

T . . . Systems . . .

rvsEVO

Version 6.02

Operator Manual

rvsEVO
Version 6.02
Operator Manual

© 2020 by T-Systems International GmbH
Holzhauser Straße 4 - 8
13509 Berlin

This manual is protected by copyright. All rights reserved. No part of this book may be used or reproduced in any form or by any means including photocopies, microfilm or any other means or stored in a database or retrieval system without obtaining prior permission from T-Systems. Rights are also reserved as far as lectures, radio and television is concerned.

We reserve the right to make changes to the content of this manual without giving prior notice. T-Systems is not liable for technical or printing errors or defects in this manual. Moreover, T-Systems shall not be liable for damage which is directly or indirectly caused by delivery, performance and use of this material.

Content

1	Introduction	7
1.1	Representation Means	7
2	Technical Pre-Conditions	9
2.1	System Requirement.	9
2.2	Networks	9
2.2.1	ISDN System Requirement.	9
2.2.2	XOT Systems Requirement.	11
3	Migration from rvs[®] portable to rvsEVO	13
3.1	Migration Procedur	13
4	Fall Back after Update Installation.	15
4.1	Fall Back after Update of rvsEVO Version 5.xx to Version 6.xx	15
4.2	Fall Back after Update of rvsEVO with Derby Database.	15
4.3	Fall Back after Update of rvsEVO with Oracle or MS SQL Database	15
5	Server Farm of rvsEVO	17
5.1	System Requirements.	17
5.2	Architecture of a Server Farm	17
5.3	Installation of an rvsEVO Server Farm	19
5.4	Setting up of a Server Farm	25
5.5	Update of rvsEVO Server Farm	26
5.6	Server Farm with rvs [®] OFTP Proxy Implementation.	26
6	Multitenancy Functionality of rvsEVO.	27
6.1	System Requirements.	27
6.2	Installation of rvsRuntime	27
6.2.1	Obtaining a License.	27
6.2.2	New Installation of rvsRuntime	27
6.2.3	Update Installation.	29
6.2.4	rvsRuntime Installation Folder.	30
6.3	Graphical User Interface of rvsRuntime	31
6.3.1	Starting of rvsRuntime GUI (RCP-GUI MI)	32
6.4	Managing of rvsEVO Tenants.	32
6.4.1	Creation of rvsEVO Tenant.	34
6.4.2	Distribution of Licensekey or Station Configuration	38
6.4.3	Deletion of rvsEVO Tenant	40
7	Customizing rvsEVO	43
7.1	Customizing rvsEVO via rvs-system.properties File	43
7.2	Customizing rvsEVO via rvsEVOClient.prefs File	48
7.3	Customizing the Log Files.	51
7.3.1	Customizing the Monitor and rvs Log Files via rvsLogger.xml File	51
7.3.2	Customizing the rvsClient Log File via rvsClientLogger.xml File.	51
7.3.3	Customizing the rvsClientRCP Log File via rvsClientRCPLogger.xml	52
7.4	Customizing RMI Connection over TLS	52
7.4.1	Setting up of an RMI Connection via TLS Protocol.	52
7.5	Pass Parameters to Jobstart Process.	53
7.6	Conversion Tables	56
7.6.1	Code Tables Offered by rvsEVO.	56
7.6.2	How to Add Your Own Conversion Table	56
7.7	Connection Pooling	57
7.8	Operating with OFTP Routing Rules.	58

8	Maintenance of rvsEVO	61
8.1	rvsEVO Installation Folder	61
8.2	Maintenance of Directories	62
8.3	Maintenance of rvsEVO Database	63
8.3.1	Deletion of Job Data from Database	63
8.3.2	Deletion of Monitor Log Data from Database	63
8.3.3	Deletion of Jobs and Monitor Log Data from the Database	64
9	ODETTE Protocol	67
9.1	RFC 5024: OFTP Commands for OFTP File Transfer	67
9.1.1	Message Flow of Odette Session	68
10	Monitoring of rvsEVO	71
10.1	What Gets Logged	71
10.2	Log Files	71
10.2.1	monitor.log	71
10.2.2	rvs.log	74
10.2.3	rvsClient.log	75
10.2.4	rvsClientRCP.log	75
10.3	Tracing	75
10.3.1	Odette Tracing	75
10.3.2	TLS Tracing	77
10.3.3	API Tracing	77
11	Error Situations of rvsEVO	79
11.1	Debugging the Installer	79
11.2	Failure of Installation	79
11.2.1	Wrong Permission of DB Scripts	79
11.2.2	Exception during Installation of rvsEVO and Executing DB Script „schema-create.sql“	79
11.3	Aborting of Installation on Unix Systems	80
11.4	Failure of rvsEVO Server Start	81
11.4.1	Failure of rvsEVO Server Start with Derby Database	81
11.5	Failure of Listener Start	82
11.5.1	Port is already in use	82
11.6	Connection Failure	83
11.6.1	Connection to Proxy Bastion Fails	83
11.6.2	Examples with outgoing call, TCP Connection	85
11.6.3	Examples with outgoing call, TLS Connection	87
11.7	Failure during Build-up of OFTP Session to Neighbour Station	88
11.8	Failure of File Transmission	90
11.8.1	SFNA and EFNA	90
11.8.2	SFNA: REAST=15 „Cipher suite not supported“	90
11.8.3	Error Messages via NERP	91
11.8.4	Outgoing Call, EERP is Missing	92
11.9	File Transmission: Pre- or Post-Processing Fails	92
11.9.1	Key or Certificate is Missing	92
11.9.2	Jobstart After Receive Fails on Receiver Side	93
11.9.3	Failure of OFTP2 Conversion with Cluster Functionality on Linux and UNIX Systems	94
11.10	Failure of Connection from rvsEVO Client to rvsEVO Server	94
11.10.1	Failure of Connection to Server on Linux Systems	94
11.10.2	Failure of Connection to Server with rvsEVO Server Farm	96
11.11	Failure during Launching Scripts in „tools“ Directory	97

11.12	Failure of rvsRuntime Commands.....	97
11.12.1	Failure of starting a tenant.....	97
12	rvsEVO Support.....	99

1 Introduction

The rvsEVO Operator Manual is an addition to the rvsEVO User Manual. It is meant for operators and administrators.

This manual contains information that goes beyond basic usage information like first installation or sending and receiving files. It provides:

- detailed technical information
- detailed description of
 - Migration from rvs[®] portable to rvsEVO
 - Fall back scenario
 - Application cluster functionality
 - Multitenancy functionality
 - Customizing rvsEVO
 - Maintenance of rvsEVO
 - Monitoring
- Failure solutions.

For information of topics relating to the normal usage please have a look in the rvsEVO User Manual.

1.1 Representation Means

This chapter describes the typographic conventions used in this manual and explains the meaning of specially highlighted expressions.

Typographic Conventions

- Instructions begin with a bullet.
- Other lists begin with the en dash.

Character styles

<code>Courier</code>	Commands, menu commands, file names, path names, programs, examples, scripts, options, qualifiers, data sets, fields, modes, window names, dialog boxes and statuses
BOLD and UPPER CASE	Parameters, environment variables, variables
"Inverted comma"	Links to other manuals, sections and chapters, literature
Bold	Important terms, names of operating systems, proper names, buttons, function keys

Installation Directory:

\$RVS_HOME As user directories are found on different locations for the different operating systems we use the variable **\$RVS_HOME** in this manual. Default values are:

- C:\Programs\rvsEVO for **Windows**.

Substitute the variable with your correct path.

2 Technical Pre-Conditions

This chapter describes the system requirements and other information about technical pre-conditions.

2.1 System Requirement

To successfully operate rvsEVO you need the following software:

- Software
- Operating system: **Windows 8.1 / Windows 10 / Windows Server 2012 R2 / Windows Server 2016 / Windows Server 2019** or **UNIX** (AIX, Solaris/SunOS, HPUX) as well as **Linux** und **zLinux**.
 - X11-Server (Linux or UNIX operating systems): for graphics mode of rvsEVO installer and for Swing GUI.
 - X11-Server and GTK-2.4 (Linux): for RCP GUI.
 - Java runtime environment Version (JRE) 1.8_XX or higher.

Hints:

- With Windows and Linux systems the installer distributes the required Java runtime environment version into the rvsEVO installation folder.
- With UNIX systems you can select an installed JRE during the installation.
- With AIX you need the following java version "1.8.0" or higher:


```
Java(TM) SE Runtime Environment (build pap6480sr4fp6-
20170518_02(SR4 FP6))
IBM J9 VM (build 2.8, JRE 1.8.0 AIX ppc64-64 Compressed Refer-
ences 20170516_348050 (JIT enabled, AOT enabled)
J9VM - R28_20170516_1905_B348050
JIT - tr.r14.java_20170516_348050
GC - R28_20170516_1905_B348050_CMPRSS
J9CL - 20170516_348050)
JCL - 20170516_01 based on Oracle jdk8u131-b11
```
- Initially, you need at least 400 MB free space on your hard disk. Depending on the amount of usage, the retention period for old entries, and the time between database cleanups, the space requirement may be considerably larger.

2.2 Networks

Communication line based on TCP/IP, TLS, ISDN, XOT, Proxy TCP/IP und Proxy TLS.

Hint: ISDN is only available with Windows32 installer.

2.2.1 ISDN System Requirement

ISDN is only for Windows platform available. For Unix platforms XOT should be used.

If you want to use rvsEVO to exchange data via the **ISDN network** you need the following equipment:

- ISDN type telephone connection with s₀ bus having at least two data channels (B-channel) and one control channel (D-channel)
- ISDN adapter
- CAPI 2.0 driver software for operation of the ISDN card under Windows.

How to install the ISDN interface:

- Install the ISDN card into your computer and attach the ISDN card to the ISDN connection.
- Make sure that the ISDN card works properly.

Note: Many card manufacturers supply suitable software for a self-test, such as a call from one data channel to another. Restart your computer after installation and test in order to make the CAPI 2.0 driver available for other applications.

External ISDN
router

If you use an external ISDN router with remote CAPI Interface (e.g. BinTec Brick), you do not need an internal ISDN card. In this configuration, several applications can share the same Brick router. The Brick router supports the "Remote CAPI" interface. This means that every computer in your LAN uses the Brick router as if it were a local ISDN card in the computer.

Note: T-Systems has successfully tested the following ISDN devices for use with rvsEVO:

ISDN Devices

Device	Manufacturer
EICON DIVA Server BRI-2M	Eicon
EICON DIVA Pro 2.02 PCI	Eicon
EICON DIVA 2.0	Eicon
Longshine LCS-8051A	Longshine, http://www.longshine.de
BIANCA/BRICK-XS, -XM, -XL2; X4000, X4100, X4300, R1200, R4100 und R4300	Funkwerk EC

rvs[®] Service Support Center will provide you with the current list of tested devices. See chapter 12 "rvsEVO Support" for contact.

2.2.2 XOT Systems Requirement

On demand we shall send you a separate document with examples how to configure CISCO or BinTec router for XOT.

To use the XOT functionality in rvsEVO you need a TCP/IP connection to the XOT-capable router e.g. CISCO 801, CISCO 2600 or BINTEC X4300, R1200, R4100 and R4300. (CISCO: <http://www.cisco.de>; BINTEC).

Please see the rvsEVO User Manual for Information about new installation and update installation of rvsEVO.

3 Migration from rvs[®] portable to rvsEVO

This chapter describes the migration from rvs[®] portable to rvsEVO.

Please have a look at rvsEVO User Manual for information about first installation and usual update installation.

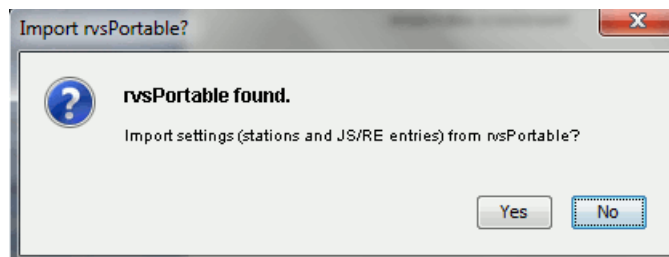
Requirements For migration of settings from rvs[®] portable you have to consider the following requirements:

- an executable rvs[®] portable must be installed on your system
- migration should only be started if the absence of communication has been assured (no file transmission and no encryption/compression)
- The system variable **RVSENV** has to be set for logged on user. Set **RVSENV** by choosing Start -> Control Panel -> System -> Advanced -> Environment Variables.

3.1 Migration Procedur

The following steps need to be done for migration from rvs[®] portable to rvsEVO:

- Install rvsEVO on the same machine where rvsXP / rvsX is installed like described in the rvsEVO User Manual.
- The installer searches through your system for an existing rvs[®] portable. If you are asked whether you wish to import the settings from rvs[®] portable, press the **Yes** button.



- Only with encrypted transmission: import the public keys of your partners.
- Create the users which were defined in rvs[®] portable. They will not be created automatically.

Note: Use the following command tools in \$RVS_HOME/tools directory for importing the settings into an existing rvsEVO:

```
$RVS_HOME/bin/portable2codetable.bat
$RVS_HOME/bin/portable2jobstart.bat and
$RVS_HOME/bin/portable2stationlist.bat
```

portable2codetable Use this tool for converting a codetable from rvs[®] portable (rtc format) to xml format.

Syntax:

```
portable2codetable <rtc file> <xml file>
```

`rtc file` is the name of the codetable in rvs[®] portable. `xml file` is the name of the codetable in rvsEVO.

portable2jobstart Use this tool for creating a jobstart list (xml file) from rvs[®] portable. rvs[®] portable must be launched.

Syntax:

```
portable2jobstart <file>
```

`file` is the xml file name. The file can be imported into the rvsEVO database via `$RVS_HOME/tools/importJobstarters` script.

portable2stationlist Use this tool for creating a stationlist (xml file) from rvs[®] portable. rvs[®] portable must be launched.

Syntax:

```
portable2stationlist <file>
```

`file` is the xml file name. The file can be imported into the rvsEVO database via `$RVS_HOME/tools/importStationlist` script.

Important: rvsEVO supports only tools which are provided by rvsEVO and no other tools which are supported by rvsX / rvsXP. Please see rvsEVO User Manual, chapter 'Command Line Tools' and 'Database Maintenance' for getting information about the supported tools.

Communication with rvs400:

In case of transmitting files bigger than 2 GB with AS/400 machine, EFID gets a negative value. Thus an EFNA is sent to the sender of the file and the transmission is restarted.

Set the denifinition

`rvs_evo.receive_job.ignore_illegal_efid_values=true` in `$RVS_HOME/conf/rvs-system.properties` file to circumvent this problem.

(In rvs[®] portable this problem is fixed via `RVSNOSIZECHECK` environment variable.)

4 Fall Back after Update Installation

This chapter describes a fall back scenario. In case of a faulty update you can fall back to your previous rvsEVO version. The procedure depends on the rvsEVO version and the database.

Please have a look at rvsEVO User Manual for information about first installation and usual update installation.

4.1 Fall Back after Update of rvsEVO Version 5.xx to Version 6.xx

The installer is saving the installation directory in a directory named **\$RVS_HOME** with added time stamp (Format: `$RVS_HOME.backu-pYYYY-MM-DD_hh-mm-ss`).

Fall Back:

For falling back to your origin installation delete or rename the updated rvsEVO and rename the saved installation directory to **\$RVS_HOME**.

4.2 Fall Back after Update of rvsEVO with Derby Database

Preparation Prior to Update:

Save a copy of your rvsEVO installation directory.

Fall Back:

- Delete or rename your updated rvsEVO installation.
- Rename your saved rvsEVO installation directory into the origin **\$RVS_HOME** directory name.
- Check whether rvsEVO is correctly installed as Windows service and the entry in your **\$RVS_HOME** variable.

4.3 Fall Back after Update of rvsEVO with Oracle or MS SQL Database

Preparation Prior to Update:

- Back up the rvsEVO data. Please see the rvsEVO User Manual chapter 'Backing Up and Recovering rvsEVO Data' for more information about the back up procedure.
- Save a copy of your rvsEVO installation.

Fall Back:

- Delete or rename your updated rvsEVO installation.
- Rename your saved rvsEVO installation directory into the origin **\$RVS_HOME** directory name.
- Recover the rvsEVO data from the back up file. Please see the rvsEVO User Manual chapter 'Backing Up and Recovering rvsEVO Data' for more information about the recover procedure.

- Check whether rvsEVO is correctly installed as Windows service and the entry in your **\$RVS_HOME** variable.

Hints:

- During the update procedure the rvsEVO installer saves a back up of the rvsEVO data in the users temp directory. The directory with backup files is called 'rvsdata' plus time stamp [rvsdata.YYYY-MM-DD-hh-mm-ss]. Windows:
C:\Users\user_name\AppData\Local\Temp.
- In case of a faulty update installation the rvsEVO installer saves an update log file in the users temp directory.
- For each rvsEVO installation/update installation a log file is also saved in **\$RVS_HOME/UninstallerData** directory.

5 Server Farm of rvsEVO

rvsEVO provides the application cluster functionality with a server farm.

See the rvsEVO User Manual Chapter „**Server Farm of rvsEVO**“ for information about provided functionalities and benefits of application cluster capability.

5.1 System Requirements

System require-
ments

To successfully operate the server farm you need the following components:

- Operating systems: Windows, Unix or Linux. The same operation system should be installed on all nodes.
- Database: Oracle 12 and higher, MS SQL
- IP load balancer
- network file System/network drive (e.g. NFS or CIFS)

Hint: The system clock of all nodes of the server farm has to be synchronized via NTP.

5.2 Architecture of a Server Farm

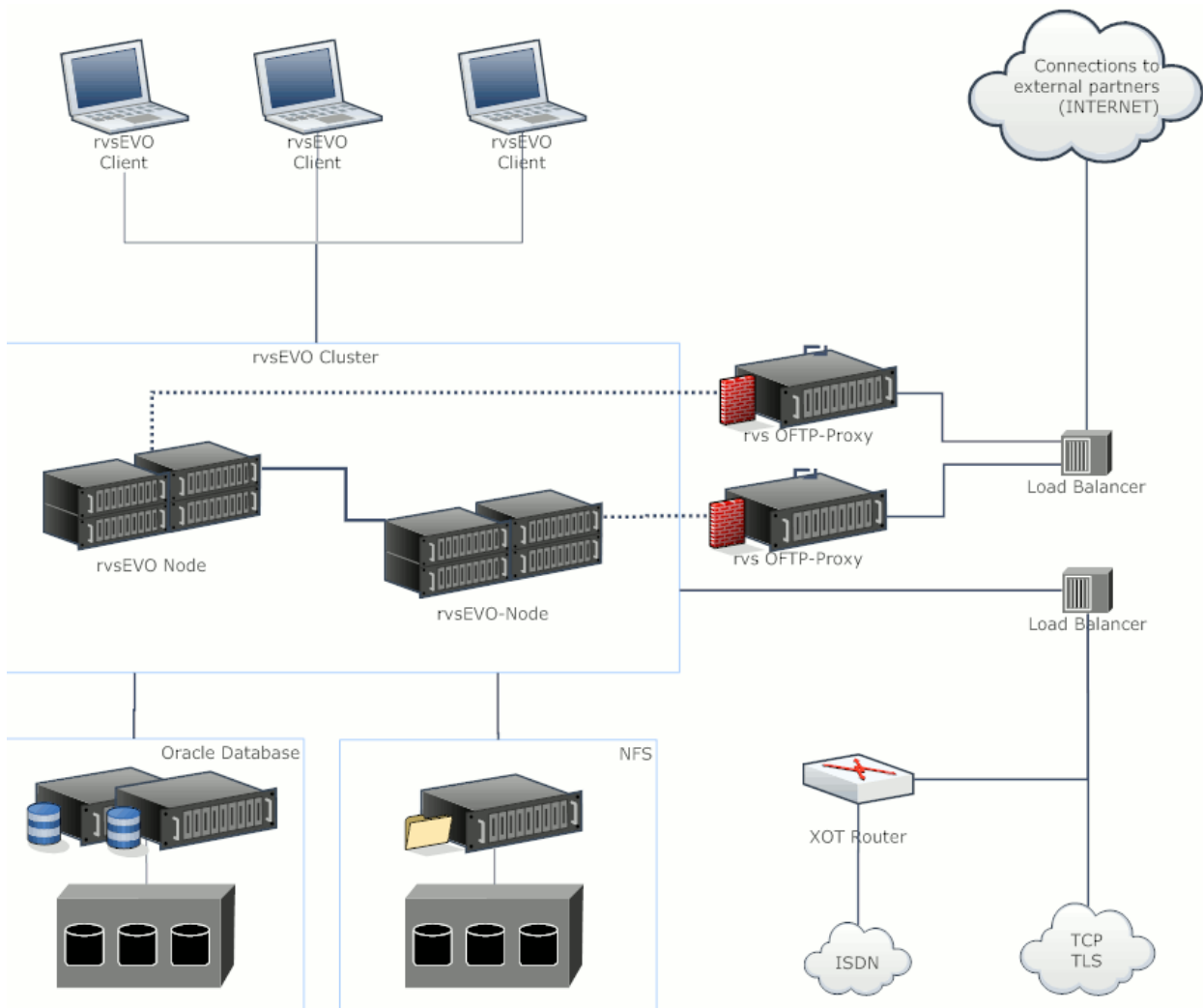
System architecture

Each rvsEVO node is equipped with all functions and needs access to all resources in the cluster systems (resource sharing).

