML Config

Create AWD object list with update status

Call SP ZSPNOMATCH

Calculate dates for SP

If report file of current date exists

Time >= Start time

Is business day?

Get Current datetime

Read AWD Calendar XML

Get Password

# Installation and configuration

After the NoMatch package has been in place and extracted to the location it should be run, there should be three directories as follows:

* config - this directory contains a sample configuration file.
* dist - this directory contains nomatch.jar and the scripts which use to run the program.
* lib - this directory contains the library used in this program.

The following steps are required to set up the program.

* The machine running the program should have at least JRE 1.6.
* If the program is on Microsoft Windows, the NoMatchEnvironment.cmd file in the dist directory should be updated with the following values. If it is on Linux, the NoMatchEnvironment.sh should be updated.

| Parameter | Description |
| --- | --- |
| DT\_JAVA\_PATH | The path to the Java version to be used for the import program i.e. C:\Program Files\Java\jre6\bin |
| DT\_HOME | The location of the program. |
| DT\_CONFIG | The path and name of the configuration file i.e. %ST\_HOME%\config\config.xml |

The configuration file, config.xml in config directory, must be updated with appropriate values. The following table describes each field in the configuration file.

| Parameters | Description |
| --- | --- |
| userID | The userID is AWD user ID |
| dbHostAddr | The database host address (DB2) |
| dbHostPort | The database host port. |
| dbUserID | The database user ID to connect the DB2 database. |
| workingDirectory | The working directory that will use to store the file. |
| propertiesFile | The property file location, this property file will keep KeyStore file location, KeyStore password, AWD database name and AWD web service wsdl location. |

The properties file is common use with other jobs, and it keep property values below

| Properties | Description |
| --- | --- |
| KEYSTORE\_PATH | KeyStore file location |
| KEYSTORE\_PASSWORD | KeyStore file password |
| AWD\_Schema\_name | AWD database name |
| AWDprocess\_service\_wsdl | AWD web service wsdl location. |

# Execution

The program can be run by the script NoMatch.cmd or NoMatch.sh in the dist directory or use the following command.

java -cp nomatch.jar;../lib/\* com.dstawd.custom.axa.nomatch.CollectionsNoMatchApp -f<path to configuration file>

-f = path to the configuration file

* + 1. Revision History

| Version | Chapter | Date | Author | Change Description |
| --- | --- | --- | --- | --- |
| 1.0 | All | 4/3/2020 | Teerachai Witchudakornkul | Initial Release |