



AccountingIntegrator

Version 2.2.1

z/OS

1 July 2019

Installation Guide



Copyright © 2019 Axway

All rights reserved.

This documentation describes the following Axway software:

Axway AccountingIntegrator 2.2.1

No part of this publication may be reproduced, transmitted, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of the copyright owner, Axway.

This document, provided for informational purposes only, may be subject to significant modification. The descriptions and information in this document may not necessarily accurately represent or reflect the current or planned functions of this product. Axway may change this publication, the product described herein, or both. These changes will be incorporated in new versions of this document. Axway does not warrant that this document is error free.

Axway recognizes the rights of the holders of all trademarks used in its publications.

The documentation may provide hyperlinks to third-party web sites or access to third-party content. Links and access to these sites are provided for your convenience only. Axway does not control, endorse or guarantee content found in such sites. Axway is not responsible for any content, associated links, resources or services associated with a third-party site.

Axway shall not be liable for any loss or damage of any sort associated with your use of third-party content.

Contents

| | |
|---|-----------|
| Preface | 5 |
| Who should read this document | 5 |
| Related documentation | 5 |
| Accessibility | 6 |
| Screen reader support | 6 |
| Support for high contrast and accessible use of colors | 6 |
| 1 Overview | 7 |
| Components available in the installation kit | 7 |
| Recommendations | 7 |
| 2 Installation medium | 8 |
| Delivery medium | 8 |
| RDJHOME – Delivery environment | 8 |
| RDJEXEC – Execution environment | 8 |
| MQSeries objects | 9 |
| SIC – Validation test deck | 10 |
| 3 Product installation | 11 |
| Step 1: Creating the RDJHOME environment | 11 |
| Copying the file in the current library | 11 |
| Updating z/OS parameters | 11 |
| Transferring, customizing and submitting the installation JCL | 12 |
| Automatic mode (FTP) | 12 |
| Manual mode | 13 |
| Checking the execution | 14 |
| Step 2: Creating the RDJEXEC environment | 14 |
| Updating the environment parameters | 14 |
| Executing the CLIST PMAJEXFR/EN | 16 |
| Executing the JCL <RDJEXEC>.JCLLIB(RDJDEF) | 17 |
| Creating MQSeries objects | 17 |
| Step 3: Implementing and validating (SIC) | 18 |
| Submitting the main JCL | 18 |
| Managing errors | 19 |
| 4 Uninstalling an environment | 22 |
| Execution environment | 22 |
| Delivery environment | 22 |

| | |
|--|-----------|
| 5 Appendix | 23 |
| Contents of the delivery environment (RDJHOME) | 23 |
| List of created MQSeries objects (MQSeries mode) | 28 |
| Contents of the standard test deck (SIC) | 30 |
| Overview | 30 |
| Details on each JCL Sxx | 32 |

Preface

The purpose of this document is to guide you through the installation of the product in a z/OS environment and the validation of this installation by running the test deck provided by Axway.

Who should read this document

This document is intended for users who will install product Rule Engine.

In this guide, we assume that you have a good understanding of the previous Rule Engine versions and of the z/OS technical environment.

Related documentation

The AccountingIntegrator documentation set includes the following documents:

- *AccountingIntegrator User Guide*
- *AI Enabler User Guide*
- *AccountingIntegrator Installation Guide*
- *Rule Engine Reference Guide*
- *Rule Engine Operations Guide z/OS*
- *Rule Engine Exits and External Calls*
- *Rule Engine Error Messages Guide*
- *Rule Engine Installation Guide z/OS*
- *Rule Engine Installation Guide OS/400*

Accessibility

Axway strives to create accessible products and documentation for users.

This documentation provides the following accessibility features:

- Screen reader support
- Support for high contrast and accessible use of colors

Screen reader support

- Alternative text is provided for images whenever necessary.
- The PDF documents are tagged to provide a logical reading order.

Support for high contrast and accessible use of colors

- The documentation can be used in high-contrast mode.
- There is sufficient contrast between the text and the background color.
- The graphics have the right level of contrast and take into account the way color-blind people perceive colors.

This chapter lists the components to install and the installation recommendations.

Components available in the installation kit

The components that you can install on z/OS are the following:

- Rule Engine File for the File Engine (EBCDIC encoding)
- Rule Engine MQSeries for the MQSeries Engine (EBCDIC encoding)
- Rule Engine File UTF-16 for the File Engine (UTF-16 encoding)

Rule Engine Composer and AI Enabler installation procedures are delivered in separate installation kits and documented in specific manuals.

Recommendations

Before you start the installation, we recommend that you check the validity of your technical environment against the *AI Suite Installation and Prerequisites Guide*.

We also recommend that you install the product in an empty site to avoid any interaction with a previous installation and that you use new prefixes for file names.

Installation medium

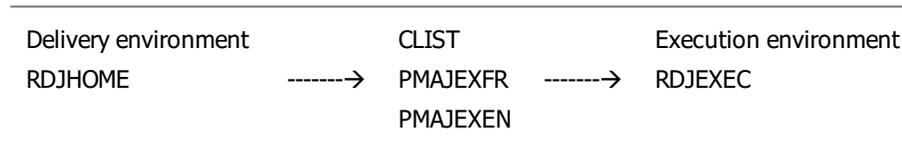
2

This chapter describes the installation media, how to define the reference (RDJHOME) and execution (RDJEXEC) environments and also describes the content of the standard test deck (SIC) used to validate the installation.

Delivery medium

The product is delivered on CD-ROM.

To install the product, download the contents of the file delivered on the support called "product file" in the environment called "delivery environment".



RDJHOME - Delivery environment

The environment created after download from the provided support is referred to as < RDJHOME > in the rest of the document. It is made up of all the files that will enable you to create one or more execution environments and the product common files not copied in each execution environment.

For more information on these files, refer to section [Contents of the delivery environment \(RDJHOME\)](#) on page 23 in the appendix.

RDJEXEC - Execution environment

From the RDJHOME environment, you can automatically generate as many execution environments as you need via the PMAJEXFR CLIST in French or the PMAJEXEN CLIST in English. You can define for example a development environment, a test environment, an approval environment, or a production environment.