An rvsEVO cluster comprises the following components:

- several rvsEVO nodes (rvsEVO server with rvsEVO client and Oracle client).
- Network file system

The illustration below shows the system architecture of the operation with application cluster functionality.



Note:

If the network share for data storage is supplied by **CIFS** (Windows: SMB / Linux: SAMBA) **opportunistic locking** must be disabled on the file server.

Proceed as follows to disable **opportunistic locking**:

Unix/Linux Systems: Add in Samba Config for the appropriate share:

```
oplocks = no
```

Windows Server Systems: With Windows Server systems additional the **SMB feature level** is to set down to version 1. This will have effect on the performance.

<https://support.microsoft.com/en-us/kb/296264>

<https://support.microsoft.com/en-us/kb/2696547>)

Perform the following settings via PowerShell:

Windows Server
2012 upwards:

- `Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" EnableOplocks -Type DWORD -Value 0 -Force`
- `Set-SmbServerConfiguration -EnableSMB1Protocol $true`
- `Set-SmbServerConfiguration -EnableSMB2Protocol $false`
- `Restart-Service server -PassThru -Force`
or restart your system.

5.3 Installation of an rvsEVO Server Farm

Preparation Please observe the pre-conditions for using an Oracle or MS SQL database (see the „rvsEVO User Manual“ chapter „rvsEVO Database“).

Define an alias name for the local IP address of rvsEVO server. Alias name is a pseudonym for the IP address and must be the same on all nodes. Assign an IP address to the alias name in `\etc\hosts` file (on Windows systems in `C:\Windows\System32\drivers\etc` directory).

Example:

```
# rvsEVO alias
102.54.94.97      EVOCLUSTER
```

You also need an alias name for setting up of local receiver. If the IP address of the local receiver is identical to the IP address of rvsEVO server you should use the same alias name.

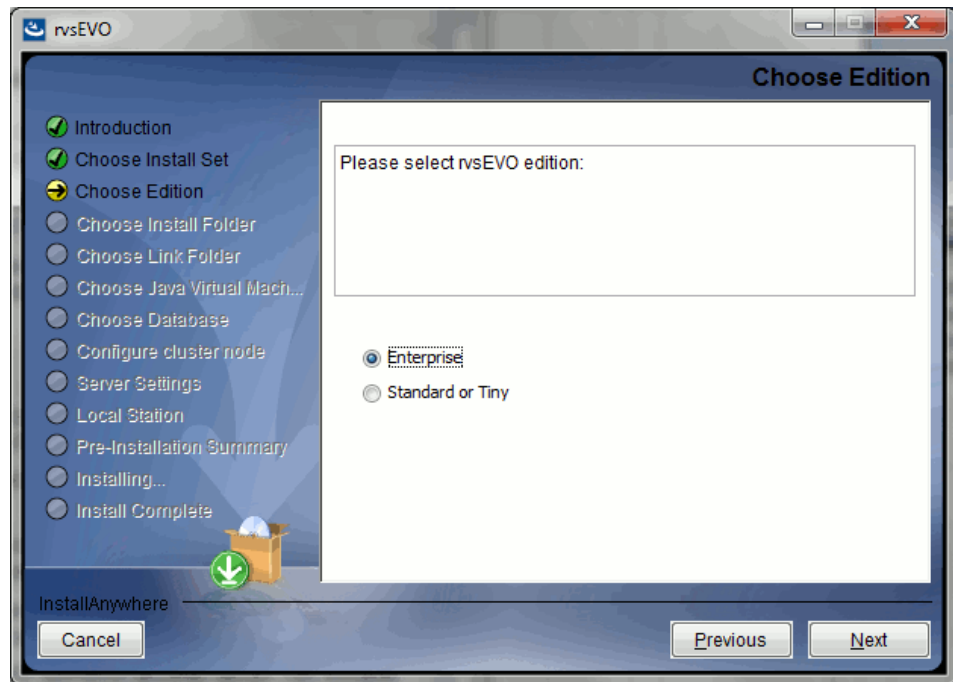
Hint: With rvsEVO version 6.02.02 and higher, alternatively you can use `host-alias.properties` file in `$RVS_HOME/conf` directory:

After successful completed installation, activate element `RVSA-LIAS=rvsevohost` in `host-alias.properties` file and replace `RVSALIAS` with your alias name and `rvsevohost` with the host name or IP address of your machine.

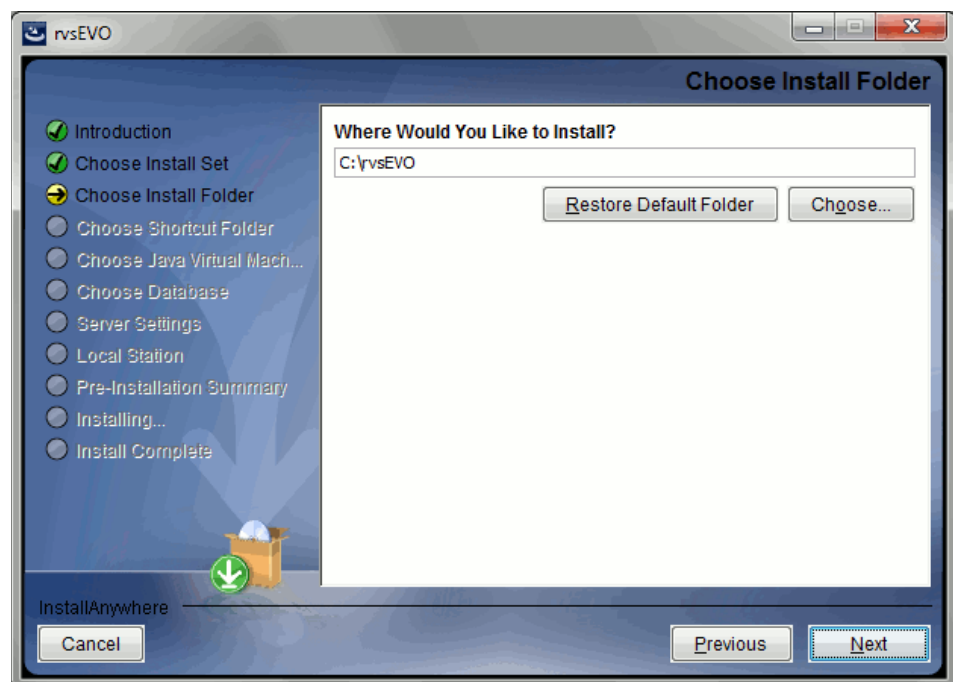
Replace the value of `<entry key="client.RMIServiceHost" value=<Alias name>/>` element in `$RVS_HOME/con/rvsEvoClient.prefs` file with the IP address or hostname of your machine.

Installation Installation of a server node runs nearly analog to an installation of a single instance (see the rvsEVO User Manual, chapter 'New Installation of rvsEVO'). Please notice the following points:

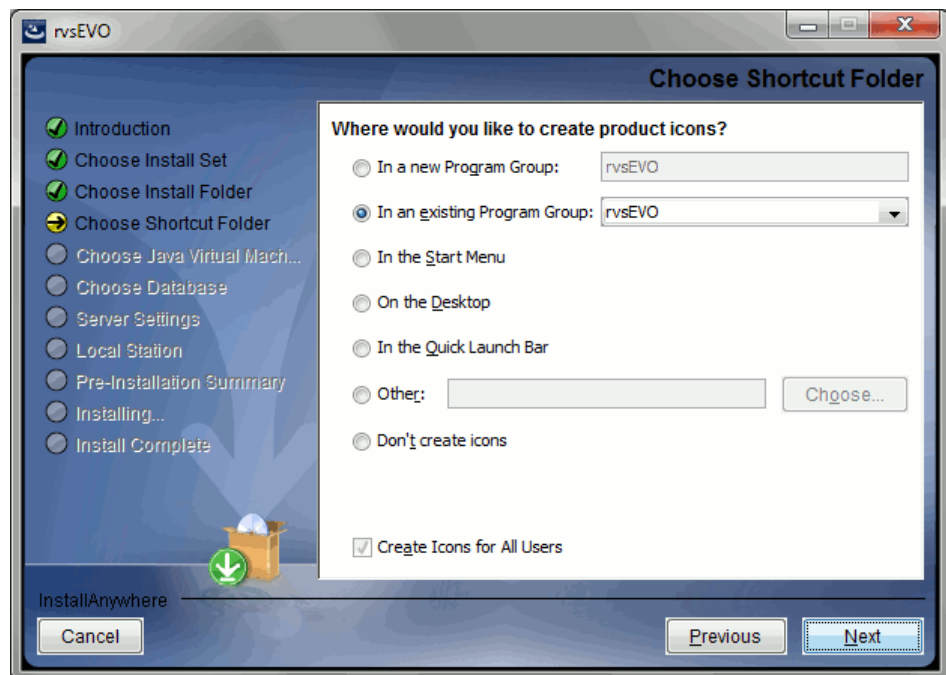
- Install the Enterprise Edition of rvsEVO. The Standard or Tiny Edition does not support application cluster functionality.



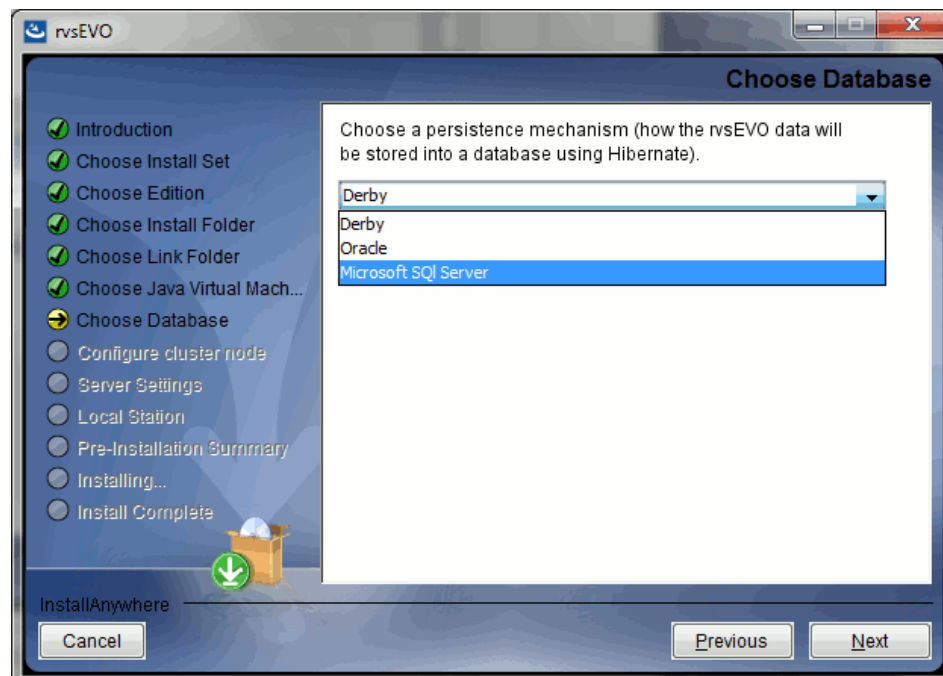
- Use the same installation path for all rvsEVO nodes. This is required because the directory paths are defined via rvsEVO database.



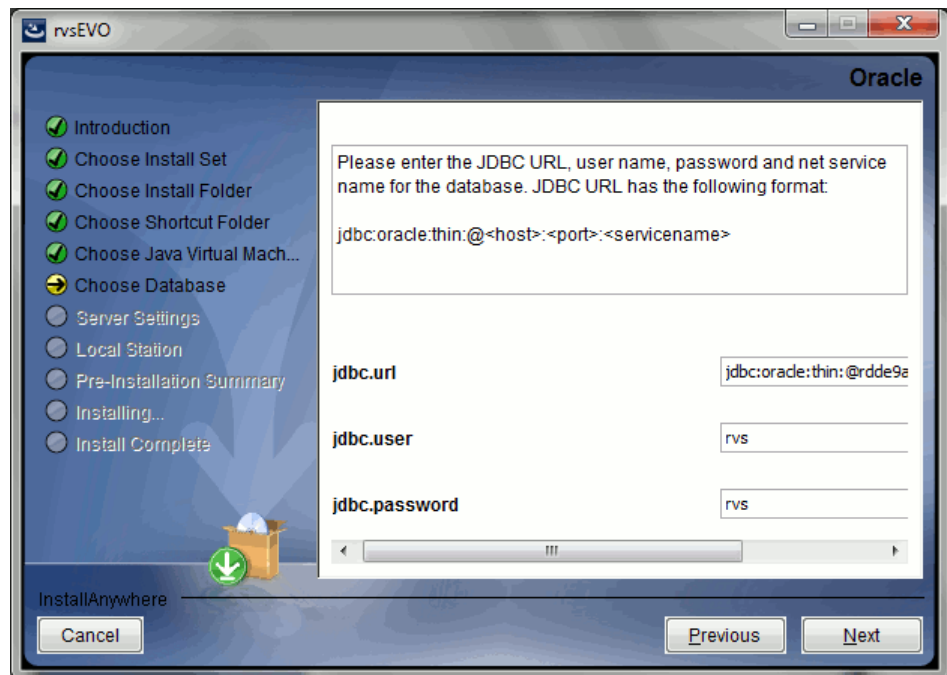
- Define the program group to install rvsEVO icons like described in the rvsEVO User Manual.



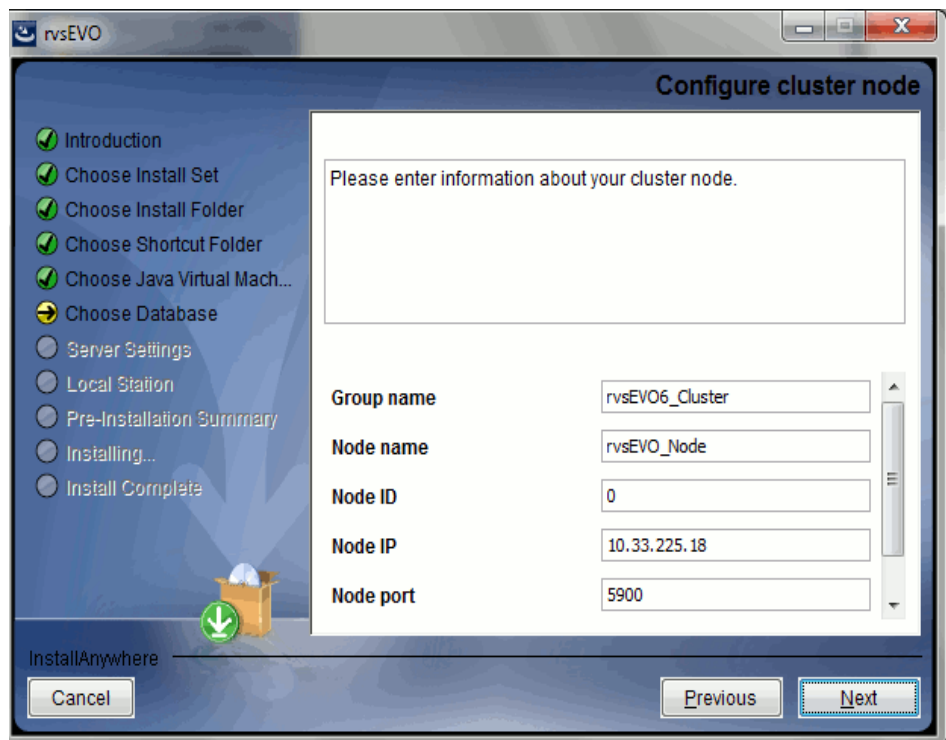
- Use an external database (Oracle or MS SQL)



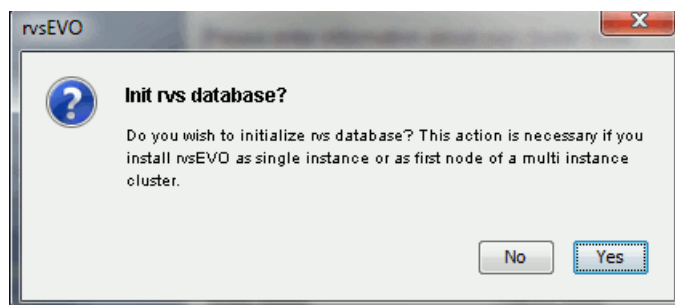
- Set the Oracle or MS SQL connection parameters like described in the rvsEVO User Manual.



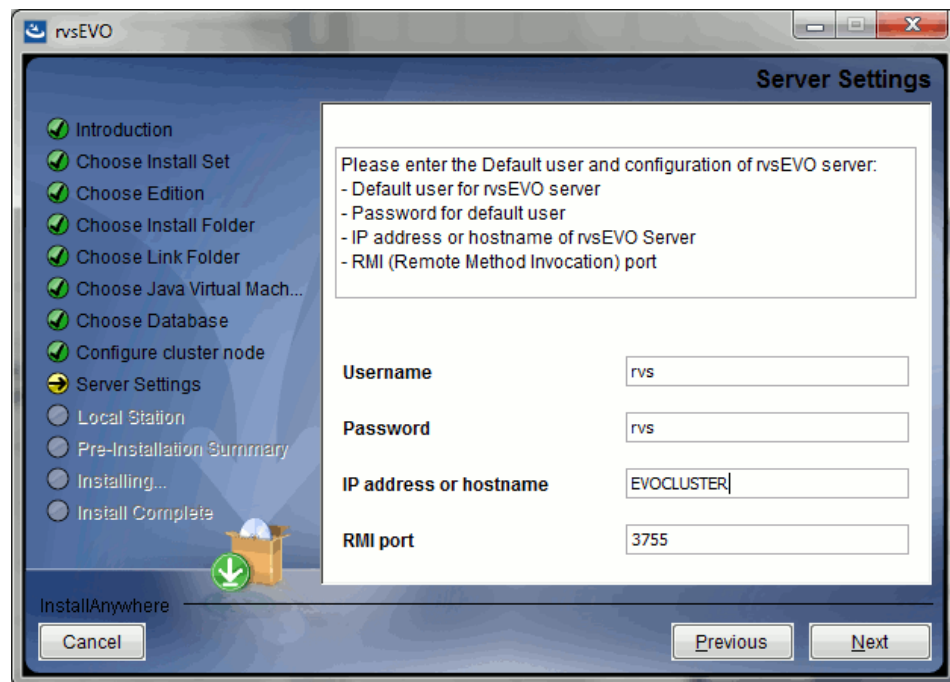
- Define the group name with installation of the first node in window **Configure cluster node**. Use the same group name for each node within the server farm.
Node Name and **Node ID** must be unique values in your rvsEVO system.
Node IP is the IP address of the currently installed node.
 Insert the IP addresses of the other nodes within the server farm via **Nodes** parameter. Multiple entries are separated by commas.



