TIBCO Rendezvous®

z/OS Installation and Configuration

Version 8.8.0 | February 2025

Contents

Introduction

This manual provides information concerning TIBCO Rendezvous® software for IBM z/OS systems and is primarily intended for system programmers and administration personnel who are responsible for installing and maintaining the software. Some of the information contained here may also be useful for application programmers involved in creating or modifying programs that work in conjunction with this product. We recommend that you reading through this manual before attempting to install the product, and keep a copy handy for future reference.

Overview

Rendezvous software allows applications running on IBM z/OS systems to pass information between Rendezvous applications that communicate through Rendezvous message passing facilities. This message passing is implemented through one or more long running batch tasks (described as network daemons in other TIBCO publications) that receive Rendezvous messages from the network. These daemons can be run under the traditional z/OS MVS environment, or alternatively can be run in the Unix System Services USS (sometimes referred to as Open Edition) environments. Rendezvous daemons filter incoming messages according to their subjects and route these messages to applications (that register interest in a subject or group of subjects).

Applications that incorporate Rendezvous API function calls into their processing logic must supply the Rendezvous API library during the pre-link edit phase of the compilation. Application programs can then publish and subscribe to messages through the batch tasks running on z/OS to other Rendezvous applications anywhere in the network.

Please refer to TIBCO Rendezvous Concepts for a complete description of Rendezvous operations.

Hardware Requirements

Before you can begin using the Rendezvous product, your system must have the appropriate network hardware to support IBM TCP/IP (for example, an ethernet adapter card) should be installed and properly configured.

The network **should** be broadcast-capable. For messages to be seen in the local subnet, the network interface **must** support UDP broadcast or multicast. If your network interface **does not** support UDP broadcast, the Rendezvous software will not function to its full capability.

Storage Requirements

Rendezvous for the MVS environment requires approximately 350 cylinders of Direct-Access Storage Device (DASD) space. This spaces calculation is based on an IBM 3390 DASD device.

System Software Requirements

The Rendezvous software runs as an MVS batch task or as a Unix Systems Services (USS) process. You can install both the MVS and USS versions from the installation materials supplied by TIBCO Software Inc. Both products are supported on z/OS 1.6 and later (64-bit).

MVS Environment

Before installing Rendezvous software in an MVS environment, you must ensure that the following software components are installed and properly configured:

TCP/IP

Rendezvous z/OS only supports the IBM TCP/IP environment. A TIBCO messaging client for z/OS is expected to be installed and operated within a previously existing, fully functional and integrated network. At a minimum, this includes the following:

- The TCP/IP protocol stack within the z/OS Communications Server.

- A DNS server that provides the host name and host address of the TIBCO Rendezvous daemon (if the daemon is not running on the same host).

- Unrestricted network connectivity between the DNS, the TIBCO Rendezvous Client, and the TIBCO Rendezvous daemon (if the daemon is not running on the same host).

Customers running z/OS version 1.6 and above should be running a properly configured and customized RESOLVER address space.

Customers should review their TCPCONFIG parameters to ensure that they meet the installation's needs. If the customer chooses to run the TIBCO messaging client non-APF authorized, the RESTRICTLOWPORTS parameter should be used. If customers want to control TCP buffering, to limit storage usage, or to support large bandwidth devices, the TCPSENDBFRSIZE, TCPRCVBUFRSIZE, and TCPMAXRCVBUFRSIZE parameters may be needed. TIBCO Software typically recommends that customers override the IBM default of 16K by setting TCPRCVBUFRSIZE and TCPSENDBFRSIZE to a minimum value of 48K.

Next, you must make sure that the following required or recommended TCP and UDP parameters are appropriately set in the TCP Profile:

TCP and UDP parameters in the TCP Profile

Parameter	Setting
TCPCONFIG	RESTRICTLOWPORTS
	TCPSENDB 48K
	TCPRCVB 48K
UDPCONFIG	RESTRICTLOWPORTS
	UDPCHKSUM
	UDPQUEUELIMIT
	UDPSENDB 32768
	UDPRCVB 32768

LE

LE is the language environment. The C language and/or COBOL for MVS compilers are required for application development under this environment.