You specify in the CLIST the parameters specific to the execution environment to generate, then you execute it. It generates a JCL and submits it.

Each environment prefix RDJEXEC is referred as <RDJEXEC>.

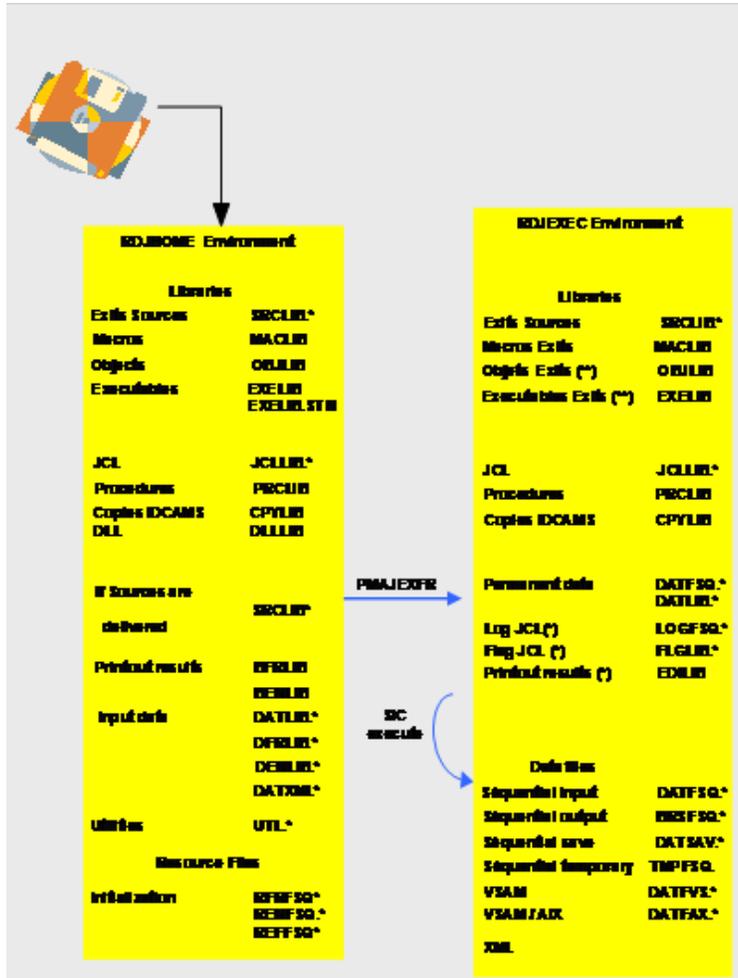


Figure 1. Figure 1: RDJHOME and RDJEXEC environments

Libraries and files indicated by:

- (*) Are copied as empty members (member \$\$CRIGHT) and completed when the SIC (test deck) is submitted
- (***) Are copied as empty members (member \$\$CRIGHT) and completed when the client exits are compiled and link edited

Note Although delivery and execution environments can make up a single environment, however we recommend that you create two separate environments.

MQSeries objects

In MQSeries mode and in the customized execution environment, you can via a JCL, automatically:

- Create queues and triggering processes by setting parameters with the name of the queue manager and the prefix of the created objects. In this way, you can distinguish MQSeries

objects from the objects of another execution environment.

- Delete these MQSeries objects.

For more information on these objects, refer to section [List of created MQSeries objects \(MQSeries mode\) on page 28](#) in the appendix.

SIC - Validation test deck

To validate the product installation in the execution environment (RDJEXEC), you submit the SIC (test deck). Result files generated in RDJEXEC are compared with the result files delivered by the Rule Engine in RDJHOME.

For more information on the test deck, refer to section [Contents of the standard test deck \(SIC\) on page 30](#) in the appendix.

This chapter describes how to create the environments, implement and validate them.

Step 1: Creating the RDJHOME environment

This step consists in downloading the RDJHOME environment to the z/OS platform from the product file.

Copying the file in the current library

On Windows, copy the files in <DVD_Drive>:\Software\[component_name]\install into the current library that contains the following files:

- envMvs.bat: z/OS variables to be set
- setupMvs.bat: Start procedure
- setupMvs.jcl: Installation JCL, not customized
- setupMvs_fil.xmi: Product file – file mode
- setupMvs_mqs.xmi: Product file – MQSeries mode
- setupMvs_fil16.xmi: Product file – UTF-16 file mode

Updating z/OS parameters

Still on Windows, update the z/OS parameters in the file envMvs.bat.

| Parameter | Description |
|-----------|---|
| JOBC1 | First card of the installation JCL. It ends with a comma. |
| JOBC2 | Second card of the installation JCL. It starts with a space. |
| UPLIB | Installation library of the product file and the customized JCL. |
| ENV | <RDJHOME> prefix of the RDJHOME delivery environment. Modify at the most the 3 first qualifiers using the RENAMEUNC parameter of ADDRSSU. |

| Parameter | Description |
|------------|---|
| EUNIT | Unit type of <RDJHOME>** files.(6 characters - optional) |
| EVOLSER | Volume name of <RDJHOME>** files. (6 characters - optional) |
| EMGMTCLASS | Management class of <RDJHOME>** files.(8 characters - optional)Specify NULL for NULLMGMTCLAS. |
| ESTORCLASS | Storage class of <RDJHOME> files(8 characters - optional)Specify NULL for NULLSTORCLAS. |

Then, and only if the client uses FTP, complete the following parameters:

- HOST:IP address of the z/OS host
- USERID: z/OS user account
- PASSWORD: associated password

Transferring, customizing and submitting the installation JCL

If the client uses FTP, the following actions are executed automatically:

- Transfer with pre-allocation of the product file in the <UPLIB> library
- Customization and submission of the installation JCL with transfer in the UPLIB library

If the client does not use FTP, you must enter the commands manually.