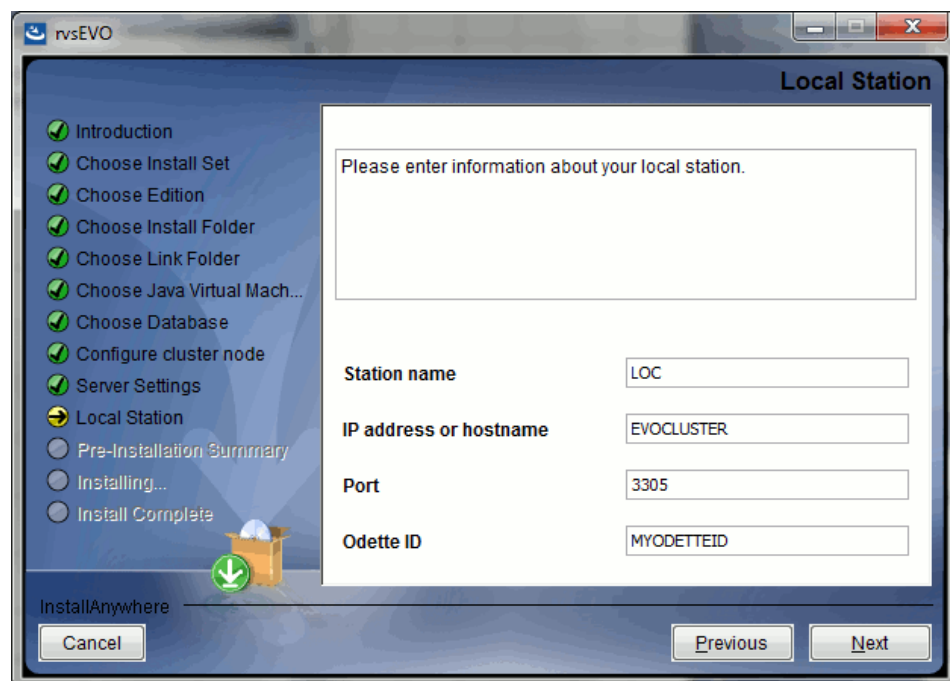
- Initialize the database only with the installation of the first node.



- Enter the server settings like described in the rvsEVO User Manual. Insert in **IP address or hostname** parameter the same alias name (see above) on all rvsEVO nodes and use the same RMI port on all nodes..



- Setting up of the local receiver is only required during installation of the first node. Insert the alias name for IP address or host name.



After installation:

- Save the directory `$RVS_HOME/files` and its subdirectories on the network file system. The path of the subdirectories `/inbox`, `/outbox` und `/temp` are to be set up in **DIRECTORY** parameter

group (see rvsEVO User Manual chapter 'Customizing the Global rvsEVO Parameters').

- Create directories for jobstart scripts and Observer on the network file system and provide the read and write access to all nodes.
- In case of using `host-alias.properties` file, edit the `host-alias.properties` file and `rvsEvoClient.prefs` file like described above.

5.4 Setting up of a Server Farm

During installation the cluster node has been configured. You can set up the configuration in `$RVS_HOME/conf/cluster.properties` file. This is required after adding or removing nodes. In this case on each node of the server farm the IP addresses of the added or removed nodes has to be added or deleted in `cluster.nodes` definition.

Example of `cluster.properties` file:

```
cluster.group.name=rvsEVO6_Cluster
cluster.node.name=rvsEVO_Node
cluster.node.id=0
cluster.node.ip=10.177.191.220
cluster.node.port=5900
cluster.nodes=10.177.191.1,10.177.191.2,10.177.191.3
```

Definitions of Cluster.Properties

Definition	Description
cluster.group.name	Name of the server farm (has to be identical on each node)
cluster.node.name	Name of the node
cluster.node.id	Unique node identifier. Possible values: 0-9 e.g. first node: <code>cluster.node.id = 0</code> ; second node: <code>cluster.node.id = 1</code> ; etc. You can also assign value 1 to the first node; value 2 to the second node, etc. (Default is 0). This value is part of the job ID (fourth last position).
cluster.node.ip	IP address of the node
cluster.node.port	Port of the node (default: 5900)
cluster.node	IP addresses of the other nodes within the server farm, separated by commas

5.5 Update of rvsEVO Server Farm

Update of a server node runs nearly analog to an update of a single instance (see rvsEVO User Manual chapter 'rvsEVO Update Installation').

Please notice the following points:

- Each node of the server farm has to be updated separately.
- Before updating you do not need to stop all servers of the farm but only the server to be updated.
- Update the database structure with updating the last node.
- During the update installation a `$RVS_HOME/conf/host-alias.properties.new` file is added. Please take over the configurations of your old `host-alias.properties` file to the new one and rename `host-alias.properties.new` in `host-alias.properties`.
- Do not start the rvsEVO server before the last node was updated.

5.6 Server Farm with rvs[®] OFTP Proxy Implementation

If rvs[®] OFTP is implemented in the rvsEVO server farm you have to use the same alias name for the IP addresses of Proxy servers and Proxy listeners on each node. Alias name is a pseudonym of IP address. The alias name of the Proxy listeners must also be assigned on the Proxy hosts in `etc/hosts`!

Hint: With version 6.02.02 and higher, alternatively you can use `host-alias.properties` file in `conf` folder of rvs[®] OFTP Proxy installation directory and **\$RVS_HOME** directory:

After successful completed installation of rvs[®] OFTP Proxy, activate element `PROXYALIAS=rvsproxyhost` in `host-alias.properties` file in `conf` folder of rvs[®] OFTP Proxy installation directory and replace `PROXYALIAS` with your alias name and `rvsproxyhost` with the host name or IP address of the Proxy listener.

Subsequently, add one element each for the alias name of the Proxy bastion and the RMI Sender-IP to `host-alias.properties` file in `conf` folder of **\$RVS_HOME** directory (Format: `PROXYALIAS=rvsproxyhost`).

For encrypting the line between rvs[®] OFTP Proxy and rvsEVO via TLS you need to use the same keystore for each rvs[®] OFTP Proxy installation and the same truststore for each rvsEVO node.

Hint for Update-Installation

During the update installation a `host-alias.properties.new` file is added to `conf` directory. Please take over the configurations of your old `host-alias.properties` file to the new one and rename `host-alias.properties.new` in `host-alias.properties`.

6 Multitenancy Functionality of rvsEVO

From version 6.02 upward rvsEVO provides the multitenancy functionality via rvsRuntime.

See the rvsEVO User Manual Chapter "Multitenance Functionality of rvsEVO" for information about provided functionalities and benefits of multitenancy capability.

6.1 System Requirements

System requirements

To successfully operate rvsRuntime you need the same system requirements as for rvsEVO. Please see the rvsEVO User Manual, chapter "Installation" / "System Requirements".

6.2 Installation of rvsRuntime

Installation Directory

As user directories are found on different locations for the different operating systems we use the variable **\$RVS_RUNTIME_HOME** in this manual. Default value for Windows is `C:\rvsEVO`.

Substitute the variable with your correct path.

6.2.1 Obtaining a License

The rvsEVO Server needs to be enabled with a license component to be accepted as a tenant of rvsRuntime.

rvs[®] After Sales Service

Please contact the rvs[®] Service Support Center.

From Germany for free, telephone: 0800 664 77 45

From other contries, telephone: +49 375 606 19 902

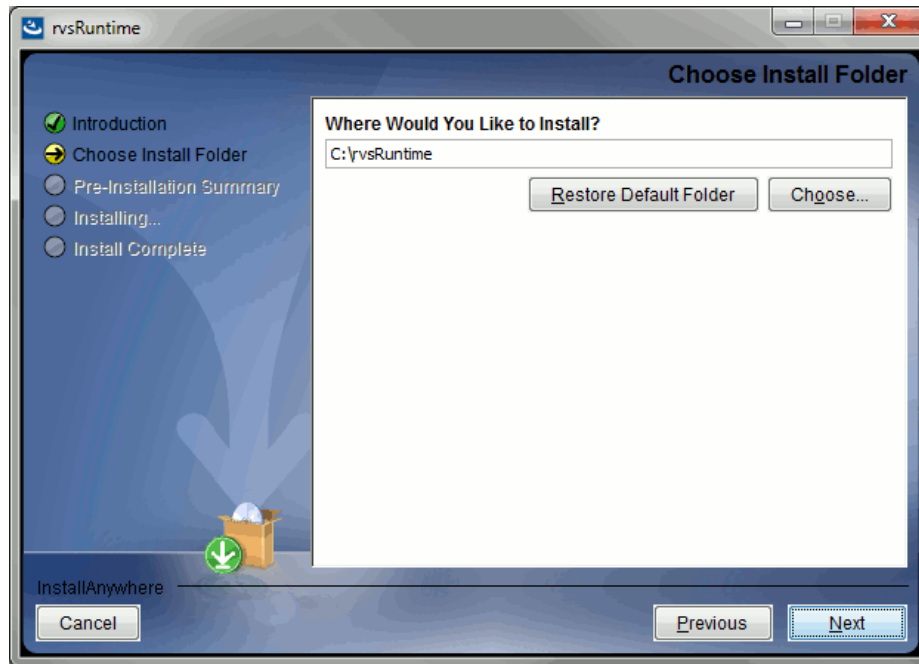
E-mail: rvs-service@t-systems.com

6.2.2 New Installation of rvsRuntime

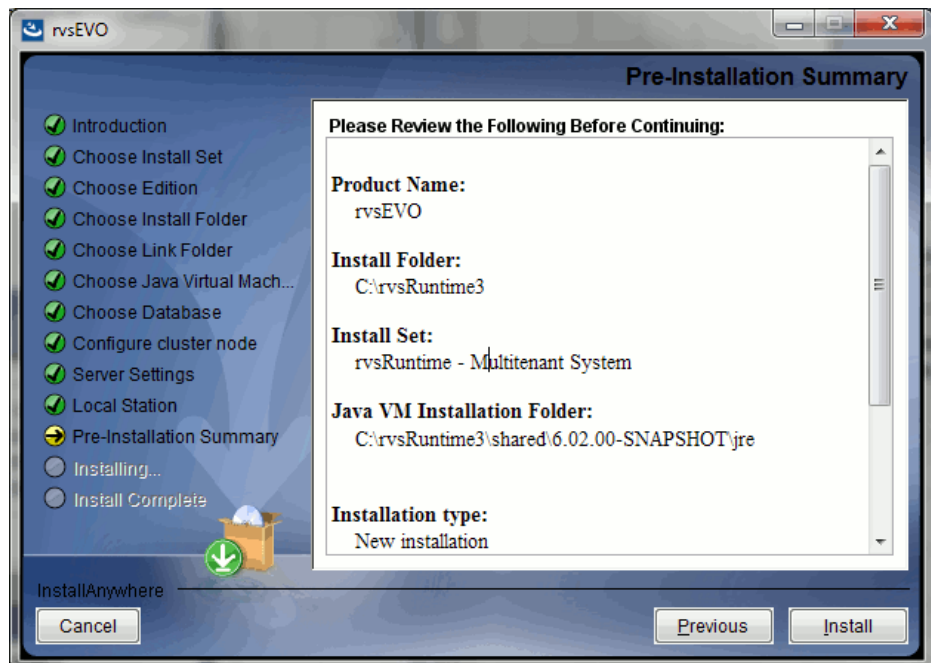
Start the installation process like described in the rvsEVO User Manual, chapter „Installation“ / „ New Installation of rvsEVO“.

After the dialog which invites you to choose the install set you will be asked to indicate the rvsRuntime destination directory (**\$RVS_RUNTIME_HOME**). Alphanumeric characters without special characters, underscores, hyphens and spaces are permitted for the directory name.

If you choose an existing installation folder of rvsEVO or rvsRuntime a warning is displayed and via a mouse-click on **OK** button you can indicate another directory name.



- In the next dialog you are given a brief overview of selections you have made (installation directory, installation type). The required and the currently available disk space is also displayed. Press the **Install** button to start installation and to copy the installation files into the directory you specified.



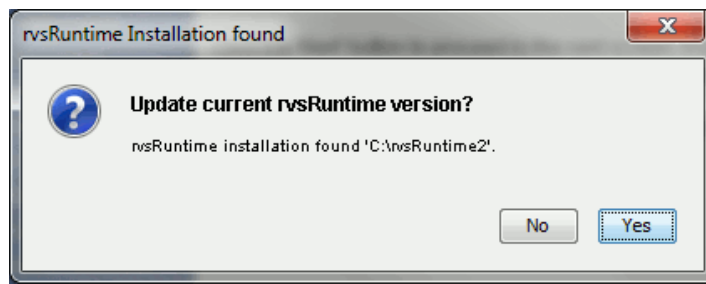
- The last dialogs informs you about the successful installation of rvsRuntime.

Hint: After installation process you need to save the license key file in the `$RVS_RUNTIME_HOME\conf\` directory as `license.properties`.

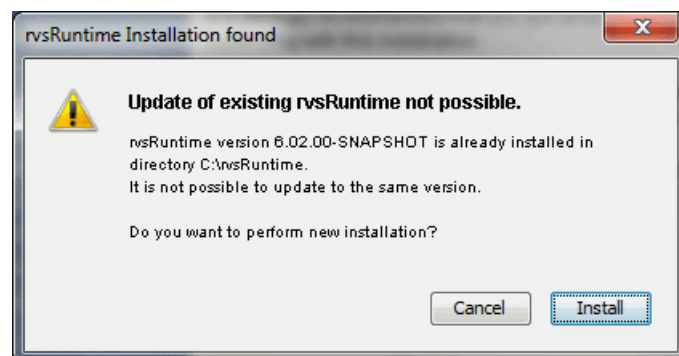
6.2.3 Update Installation

Requirement The system variable **\$RVS_RUNTIME_HOME** (please see chapter 6.2 "Installation of rvsRuntime") has to be set for the logged on user. Set **\$RVS_RUNTIME_HOME** by choosing Start -> Control Panel -> System -> Advanced -> Environment Variables.

- Start the `rvsEVO-setup-[version]-.exe` installer by double-clicking or using the Windows command: Start -> Run.
- Click the **Yes** button if the installer asks you whether you would like to update your current rvsRuntime.



Hint: The version of the installer must be newer than the version of your current installation. If the versions are identical you get the following message:



- In the next dialog you are given a brief overview of selections you have made (installation directory, link directory). The required and the currently available disk space is also displayed. Press the **Install** button to start the update installation.
- The last window informs you about the successfully installation of rvsRuntime. Finish the installation routine by pressing **Done** button.

6.2.4 rvsRuntime Installation Folder

During installing process the following subdirectories are installed in **\$RVS_RUNTIME_HOME** directory. Please see chapter 8.1 "rvsEVO Installation Folder" for information about the folders, which are not described in this section.

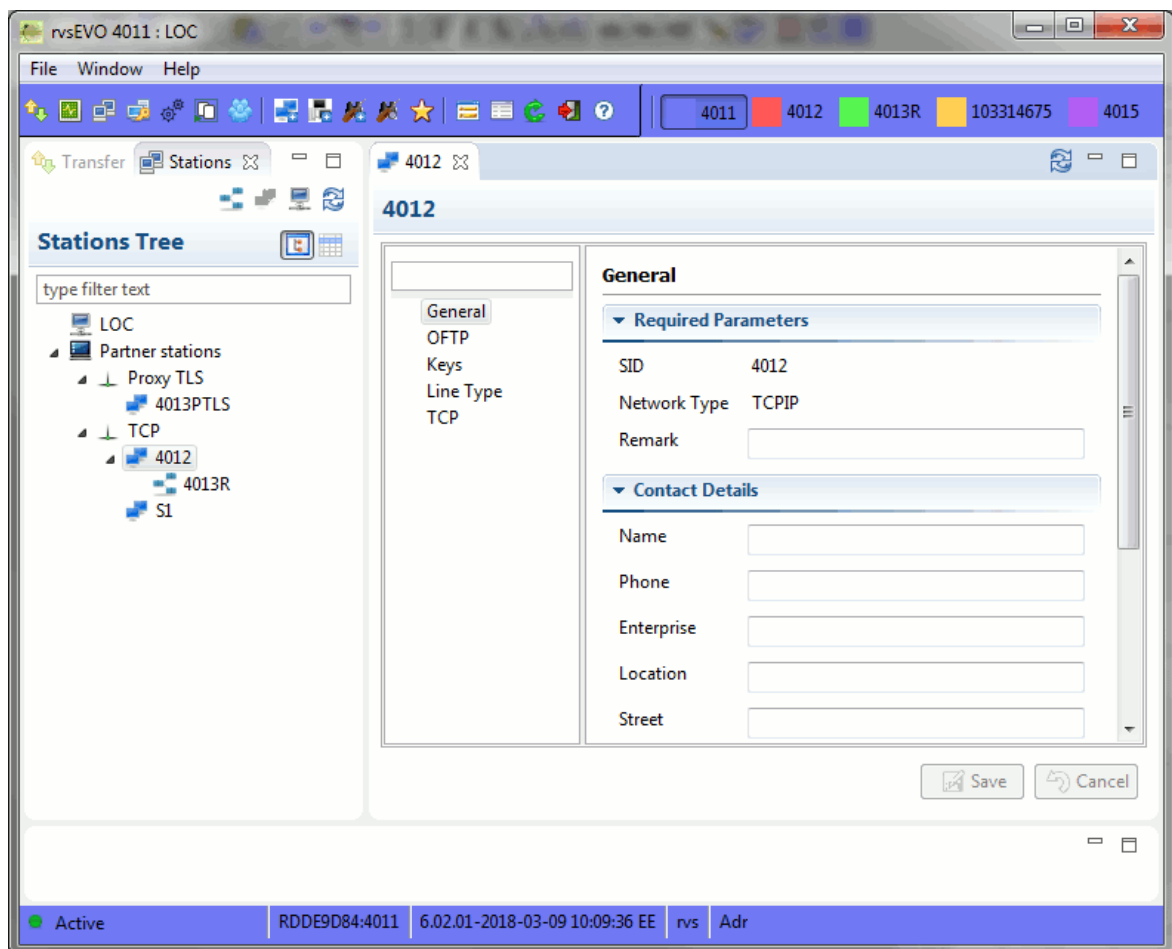
- **bin** contains the `rvsrt` tool
- **conf** contains the licensekey and configuration files of rvsRuntime.
- **docs**
 - **Manuals**
 - **jobstart-examples**
 - **rvsEVOClientAPI**
- **log** contains the installation log file and `rvsClientRCP.log` file which contains error messages from RCP-GUI MI.
- **shared:** contains data which are identical for all tenants
 - **<rvsRuntime installation version>**
 - **clib**
 - **jre**
 - **lib**
 - **rcpgui**
 - **system**
 - system/conf**
 - system/data**
 - **tenants:** contains the tenant installation directories
 - **<tenant ID>:** contains the special data for the tenant
 - **archive**
 - **bin** contains the command line tools and `derby.log` file
 - **conf:** contains configuration files and the licensekey of the tenant
 - conf/conversiontables:** conversiontables
 - **db**
 - **files**
 - **jobs**
 - **log**
 - **system**
 - system/db**
 - **tools**

- **UninstallerData** contains the files, needed for uninstalling rvsRuntime

6.3 Graphical User Interface of rvsRuntime

The GUI of rvsRuntime (RCP-GUI MI (**M**ulti **I**nstance)) is nearly analog to rvsEVO GUI (RCP-GUI SI (**S**ingle **I**nstance)). The RCP-GUI SI is described in chapter "Graphical User Interface" of rvsEVO User Manual.

The illustration below shows the basic structure of the GUI in Stations perspective.