Automatic mode (FTP)

On Windows

1. Run the following command in the current library:

```
>setupMvs -file for the file mode
```

```
>setupMvs -mq5 for the MQSeries mode
```

```
>setupMvs -file16 for the UTF-16 file mode
```

The next time you run the command or in case of failure, since the product file has already been transferred, run the following command that only customizes and submits the installation JCL:

```
>setupMvs -sub
```

2. Check that the:

- Message "Procedure setupMvs successful" is displayed
- Generated installation JCL setupMvs.jcl_gen exists in the current library

On z/OS

1. Check that the:
 - Product file exists in the member <UPLIB>(SETUPXMI)
 - Customized installation JCL exists in the member <UPLIB>(SETUPJCL)

Manual mode

On Windows

1. Run the following command in the current library to customize the installation JCL:>setupMvs – gen
2. Check that the:
 - Message "Procedure setupMvs successful" is displayed
 - Generated installation JCL setupMvs.jcl_gen exists in the current library

On z/OS

1. Allocate the PDS <UPLIB> with a DCB=(RECFM=FB, LRECL=80) and a size of 30 cylinders.

On Windows

1. Transfer from Windows to z/OS the:
 - Product file in binary mode:setupMvs_fil.xml ---> member <UPLIB>(SETUPXMI) for file modesetupMvs_mqs.xml ---> member <UPLIB>(SETUPXMI) for MQSeries mode
 - setupMvs_fil16.xml à member <UPLIB>(SETUPXMI) for UTF-16 file mode
 - customized installation JCL in ASCII mode:setupMvs.jcl_gen ---> member <UPLIB>(SETUPJCL)

On z/OS

1. Check that the:
 - Product file exists in the member <UPLIB>(SETUPXMI)
 - Customized installation JCL exists in the member <UPLIB>(SETUPJCL)
2. Submit the customized installation JCL.

<UPLIB> (SETUPJCL)

Checking the execution

On z/OS, check that the:

- Installation job ended successfully (RC=00)

Note: In some cases, the step ADENVO of the program ADRDSSU returns a code that is different from zero (RC=04) and the following warning message is displayed in the SYSPRINT for each file that is restored temporarily before it is renamed with the <RDJHOME> prefix. This message is just a warning and you can continue.

```
ADR755W (001)-PROTD(01), SOURCE DATA SET ARDJD.XRDISNNN... WAS
GENERALLY PROTECTED. THE TARGET DATA SET
ARDJD.XRDISNNN... IS NOT PROTECTED BY ANY PROFILE
```

- All <RDJHOME> files exist:
 - 60 libraries and files for file mode
 - 65 libraries and files for MQSeries mode
 - 54 libraries and files for UTF-16 file mode

For more information on libraries and files, refer to the section [List of created MQSeries objects \(MQSeries mode\)](#) on page 28 of the Appendix.

Step 2: Creating the RDJEXEC environment

The rest of the installation takes place on z/OS and must be repeated for all the other environments.

This step generates and customizes RDJEXEC execution environment via the update and running of the CLIST called PMAJEXFR/EN:

- <RDJHOME>.UTL.CLIB(PMAJEXFR) for an installation in French
- <RDJHOME>.UTL.CLIB(PMAJEXEN) for an installation in English

Updating the environment parameters

Update the parameters of the execution environment in the CLIST PMAJEXFR/G.

The values of the parameters listed below must be specified between the two parentheses of &STR (...). If no value is specified, the parameter is ignored. For legibility, relevant lines start in column 1.

Delivery environment definition paragraph

In the SET RDJHOME line, replace the value set by default by the client value <RDJHOME>. This value must be the same as the value of the parameter ENV completed previously (see paragraph [Updating z/OS parameters on page 11](#)).

Execution environment definition paragraph

| Parameter | Description | Values / Limits |
|------------------|---|--|
| RDJEXEC | <RDJEXEC> Prefix of the execution environment. | 21 characters maximum |
| QMGNAM | Name of the Queue Rule Engine MQ (MQSeries mode) | 4 characters maximum |
| QMGRPF | Prefixes of the MQ objects (MQSeries mode) | 25 characters |
| JOBC1N | Job name common to all JCLs | 8 characters maximum |
| JOBRDM | Job name specific to the JCLs that send administration messages (MQSeries mode) | 8 characters maximum |
| JOBTGG | Job name specific to the JCL launched by the triggering process (MQSeries mode) | 8 characters maximum |
| JOBTGP | Job name specific to the JCL that stops the trigger (MQSeries mode) | 8 characters maximum |
| JOBTGR | Job name specific to the JCL that starts the trigger (MQSeries mode) | 8 characters maximum |
| JOBSTC | Job name specific to the JCLs in Start Task mode (MQSeries mode) | 8 characters maximum |
| JOBC1R | Remainder of the 1 st JOB card | |
| JOBC2T | 2 nd JOB card (**) | 1 st character = space or "*" |
| JOBC3T | 3 rd JOB card | 1 st character = space or "*" |
| MQLOAD | MQ SCSQLOAD library (MQSeries mode) | (*) |
| MQAUTH | MQ SCSQAUTH library (MQSeries mode) | (*) |
| MQANLE | MQ SCSQANLE library (MQSeries mode) | (*) |
| MQC370 | MQ SCSQC370 library (MQSeries mode) | (*) |
| STCLIB | System Proclib in Start Task mode (MQSeries mode) | (*) |

| Parameter | Description | Values / Limits |
|-----------|---|-----------------|
| STCEXP | Name of the Start Task in Start Task mode (MQSeries mode) | (*) |
| STNLIB | API Sentinel library | (*) |
| RUNLIB | Run-Time SCEERUN library | (*) |
| UNLIB | Libraries unit type | (*) |
| UNBRFSQ | Unit type of the sequential result files | (*) |
| UNDATFSQ | Unit type of the permanent sequential files | (*) |
| UNDATSAV | Unit type of the backup sequential files | (*) |
| UNTMPFSQ | Unit type of the temporary catalogued sequential files | (*) |
| UNTMPJCL | Unit type of the temporary non-catalogued sequential files (&&) | (*) |
| VLLIB | Libraries volume name | (*) |
| VLBRFSQ | Volume name of the result sequential files | (*) |
| VLDATFSQ | Volume name of the permanent sequential files | (*) |
| VLDATSAV | Volume name of the backup sequential files | (*) |
| VLTMPFSQ | Volume name of the temporary catalogued sequential files | (*) |
| VLTMPJCL | Volume name of the temporary non-catalogued sequential files (&&) | (*) |
| VLDATFVS | Volume name of the VSAM files | (*) |
| VLDATFAX | Volume name of the AIX / VSAM files | (*) |

Note (*)Optional parameter, not completed if not used.

Note (**)Default value of the region parameter: 256 (REGION=256M).

Executing the CLIST PMAJEXFR/EN

Procedure:

1. In PDS <RDJHOME>.UTL.CLIB, type EXEC in the CLIST PMAJEXFR/EN line.
2. Check the contents of the substituted parameters.

3. Answer the final message:

- Confirm: (O/N) or (Y/N)

| If you confirm, then... | is... |
|---|---|
| the job submitted to update the execution environment | xxxxxxx (where xxxxxxx = value of JOBC1N) |
| the JCL resolved | <RDJHOME>.UTL.SLIB (MAJEXE) |
| the JCL generated and submitted | <RDJHOME>.UTL.SLIB (MAJEXESB) |

1. Check that the:

- Job has ended successfully (RC = 04)
- Following line exists in SDSF in the card ddname FO – step JCLOK--> JCL MAJEXESB : RC=00 - SUCCESSFUL
- Generated execution environment exists: <RDJEXEC>.*
- Following libraries <RDJEXEC>.JCLLIB, JCLLIB.SIC, JCLLIB.STC (MQSeries mode), PRCLIB exist and are customized and their parameters replaced by the values entered when updating the environment parameters.

Caution Some test environments may force you to add new parameters (or norms), even insert new JCL cards. In that case, you must complete the customization manually by adapting the libraries JCLLIB, JCLLIB.SIC and/or PRCLIB.

Executing the JCL <RDJEXEC>.JCLLIB (RDJDEF)

This JCL can only be submitted in a JES2 environment.

In a JES3 environment :

1. Run the JCL <RDJEXEC>.JCLLIB(RDJDEF) twice:
 - The first time with only the part that concerns the Define cluster. The comment « FOR JES3 ...» indicates the division.
 - The second time with the complete procedure <RDJEXEC>.PRCLIB(RDJDEF).
2. Check that the two jobs have ended successfully (RC=00).

Creating MQSeries objects

This step is only executed if you are in MQSeries mode.

Note The JCL ADMMQSDF and ADMMQSDL require MQSeries administration rights to create and delete MQSeries objects via the CSQUTIL program.

1. Modify, if necessary, the following parameters in the file <RDJEXEC>.PRCLIB(\$\$SETQMG) :
 - Name of the Queue Rule Engine: value of the JCL variable set by default in QMGNAM
 - Prefix of the MQSeries objects: value of the JCL variable set by default in QMGPRF
2. Submit the creation JCL ** <RDJEXEC>.JCLLIB(ADMMQSDF)
3. Check that the:
 - Job has ended successfully (RC = 00)
 - Following lines are displayed at the end of the sysprint of procstep CSQUTIL:

```
CSQU057I CSQUCMDS 42 commands read
CSQU058I CSQUCMDS 42 commands issued and responses
received, 0 failedCSQU143I CSQUTIL 1 COMMAND statements
attempted CSQU144I CSQUTIL 1 COMMAND
statements executed successfully CSQU148I CSQUTIL
Utility completed, return code=0
```
 - 42 MQSeries objects (41 local queues + 1 process) exist These objects are described in the Appendix, section [List of created MQSeries objects \(MQSeries mode\) on page 28](#)

In case of prefix error or if you want to delete the created MQSeries objects:

1. Submit the delete JCL <RDJEXEC>.JCLLIB(ADMMQSDL).
2. Check that the execution report at the end of sysprint of procstep CSQUTIL is similar to the one for ADMMQSDF.

Step 3: Implementing and validating (SIC)

This step checks that the translation has been carried out correctly and validates the installation. Specific JCLs from the SIC test deck in the PDS <RDJEXEC>.JCLLIB.SIC are executed, and call standard procedures in the PDS <RDJEXEC>.PRCLIB.

Note At the end of the process, the SIC executes again the initialization procedure (RDJRAZ) of the generated data files of the repository. You can however submit directly the standard JCL in the PDS <RDJEXEC>.JCLLIB with the input files (USR members) that contain the SIC data.

Submitting the main JCL

To implement and validate the installation:

1. Submit the main JCL <RDJEXEC>.JCLLIB.SIC(RDJSIC).
2. Wait until all the JCLs Sxx have been submitted successively.
3. Check that the following information is in the file LOG <RDJEXEC>.LOGFSQ.LOGJCL.SIC.

```

BROWSE      ARDJD.XREXE130.MQS.LOGF3Q.LOGJCL.SIC      Line 00000007 Col 001 080
Command ==>                                     Scroll ==> CSR
Standard test pack :
--> JCL RDJ3IC.. : RC=04 - SUCCESSFUL
--> JCL 300DEF.. : RC=00 - SUCCESSFUL
--> JCL 301RAZ.. : RC=04 - SUCCESSFUL
--> JCL 302MNT.. : RC=00 - SUCCESSFUL
--> JCL 303RCH.. : RC=00 - SUCCESSFUL
--> JCL 304MAJ01 : RC=08 - SUCCESSFUL
--> JCL 305MAJ02 : RC=08 - SUCCESSFUL
--> JCL 306MAJ03 : RC=04 - SUCCESSFUL
--> JCL 307EDR.. : RC=00 - SUCCESSFUL
--> JCL 308CRE.. : RC=00 - SUCCESSFUL
--> JCL 309CLN.. : RC=00 - SUCCESSFUL
--> JCL 309PUT.. : RC=00 - SUCCESSFUL
--> JCL 310EXP.. : RC=04 - SUCCESSFUL
--> JCL 311GET.. : RC=04 - SUCCESSFUL
--> JCL 312CR3.. : RC=00 - SUCCESSFUL
--> JCL 313TRF.. : RC=00 - SUCCESSFUL
--> JCL 399VALID : RC=04 - NO ERROR DETECTED - RULES FILES REINITIALIZED

*****
* INSTALLATION TEST PACK SUCCESSFUL *
*****

***** Bottom of Data *****

```

Note The lines S09CLN, S09PUT and S11GET are only displayed for the MQSeries mode.