There are the following differences to rvsEVO RCP-GUI SI:

- The title bar displays the title („rvsEVO“, tenant ID of the connected server and SID of the local station).
- The menu bar offers the functionalities:
 - **File**: Change Password, Exit, Logout (not available)
 - **Window**: Reset Perspective, Show View (Selection of perspective or special view)
 - **Help**: rvsEVO online help, rvsEVO version

- On right-hand side, the function bar contains tenant switchers for choosing the tenant you want to connect to.
Via right-click on the buttons you get the following options:
 - **Reset**: only with started server, to reset the tenant perspective to default
 - **Close**: for removing the tenant from RCP-GUI
 - **Show Text**: for displaying / hiding the tenant ID.
- Via left-hand symbols of the function bar you can select the view of the tenant (Transfer, Monitoring, Stations, Key Management, Automation, Parameter, User Management).
- The Reset button in the function bar resets the visible tenant to its defaults: Navigation tree of Transfer view is shown.
- The function bar and status line are highlighted in different colors depending of the tenant. The setting is done in `rvsServerInstanceList.xml` file during creation of the tenant.
- The function bar provides an additional button for logging out from server (Currently not available):



6.3.1 Starting of rvsRuntime GUI (RCP-GUI MI)

Hints:

Before the graphical user interface can be started one rvsEVO tenant server has to be started.

Only the RCP-GUI is supported by rvsRuntime.

Start the RCP-GUI MI (Multi Instance) via `startgui` command of `rvsrt` tool.

Syntax:

```
rvsrt [<rvsrt options>] startgui [-h]
```

Hint: In case of GUI start failure an `rvsClientRCP.log` file is written into `$RVS_RUNTIME_HOME/log` directory. The file settings can be done via `rvsClientRCPLogger.xml` file in `$RVS_RUNTIME_HOME/conf` directory like described in chapter 7.3.3 "Customizing the rvsClientRCP Log File via `rvsClientRCPLogger.xml`".

6.4 Managing of rvsEVO Tenants

A tenant works like a single rvsEVO instance. Please see the rvsEVO User Manual for more information.

The `rvsServerInstanceList.xml` file

The `rvsServerInstanceList.xml` file in `$RVS_RUNTIME_HOME/conf/` directory contains the information needed for server connection (tenant ID, tenant name, host name ...).

With creation of a tenant the tenant information is added to `rvsServer-InstanceList.xml` file and with deletion of a tenant it is removed from the list.

rvsrt Tool The `rvsrt.bat` tool in `$RVS_RUNTIME_HOME/bin` directory and its commands provides creation, deletion and managing of rvsEVO tenants.

Syntax:

```
rvsrt [-h ] [-p] [-v] [-V] <command> [command-options]
[<tenant-id>]
```

Attention: For successful execution of commands the sequence of the parameters must be followed as given in the syntax: „`rvsrt`“ `<rvsrt parameters>` `<command>` `<command options>` `<tenant ID>`.

rvsrt Parameters:

Parameter	Description
-h -? --help	Displays the description of the command.
-p --plain-output	Parameter for plain output.
-V --version	Displays the installed rvsRuntime version.
-v --verbose	Disables colorised ANSI output

The table below shows the `rvsrt` commands and its functionality.

rvsrt Commands:

Command	Description
create	Creates a new rvsEVO tenant (see chapter 6.4.1 "Creation of rvsEVO Tenant")
delete	Removes an rvsEVO tenant (see chapter 6.4.3 "Deletion of rvsEVO Tenant")
list	Lists the state of all rvsEVO tenants. Syntax: <code>rvsrt [<rvsrt options>] list [-c=<column-list> --columns=<column-list>] [--help -h -?]</code> Hint: You can list the columns to be shown comma-separated. The following values are allowed: <code>name</code> , <code>version</code> , <code>rmi</code> , <code>userid</code> , <code>run-mode</code> , <code>directory</code> , <code>command</code> , <code>color</code> . Default: <code>tenant ID</code> , <code>state</code> , <code>tenant name</code>

rvsrt Commands:

Command	Description
rollout	For distribution of licensekey (available for local instance, only) or station configuration from one rvsEVO tenant to others (see chapter 6.4.2 "Distritution of Licensekey or Station Configuration").
status	Issues a rvsEVO tenant status Syntax: rvsrt [<rvsrt otpions>] status [--help -h -?] <tenant ID> You get the following information: status, tenant ID, tenant name, version, RMI, user ID, run mode, installation directory, start command, color, login status.
start	Starts an rvsEVO server (available for local instance, only). The start command is defined in \$RVS_RUNTIME_HOME/conf/rvsServerInstanceList.xml file. Via XML element startServerCommand the command can be adjusted to your requirements by the superuser. Hint: On Windows systems the command prompt has to be opened as administrator. Syntax: rvsrt [<rvsrt otpions>] start [--help -h -?] <tenant ID>
startgui	Starts the RCP-GUI MI Syntax: rvsrt [<rvsrt otpions>] startgui [--help -h -?]
stop	Stops an rvsEVO server (available for local instance, only). Syntax: rvsrt [<rvsrt otpions>] stop [--help -h -?] <tenant ID>
update	Updates an rvsEVO tenant (available for local instance, only). Hints: Multiple tenants should be updated successively. Syntax: rvsrt [<rvsrt otpions>] update [--help -h -?] <tenant ID>

6.4.1 Creation of rvsEVO Tenant

You can import the parameters for creating a tenant via input file or enter the parameters into the command prompt. An example of input file is given at the end of this section.

Use the `rvsrt create` command for creating an rvsEVO tenant and to add the tenant to the server instance list in `rvsServerInstance-`

List.xml. The installation directory of the tenant is
`$RVS_RUNTIME_HOME/tenants/<TENANT-ID>`

On Windows systems rvsEVO tenant will be installed as Windows service, named „rvsEVO“ <TENANT-ID>, by default.

Hints:

- Please see the „rvsEVO User Manual“ chapter „rvsEVO Database“ before creating a tenant with Oracle or MS SQL database.
- The creation process can be terminated via entering `exit` or `quit` command.

Attention:

The sequence of the parameters must be followed as given in the syntax:
`„rvsrt“ <rvsrt parameters> <command> <command options> <tenant ID>`.

On Windows systems: The command prompt has to be opened as administrator.

A restart of RCP-GUI MI is necessary in order to display the created tenant.

Syntax:

`rvsrt [<rvsrt options>] create [-r] [-h] [-i <file-name>]`

Optional `create` Parameters:

Parameter	Description
<code>-r</code> <code>--remote</code>	With this parameter you can add an existing rvsEVO installation (on remote machine) to the instance list in <code>\$RVS_RUNTIME_HOME/conf/rvsServerInstanceList.xml</code> file.
<code>-h</code> <code>-?</code> <code>--help</code>	Displays a description of the current command.

During creation process the following parameters are requested. The values in parentheses are to be used in input file (see the example below the table).

Requested create Parameters:

Parameter	Description
License file [LICENSE_FILE]	Path and name of license file that will be distributed to the tenant. Default: \$RVS_RUNTIME_HOME/conf/license.properties
Tenant ID [TENANT_ID]	Unique ID of rvsEVO tenant, max 16 characters. Permitted characters: Alphanumeric characters (upper-case letters) without special characters, underscores, hyphens and spaces. Default: RVS001
Tenant name [TENANT_NAME]	Unique name of rvsEVO tenant. Default: „RVS_“ & <specified Tenant ID>
Database type [DATABASE_TYPE]	With creation of a local tenant, only. rvsEVO Database type. Possible values: - derby (default) - oracle (only with Enterprise Edition) - mssql (only with Enterprise Edition)
Username [DEFAULTUSER_NAME]	User name of the default user with administrator rights Default: rvs Hint: Please use unique user names.
Password [DEFAULTUSER_PASSWORD]	Password of the default user with administrator rights Default: rvs
IP address or hostname [RMI_HOST]	IP address or host name for RMI connection Default: host name of your machine
RMI Port [RMI_PORT]	Each tenant needs a unique RMI port Default: 3755

create Parameters for Oracle or MS SQL Database:

Parameter	Description
Database host [DATABASE_HOST]	Host Name or IP address for database connection
Database port [DATABASE_PORT]	Port for database connection Default for Oracle: 1521 Default for MS SQL: 1433

create Parameters for Oracle or MS SQL Database:

Parameter	Description
Database name [DATABASE_NAME]	Name of Oracle or MS SQL database Default: EVODB
Database user [DATABASE_USER]	User setting up on Oracle or MS SQL database Default: rvs
Database password [DATABASE_PASSWORD]	Password of database user Default: rvs
Group name [CLUSTER_GROUP_NAME]	Name of server farm Default: rvsEVO6_Cluster
Node name [CLUSTER_NODE_NAME]	Unambiguous designation of the node to be installed Default: rvsEVO6_Node
Node ID [CLUSTER_NODE_ID]	Unambiguous ID of the node to be installed Default: 0
Node IP [CLUSTER_NODE_IP]	IP-Adress or Host Name where the node is to be installed Default: IP address of your machine
Node port [CLUSTER_NODE_PORT]	Port for the node Default: 0
Nodes [CLUSTER_NODES]	IP addresses or host names of the other nodes of the server farm: Default: Host name of your machine

Below you find examples of input files for importing the `create` parameters via input file:

Example of input file for tenants with derby Database:

```
TENANT_ID           =4011
TENANT_NAME        =RVS_4011
DATABASE_TYPE      =derby
DATABASE_HOST      =
DATABASE_PORT      =
DATABASE_USER      =rvs
DATABASE_PASSWORD  =rvs
DATABASE_NAME      =0
CLUSTER_GROUP_NAME =rvsEVO6_Cluster
```

```

CLUSTER_NODE_NAME      =rvsEVO_Node
CLUSTER_NODE_ID        =0
CLUSTER_NODE_IP        =BE1DY430
CLUSTER_NODE_PORT      =0
CLUSTER_NODES          =
DEFAULTUSER_NAME       =rvs
DEFAULTUSER_PASSWORD   =rvs
RMI_HOST               =BE1DY430
RMI_PORT               =4011
LICENSE_FILE           =c:/rvsRT/conf/license.properties

```

Example of input file for tenants with Oracle- or MS-SQL Database:

```

TENANT_ID              =SQL4011
TENANT_NAME            =SQL4011
DATABASE_TYPE          =mssql
DATABASE_HOST          =<hostname>
DATABASE_PORT          =1433
DATABASE_USER          =dbuser
DATABASE_PASSWORD      =userpw
DATABASE_NAME          =evodb
CLUSTER_GROUP_NAME     =rvsEVO6_Cluster
CLUSTER_NODE_NAME     =rvsEVO_Node
CLUSTER_NODE_ID       =0
CLUSTER_NODE_IP       =10.177.63.220
CLUSTER_NODE_PORT     =2988
CLUSTER_NODES         =10.177.63.220
DEFAULTUSER_NAME       =rvs
DEFAULTUSER_PASSWORD   =rvs
RMI_HOST               =RDDE9D84
RMI_PORT               =3755
LICENSE_FILE           =c:/rvsRT/conf/license.properties

```

6.4.2 Distribution of Licensekey or Station Configuration

Via `rollout` command you can distribute a licensekey to specified rvsEVO tenants or station configurations from one rvs EVO tenant to others.

Hints:

In case of distributing station configurations the rvsEVO servers must be started.

A licencekey can be distributed to tenants on local machine, only.

Syntax:

```
rvsrt [rvsrt options] rollout [--help | -h | -?]
[-l | --license=<license-file>]
[-u | --update-if-exists]
[-r | --receiver-change-enabled]
[-n | --network-change-enabled]
[-o | --origin=<tenant-id>]
[-s | --station=<station-ID>] <target-list>
```

Attention: The sequence of the parameters must be followed as given in the syntax: „rvsrt“ <rvsrt parameters> <command> <command options> <tenant ID>.

Example for distribution of a license:

```
rvsrt -v rollout -l c:\Daten\license.properties
4001,ora4002,mssql4003
```

Examples for distribution of station configuration:

```
rvsrt -v rollout -s PS1 -o PS4005 -u -r -n
PS4006,PS4001
```

In the example above station „PS1“ is distributed from tenant „PS4005“ to tenants PS4006 and PS4011. Existing stations shall be updated (also network type and receiver number)

```
rvsrt -v rollout -s PS1 -o PS4005 @all
```

In the example above station „PS1“ is distributed from tenant „PS4005“ to all other tenants.

Required rollout Parameter:

Parameter	Description
<target-list>	Destination tenant IDs. Possible values: - comma-separated list of tenant IDs - „@all“ for all tenants except origin

Optional rollout Parameters:

Parameter	Description
-l --license	Path and name of license file which is to be distributed

Optional `rollout` Parameters:

Parameter	Description
-n --network-change-enabled	Enables change of network type of station if station ID already exists. Only if parameter -u and -s is set.
-o --origin	Tenant from which the station configuration is to be read. Only if parameter -s is set
-r --receiver-change-enabled	Enables change of receiver number of station if station ID already exists. Only if parameter -u and -s is set.
-s --station	ID of the station which is to be distributed
-u --update-if exists	Will update station at target in case station already exists. Only if parameter -s is set
-h -? --help	Displays a description of the current command.

6.4.3 Deletion of rvsEVO Tenant

Use the `rvsrt` command `delete` for removing an rvsEVO tenant. The tenant to delete must be in STOPPED or START-FAIL status.

Hints:

A tenant on remote machine can be deleted from instance list, only. Database and filesystem cannot be removed.

On Windows systems: The command prompt has to be opened as administrator.

Syntax:

```
rvsrt [rvsrt options] delete [-c] [-d] [-s] [-h]
<tenant ID>
```

Attention:

The sequence of the parameters must be followed as given in the syntax: „rvsrt“ <rvsrt parameters> <command> <command options> <tenant ID>.

A restart of RCP-GUI MI is necessary in order to delete the removed tenant from the interface.

In case of using the `delete` command only with <tenant-id> parameter the program guides you through the deletion process step by step:

- First you are asked whether you want to delete the tenant definition. Enter „Yes“ or „y“ to delete the tenant from instance list

(`rvsServerInstanceList.xml` file in `$RVS_RUNTIME_HOME/conf` directory) and to move to the next step.

- Subsequent you are asked whether you want to delete the database and filesystem of the tenant.
Enter „Yes“ or „y“ to finish the deletion of the tenant.

With Oracle or MS-SQL database you may delete filesystem and database content separately.

Required delete Parameter:

Parameter	Description
<tenant ID>	ID of the tenant to be removed

Optional delete Parameters:

Parameter	Description
-c --clean-database	This parameter deletes the database content of the tenant, incl. transfer jobdata
-d --delete-files	This parameter deletes the tenant from <code>rvsServerInstanceList.xml</code> file in <code>\$RVS_RUNTIME_HOME/conf</code> directory and the file system of the tenant incl. transfer files
-s --silent	This parameter deletes the tenant from <code>rvsServerInstanceList.xml</code> file in <code>\$RVS_RUNTIME_HOME/conf</code> directory without interaction.
-h -? --help	Displays a description of the current command.

7 Customizing rvsEVO

In this chapter you find information about configuration beyond standard configuration via rvsEVO GUI.

7.1 Customizing rvsEVO via rvs-system.properties File

Below you can find the description of the definitions of `rvs-system.properties` file.

Please enable the settings with re-starting the rvsEVO server.

- `rvs_evo.jobstarter.keep_temp_files=true`

This definition refers to the jobstarts. If the value of **Parameter Handling** parameter is REPLACE the jobstarter creates temporary files including the job data which substitutes the placeholders of the script. You can define whether the temporary files are to be deleted or not. Possible values:

`true`: the files are kept

`false`: the files are deleted (default)

- `rvs_evo.serviceprovider.keep_temp_files=true`

Via this definition you can define whether temporary files created by the service provider are to be deleted or not. Possible values:

`true`: the files are kept

`false`: the files are deleted (default)

- `rvs_evo.protocol.restart=false`

Via this definition you can define whether a restart flag is to be set up in SSID and the job is restarted.

`true`: restart flag is set up (default)

`false`: no restart flag

- `rvs_evo.routed_job.fsp.dont_keep.receive_file=true`

By default, files received via FileServiceModul feature are kept until receiving the EERP. If

`rvs_evo.routed_job.fsp.dont_keep.receive_file=true`, this function is deactivated and the files are deleted after successful routing.

- `observer.sfiddesc.usefilename=true`

If the file transmission takes place via Observer you can define whether the original filename is to be used as file description (SFID-DESC).

- `rvs_evo.controller.underscore_in_vdsn_allowed=true`

Use this definition to allow underscores in VDSN. Possible values:

`true`: underscores are admitted in VDSN

`false`: underscores are not admitted in VDSN (default)

- `useKeyUsage=false`

Via this definition you can define whether the certificate usage of X.509 certificate is to be considered or not.

`true`: the key usages for each function has to be set in the X.509 certificate (default)

`false`: the settings of X.509 certificate will not be checked.

- `javax.net.debug=ssl`

Via this definition you can activate the debugging for TLS connections. By default, this is disabled.

- `java.rmi.server.hostname=<IP>`

The value of this property represents the host name string that should be associated with remote stubs for locally created remote objects, in order to allow clients to invoke methods on the remote object. The default value of this property is the IP address of the local host, in "dotted-quad" format.

- `rvs_evo.network.remote_callback_handler.verification_initial_delay=30000`

Use the definition above to define the wait time in milli seconds after server start before end-to-end check between rvsEVO and rvsProxy will start.

- `rvs_evo.network.remote_callback_handler.verification_interval=60000`

Use the definition above to define the wait time in milli seconds between two end-to-end checks between rvsEVO and rvsProxy.

- `rvs_evo.tracing.api.enabled = true`

Use this definition to enable tracing function for API.

```
- rvs_evo.tracing.api.subjects = *.*
```

Use this definition to configure tracing function for API. Enter a comma separated list of classes and methods.

. : enables tracing of all classes with all of its public methods.

Rvs.login : enables tracing of method "login" in class "Rvs".

StationList.* : enables tracing of all methods in class "StationList".

```
- rvs_evo.receive_job.keep_temp_files_on_failure = true
```

Enable this definition for keeping temporary files, generated during receiving a job, in case of failure in `$RVS_HOME/files/temp` directory and display the file name in the detailed view of the receive job.

```
- rvs_evo.networkaddress.cache.ttl=30
```

Enable this property to set the validation (ttl) in seconds for caching DNS entries.

Default value: 30

```
- rvs_evo.receive_job.ignore_jobstart_and_delivery_error=false
```

With this property is possible to force the application to ignore error while delivering files or executing jobstart after receive.

Default value: false

```
- rvs_evo.receive_job.ignore_illegal_efid_values=false
```

Only relevant in case of transmitting files with AS/400 machine: With this property the application can be forced to ignore illegal values in EFID.

Default value: false

```
- rvs_evo.serviceprovider.status_update.interval=1
```

With this property it is possible to adjust time service provider will write progress information to monitor log.

Default value: 1 minute

- `rvs_evo.network.tls.handshake_timeout=15000`

Timeout in milliseconds for SSL handshake.

Sets `SoTimeOut` when a `SSLSocket` is created before handshake is initiated.

IMPORTANT: timeout must be less than `OFTP_TIMEOUT`

- `rvs-evo.controller.sendqueue.polling.interval=200`

This property allows to change the length of the interval the send queue uses to poll for new jobs that needs sending.

Default value: 200 milliseconds

- `rvs-evo.controller.activejob.status.interval=600`

This property allows to change the interval used to write actual job workload to monitor log.

Default value: 600 milliseconds

- `rvs-evo.controller.archiveJob.txtLevel=2`

This property provides additional writing of `RevisionLog.txt`. (TAB separated values; in values: TAB is replaced by SPACE, leading and closing whitespaces are removed; every 100 rows a headline is printed). Possible values:

- 0 : do not write `RevisionLog.txt`
- 1 : basic
- 2 : extended (default)
- 3 : same as `revisionLog.xml` but TAB separated values
- 4 : customized by the user via element `rvs-evo.controller.archiveJob.txtHeader` (see below)

- `rvs-evo.controller.archiveJob.txtHeader=`

This property is mandatory in case of `rvs-evo.controller.archiveJob.txtLevel=4`

`txtHeader` is a comma separated list of valid column headers. First column header is set implicit to `JobID`, so `JobID` is not to be set.