Managing errors

If the message "INSTALLATION TEST PACK NOT SUCCESSFUL" is displayed at the end of the LOG file, you can choose, after correcting the error, to submit your JCLs:

- Partially in manual mode
- In start-over mode

Correcting the error

Correct the JCL Sxx that was in error and from which the message is displayed "NOT EXECUTED OR IN ERROR" or "NOT SUCCESSFUL".

Example: JCL S10EXP "NOT SUCCESSFUL"

```

BROWSE      ARDJD.XREXE130.MQS.LOGF3Q.LOGJCL.SIC      Line 00000020 Col 001 080
Command ==>                                     Scroll ==> CSR
--> JCL S10EXP.. : RC=99 - NOT SUCCESSFUL - CHECK CONDITIONS ON STEP JCLOK
--> JCL S11GET.. : RC=99 - NOT RUN - CONDITIONAL PREVIOUS JCL
--> JCL S12CR3.. : RC=99 - NOT RUN - CONDITIONAL PREVIOUS JCL
--> JCL S13TRF.. : RC=99 - NOT RUN - CONDITIONAL PREVIOUS JCL
--> JCL 399VALID : RC=99 - PREVIOUS JCL(3) NOT SUCCESSFUL

*****
* INSTALLATION TEST PACK NOT SUCCESSFUL *
*****

***** Bottom of Data *****

```

Submitting in manual mode

1. Submit again the JCL Sxx that was in error then all the following JCLs one after another and in the same sequence including the JCL S99VALID. Warning: Special case – in MQSeries mode, if one of the JCL S09PUT, S10EXP or S11GET is in error, you must again submit all the JCLs from JCL S09CLN (included) onwards.
2. Every time you submit a JCL, check that in the information added at the end of the LOG file the:
 - Last current line displays "SUCCESSFUL"
 - Last message after completion of the JCL S99VALID is: "INSTALLATION TEST PACK SUCCESSFUL". Going back to the preceding example with the JCL S10EXP in error:

```

BROWSE      ARDJD.XPXE130.MQ3.LOGP3Q.LOGJCL.SIC      Line 00000030 Col 001 060
Command ==>                                     3scroll ==> C3R
--> JCL S09CLN.. : RC=00 - SUCCESSFUL
--> JCL S09PUT.. : RC=00 - SUCCESSFUL
--> JCL S10EXP.. : RC=04 - SUCCESSFUL
--> JCL S11GET.. : RC=04 - SUCCESSFUL
--> JCL S12CR3.. : RC=00 - SUCCESSFUL
--> JCL S13TRF.. : RC=00 - SUCCESSFUL
--> JCL S99VALID : RC=04 - NO ERROR DETECTED - RULES FILES REINITIALIZED

*****
*  INSTALLATION TEST PACK SUCCESSFUL  *
*****

***** Bottom of Data *****

```

2. If you want to have a complete report in one set, as soon as all JCLs have been executed correctly, submit again the main JCL. To do so, refer to the initial procedure (see section [Submitting the main JCL on page 18](#)).

Submitting in restart mode

1. Specify "Y" in the RESTART variable that was set by default to "N" in <RDJEXEC>.JCLLIB.SIC (RDJSIC).
2. Submit again the complete JCL <RDJEXEC>.JCLLIB.SIC(RDJSIC).
3. Submit again the JCL Sxx that was in error then all the following JCLs one after another and in the same sequence including the JCL S99VALID. Wait until all the JCL Sxx have been submitted.
4. Check the information added at the end of the LOG file.
5. Going back to the preceding example with the JCL S10EXP in error:

```
BROWSE      ARDJD.XREXE130.MQ3.LOGP3Q.LOGJCL.SIC      Line 00000030 Col 001 080
Command ==>                                     Scroll ==> CSR
Standard test pack (RESTART) :
--> JCL RDJ3IC.. : RC=04 - SUCCESSFUL
--> JCL S09CLN.. : RC=00 - SUCCESSFUL
--> JCL S09PUT.. : RC=00 - SUCCESSFUL
--> JCL S10EXP.. : RC=04 - SUCCESSFUL
--> JCL S11GET.. : RC=04 - SUCCESSFUL
--> JCL S12CR3.. : RC=00 - SUCCESSFUL
--> JCL S13TRP.. : RC=00 - SUCCESSFUL
--> JCL S99VALID : RC=04 - NO ERROR DETECTED - RULES FILES REINITIALIZED

*****
* INSTALLATION TEST PACK SUCCESSFUL *
*****

***** Bottom of Data *****
```

Uninstalling an environment

4

This chapter describes how to delete the execution and delivery environments.

Execution environment

To delete the libraries of the execution environment with the <RDJEXEC> prefix:

1. Copy the JCL <RDJEXEC>.JCLLIB(ZDELEXEC) outside of the <RDJEXEC> environment.
2. Delete the security of the JCL copied://* SECURITY : PUT * BELOW TO SUBMIT THE JCL//* THEN YOU LEAVE FROM THIS SAME JCLLIB BEING DELETED //*
3. Submit the JCL.
4. In TSO (choice 3.4), check that all <RDJEXEC>.* files have been deleted.

Delivery environment

You must delete the files of the delivery environment manually:

1. Open the TSO choice 3.4.
2. Select all the <RDJHOME>.* files.
3. Type "d" (delete option) in front of all the files concerned.
4. Validate.

This chapter contains additional information about the product installation.

Contents of the delivery environment (RDJHOME)

This environment is made up of the following files with the <RDJHOME> prefix.