Valid column headers are:

`FileName` `FileNameSrc` `SourceFileLength` `VDSN` `SidOriginator` `SidNeighbour` `SidDestination` `OIDOriginator` `OIDNeighbor` `OIDDestination` `Direction` `NoCopy` `Routing` `SFIDDateTime` `RecordLength` `RecordFormat` `Conversi-`

```

onTable InputCharacterSet OutputCharacterSet
Serialize Disposition SecurityFeatureSet Compres-
sion Encryption SignFile SignEERP EnvFormat SFIDDes-
criptionHex SFIDDescriptionText EncryptionAlgorithm
EncryptionCertificateIssuerName EncryptionCertifi-
cateSerialNumber SignCertificateIssuerName SignCer-
tificateSerialNumber ExternalJobID ScheduleDateTime
RecordMode Label RetryCount TimeStartFile TimeEnd-
File RestartPos CurrentRecordLength MarkedForCom-
plete RecordCount FilePos TransmittedBytes
LastByteRead LastByteSend SendAttempts LastChange
WaitTime Category State ResponseCreatorSID Response-
CreatorOID NerpReasonCode NerpReasonText AssignERP
UniformatFilename CompressedFilename EncryptedFile-
name SpCompleted SpCmsFormatCompleted SpCmsSignCom-
pleted SpCmsCompressCompleted
SpCmsEnvelopeCompleted SpCmsTempFileName ErpSigna-
ture ErpType ErpVDSN ErpDate ErpTime ErpDestination
ErpOriginator ErpHash SyncSendAttempts SyncTimeout
ErrorType ErrorID ErrorText

```

- `rvs-evo.controller.log_to_console=true`

This property allows to enable logging to console after server start.

Default value: `false`

- `rvs_evo.protocol.deactivate_sfid_datetime_check=false`

This property deactivates check of SFIDDATE and SFIDTIME.

For example: invalid is DATE 20180199 (2018/01/99) or TIME 080070 (8:00:70)

Default value: `false`

- `rvs_evo.serviceprovider.compression_level=1`

This property allows to adjust the compression level that the service provider will use for compression.

Available values from -1 to 9.

Default value: 1

- `rvs_evo.serviceprovider.maximum_buffer_size=128`

This property allows to change the buffer size used for pre- and post-processing.

Unit: kilo bytes

Default value: 128

- `rvs_evo.controller.pre_and_post_processing.number_of_used_threads=40`

This property controls the number of threads which are used for pre- and postprocessing. We do not recommend to increase the value.

Default value: 40

- `rvs_evo.controller.disable_erp_routing=false`

This property allows to disable termination of routing jobs after receiving EFPA for the routing send jobs and to wait for EERP/NERP to terminate (default behaviour up to version 6.02.01).

Default value: false

- `rvs_evo.controller.sfna_retry_no.automatic_fail=true`

This property allows to switch back to automatic FAILED_WITH_NERP behaviour, which was implemented up to version 6.02.03, on reception of SFNA RETRY=NO for routing send jobs.

Possible values:

- `true`: NERP will be sent to originator and routing send and receive jobs will terminate with FAILED_WITH_NERP (category FAILED)
- `false`: routing send and receive job will not terminate to allow operator restart or deletion of routing jobs with or without NERP

Default value: false

Hint: During the update installation the `rvs-system.properties` file is not upgraded but an `rvs-systems.properties.new` file is added to `conf` directory. For making settings via an new `rvs-system.properties` definition, please take over the needed element from the new `properties` file to your old one. If your `rvs-system.properties` file was not edited up to now, you can rename `rvs-system.properties.new` to `rvs-systems.properties`.

7.2 Customizing rvsEVO via rvsEVOClient.prefs File

Below you can find the description of the definitions of `rvsEVOClient.prefs` file in `$RVS_HOME/conf` directory.

```
- <entry key="client.RMIServiceName" value="rvsEVO"/>
```

This definition sets client RMIServiceName which is used in local mode only and must be equal to the database entry for the related server.

```
- <entry key="client.RMIServiceHost" value="RDDE9D84"/>
```

This definition sets client RMIServiceHost which is used in local mode only and must be equal to the database entry for the related server.

```
- <entry key="client.RMIServicePort" value="3755"/>
```

This definition sets client RMIServicePort which is used in local mode only and must be equal to the database entry for the related server. `client.RMIServicePort` is a integer between 0 and 65535 with default 3755.

```
- <entry key="client.account" value="rvs"/>
```

This definition sets client account which is used in local mode only. Is set usually during installation. Should be changed by specialists only.

```
<entry key="client.passwd" value="ZOKWCALCZGKW-WICB"/>
```

This definition sets client password which is used in local mode only. Is set usually during installation. Should be changed by specialists only

```
- <entry key="client.MaxLoginHistory" value="10"/>
```

This definition sets the number of maximum login history details which were shown in drop down list of **Login History Details** field during starting remote GUI.

```
- <entry key="client.FirstLanguage" value="en"/>
```

Via this definition you can define the GUI language. This setting does not concern the log messages. Changes only take effect after restarting rvsEVO GUI. Possible values:

en: english

de: german

```
- <entry key="client.RemoteAliasMode" value="no"/>
```

Via this definition you can allow remote login with the alias name defined in `remote-login-alias.properties` file (for remote GUI only).

- `<entry key="client.remotefileloader.port" value="3756" />`

Port for uploading files in remote GUI

- `<entry key="client.sendjobs.allow.OID.addressing" value="yes"/>`

With this definition you can activate parameter **Addressing Mode** in the wizard for creating a send job via rvsEVO GUI.

- `<entry key="client.transfer.sendJobWizard.externalJobId" value="mandatory"/>`

This definition sets up **External Job ID** to mandatory parameter in the wizard for creating send jobs via rvsEVO GUI.

- `<entry key="client.monitorlog.livelog.pagesize" value="1000"/>`

This definition sets up the number of messages that should be displayed in Live Monitor Log

- `<entry key="client.monitorlog.[INF|WRN|ERR].colour.[r|g|b]" value="[255|0]"/>`

This definition highlights the display of information (INF), warnings (WRN) and error messages (ERR).

Example:

```
<entry key="client.monitorlog.WRN.colour.g" value="255"/>
```

In the example above warnings are displayed in green (colour.g). Changes only take effect after restarting rvsEVO GUI.

- `<entry key="client.startjob" value="" />`
`<!-- used in swing GUI only -->`
- `<entry key="client.home.dir" value="" />`
`<!-- used in swing GUI only -->`
- `<entry key="client.conf.dir" value="" />`
`<!-- used in swing GUI only -->`
- `<entry key="client.engdat.conf.dir" value="" />`
`<!-- used in swing GUI only -->`

- `<entry key="client.arc.dir" value="" />`
`<!-- used in swing GUI only -->`
- `<entry key="client.help.file" value="" />`
`<!-- used in swing GUI only -->`

7.3 Customizing the Log Files

The maximum file size and maximum number of log files can be defined via `$RVS_HOME/conf/rvsLogger.xml`, `rvsClientLogger.xml` and `rvsClientRCPLLogger.xml` files.

7.3.1 Customizing the Monitor and rvs Log Files via rvsLogger.xml File

`monitor.log` The Monitor messages are saved in the file `$RVS_HOME/log/monitor.log`. A new `monitor.log` file is generated daily and the old `monitor.log` file is renamed in „`monitor.log`“ plus date pattern plus counter. Also a new file is generated if the maximum file size is reached. XML-Element `<appender name="monlog" ...>` contains the definitions of `monitor.log` file settings.

- Use function **MaxFileSize** to set the maximum size:

Syntax:

```
<param name="MaxFileSize" value="10MB"/>
```

- Use function **MaxRollFileCount** to set the maximum number of log files:

Syntax:

```
<param name="MaxRollFileCount" value="50"/>
```

`rvs.log` XML-Element `<appender name="rvslog" ...>` contains the definitions of `rvs.log` file settings.

- Use function **maxFileSize** to set the maximum size (in byte):

Syntax:

```
<param name="maxFileSize" value="2097152"/>
```

- Use function **MaxRollFileCount** to set the maximum number of log files:

Syntax:

```
<param name="maxBackupIndex" value="10"/>
```

7.3.2 Customizing the rvsClient Log File via rvsClientLogger.xml File

XML-Element `<appender name="Standard" ...>` of `$RVS_HOME/conf/rvsClientLogger.xml` contains the definitions of `rvsClient.log` file settings.

- Use function **maxFileSize** to set the maximum size (in byte):

Syntax:

```
<param name="maxFileSize" value="2097152"/>
```

- Use function **maxBackupIndex** to set the maximum number of log files:

Syntax:

```
<param name="maxBackupIndex" value="10"/>
```

7.3.3 Customizing the rvsClientRCP Log File via rvsClientRCPLogger.xml

XML-Element `<appender name="Standard" ...>` of `$RVS_HOME/conf/rvsClientRCPLogger.xml` contains the definitions of `rvsClientRCP.log` file settings.

- Use function **maxFileSize** to set the maximum size (in byte):

Syntax:

```
<param name="maxFileSize" value="2097152"/>
```

- Use function **maxBackupIndex** to set the maximum number of log files:

Syntax:

```
<param name="maxBackupIndex" value="10"/>
```

7.4 Customizing RMI Connection over TLS

From rvsEVO version 6.02.05 upwards the communication between rvsEVO client and server can be encrypted via TLS protocol. Please see below how to customize rvsEVO for secure RMI communication.

7.4.1 Setting up of an RMI Connection via TLS Protocol

Necessary steps for line encryption via TLS protocol between client and server:

- Generate keystore file via `keytool` script. In rvsEVO User Manual chapter „Database Maintenance and further Administrator Tools“ you find a description of `keytool` script. The X.509 certificate is exported into the truststore automatically.
- Keystore and truststore must be saved in `$RVS_HOME/conf` directory. At all clients the truststore should be saved in `$RVS_HOME/conf` directory too. For connecting rvsRuntime GUI via TLS truststore is to be saved in `$RVS_RUNTIME_HOME/conf` directory.
- Set up the rvsEVO server for RMI over TLS via **RMITLSEnabled**, **RMITLSKeystore** and **RMITLSKeystorePassword** parameters in

Parameter Group BASIC (please see rvsEVO User Manual, chapter „Customizing the global rvsEVO Parameters“ for more information).

- Set up the rvsEVO clients via setting the TLS definition in `$RVS_HOME/conf/api-rmi-connection.properties` file:

Syntax:

```
<remote-ref-hostname>, <remote-ref-port>
=@TLS, <trust-store-path>
```

Hint: In `api-rmi-connection.properties` file you find examples and description for customizing dynamic SSH tunneling for RMI connection.

7.5 Pass Parameters to Jobstart Process

In „rvsEVO User Manual“ chapter „Automation with rvsEVO“ you get information about jobstarts after transmission and jobstarts after failure.

This chapter provides a detailed discription of Jobstart parameter **Parameter Handling** with settings **Parameter Handling=ENV** and **Parameter Handling=REPLACE**.

Parameter Hand-
ling=ENV

Parameter Handling=ENV: job data are set as environment variables. The following environment variables are specified (in angle brackets: name in XML file, in brackets: GUI name):

- **RVS_COMPRESSION** <compression> (Compression)
- **RVS_CONVERSION_TABLE** <conversionTable> (Code Table)
- **RVS_CREATED_AT** <creationDate> (SFID Date/Time)
- **RVS_DATE** Date of <creationDate>
- **RVS_DESCRIPTION** <fileDescription> (File Description)
- **RVS_DIRECTION** <direction> (Direction)
- **RVS_DISPOSITION** <disposition> (Disposition)
- **RVS_ENCRYPTION** <encryption> (Encryption)
- **RVS_ENCRYPTION_ALGORITHM** <encryptionAlgorithm> Encryption Algorithm)
- **RVS_ERROR_ID** <errorID> (Error Number)
- **RVS_ERROR_TEXT** <errorText> (Description)
- **RVS_EXTERNAL_JOB_ID** <externalJobId> (External Job ID)
- **RVS_FILE_BASENAME** (File name)
- **RVS_FILE_NAME** <filename> (Path and name of the file in outbox directory)
- **RVS_FILE_NAME_SRC** <filenameSrc> (Source File Name)
- **RVS_JOB_ID** <jobNumber> (ID)

- **RVS_LABEL** <label> (Label)
- **RVS_LAST_CHANGE** <lastStateChange> (Last Change)
- **RVS_NERP_CREATOR_ODETTE_ID** <nerpCreatorOdetteID> (Nerp Creator OdetteID)
- **RVS_NERP_REASON_CODE** <nerpReasonCode> (Nerp Reason Code)
- **RVS_NERP_REASON_TEXT** <nerpReasonText> (Nerp Reason Text)
- **RVS_PRIORITY** <priority> (Priority)
- **RVS_RECORD_FORMAT** <recordFormat> (Format)
- **RVS_RECORD_LENGTH** <recordLength> (Record Length)
- **RVS_RESTART_POSITION** <restartPos> (Restart Position)
- **RVS_SECURITY_FEATURE_SET** <securityFeatureSet> (Security Feature Set)
- **RVS_SEND_ATTEMPTS** <sendAttempts> (Send Attempts)
- **RVS_SERIALIZE** <serialization> (Serialization)
- **RVS_SID** <SID>
- **RVS_SID_DESTINATION** <sidDestination> (SID Destination)
- **RVS_SID_ORIGINATOR** <sidOriginator> (SID Originator)
- **RVS_SIGNATURE** <sign> (File Signature)
- **RVS_SIGNATURE_EERP** <signEERP> (Request Signed EERP/NERP)
- **RVS_STATUS** <status> (Status)
- **RVS_TIME** Time of <creationDate>
- **RVS_TIME_START_FILE** <timeStartFile> (Time Start File)
- **RVS_TRANSMITTED_BYTES** <transmittedBytes> (Bytes Transmitted)
- **RVS_VDSN** <VDSN> (VDSN)

Please see the rvsEVO User Manual, table "Explanation of the job attributes" in chapter 'Command Line Tools' for description of the environment variables .

Parameter Handling=REPLACE

Parameter Handling=REPLACE: A copy of the denoted file is created and discarded into `temp` directory. The placeholders of the scripts are substituted for job data. The placeholders are encased by ? (e.g. "move ?DSN? /home/rvsEVO/incomming" becomes "move /home/rvsevo/rvsEVO/files/inbox/TEST.TXT /home/rvsevo/incomming").

With MAINTENANCE parameter **PermissionTempScript** you can set up the file permissions of the temporary file. (Default: read/write/execute-permission for owner and group).

Placeholder of all jobstarts:

- **?DSN?**: DataSetName (name of local data set, where received information has been stored)
- **?VDSN?**: Virtual data set name under which the data set was transmitted
- **?DTAVAIL?**: Date, when the data set was available for sending;
Format TT/MM/JJ HH:MM:SS
- **?FORMAT?**: Record format of data set
- **?BYTES?**: Number of transmitted bytes
- **?RECORDS?**: Number of transmitted records for F and V format data sets; always zero for T und U format data sets.
- **?DTRCV?**: Date, when data set was delivered to local user in format
TT.MM.JJ HH:MM:SS
- **?UID?**: User ID
- **?SID?**: Station ID
- **?DSNTEMP?**: Name of temp. data set

Placeholder of jobstart type SND:

- **?MAXRECL?**: The meaning of this field depends upon the record format of the transmitted data set:
 - - F format: length of each record
 - - V format: maximum length a record may have
 - - T und U format: always 0 (zero)
- **?LABEL?**: string if the send command contained a LABEL parameter. Can be used to identify the send command.
- **?SECN?**: rvsEVO job ID (number of send command SE in rvs[®] portable)
- **?SKCN?**: rvsEVO job ID (number of send command SK in rvs[®] portable)
- **?SIDORIG?**: Station ID of originator
- **?SENDATT?**: number of unsuccessful attempts after which the program is to be started

Placeholder of jobstart type RCV:

- **?MAXRECL?**: The meaning of this field depends upon the record format of the received data set:
 - - F format: length of each record
 - - V format: maximum length a record may have
 - - T und U format: always 0 (zero)

- **?CNQS?**: rsvEVO job ID (command number of EERP (End-to- End-Response) for received file in rvs[®] portable).
- **?CNIE?**: rsvEVO job ID (command number of IE for received file in rvs[®] portable).
- **?CNIZ?**: rsvEVO job ID (command number of IZ for received file in rvs[®] portable)

7.6 Conversion Tables

The file transfer provides the possibility of code conversion. This chapter gives a generally explanation of the conversion tables and describes the process of creating own conversion tabels.

7.6.1 Code Tables Offered by rsvEVO

Text files are stored on most systems in one of two computer codes, namely ASCII (American National Standard Code for Information Interchange) or EBCDIC (Extended Binary Coded Decimal Interchange Code).

ASCII is the standard code for UNIX and DOS/Windows systems. EBCDIC was developed for IBM Mainframe computers.

ASCII: US-ASCII ISO 646; the ASCII character set defines 128 characters (0 to 127 decimal). This character set is a subset of many other character sets with 256 characters, including the ANSI character set of MS Windows.

ANSI: Windows ANSI, Values 0 to 127 are the same as in the ASCII character set, values 128 to 255 are similar to the ISO Latin-1 character set.

EBCDIC 037: support characters, which are used in the following countries: Australia, Brazil, Canada, New Zealand, Portugal, South Africa, USA.

EBCDIC 273: supports characters (especially umlauts), which are used in the following countries: Germany, Austria and Switzerland.

7.6.2 How to Add Your Own Conversion Table

In order to use your own conversion table, please follow the procedure given below:

- Create a new conversion table. The content of this file is described in the following XML schema:

```
<conversiontable codein=Original-Characteraset
codeout=Ziel-Characteraset>
  <description>Table to converts signs from ... to
  ...</description>
  <char value="0">0</char>
```



```

    <char value="1">1</char>
    .
    .
</conversiontable>

```

Note: `char value` is equivalent to the source code and the value in angle brackets is equivalent to the target code.

- Store the XML file in the directory `$RVS_HOME/conf/conversiontables`
- To add your table to the list of conversion tables, edit the XML file `$RVS_HOME/conf/conversiontables/characterSetConverters.xml`.

Next you find an abridged version of the file

`characterSetConverters.xml`:

```

<characterSetConverters coding="">
  <!-- -->
  <converter cid="ANSIIBM037"
    >conversion_ANSI_IBM037.xml</converter>
  <converter cid="IBM273-ASCII"
    >conversion_IBM273_ASCII.xml</converter>
  .
  .
</characterSetConverters>

```

In the example above `cid="ANSI-IBM037"` (`cid="IBM237-ASCII"`) is the used name in `rvsEVO` and `conversion_ANSI_IBM037.xml` (`conversion_IBM237_ASCII.xml`) is the name of your XML file.

- Please restart `rvsEVO` after your complement.

7.7 Connection Pooling

When using MS SQL- or Oracle database `rvsEVO` provides the function **connection pooling**.

A connection pool is a cache with existing data base connections. If a connection is needed, one of the pool can be used and it is not necessary to establish a connection for every access. `rvsEVO` uses `C3P0` (Hibernate) for connection pooling.

The settings for Connection Pooling can be done via `$RVS_HOME/conf/jdbc.properties` file.

Example of configuration with default values:

```
c3p0.minPoolSize=5
```

```

c3p0.maxPoolSize=25
c3p0.checkoutTimeout=0
c3p0.maxIdleTime=0
c3p0.idleConnectionTestPeriod=60
c3p0.automaticTestTable=POOLING_TEST
c3p0.acquireRetryAttempts=30
c3p0.acquireRetryDelay=1000
c3p0.breakAfterAcquireFailure=false
    
```

The most important parameter is `c3p0.idleConnectionTestPeriod`. With this parameter you define the time interval (in seconds) for considering the connections of the pool. The parameter `maxPoolSize` defines the maximum number of connections to the database and `minPoolSize` defines the initial number of connections.

For further information please read the C3P0 documentation:

<http://community.jboss.org/wiki/HowToconfiguretheC3P0connectionpool>

7.8 Operating with OFTP Routing Rules

From version 6.02.05 onwards rvsEVO provides the possibility of operating with internal routing rules on router side. That means you can route incoming files to another destination or convert the transmitted stations data of originator in routing send jobs for outgoing transmissions.