In this table:

- The value 1/15 indicates that it is a track whose size is 1/15th of the size of the cylinder
- ** next to a file indicates that it is only used for the Rule Engine component MQSeries

| File | | | Length | | | | |
|-----------|----------------|--|--------|--------|---------------|--------|------|
| Qualif. 1 | Qualif. 2 et 3 | Label | ASCII | EBCDIC | UTF-16 | Format | Cyl |
| BFRLIB | - | PDS of the label for the printouts in French (results) | | 133 | 266 | FBA | 1 |
| BENLIB | | PSDS of the label for the printouts in English (results) | | | | | |
| CPYLIB | - | PDS of the SYSIN cards | | 80 | 80 | FB | 1 |
| DATFSQ | MVTXPL.INI | COBOL execution parameters | | 80 | 80 | FB | 1/15 |
| | STNEVT.INI | XML file of the Sentinel parameters | | 5004 | Not available | VB | 1/15 |
| | TGGCNF.INI** | MQSeries Trigger configuration file | | 5004 | | VB | 1/15 |

| File | | | Length | | | |
|--------|--------------------------|---|--------|---------------|----|---|
| DATLIB | ADMMQS** | PDS of the MQSeries administration commands | 80 | 80 | FB | 1 |
| | DEMTRV | PDS of the work requests | 255 | 255 | FB | 1 |
| | IEVASC | PDS of the events in ASCII input format | 4000 | Not available | FB | 1 |
| | IEVEBC (IEVENV in UTF16) | PDS of the events in EBCDIC input format | 4004 | 8004 | VB | 1 |
| | MVTCTX | PDS of updating files of referential / context | 5004 | 10004 | VB | 1 |
| | MVTRDJ | PDS of updating files of referential / context | 105 | 210 | FB | 1 |
| | RCHRES | PDs of the data for "resource" context | 5004 | 10004 | VB | |
| | RDJENV | PDS of the environment variables | 5004 | 5004 | VB | 1 |
| | RDJSTN | PDS of the initialization file for the Sentinel API | 5004 | Not available | VB | 1 |
| | REFEXP | PDS of the repository data | 5004 | 10004 | VB | 1 |
| | TSCDES | PDS of the data for TSC descriptors | 5004 | 10004 | VB | 1 |

| File | | | Length | | | |
|--------|--------|--|--------|-------|----|----|
| DFRLIB | FMTEDI | PDS of the | 5004 | 10004 | VB | 1 |
| DENLIB | | printout formats in French | | | | |
| | | PDS of the | | | | |
| | | printout formats in English | | | | |
| | | For UTF16 mode | | | | |
| | MVTPAR | PDS of the input events of the PARAME file in French | 105 | 105 | FB | 1 |
| | | PDS of the input events of the PARAME file in English | | | | |
| | SCRIPT | PDS of script data in French | 5004 | 10004 | VB | 1 |
| | | PDS of script data in English | | | | |
| EXELIB | - | PDSE of the executables | 0 | 0 | U | 12 |
| JCLLIB | - | PDS of the JCLs | 80 | 80 | FB | 1 |
| | SIC | PDS of the JCLs / SIC launch | 80 | 80 | FB | 1 |
| | STC** | PDS of the JCL / MQSeries start task mode | 80 | 80 | FB | 1 |
| MACLIB | - | PDS of the includes for link execution | 80 | 80 | FB | 1 |
| OBJLIB | - | PDS of the compiled objects | 80 | 80 | FB | 14 |
| PRCLIB | - | PDS of the procedures called by the JCLs | 80 | 80 | FB | 1 |

| File | | | Length | | | |
|------------------|------------|---|--------|------------------|----|------|
| RESFSQ | DEFVAR.INI | Initialization files for repository / transformation | 320 | 596 | FB | 1/15 |
| | DESCEE.INI | « | 81 | 101 | FB | 1/15 |
| | MAPCRE.INI | « | 4012 | 4012 | FB | 1/15 |
| | RGCLCR.INI | « | 97 | 125 | FB | 1/15 |
| | RGCLLE.INI | « | 102 | 102 | FB | 1/15 |
| | RGCLIN.INI | « | 5500 | 10930 | FB | 1/15 |
| | RGCLSC.INI | « | 710 | 710 | FB | 1/15 |
| | RGTRDN.INI | « | 9598 | 9598 | FB | 1/15 |
| | RGTRDT.INI | « | 9598 | Not available | FB | 1/15 |
| | RGTRIN.INI | « | 32 | 32 | FB | 1/15 |
| | TABLEL.INI | « | 431 | 794 | FB | 1/15 |
| | TABLES.INI | « | 320 | Not available | FB | 1/15 |
| | XREFRG.INI | « | 32 | 32 | FB | 1/15 |
| | RDJKEY.SYS | System file / Rule Engine protection key | 128 | 128 | FB | 1/15 |
| | TBLMON.SYS | Currency conversion rates (EuroConverter) | 100 | Not available | FB | 1/15 |
| RFRFSQ RENFSQ | ERREUR.SYS | COBOL error messages in French COBOL error messages in English | 80 | 80 | FB | 2/15 |

| File | | Length | | | | |
|--------|------------|---|------|-------|----|------|
| | ERREXP.SYS | RDJEXP system error messages in French RDJEXP system error messages in English | 5004 | 10004 | VB | 2/15 |
| | LIBXPL.SYS | RDJ operating counters labels in French RDJ operating counters labels in English | 80 | 80 | FB | 3/15 |
| | PARAME.SYS | File of the compilation parameters in French File of the compilation parameters in English | 320 | 320 | FB | 5/15 |
| | STCCTX.SYS | Encrypted system context file in French Encrypted system context file in English | 5004 | 10004 | VB | 3/15 |
| SRCLIB | COB | PDS of the user exit source in COBOL | 80 | 80 | FB | 1 |
| | COB.RDG | PDS of the user exit source in COBOL COPY clause procedure | 80 | 80 | FB | 1 |
| | COB.STR | PDS of the user exit source in COBOL COPY clause working | 80 | 80 | FB | 1 |

| File | | Length | | | | |
|------|---|--------|------|----|---|--|
| C | PDS of the user exit source in C language | 5004 | 5004 | VB | 1 | |
| H | PDS of the Header of the client exits sources | 5004 | 5004 | VB | 1 | |
| UTL | - | 80 | 80 | FB | 1 | |
| CLIB | PDS of the CLIST utilities | 80 | 80 | FB | 1 | |
| SLIB | PDS of the resolved JCLs utilities | 80 | 80 | FB | 1 | |

Note (**) Rule Engine MQSeries only

Note The PDS and resource files RES*, RFR* and REN* in the preceding table contain internal operating parameters and must not be modified.

List of created MQSeries objects (MQSeries mode)

The prefix of these objects is <QMGPRF>.

| MQSeries object name | Description |
|----------------------|-------------|
| <QMGPRF>OSEGT0 | ME output 0 |
| <MQSPRF>OSEGT1 | ME output 1 |
| <MQSPRF>OSEGT2 | ME output 2 |
| <QMGPRF>OSEGT3 | ME output 3 |
| <QMGPRF>OSEGT4 | ME output 4 |
| <QMGPRF>OSEGT5 | ME output 5 |
| <QMGPRF>OSEGT6 | ME output 6 |

| MQSeries object name | Description |
|-----------------------------|------------------------|
| <QMGRPF>OSEG7 | ME output 7 |
| <QMGRPF>OSEG8 | ME output 8 |
| <QMGRPF>OSEG9 | ME output 9 |
| <QMGRPF>OSEGTA | ME output A |
| <QMGRPF>OSEGTB | ME output B |
| <QMGRPF>OSEGTC | ME output C |
| <QMGRPF>OSEGTD | ME output D |
| <QMGRPF>OSEGTE | ME output E |
| <QMGRPF>OSEGTF | ME output F |
| <QMGRPF>OSEG TG | ME output G |
| <QMGRPF>OSE GTH | ME output H |
| <QMGRPF>OSEG TI | ME output I |
| <QMGRPF>OSEG TJ | ME output J |
| <QMGRPF>OSEG TK | ME business output |
| <QMGRPF>OSEG TZ | ME default output |
| <QMGRPF>IEVENT | CRE input |
| <QMGRPF>IEVRDI | CRE redirected I |
| <QMGRPF>IEVRDJ | CRE redirected J |
| <QMGRPF>IEVREJ | CRE rejected |
| <QMGRPF>IEVANO | CRE in anomaly |
| <QMGRPF>IEVMOD | CRE modified |
| <QMGRPF>TRACEI | CRE audit traces |
| <QMGRPF>TRACEO | ME audit traces |
| <QMGRPF>TRACES | CRE et ME audit traces |

| MQSeries object name | Description |
|--------------------------|---|
| <QMGPRF>GRPANO | Incomplete groups in file |
| <QMGPRF>CNTRUL | Rules counter |
| <QMGPRF>CNTTRA | Transformation counters |
| <QMGPRF>DETANO | Printout of the anomalies and rejected events |
| <QMGPRF>DETRED | Printout of the Input Event redirections |
| <QMGPRF>DETTRA | Printout of transformation details |
| <QMGPRF>UNREAD | CRE not processed by Rule Engine |
| <QMGPRF>LOGACC | Accounting journal |
| <QMGPRF>TGGINI | Triggering / initialization |
| <QMGPRF>TGGSEND | Triggering / sender |
| <QMGPRF>TGGSES (process) | Triggering |

Contents of the standard test deck (SIC)

To make it easier for you to run the test deck, the standard procedures of the PDS <RDJEXEC>.PRCLIB are called from the JCL "SIC test deck" contained in the specific PDS <RDJEXEC>.JCLLIB.SIC.

Overview

The JCL RDJSIC enables you to run the entire set of application JCLs of the Rule Engine test deck and the JCL of final validation S99VALID in normal mode (RESTART='N') or in restart mode (RESTART='Y').

Caution z/OS only accepts a maximum of 255 EXEC commands in a single JCL, and that forbids any direct execution from all the procedures in the RDJSIC test deck.

SIC LOG file

The file <RDJEXEC>LOGFSQ.LOGJCL.SIC enables you to display the execution report of the SIC related to the execution environment. The result of the processing of each JCL Sxx is displayed on one line and the final message of successful execution is displayed at the end of the file.

In normal mode, the LOG file is reinitialized in the JCL RDJSIC. In start up mode, the LOG file is completed for the JCL RDJSIC and the JCL Sxx re-executed.

“flags” members of the SIC

The PDS <RDJEXEC>FLGLIB.SIC contains the different flag members of the execution concerning each of the JCL Sxx excluding the JCL S99VALID.

| If the execution... | then... |
|--|---|
| That precedes the JCL Sxx does not end correctly, that is the preceding Sxx flag is not created, | No flag is created, the Sxx JCL returns a 99 code and the LOG file displays: --> JCL Sxx : RC=99 - NOT RUN - CONDITIONAL PREVIOUS JCL |
| Of the JCL Sxx ends correctly, | The Sxx flag Sxx is created, the Sxx JCL returns the expected code yy and the LOG file displays: --> JCL Sxx : RC=yy - SUCCESSFUL |
| Of the JCL Sxx ends abnormally or returns unexpected codes, | The KO flag is created, the Sxx JCL returns a 99 code and the LOG file displays: --> JCL Sxx : RC=99 - NOT SUCCESSFUL - CHECK CONDITIONS ON STEP JCLOK |
| Of Sxx is not executed or has a JCL error, | No flag is created |

In normal mode, the flags are re-initialized/deleted in the JCL RDJSIC.

SIC final validation

The final validation is done in the JCL S99VALID.

| If... | then... |
|--|--|
| The KO flag does not exist and all the flags Sxx exist | the installation is validated, the files of the repository are re-initialized, the JCL S99VALID returns a code 04 and the LOG file displays: --> JCL S99VALID : RC=04 - NO ERROR DETECTED - RULES FILES REINITIALIZEDINSTALLATION TEST PACK SUCCESSFUL |

| If... | then... |
|---|---|
| The KO flag does not exist and if one of the flags does not exist | the installation is not validated, the files of the repository are not re-initialized, the JCL S99VALID returns a code 99 and the LOG file displays for each JCL Sxx concerned: --> JCL Sxx : NOT EXECUTED OR IN ERROR then --> JCL S99VALID : RC=99 - PREVIOUS JCL(S) NOT SUCCESSFUL INSTALLATION TEST PACK NOT SUCCESSFUL |
| the KO flag exists | the installation is not validated, the files of the repository are not re-initialized, the JCL S99VALID returns a code 99 and the LOG file displays: --> JCL S99VALID : RC=99 - PREVIOUS JCL(S) NOT SUCCESSFUL INSTALLATION TEST PACK NOT SUCCESSFUL |

Details on each JCL Sxx

The JCL RDJSIC enables you to execute automatically and in sequence first the JCL Sxx (listed below) then the JCL of final validation S99VALID.