- | | |
|--------------|---|
| Application | You decide to perform your rvsEVO processes from an new machine. All your partners must take over your new data into their systems. You can circumvent this procedure by applying OFTP routing rules. OFTP routing rules can be created, updated, displayed, deleted and listed via <code>rvscli</code> feature. (See rvsEVO User Manual, chapter 'Command Line Tools' for more information about <code>rvscli</code>) |
| Requirements | <ul style="list-style-type: none"> – All affected stations are reachable via router (gateway) – Feature File Service Module is linked to your rvsEVO system which functions as router. <ul style="list-style-type: none"> – OFTP parameter File Service Module of internal stations is set to <code>INTERNAL</code> – OFTP parameter File Service Module of external stations is set to <code>EXTERNAL</code>. |

Hint:

De- and encryption for stations set up as `INTERNAL` is processed by routing station. Please see the rvsEVO User Manual, chapter „File Service Module“ for further information about File Service Module.

Please see the rvsEVO User Manual, chapter „Management of Stations“ -> „Routed Stations“ for information about setting up of a routed station.

Examples:

- Internal station INT1 (destination) receives files from external station EXT (originator) via station ROU (gateway). Station INT1 is replaced with internal station INT2.

With rvscli command `createOfftpRoutingRule` the files can be routed to new station INT2:

```
createOfftpRoutingRule --type INBOUND --newSid INT2
--originator EXT --destination INT1 --vdsn *
```

- Internal station INT2 (originator) transmits files to external station EXT (destination) via gateway ROU. Partnerstation INT1 is configured with EXT. The data of INT2 are unknown with EXT. Create the following routing rule to convert the transmitted stations data of routing send job from INT2 to INT1:

```
createOfftpRoutingRule --type OUTBOUND --newSid INT1
--originator INT2 --destination EXT --vdsn *
```

- All incoming jobs with VDSN="TEST" shall be routed to INT2. This can be obtained with the following command:

```
createOfftpRoutingRule --type INBOUND --newSid INT2
--destination INT1 --vdsn TEST
```

Hint:

Routing rules are to be created by the gateway station.

In case of using an OFTP routing rule no particular ERP job is generated.

No changes of settings are necessary at the external station.

Further rvscli commands concerning OFTP routing rules:

- `updateOfftpRoutingRule`
- `displayOfftpRoutingRule`
- `listOfftpRoutingRule`
- `deleteOfftpRoutingRule`

The parameters are described in the usages.

8 Maintenance of rvsEVO

In this chapter you find a description of the rvsEVO installation folder and information about required maintenance of directories and your stored data.

Via parameter group **MAINTENANCE** in Parameter perspective of rvsEVO GUI settings for automatic backup and cleanup can be configured. (see rvsEVO User Manual, chapter „Customizing the Global rvsEVO Parameters“ for more information).

8.1 rvsEVO Installation Folder

During installing process the following subdirectories are installed in **\$RVS_HOME** (see chapter 1.1 for further information) directory:

- **archive:** directory for archive and backup files
 - RevisionLog.xml and
RevisionLog.txt files contain the entries of processed send or receive jobs
 - jar files contain the backup data.
- **bin:** contains scripts (see chapter 'Command Line Tools' in rvsEVO User Manual for more information)
- **clib:** contains C-Libraries (only existing with Win32 installer)
- **conf:** contains configuration files
 - **conf/conversiontables:** conversion tables
- **db:** contains derby data base files (only existing with derby embedded data base)
- **docs:** contains readme file, user manual, operator manual, ClientAPI documentation
 - **conf/jobstart-examples:** examples of jobstart scripts
 - **docs/rvsEVOClientAPI:** ClientAPI libraries (javadoc, examples)
- **files:** contains the subdirectories used for transmission
 - **files/inbox:** default folder of received files
 - **files/outbox:** default folder of files to be transmitted
 - **files/temp:** default folder of temporary files for post- and preprocessing of send jobs
- **jobs:** after installing the folder is empty. Recommended for own scripts (e.g. for automating or logging processes) because the folder is saved during back up process
- **jre and subdirectories:** default folder of JRE, provided by rvsEVO installer (not available for genericUNIX)
- **lib:** contains rvsEVO libraries

- **log:** contains log files
 - `monitor.log`: current rvsEVO messages
 - `rvs.log`: StackTraces of rvsEVO server, error log
 - `rvsClient.log`: StackTraces of rvsEVO clients, error log
 - `rvsClientRCP.log`: StackTraces of rcpGUI, error log
 - `rvsEVO_Installieren_MM_DD_YYYY_hh_mm_ss.log`: installer log
- **log/trace/odette:** .trc files (Odette traces)
- **log/trace/tls:** .trc files (TLS traces)
- **log/trace/api:** .trc files (API traces)
- **rcpgui and subdirectories:** RCP GUI workspace and libraries
- **rvsapitest and subdirectories:**
- **system:** recommended directory for PKCS#12 files of OFTP2-Proxies
 - **system/conf:** configuration files that does not belong to EVO business logic. The files that are independent from a certain tenant/mandant. The files should not be edited.
 - **system/data:** contains subdirectory `/system/data/xsd` with XML shema definitions
 - **system/db:** contains generic scripts for creating, initialazing and removing of database tables.
 - This scripts use the SQL scripts of `/system/db/derby`, `/system/db/mssql` and `/system/db/oracle` subdirectory
- **tools and subdirectories:** contains scripts for maintenance of rvsEVO database.
 - Have a look at the rvsEVO User Manual, chapter „rvsEVO Database“ for detailed information of the scripts.
- **UninstallerData:** contains files needed for uninstalling rvsEVO.

8.2 Maintenance of Directories

The following directories are not cleared by rvsEVO and have to be maintained by the rvsEVO operator to prevent filling up of the directories. In parameter group RESOURCE_CHECK you can set the options for warnings and error messages in case of low disk space (see rvsEVO User Manual, chapter „Customizing the Global rvsEVO Parameters“ - Group RECOURCES CHECK).

- **archive:** .jar files of backup and RevisionLog files
- **files:** inbox, outbox, temp.
- Directory for Observer functionality.

Hint: In `rvs-system.properties` file and during creating send jobs you can set up whether the files in outbox and Observer directory should be kept after transmission or not.

- **log:** log files and subdirectories with trace files.

8.3 Maintenance of rvsEVO Database

The job and MonitoLog data must be maintained by the rvsEVO operator to prevent filling up of the database.

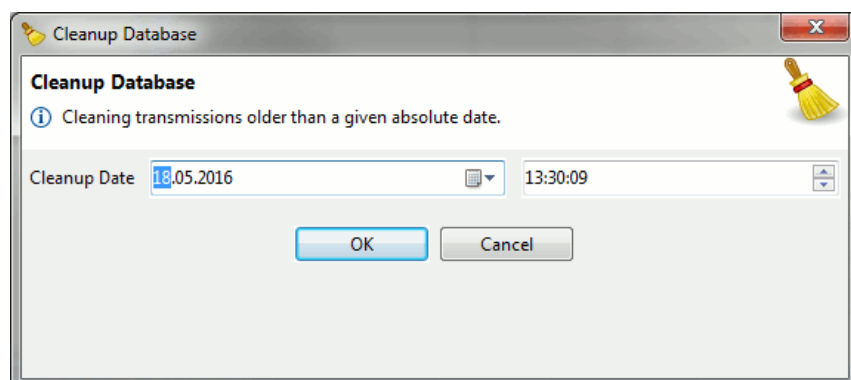
8.3.1 Deletion of Job Data from Database

There are several possibilities for deletion of job data:

- via Transfer perspective of rvsEVO GUI
- via `cleanup` cmd tool (see chapter 8.3.3 "Deletion of Jobs and Monitor Log Data from the Database")
- via MAINTENANCE parameter **CleanupDays**, **CleanupInterval**, **CleanupTime** (see rvsEVO User Manual, chapter „Customizing the Global rvsEVO Parameters“ - Group MAINTENANCE).
- via `rvscli` tool (see rvsEVO User Manual, chapter „Command Line Tools“).

rvsEVO GUI **Hint:** This function is only available for user with administrator or operator rights.

Click the **Cleanup** icon in the toolbar for deletion of ended and failed jobs from the database. The jobs which are older than the entered date and time will be deleted.



8.3.2 Deletion of Monitor Log Data from Database

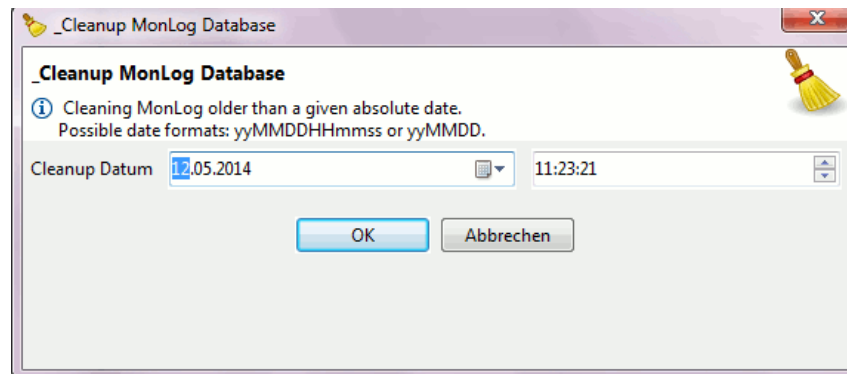
Monitor Log data are only saved in rvsEVO database if MAINTENANCE parameter **MonlogInDatabase** = `Y`

There are several possibilities for deletion of Monitor Log data:

- via Monitoring perspective of rvsRVO GUI

- via `cleanup` cmd tool (see chapter 8.3.3 "Deletion of Jobs and Monitor Log Data from the Database")
- via MAINTENANCE parameter **CleanupDays, CleanupInterval, CleanupTime, CleanupInterval, MonlogInDatabase, MonlogInDatabaseCleanup** (see rvsEVO User Manual, chapter „Customizing the Global rvsEVO Parameters“ - Group MAINTENANCE)
- via `rvscli` tool (see rvsEVO User Manual, chapter „Command Line Tools“)

rvsEVO GUI Single-click the `Cleanup Monlog` icon in the toolbar to open the dialog window for deletion of Monitor Log data. All entries are deleted which are older than defined date.



8.3.3 Deletion of Jobs and Monitor Log Data from the Database

Use the `cleanup` program to delete ended and failed transmission jobs or/and Monitor Log data which are older than the defined date.

Syntax:

```
cleanup -d <date> [-r] [-t <type>] [-help] [-?]
[-verbose|-v]
```

Required Parameter:

Parameter	Description
-d <date>	This option needs a date in the following format: <code>yyMMddHHmmss</code> or <code>yyMMdd</code> . The jobs or Monitor Log data, which are older than the entered date will be deleted from the database.

Optional Parameters:

Parameter	Description
-r	Remote, availables deletion during rvsEVO is running.

Optional Parameters:

Parameter	Description
-t -type	Type of entry to be deleted. Possible values: - JOBS (default): data of ended or failed jobs - MONLOG: Monitor Log data - ALL: jobs and Monitor Log data
-help	Requests help information.
-?	Displays a description of the current command.
-verbose (-v)	Verbose message output.

9 ODETTE Protocol

It is the purpose of the Odette File Transfer Protocol (OFTP) to ensure the reliable transfer of a data set. The OFTP enters a protocol session with the OFTP on the remote rvsEVO station which logically runs on top of the linedriver connection.

After the OFTP session has started, both sides exchange their Odette IDs and passwords, negotiate some parameters, like Odette exchange buffer size, Odette credit value (the number of buffers the sending side can send without waiting for a response), and exchange information about name, approximate size and format of the data set to be transferred.

During transfer, a compression and decompression of data is performed. After the data have been transferred, the byte count is checked between both sides. After the data set has successfully been stored, a receipt is sent to the sending station. If the transfer has been disrupted, for example by a link failure, the OFTP protocol provides a mechanism that allows to restart the transfer at the point of rupture.

'Change direction' feature allows the receiver to become sender and to send acknowledgments and data sets.

For protocol details kindly refer to the publications of the ODETTE and VDA groups: "ODETTE Specifications for File Transfer".

9.1 RFC 5024: OFTP Commands for OFTP File Transfer

Below you find the most important OFTP commands:

Session Establish-
ment

SSRM: Start Session Ready Message

This is the first command to be sent after a connection has been requested, indicating that the application is communicating with another OFTP.

SSID: Start Session InDication

This is a request for initiate a session. It contains information such user and password of the requester.

File Transfer

SFID: Start File InDication

This is a request for permission to send a file. It contains information such as the origin and destination of the file, its name and physical size.

SFPA: Start File Positive Answer

This is a positive acknowledgement for SFID request. Transfer is accepted.

SFNA: Start File Negative Answer

This is a negative acknowledgement for SFID request. Transfer is rejected.

DATA: The actual file data sent to the trading partner.

CDT: Set CreDiT

Service command for flow control.

EFID: End of File InDication

This command is sent immediately after the file has been transferred containing control totals to ensure the integrity of the sent file.

EFPA: End File Positive Answer

This is a positive acknowledgement for EFID request. Transfer is successfully completed.

EFNA: End File Negative Answer

This is a negative acknowledgement for EFID request. Transfer is cancelled.

CD: Change Direction

The CD command gives token to partner.

Confirm File
Reception

EERP: End to End ResPonse

The EERP is generated when a file reaches its ultimate destination. It is sent to the originator of the file to inform them that the file has been received.

NERP: Negative End ResPonse

The NERP is generated when a file cannot be transmitted to its ultimate destination. It is sent to the originator of the file to inform them that the file cannot be transmitted

Terminate Session

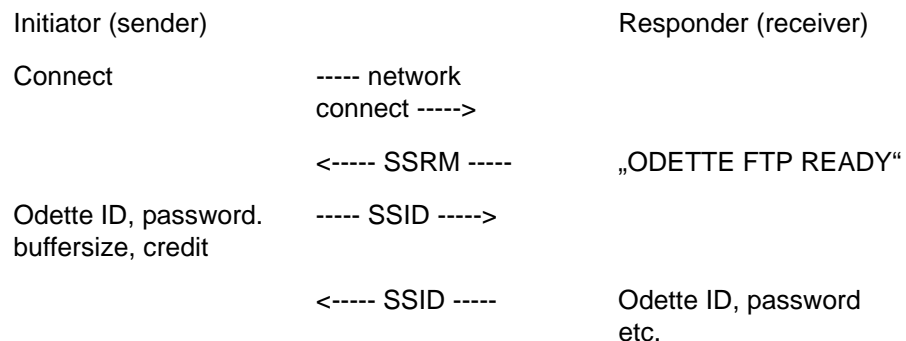
ESID: End Session InDication

The ESID requests that the communication session be terminated.

Please see chapter 11.7 "Failure during Build-up of OFTP Session to Neighbour Station" and following for information on ReasonCodes in case of failure.

9.1.1 Message Flow of Odette Session

Shown below is the general but simplified message flow within an Odette session. The sending side acts as initiator, the receiving side as responder.



StartFile ID (name, size, format)	----- SFID ----->	
	<----- SFPA -----	StartFile Positive Answer
'n' data records	----- DATA -----> ----- DATA -----> ----- DATA -----> ...	
	<----- CREDIT -----	send credit value 'n'
send 'n' data records	----- DATA -----> ----- DATA -----> . .	
EndFile ID (byte count)	----- EFID ----->	
	<----- EFPA ----- or <----- EFNA -----	EndFile Positive Answer (if store successful and byte count correct)/ EndFile Negative Answer
Change Direction	----- CD ----->	
	<----- EERP ----- or <----- NERP -----	End-to-End Response/Negative End-to-End Response (acknowledge)
	<----- ESID -----	End-Session ID
Network disconnect		Network disconnect

10 Monitoring of rvsEVO

In this chapter you find out how to display rvsEVO messages and warnings.

10.1 What Gets Logged

In rvsEVO, messages from rvsEVO server and rvsEVO GUI and the jobstart processes are logged by default and saved in log files.

Additionally the transmitted Odette data and the TLS and Proxy TLS connection is traced and saved in trace files, if configured.

10.2 Log Files

The log files contain the messages from rvsEVO server and rvsEVO GUI. They are saved in `$RVS_HOME/log` directory.

Please see chapter 7.3 "Customizing the Log Files" for information about setting up the maximum file size and maximum number of log files.

There are the following log files:

- `monitor.log` (monitor messages)
- `rvs.log` (error messages and warnings from rvsEVO server)
- `rvsClient.log` (messages from the command prompt)
- `rvsClientRCP.log` (messages of RCS GUIs)

10.2.1 monitor.log

The `monitor.log` file contains all messages which are displayed in rvsEVO GUI: Errors (ERR), warnings (WRN) and information (INF).

A new `monitor.log` file is generated daily and the old `monitor.log` file is renamed in „monitor.log“ plus date pattern plus counter. Also a new file is generated if the maximum file size is reached.

The list below shows the message sending rvsEVO modules:

rvsEVO Modules and its Kind of Information:

Module	Kind of Message
ACX	Automatic exchange of certificates
CTRL	Controller, e.g. cleanup, Proxy connection, start-up listeners, stopping receive channel
NET	Network, e.g. network connection information, expire of certificates
OBSR	Observer

rvsEVO Modules and its Kind of Information:

Module	Kind of Message
OFTP	Odette protocol, e.g. Odette session information
PROX	Proxy
SP	Service provider, e.g. pre- and post processing

The list below shows examples of short messages that are saved in `monitor.log` file. All short messages are explained in Text column. For information about error messages see chapter 11 "Error Situations of rvsEVO".

Messages from ACX module:

- ACX_CERT_FOR_REF_FOUND
- ACX_CERT_FOR_REF_NOT_FOUND
- ACX_CERTIFICATE_IMPORT_DENIED
- ACX_CERTIFICATE_NOT_IMPORTED
- ACX_CERTIFICATE_REPLACED
- ACX_CERTIFICATE_USAGE_ADOPTED
- ACX_COMMAND_FAILED
- ACX_DELIVER
- ACX_DELIVER_FAILED_MISSING_CERT
- ACX_DELIVER_WITH_REF
- ACX_IMPORT_SUCCESSFUL
- ACX_RECEIVE_DELIVER
- ACX_RECEIVE_DELIVER_DENIED
- ACX_RECEIVE_DELIVER_FAILED
- ACX_RECEIVE_DELIVER_REF_DENIED
- ACX_RECEIVE_DELIVER_WITH_REF
- ACX_RECEIVE_REPLACE
- ACX_RECEIVE_REPLACE_DENIED
- ACX_RECEIVE_REPLACE_FAILED
- ACX_RECEIVE_REPLACE_REF_DENIED
- ACX_RECEIVE_REPLACE_WITH_REF
- ACX_RECEIVE_REPLACE_WITHSFIDCERT
- ACX_RECEIVE_REQUEST
- ACX_REPLACE
- ACX_REPLACE_FAILED_INVALID_CERT

- ACX_REPLACE_WITH_REF
- ACX_REQUEST
- ACX_REQUEST_FAILED_MISSING_CERT

Messages from CTRL module:

- CREATE_SENDJOB
- JOB_SET_TO_SEND_NEXT
- JOB_SEND_NEXT_NOT_DONE
- PREPROCESSING_ENDED
- OUTGOING_CALL_INITIATED
- EERP_ASSIGNED
- TERMINATE_SENDJOB
- CLEANUP_STARTED
- REINIT_JOBLIST
- CLEANUP_COMPLETED
- SERVER_STARTED
- CERTIFICATE_EXPIRATION
- RES_NOT_AVAIL_DIR

Messages from NET module

- LISTENER_START_SUCCESS
- CONNECTION_CREATE_SUCCESS_1
- CONNECTION_CLOSE_SUCCESS
- INCOMING_CALL_RECEIVED

Messages from OBSR module:

- OBSERVER_STOPPED
- OBSERVER_STARTED
- OBSERVER_SE_CREATED

Messages from OFTP module:

- SENT_F_PDU
- RECEIVED_F_PDU
- SESSION_CREATED
- OFTP_TIMEOUT

Messages from PROX module:

- LISTENER_START_SUCCESS
- CONNECTION_CREATE_SUCCESS_1
- CONNECTION_CLOSE_SUCCESS

Messages from SP module:

- SERVICE_PROVIDER_CHAIN_STARTED
- START_CONVERSION2_OFFTP2FORMAT
- FINISH_CONVERSION2_OFFTP2FORMAT
- CMS_SIGNATURE_START
- CMS_SIGNATURE_FINISHED
- CMS_COMPRESSION_START
- CMS_COMPRESSION_FINISHED
- CMS_ENVELOPING_START
- CMS_ENVELOPING_FINISHED
- MESSAGE_DIGEST_BUILD_START
- MESSAGE_DIGEST_BUILD_FINISHED
- SERVICE_PROVIDER_CHAIN_FINISHED
- CMS_ERP_VERIFY_SIGNATURE_START
- CMS_ERP_VERIFY_SIGNATURE_FINISHED

Messages of server start:

- CLEANUP_STARTED
- OBSERVER_STARTED
- REINIT_JOBLIST
- CLEANUP_COMPLETED
- OUTGOING_CALL_INITIATED
- LISTENER_START_SUCCESS
- SERVER_STARTED

Messages of resource check:

- CERTIFICATE_EXPIRATION
- RES_NOT_AVAIL_DIR

Hint: Please see the rvsEVO User Manual, chapter „Monitoring“ for further information about Monitor Log.

10.2.2 rvs.log

The `rvs.log` file contains error messages and warnings from rvsEVO server. In detail the following information are saved ...

- ... with each start-up of rvsEVO server:
 - java.properties
 - rvsEVO properties
 - license validation data

- ... in on-going operation processes:
 - error messages from rvsEVO server, by default
 - warnings from rvsEVO Server, if configured

The messages in `rvs.log` are also displayed via console.

Please see chapter 11 "Error Situations of rvsEVO" for more information about error messages and warnings.

10.2.3 rvsClient.log

The warnings and error messages of rvsEVO SWING-GUI and of command line tools are saved in `$RVS_HOME/log/rvsClient.log` file.

10.2.4 rvsClientRCP.log

The warnings and error messages of rvsEVO RCP-GUI are saved in `$RVS_HOME/log/rvsClientRCP.log` file.

10.3 Tracing

The trace files are saved in `$RVS_HOME/log/trace` directory.

10.3.1 Odette Tracing

Hint: Please see the rvsEVO User Manual, chapter „Log and Trace Files“ for information about enabling Odette tracing.

The Odette trace files contain the transmitted Odette data packages. Below you find a description of the saved Odette parameters:

- | | |
|---------------|--|
| Start Session | <ul style="list-style-type: none"> – SSIDLEV: OFTP Release – SSIDCODE: Initiator / Responder Odette ID – SSIDPSWD: Receive / Send password – SSIDSDEB: Exchange Buffer Size – SSIDSR: Transmission capability (Send, Receive, Both) – SSIDCMPR: Compression – SSIDREST: Restart – SSIDSPEC: Special Logic – SSIDCRED: Exchange Buffer Credit – SSIDAUTH: Authentication – SSIDRSV1: No function – SSIDUSER: Special user information – SSIDCR: Carriage return |
| Start File | <ul style="list-style-type: none"> – SFIDDSN: VDSN |

- SFIDRSV1: Without function
- SFIDDATE: SFID Date
- SFIDTIME: SFID Time
- SFIDUSER: Special user information
- SFIDDEST: Destination Odette ID
- SFIDORIG: Originator Odette ID
- SFIDFMT: Format
- SFIDLRECL: Record Length
- SFIDFSIZ: File Size
- SFIDOSIZ: Original File Size
- SFIDREST: Restart Position
- SFIDSEC: Security Feature Set
- SFIDCIPH: Encryption Algorithm
- SFIDCOMP: Compression
- SFIDENV: Enveloping format
- SFIDSIGN: Signed EERP/NERP requested
- SFIDDESCL: Length of File Description
- SFIDDESC: File Description
- Start File Positive Answer - ACNT: Answer Count (for Restart Position)
- EFPACD: Change directory
- End File - EFIDRCNT: Record count
- EFIDUCNT: Transmitted Bytes
- End to End Response - EERPDSN: VDSN
- EERPERSV1: No function
- EERPDATE: Date (YYYYMMDD) for file date stamp
- EERPRTIME: Time counter (hhmmsscccc) for file time stamp
- EERPUSER: Special user information
- EERPDEST: Odette ID of destination
- EERPORIG: Odette ID of originator
- EERPHSHL: File hash length with signed EERP
- EERPHSH: File hash
- EERPISGL: Length of EERP signature
- EERPISIG: EERP signatur
- End Session - ESIDREAS: Reason code

- ESIDREASL: Length of reason text
- ESIDREAST: Reason text
- ESIDCR: Carriage return

10.3.2 TLS Tracing

Please see the rvsEVO User Manual, chapter „Log and Trace Files“ for information about enabling TSL tracing.

10.3.3 API Tracing

API tracing function allows tracing of methods of the following API classes:

- Configuration
- JobList
- KeyList
- RVS
- StationList
- UserList
- UserManager
- UserProfile

See the javadoc in `$RVS_HOME/docs/rvsEVOClientAPI/rvs-evo-api-6.01.07-javadoc.jar` for more information about the API classes and its methods.

Enable API tracing function via definition

```
rvs_evo.tracing.api.enabled = true in $RVS_HOME/conf/  
rvs-system.properties file.
```

Configure API tracing function via definition

```
rvs_evo.tracing.api.subjects = *.* in rvs-system.properties  
file. (See also chapter 7.1 "Customizing rvsEVO via rvs-system.proper-  
ties File").
```


11 Error Situations of rvsEVO

This chapter presents error situations with related entries in `monitor.log`, `rvsClient.log`, `rvsClientRCP.log` and `rvs.log` and proposes possible solutions. The `monitor.log` entries are also displayed via rvsEVO GUI and the `rvs.log` entries can be followed via console.

11.1 Debugging the Installer

Before starting the installer, set the environment variable **LAX_DEBUG** to record the error messages to the console.

Linux / Unix:

```
export LAX_DEBUG=true
```

Windows:

```
set LAX_DEBUG=true
```

11.2 Failure of Installation

11.2.1 Wrong Permission of DB Scripts

Rights for DB scripts were set wrong by the installer.

Error:

```
/tmp/174067.tmp/executeScriptTmp0 [2]: /home/qsevo/  
rvsEVO/system/db/create.sh:
```

```
0403-006 Execute permission denied.
```

Reason: The user temp directory respectively **\$IATEMPDIR** (see rvsEVO User Manual, chapter „New Installation of rvsEVO“ for more information) does not have enough free space.

Solution:

New installation: Clean up your temp directory, uninstall / delete your rvsEVO installation and start the installer again.

Update installation: Clean up your temp directory, recover your rvsEVO backup which was created before update installation and start the installer again.

11.2.2 Exception during Installation of rvsEVO and Executing DB Script „schema-create.sql“

Error:

```
A JNI error has occurred, please check your installa-  
tion and try again
```

```
Exception in thread "main" java.lang.NoClassDefFound-  
Error:org.apache.tools.ant.BuildException at
```

```

java.lang.J9VMInternals.prepareClassImpl (Native
Method) at
java.lang.J9VMInternals.prepare (J9VMInternals.java:29
1)
at java.lang.Class.getMethod (Class.java:1216)
at sun.launcher.LauncherHelper.validateMain-
Class (LauncherHelper.java:556)
at sun.launcher.LauncherHelper.checkAndLoadMain (Laun-
cherHelper.java:538)

Caused by: java.lang.ClassNotFoundException:
org.apache.tools.ant.BuildException
at java.net.URLClassLoader.findClass (URLClas-
sLoader.java:609)
at java.lang.ClassLoader.loadClassHelper (Clas-
sLoader.java:850)
at java.lang.ClassLoader.loadClass (Clas-
sLoader.java:829)
...

```

Reason: The user temp directory respectively **\$IATEMPDIR** (see rvsEVO User Manual, chapter „New Installation of rvsEVO“ for more information) does not have enough free space.

Solution:

New installation: Clean up your temp directory, uninstall / delete your rvsEVO installation and start the installer again.

Update installation: Clean up your temp directory, recover your rvsEVO backup which was created before update installation and start the installer again.

11.3 Aborting of Installation on Unix Systems

Occasionally the installer aborts during the launching process after the messages below:

```

[qsevo@q4de3esy426:~/installer]$ ./rvsEVO-setup-
6.01.08-GenericUnix.bin

Preparing to install...

Extracting the installation resources from the
installer archive...

Configuring the installer for this system's environ-
ment...

Launching installer...

[qsevo@q4de3esy426:~/installer]$

```


Solution: Make the following Install Anywhere setting via environment variable **DO_NOT_FORK**:

```
export DO_NOT_FORK=true
```

Thereafter the installation should be executed completely.

11.4 Failure of rvsEVO Server Start

License over

monitor.log:

```
2016-11-16 08:19:32,462 | ERR | CTRL | LICENSE_OVER
|                               |                               |
The license [{0}] has been expired.
... [ERROR ] [main ]
```

rvs.log:

```
2016-11-16 08:19:32,459 [DEBUG ] [main      ]
com.tsystems.rvs.evo.controller.licenses.LicenseCon-
troller.?( ): Validation Result           : [NOT VALID]
2016-11-16 08:19:32,459 [ERROR ] [main      ]
com.tsystems.rvs.evo.controller.licenses.LicenseCon-
troller.?( ): License over!
2016-11-16 08:19:32,459 [DEBUG ] [main      ]
com.tsystems.rvs.evo.controller.licenses.LicenseCon-
troller.?( ): <-----LICENSE-----VALIDA-
TION-----
2016-11-16 08:19:32,665 [ERROR ] [main      ]
com.tsystems.rvs.evo.EVO.?( ): Error validating
license. Server startup interrupted.
```

Reason: License is expired

Solution: Save a valid license file `license.properties` in `$RVS_HOME/conf` directory.

11.4.1 Failure of rvsEVO Server Start with Derby Database

Every process accessing the derby database locks the derby data base exclusive: no other processes can access to the database.

server is already
running:

rvs.log:

```
* initialising application ...
[ERROR ] org.hibernate.tool.hbm2ddl.SchemaUpdate.?( ):
HHH000319: Could not get database metadata
java.sql.SQLException: Failed to start database '/
home/qsevo/rvsEVO/db'
with class loader sun.misc.Launcher$AppClas-
sLoader@5c647e05, see the next
exception for details.
```

Caused by: ERROR XSDB6: Another instance of Derby may have already booted the database /home/gsevo/rvsEVO.6.01.03.RC2/db.

Reason: rvsEVO server is already running.

Solution: Stop the running rvsEVO server and start the rvsEVO server again.

RMI port in use

rvs.log:

```
java.rmi.server.ExportException: Port already in use:
3755; nested exception is:
```

```
java.net.BindException: Address already in use:
JVM_Bind
```

Reason: RMI Port is in use.

Solution: If the port is not used by another process: restart your system.

If the port is used by another process: Change the RMI port in rvsEVO configuration like described below:

- export the rvsEVO configuration via `exportConfiguration` tool.
- change the value of **RMIServicePort** parameter in group **BASIC** in the xml file.
- import the xml file via `importConfiguration` tool.

Please see the rvsEVO User Manual, chapter „Database Management“ for description of the tools.

Warning without effect on rvsEVO functions

During the server start the warning below can be output. This does not have any effect on the rvsEVO functions.

```
* checking database connection...
```

```
2017-11-14 16:30:16,028 [WARN ] com.hazelcast.nio.UnsafeHelper.?(): sun.misc.Unsafe has been disabled because your platform does not support unaligned access to memory, some features might be not available.
```

11.5 Failure of Listener Start

11.5.1 Port is already in use

MSSQL and Oracle database

rvs.log:

```
2016-11-21 13:52:02,090 [ERROR ] [10.33.67.220]
com.gedas.rvs.tiny.network.tls.TLSListener.?(): run:
Exception java.net.BindException: Cannot assign
requested address: JVM_Bind
```

monitor.log:

```
2016-11-21 13:52:02,180 | WRN | NET |
LISTENER_START_FAILURE |
| Failed to start TLS Listener,
system tried the following parameters: ip=/
10.33.67.220, port=6619. Reason: Message: Cannot
assign requested address: JVM_Bind. Cause: BindExcep-
tion
```

```
2016-11-21 16:54:57,862 | WRN | NET |
LISTENER_START_FAILURE |
| Failed to start TCPIP Listener,
system tried the following parameters: ip=RDDE9D84/
10.177.191.220, port=3307. Reason: Message: Address
already in use: JVM_Bind. Cause: BindException
```

Reason: More than one TLS-listeners are generated with the same port number

Solution: Change the port number of the added listener

11.6 Connection Failure

11.6.1 Connection to Proxy Bastion Fails

Bastion not active **rvs.log:**

```
2016-11-22 15:25:31,744 [ERROR ] [RemoteCallback-
Handler] com.tsystems.rvs.evo.network.remote.call-
back.CallbackHandlerDelegate.?): Receiving of
requests got interrupted. - Unable to connect either
the primary nor the backup proxy.
```

```
2016-11-22 15:25:31,745 [ERROR ] [main      ]
com.gedas.rvs.tiny.network.impl.ResetBastion-
Rule.?): Unable to connect to proxy. (RemoteNetwork-
Host [hostName=RDDE9D84, port=3762,
serviceName=service, command=, backupHostName=,
backupPort=0, rmiSenderId=RDDE9D84, remoteClass-
Name=com.gedas.rvs.evo.network.remote.rmi.server.Remo-
teNetworkManagerImpl, tlsEnabled=false, trustStoreFi-
leName=])
```

monitor.log:

```
2016-11-22 15:25:58,486 | ERR | CTRL |
PROXY_CHECK_FAILED |
| End-to-end check of proxy 1 failed.
```

Reason: Proxy Bastion was not started.

Solution: Start the Proxy Bastion first and restart the rvsEVO server

No valid certificate **rvs.log:**

```
2017-08-08 16:09:50,444 [ERROR ] [taskScheduler-1]
com.tsystems.rvs.evo.network.remote.callback.Call-
backHandlerDelegate.?): Unable to check proxy connec-
tion. Unable to connect either the primary nor the
backup proxy.
```

```
2017-08-08 16:09:53,994 [ERROR ] [RemoteCallback-
Handler]
com.tsystems.rvs.evo.controller.networks.RemoteNet-
workHostExceptionHandler.?): Unable to connect to
RemoteNetworkHost [hostName=RDDE9D84, port=2600,
serviceName=service, command=, backupHostName=,
backupPort=0, rmiSenderId=RDDE9D84, remoteClass-
Name=com.gedas.rvs.evo.network.remote.rmi.server.Remo-
teNetworkManagerImpl, tlsEnabled=true, trustStoreFi-
leName=C:\rvsRT\tenants\4001\system\rmiTlsTrustS-
tore.p12]. NotAfter: Mon Aug 07 16:02:05 CEST 2017
```

monitor.log:

```
2017-08-08 16:09:50,446 | ERR | CTRL |
PROXY_CHECK_FAILED |
| End-to-end check of proxy 1 failed.
```

Reason: Certificate is expired.

Solution: Import a valid keypair into rvs® OFTP Proxy keystore and exchange the certificate in rvsEVO used for rvs® OFTP Proxy connection (Please see rvs® OFTP Proxy User Manual, chapter „Setting up of a TLS Connection“ for more information.)

Wrong bastion host/
port

rvs.log:

```
2016-11-22 15:42:18,530 [ERROR ] [main      ]
com.gedas.rvs.tiny.network.impl.ResetBastion-
Rule.?): Unable to connect to proxy. (RemoteNetwork-
Host [hostName=RDDE9D84, port=3765,
serviceName=service, command=, backupHostName=,
backupPort=0, rmiSenderId=RDDE9D84, remoteClass-
Name=com.gedas.rvs.evo.network.remote.rmi.server.Remo-
teNetworkManagerImpl, tlsEnabled=false, trustStoreFi-
leName=])
```

monitor.log:

```
2016-11-22 15:42:45,270 | ERR | CTRL |
PROXY_CHECK_FAILED |
| End-to-end check of proxy 2 failed.
```

Reason: The Proxy Bastion what was set up in rvsEVO is not available.

Solution: Check and correct the Proxy Bastion Parameters **Server** and **Host** in Stations perspective.

11.6.2 Examples with outgoing call, TCP Connection

monitor.log:

Wrong hostname **Message box (Activate Station):**

```
Unable to activate station 1.
Message: Unable to activate station 1.
Details:
Cause: Connect Error: Hostname 'beldy430-123' can not
be resolved.
```

monitor.log:

```
2019-04-04 18:31:37,575ERRrvsEVO_NodeNET
CONNECTION_CREATE_FAILURE_104183135300001Failed to
create TCPIP Connection, receiver: ip=Host: beldy430-
123 [lookup result => unknown host], port=3321,
sender: ip=beldy430/10.181.183.111, port=3305.
Reason: UnknownHostException/beldy430-123
```

Solution: Check and correct the hostname of your partner station;
contact your partner

Host not available **Message box (Activate Station):**

```
Unable to activate station MAIN. Message: Unable to
activate station MAIN.
```

Primary connection fails:

```
Details:
Cause: Connect Error: Primary Connection: IP connec-
tion failed for '10.33.156.111' 10.33.156.111:3305 ;
```

Alternative connection fails:

```
Alternative Connection: Hostname '11.33.222.333' can
not be resolved.
```

monitor.log:

Primary connection fails:

```
2019-04-29 13:35:29,066 | WRN | rvsEVO_Node |
NET | CONNECTION_CREATE_PRIMARY |
2913350830000 | MAIN | Failed to create
primary TCPIP Connection, receiver: ip=/10.33.156.111,
port=3305, sender: ip=beldy430/10.180.68.154,
port=3331. Reason: CONNECTION_FAILED Connection timed
out: connect.
```

Alternative connection fails:

```
2019-04-29 13:35:31,319 | ERR | rvsEVO_Node |
NET | CONNECTION_CREATE_FAILURE_1 |
2913350830000 | MAIN | TCPIP Connection
```

```
could not be created. Receiver: ip=Host: 11.33.222.333
[lookup result => unknown host], port=3305, Sender:
ip=beldy430/10.180.68.154, port=3331, Reason:
CONNECTION_FAILED Connection timed out: connect.
```

Test connection (example for Linux systems):

```
-bash-3.2$ telnet 10.33.156.111 3305
Trying 10.33.156.111...
telnet: connect to address 10.33.156.111: Connection
timed out
telnet: Unable to connect to remote host: Connection
timed out
```

Reason: Host is not available; neither primary IP address/hostname nor alternative connection

Solution: Check and correct the hostname of your partner station; contact your partner.

Port not available

Message box (Activate Station):

```
Unable to activate station 1.
Message: Unable to activate station 1.
Details:
Cause: Connect Error: IP connection failed for
'10.33.108.246' 10.33.108.246:3323
```

monitor.log:

Primary and alternative connection fail:

```
2016-11-23 10:42:38,757 | WRN | NET |
CONNECTION_CREATE_PRIMARY | 2310423730000 |
PS1 | Failed to create
primary TCP/IP Connection, receiver: ip=/10.33.146.75,
port=3309, sender: ip=RDDE9D84/10.33.231.43,
port=3307. Reason: ConnectException/Connection
refused: connect

2016-11-23 10:42:39,788 | ERR | NET |
CONNECTION_CREATE_FAILURE_1 | 2310423730000 |
PS1 | Failed to create
TCP/IP Connection, receiver: ip=/10.33.146.75,
port=3308, sender: ip=RDDE9D84/10.33.231.43,
port=3307. Reason: ConnectException/Connection
refused: connect
```

Reason: Neither primary indicated port nor alternative one is available

Solution: Check and correct the port of your partner station; contact your partner.

Wrong Odette ID **Wrong entry of Odette ID of your partner station**

monitor.log:

```
2016-11-24 08:41:58,270 | ERR | OFTP | SENT_F_PDU
| 2408415730000 | PS1 |
ESID - End Session ID sent. REAS=03 (User code not
known), REAST=""
```

```
2016-11-24 08:41:58,391 | INF | NET |
CONNECTION_CLOSE_SUCCESS | 2408415730000 |
PS1 | TCPIP Connection
was successfully closed, parameters: ip=10.33.146.75,
port=3306, localport=60324.
```

Reason: The created OFTP session is closed because of wrong entry of Odette ID in **Odette ID** parameter of partner station.