For clarity reasons, we have used the following prefixes:

<SICLIB>:<RDJHOME>.DATLIB<SLGLIB>:<RDJHOME>.DFRLIB or <RDJHOME>.DENLIB
depending on the language<DATLIB>:<RDJEXEC>.DATLIB<BRSFSQ>:<RDJEXEC>.BRSFSQ

To validate a maximum of cases, JCL Sxx are executed from several SIC files delivered and saved in SIC members in the data PDS <SICLIB>.libname of the delivery environment.

These SIC *members are copied one after another as the SIC processes the USR members of the data PDS <DATLIB>. libname of the execution environment before the standard procedure of the PDS <RDJEXEC>.PRCLIB is executed.

| Test deck JCL | Standard procedure called | Rule Engine functional description |
|----------------------|----------------------------------|---|
| S00DEF | RDJDEF | Definition of the VSAM files and the sequential files of the repository. Initialization of the VSAM files ERREUR, LIBXPL et PARAME from the delivery environment. |
| S01RAZ | RDJRAZ | Initialization of the files of the repository from the delivery environment |
| S02MNT | RDJMNT | Update of the PARAM file from the input file <SICLIB>.MVTPAR(SIC) |
| S03RCH | RDJRCH | Update of the files of the repository / context from the input file <SICLIB>.MVTCTX(SIC) |

| Test deck JCL | Standard procedure called | Rule Engine functional description |
|---------------|---------------------------|--|
| S04MAJ01 | RDJMAJ | <p>Update and compilation of the files of the repository / transformation from the input file <SICLIB>.MVTRDJ(SIC01) that contains the update induced errors:</p> <p>The ITR100 program (uncompiled rules update) terminates abnormally (RC=08)</p> <p>An anomaly report is printed</p> <p>The save/restore process of the rules is executed</p> |
| S05MAJ02 | RDJMAJ | <p>Update and compilation of the files of the repository / transformation from the input file <SICLIB>.MVTRDJ(SIC02) that contains the input forms movements deliberately in anomaly:</p> <p>The ITR100 program (uncompiled rules update) terminates normally (RC=00)</p> <p>The anomaly report is empty</p> <p>The transformation phase (ITR400) is executed and detects errors (RC=08)</p> <p>A report of transformation errors is printed</p> <p>The rules save/restore process is executed</p> |
| S06MAJ03 | RDJMAJ | <p>Update and compilation of the files of the repository / transformation from the input file <SICLIB>.MVTRDJ(SIC03) that contains correct input forms movements:</p> <p>The ITR100 program (uncompiled rules update) terminates normally (RC=00)</p> <p>The anomaly report is empty</p> <p>As all movements are correct, compilation (ITR400) starts and terminates normally (RC=00)</p> <p>The save/restore process of the rules is not executed</p> <p>The files rules and compiled tables are then ready for Input-Events transformation</p> |
| S07EDR | RDJEDR | <p>Printout of the repository / transformation from the « job request » input form file <SICLIB>.DEMTRV(ALL)*.</p> |
| S08CRE | RDJCRE | <p>Conversion of an Input-Event file <SICLIB>.IEVASC(SIC) from ASCII to EBCDIC in the file <DATLIB>.IEVEBC(CRE).</p> |
| S09CLN* | RDJCLN* | <p>List of all the MQSeries files <QMGPFR>.*</p> |
| S09PUT* | RDJPUT* | <p>Conversion of an Input-Event file <DATLIB>.IEVEBC(CRE) in the file <QMGPFR>IEVENT.</p> |

| Test deck JCL | Standard procedure called | Rule Engine functional description |
|---------------|---------------------------|--|
| S10EXP | RDJEXP | <p>File mode Translation of the Input Events in file <DATLIB>.IEVEBC (CRE) to Output segments in file <BRSFSQ>.OSEGTx.SEQ from files <SLGLIB>.SCRIPT(SIC) and <SLGLIB>.TSCDES(SIC).</p> <p>And comparison of the generated result files in PDS <RDJEXEC>.EDILIB to those delivered in the delivery environment.</p> <p>MQSeries mode Transformation of the Input Events of the queue <QMGRPF>IEVENT in the Output Events <QMGRPF>OSEGTx from the files <SLGLIB>.SCRIPT(SIC) et <SLGLIB>.TSCDES(SIC).</p> |
| S09GET* | RDJGET* | <p>Conversion/list of Input Event queues <QMGRPF>OSEGTx in the Input Event files <BRSFSQ>.OSEGTx.SEQ.</p> <p>And comparison of the generated result files in the PDS <RDJEXEC>.EDILIB to those delivered in the delivery environment.</p> |
| S12CRS | RDJCRS | <p>Conversion of an Input Event file <DATLIB>.IEVEBC(CRE) from EBCDIC into ASCII in the file <DATLIB>.IEVASC(CRS).</p> |
| S13TRF | RDJTRF | <p>Extraction of selected elements in the repository / transformation in an execution environment MVTRDJ and writing to a file input form format <SICLIB>.DEMTRV(ALL)**.</p> |

Note (*)MQSeries mode only.

Note (**) This file (ALL) must be saved as it enables you to list and extract all the elements of the repository / transformation: rules, tables, formats and variables.

The member files USR are used as input files to the standard JCL of PDS <RDJEXEC>. JCLLIB. You can then submit these standard JCLs with the files USR that contain the SIC data – particularly the contents of the last SIC03 file copied in the PDS <DATLIB>.MVTRDJ.