Solution: Check and correct the Odette ID of your partner station. Contact your partner.

Own Odette ID is unknown with the Partner station

monitor.log :

```
2019-04-04 11:41:31,226 INF rvsEVO_Node OFTP
SENT_F_PDU 0411413130000 1 SSID - Start Session ID
sent. OdetteID="OMAIN", Auth="N"
```

```
2019-04-04 11:41:31,233 ERR rvsEVO_Node OFTP
RECEIVED_F_PDU 0411413130000 1 ESID - End Session ID
received. REAS=03 (User code not known), REAST=""
```

Reason: The created OFTP session is closed because of wrong entry of our Odette ID in **Odette ID** parameter with the partner.

Solution: Communicate your correct Odette ID to your partner.

11.6.3 Examples with outgoing call, TLS Connection

Wrong IP address /
port

monitor.log:

```
... | ERR | PROX | CONNECTION_CREATE_FAILURE_1 | ... |
Failed to create TLS Connection,
```

```
receiver: ip=/10.33.101.122, port=6700, sender: ip=/
10.33.137.38, port=6720. Reason: ConnectionException/
Exception during creation of TLS socket
```

Reason: Connection to partner station fails

Solution: check and correct IP address, port and certificate of your partner

Missing partner certificate on receiver side

monitor.log:

```
2016-11-30 11:20:53,758 | INF | CTRL |
OUTGOING_CALL_INITIATED | 3011205330000 |
TLSN | | Outgoing TLS call
to TLSN initiated, parameters: ip=10.33.146.75,
port=6621, Alternative: ip=10.33.146.75, port=6621

2016-11-30 11:20:53,897 | INF | SP |
CERTIFICATE_VALIDATION_RESULT | 3011205330000 |
| | [VALID] [TLS_SERVER] CERT_PATH
validation of certificate [Subject:CN=neups SerialN-
umber:5835ad75]

2016-11-30 11:20:53,941 | ERR | NET |
CONNECTION_CREATE_FAILURE_1 | 3011205330000 |
TLSN | | Failed to create
TLS Connection, receiver: ip=/10.33.146.75, port=6621,
sender: ip=RDDE9D84/10.33.231.65, port=6622. Reason:
ConnectionException/Exception during handshake[TLS]
```

rvs.log:

```
2016-11-30 11:20:53,940 [ERROR ] [StartSessionTask:
TLSN] com.gedas.rvs.tiny.network.tls.TLSConnec-
tion.?(): Handshake failed

javax.net.ssl.SSLHandshakeException: Received fatal
alert: bad_certificate
```

Reason: Connection is build-up but TLS handshake fails, because of missing partner certificate on receiver side.

Solution: contact your partner; deliver your certificate to your partner

Receiver does not exist

monitor.log:

```
2016-11-23 13:32:40,676 | ERR | NET |
CONNECTION_CREATE_FAILURE | 2313324030000 |
TLS | | Failed to create
TLS Connection, system tried the following parameters:
ip=/10.33.146.75, port=6619. Reason: ConnectionExcep-
tion/Receiver number 1 does not exists.
```

Reason: Number of Associated Receiver Parameter of partner station is not correct.

Solution: Check and correct the **Number of Associated Receiver Parameter**; contact your partner.

11.7 Failure during Bild-up of OFTP Session to Neighbour Station

During build-up, transfer and end of OFTP session several indications are exchanged with Odette protocol.

If the building-up fails frequently the ReasonCode of **ESID** (End-Session-Indication) shows the cause of failure. rvsEVO displays the **ESID** ReasonCode in the Monitor Log:

Examples **Wrong receive password:**
Message box (Activate Station):

```
Unable to activate station 1.
Message: Unable to activate station 1.
Details:
Cause: Wrong password received.
```

monitor.log:

```
2019-04-04 17:58:39,246ERRrvsEVO_NodeOFTPSENT_F_PDU
04175839300001ESID - End Session ID sent. REAS=04
(Invalid password), REAST=""
```

Wrong send password:

Message box (Activate Station):

```
Unable to activate station 1.
Message: Unable to activate station 1.
Details:
Cause: OFTP Session aborted with ESID received: 04
(Invalid password)
```

monitor.log:

```
2019-04-04 18:04:53,818ERRrvsEVO_NodeOFTP
RECEIVED_F_PDU04180453300001ESID - End Session ID
received. REAS=04 (Invalid password), REAST=""
```

Solution: contact your partner, square the passwords and adjust the password settings in OFTP parameter group of your partner station.

Codes **Below you find a list of the most important ESID ReasonCodes (rfc5024):**

- 00: Normal session termination
- 01: Command not recognised
- 02: Protocol violation
- 03: User code not known SSIDCODE (odette ID) unbekannt
- 04: Invalid password SSIDPSWD (SEND Passwort) stimmt nicht mit RECEIVE Passwort des responders überein
- 11: Invalid challenge response
- 12: Secure authentication requirements incompatible
- 99: Unspecified Abort code

11.8 Failure of File Transmission

11.8.1 SFNA and EFNA

In case of failure of file transmission the Start File Negative Answer (**SFNA**) from sender or End File Negative Answer (**EFNA**) from receiver is displayed in the Monitor Log.

Below you find a list of SFID and EFID ReasonCodes (rfc5024) :

- 01: Invalid filename. (invalid symbol in VDSN)
- 02: Invalid destination
- 03: Invalid origin.
- 04: Storage record format not supported.
- 05: Maximum record length not supported.
- 06: File size is too big.
- 10: Invalid record count.
- 11: Invalid byte count.
- 12: Access method failure.
- 13: Duplicate file.
- 14: File direction refused.
- 15: Cipher suite not supported. (Please see also chapter 11.8.2)
- 16: Encrypted file not allowed.
- 17: Unencrypted file not allowed.
- 18: Compression not allowed.
- 19: Signed file not allowed.
- 20: Unsigned file not allowed.
- 99: Unspecified reason.

The codes below concern only EFID:

- 21: Invalid file signature.
- 22: File decryption failure.
- 23: File decompression failure.

11.8.2 SFNA: REAST=15 „Cipher suite not supported“

Reason: You get this error in case of incompatible settings.

With encrypted file transfer the settings of OFTP parameter **Hash Algorithm** of your partner station and send parameter **Encryption Algorithm** must be compatible with the settings of your partner.

Solution: Enable Odette tracing and send a test file to your partner with setting **Security Feature Set=4**. You can check the required settings in

the trace file, SFID parameter **SFIDCIPH** (Cipher Suite). Please see chapter 10.3.1 "Odette Tracing" and rvsEVO User Manual, chapter "Log and Trace Files" for more information about Odette tracing.

The following table shows the required settings for **SFIDCIPH** 1 - 6:

Cipher Suite	Encryption Algorithm	Hash Algorithm
01	DES_EDE3_CBC	SHA-1
02	AES256_CBC	SHA-1
03	DES_EDE3_CBC	SHA-256
04	AES256_CBC	SHA-256
05	DES_EDE3_CBC	SHA-512
06	AES256_CBC	SHA-512

11.8.3 Error Messages via NERP

The end of the transmission via OFTP is confirmed with End to End Response (**EERP**) or Negative End Response (**NERP**). NERP means that the transmission is ended with error.

Below you find a list of NERP ReasonCodes (rfc5024) :

- 03: ESID received with reason code '03' (user code not known)
- 04: ESID received with reason code '04' (invalid password)
- 09: ESID received with reason code '99' (unspecified reason)
- 11: SFNA(RETRY=N) received with reason code '01' (invalid file name)
- 12: SFNA(RETRY=N) received with reason code '02' (invalid destination)
- 13: SFNA(RETRY=N) received with reason code '03' (invalid origin)
- 14: SFNA(RETRY=N) received with reason code '04' (invalid storage record format)
- 15: SFNA(RETRY=N) received with reason code '05' (maximum record length not supported)
- 16: SFNA(RETRY=N) received with reason code '06' (file size too big)
- 20: SFNA(RETRY=N) received with reason code '10' (invalid record count)
- 21: SFNA(RETRY=N) received with reason code '11' (invalid byte count)

- 22: SFNA(RETRY=N) received with reason code '12' (access method failure)
- 23: SFNA(RETRY=N) received with reason code '13' (duplicate file)
- 24: SFNA(RETRY=N) received with reason code '14' (file direction refused)
- 25: SFNA(RETRY=N) received with reason code '15' (cipher suite not supported)
- 26: SFNA(RETRY=N) received with reason code '16' (encrypted file not allowed)
- 27: SFNA(RETRY=N) received with reason code '17' (unencrypted file not allowed)
- 28: SFNA(RETRY=N) received with reason code '18' (compression not allowed)
- 29: SFNA(RETRY=N) received with reason code '19' (signed file not allowed)
- 30: SFNA(RETRY=N) received with reason code '20' (unsigned file not allowed)
- 31: File signature not valid.
- 32: File decompression failed.
- 33: File decryption failed.
- 34: File processing failed.
- 35: Not delivered to recipient.
- 36: Not acknowledged by recipient.
- 50: Transmission stopped by the operator.
- 90: File size incompatible with recipient's protocol version.
- 99: Unspecified reason.

11.8.4 Outgoing Call, EERP is Missing

The file is delivered to the partner; the send job goes in state WF_EERP

Reason: The setting of **EERP Out** parameter of partner station on receiver side is NEVER or HOLD

Solution: contact the responsible person of your partner station

11.9 File Transmission: Pre- or Post-Processing Fails

11.9.1 Key or Certificate is Missing

In case of failure with pre- or post-processing the rvsEVO module ServiceProvider (SP) writes the reason in `monitor.log` file. The job stays in

ACTIVE category and the transfer can be finished after removing the problem's cause (e.g. missing or expired certificate).

Example:

monitor.log:

```
2016-11-15 14:53:53,717 | WRN | SP |
CMS_UNWRAPPING_ENVELOPE_FAILED |
PS1 | 1611151453530000 | Failed decryption
of jobId [1611151453530000] file
C:\rvsEVO\files\temp\1611151453530000: Key not found
for received serial number 58087EFE (alias=LOC).
```

```
2016-11-15 14:53:53,723 | WRN | SP |
SERVICE_PROVIDER_CHAIN_FINISHED_WIT |
PS1 | 1611151453530000 | Finished service
CMS_RCV provider chain with error.
```

```
2016-11-15 14:31:06,810 | ERR | CTRL |
POSTPROCESSING_FAILED |
PS1 | 1611151431060000 | Postprocessing
failed (JobID=1611151431060000).
```

rvs.log:

```
2016-11-15 14:31:06,822 [ERROR ] [PostProcessingTask]
com.tsystems.rvs.evo.controller.jobs.JobCon-
troller.?(): Postprocessing of ReceiveJob
1611151431060000 failed. JobDataError
```

Reason: missing keypair of local station of destination

Solution: associate the corresponding key pair to local station and activate **Restart SP**.

Hint: In case of problems on sender side the job must be restarted by the sender.

11.9.2 Jobstart After Receive Fails on Receiver Side

Outgoing call: file transmission is finished but the send job terminates with state FAILED_WITH_NERP:

```
2016-11-30 15:08:23,514 | ERR | OFTP | RECEIVED_F_PDU
| 3015082230000 | PS1 |
NERP - Negative End to End Response received.
DEST="MYOIDPS", ORIG="OPT2", VDSN="INPUT12.TXT",
DATE=20161130, TIME=1508220001, CREA="OPT2", REAS=34,
REAST="Jobstart after receive failed. File was deli-
vered."
```

Reason: Jobstart process on receiver side (partner) fails.

Solution: Contact your partner

11.9.3 Failure of OFTP2 Conversion with Cluster Functionality on Linux and UNIX Systems

Pre-processing fails on files in Observer directory with the below messages:

monitor.log:

```
../log/monitor.log:2016-06-12 12:25:54,669 | INF | SP
| START_CONVERSION2_OFFTP2FORMAT |
SFS4.EXTERN | 1606111126200001 | Starting conver-
sion of file /tbshare/lxcluster/2observe/SFS4.EXTERN/
T13.s4yd.251.1465637180395.gen to offtp2format.
```

```
../log/monitor.log:2016-06-12 12:25:54,672 | WRN | SP
| OFFTP2_CONVERSION_FAILED |
SFS4.EXTERN | 1606111126200001 |
!OFFTP2_CONVERSION_FAILED!
```

rvs.log:

```
2016-06-12 12:25:54,673 [ERROR ] [PreProcessingTask]
com.gedas.rvs.tiny.serviceprovider.command.conver-
sion.Convert2OFFTP2FormatCommand.?(): MONITORLOG:
Formating for OFFTP2 failed
```

```
java.io.FileNotFoundException: /tbshare/lxcluster/
2observe/SFS4.EXTERN/T13.s4yd.251.1465637180395.gen
(Keine Berechtigung)
```

Reason: The instance owner has no permission on .gen file in Observer directory.

Solution: Set the permissions for the .gen file via `chmod` command (`chmod 666 <file>`) and restart the send job.

(Please see also chapter 5.2 "Setting up of a Server Farm" --> "Server Farm and Observer Functionality")

11.10 Failure of Connection from rvsEVO Client to rvsEVO Server

11.10.1 Failure of Connection to Server on Linux Systems

Error message:

```
RMI Exception 'java.rmi.ConnectException' caught.
```

```
Details: First access to rvsEVO server (RCRvs.getRan-
domString) ==> take care about serverside binding of
rmi objects (remote-reference: RSRvs_Stub[UnicastRef
[liveRef:
[endpoint:[127.0.1.1:3755] (remote),objID:[2c693759:16
2b4aba17a:-7fff, -3854998514233510659]]]])
```

Cause: Connection refused: connect

```
com.tsystems.rvs.client.gui.common.exceptions.RVSUIException: Message: RMI Exception
'java.rmi.ConnectException' caught.

Details: First access to rvsEVO server (RCRvs.getRandomString) ==> take care about serverside binding of
rmi objects (remote-reference: RSRvs_Stub[UnicastRef [liveRef:
 [endpoint: [127.0.1.1:3755] (remote), objID: [2c693759:16
2b4aba17a:-7fff, -3854998514233510659]]]])

Cause: Connection refused: connect at
com.tsystems.rvs.client.gui.remote.Rvs.login(Rvs.java
:415)

    at com.tsystems.rvs.client.gui.remote.RvsConnector.<init>(RvsConnector.java:143)

    at com.tsystems.rvs.client.gui.remote.RvsConnector.access(RvsConnector.java:226)

    at com.tsystems.rvs.client.gui.common.handler.LoginHandler.checkLogin(LoginHandler.java:142)

    at com.tsystems.rvs.client.gui.common.handler.LoginHandler.login(LoginHandler.java:89)

    at com.tsystems.rvs.client.gui.coreServices.Login.performLogin(Login.java:79)

[...]

Cause: Connection refused: connect

    at com.gedas.rvs.com.rmi.client.RCRvs.getRandomString(RCRvs.java:421)

    at com.gedas.rvs.client.api.Rvs.login(Rvs.java:732)

    at
com.tsystems.rvs.client.gui.remote.Rvs.login(Rvs.java
:363)

    ... 40 more

Server Exception:

java.rmi.ConnectException: Connection refused to
host: 127.0.1.1; nested exception is:

        java.net.ConnectException: Connection
refused: connect

    at sun.rmi.transport.tcp.TCPEndpoint.newSocket(Unknown Source)

    at sun.rmi.transport.tcp.TCPChannel.createConnection(Unknown Source)
```

```
    at sun.rmi.transport.tcp.TCPChannel.newConnec-
tion(Unknown Source)
```

```
[...]
```

Solution:

Activate element `java.rmi.server.hostname` in `rvs-system.properties` file (delete „#“ at the beginning of the line) and replace **<IP>** with the value of element `client.RMIServiceHost` in `rvsEVOClient.prefs` file:

Example:

```
java.rmi.server.hostname=q4de3esy333
```

11.10.2 Failure of Connection to Server with rvsEVO Server Farm

Error Message during server start:

```
2020-11-02 16:30:26,473 [ERROR ] [main      ]
com.tsystems.rvs.evo.EVOClassicServer.?(): Failed to
resolve 'EVOCLUSTER', will use LoopbackAddress :
java.net.UnknownHostException: EVOCLUSTER
java.net.UnknownHostException: EVOCLUSTER
    at java.base/java.net.InetAddress$Cache-
dAddresses.get(Unknown Source)
    at java.base/
java.net.InetAddress.getAllByName0(Unknown Source)
    at java.base/java.net.InetAddress.getAll-
ByName(Unknown Source)
    at java.base/java.net.InetAddress.getAll-
ByName(Unknown Source)
    at java.base/java.net.InetAddress.getBy-
Name(Unknown Source)
    at com.tsystems.rvs.evo.EVOClassic-
Server.initRMICommunication(EVOClassic-
Server.java:297)
    at com.tsystems.rvs.evo.EVOClassic-
Server.postServerStartup(EVOClassicServer.java:127)
    at
com.tsystems.rvs.evo.EVO.main(EVO.java:59)
```

Reason:

Alias name for **RMIServerName** cannot be resolved if `$RVS_HOME/conf/host-alias.properties` file is used for defining the alias name..

Solution:

Use `etc/host` file for defining the alias name of rvsEVO server or set **RMHostname** to „0.0.0.0“:

- stop rvsEVO server
- export configuration into xml file via `exportConfiguration` tool
- change value of **RMHostname** parameter in xml file to „0.0.0.0“
- import changed xml file via `importConfiguration` tool

11.11 Failure during Launching Scripts in „tools“ Directory

With derby database you can get the following error message during starting a script in `$RVS_HOME/tools` directory:

```
# Another instance of Derby may have already booted
the database C:\rvsRT1\tenants\4011\db.
```

```
# ...
```

```
Error creating bean with name 'config': Invocation of
init method failed; nested exception is org.springframework
transaction.CannotCreateTransactionException:
Could not open JPA EntityManager for transaction;
nested exception is org.hibernate.exception.GenericJDBCException:
Could not open connection
```

Reason: Connection to database fails because the rvsEVO server is running.

Solution: Some scripts can be started with `--api` component though the server is running (please see the rvsEVO User Manual, chapter „rvsEVO Database“).

Stop the rvsEVO server if `--api` is not supported.

11.12 Failure of rvsRuntime Commands**11.12.1 Failure of starting a tenant****Message:**

```
Execution of StartCommand failed.
Command is not allowed in license file
'C:\rvsrt1\tenants\4015\conf\license.properties'.
Missing 'V' in licensed components: FAISCJMYKOBRT.
```

Reason: The license key of the tenant is not valid for multitenancy functionality.

Solution: Contact the rvs® Service Support Center

Only Windows OS: **Message:**

```
c:\rvsRT1\bin>rvsrt start 4011
Systemfehler 5 aufgetreten.
```

```
Zugriff verweigert  
Start script for tenant '4011' returned with error.  
Executed script : "net start "rvsEVO 4011"  
Return code      : 2
```

Reason: The command box was not opened as administrator.

Solution: Open the command box as administrator

12 rvsEVO Support

Download You can download manuals, instructions and software via the following links:

– **german:**

<https://system-communication-products.t-systems-service.com/de/de/data-communication/rvs/rvs-downloads/rvs-user-manual/user-manual-rvsevo/download-735286>

– **english:**

<https://system-communication-products.t-systems-service.com/de/en/data-communication-suite/rvs/rvs-downloads/manuals/rvsevo/manuals-742534>

Instruction:

- click on **Data communication** in the menu bar
- move the cursor over **rvs®** (in the overview)
- move the cursor over **Downloads** in the new opening drop-down list
- move the cursor over **Manuals** or **Software** and then select the product you wish to download.

Hint: The Release Notes are assigned to software

rvsEVO support Please make the following data available to the rvsEVO support:

- contact person
- license number
- rvsEVO version
- clear description of the problem
- log files as zip or tar file

Contact Service support contact data:

- from Germany for free, telephone: 0800 664 77 45
- from other countries, telephone: +49 375 606 19 902
- e-mail: rvs-service@t-systems.com