> **Note** IBM TCP uses the OE socket interface. Therefore, all user id's running Rendezvous daemons or client programs must have an OMVS UID and GID. Define these values in the OMVS segment of the USERID and GROUP records in RACF.

For more information, please refer to User Profile Requirements .

TLS

Rendezvous z/OS supports the following certificate formats:

- PEM (Privacy Enhanced Mail)
- PKCS#12

For details on TLS usage in Rendezvous, see TIBCO Rendezvous Concepts and TIBCO Rendezvous Administration.

Installation

This section describes how to install the TIBCO Rendezvous z/OS software.

Before Installation

General Variables provides a checklist of variables and data set HLQs (High Level Qualifiers) that you should obtain before performing the installation process. All Rendezvous supplied JCL and procedures have a standard naming convention. You must replace TIBCO site-specific installation-defined variables such as JCL SET statements, procedure overrides, SYSIN data and system data set HLQ with user values. You must also replace TIBCO Software variables before the JCL's can be submitted for execution.

It is recommended that you make a copy of the General Variables table below and then review and complete the entries. Use the User Value column to insert appropriate values for your site.

Here is a description of the table column headings:

The **Description** column provides instructional information.

The **Variable** column supplies the variable name, as they will appear in the provided JCL(S) and/or the member names. There may be several instances of the same variable within a specific member.

The **User Value** column is left blank so you can insert the appropriate replacement value.

General Variables

Description	Variable	User Value
Obtain a High Level Qualifier for your Rendezvous libraries. Typically, TIBCO.RVB is used.	*#USERHLQ*	
Obtain a hard disk Volume Serial where the Rendezvous files are to be installed.	*#USERVOL*	
Replace these <JOBNAME> variables with user defined values.	*#JOBNAME*	

Distribution Media and Contents

The most convenient way to get a copy of the Rendezvous software is to download it directly from the TIBCO web site, or obtain the product on a CD. Rendezvous mainframe software is distributed in IBM XMIT format.

Product distribution for IBM platforms that do not have DFSMSdss are supplied only upon request. Please request a downloadable package from TIBCO Support (see TIBCO Documentation and Support Services).

Complete Replacement Package

By default, the Rendezvous software is delivered as a non SMP/E stand-alone, base function package. Subsequent modifications and updates are version and product dependent. Therefore, a complete product replacement is preformed with each installation.

Component Code

The component code for the TIBCO Rendezvous products is RVB. This component code is registered with the IBM product codes division.

Obtaining the Installation Media

To obtain the Rendezvous software, you can download it from the web or you can request a cartridge.

Download from the Web Site

To download the Rendezvous software from the TIBCO web site, follow these steps:

- Contact TIBCO Software Inc. for a password, directory information, and so forth.

- Connect to the TIBCO web site with the required information.

- Download the appropriate files, which are in the standard IBM XMIT format.

Product	Installation Files for z/OS Version 1.6 and Later
TIBCO Rendezvous for DFSMSdss	Download this file: TIB_rv_8.8.0_zos16mvs_zSeries.xmit.zip The zip file contains these files: ``` TIB_rv_8.8.0_zos16_mvs.xmit.inst TIB_rv_8.8.0_zos16_mvs.xmit.main ```
TIBCO Rendezvous for non-DFSMSdss	Download this file: TIB_rv_8.8.0_zosmvs_zSeries.nodfs.xmit.zip The zip file contains all non-DFSMSdss install files.

Requesting a CD

If you do not want to download the software over the network, you can obtain a CD containing the Rendezvous software and load it directly onto your system. The minimum time period for delivery of a CD is five working days. Contact TIBCO Support and request the desired media for your environment. For details, see TIBCO Documentation and Support Services.

Checklist for Rendezvous Installation

Rendezvous Installation Checklist provides a checklist of variables or required items that should be obtained before performing the installation process. The data set suffix and member names are included as reference to where they occur.

Here is a description of the table column headings:

- The **Description** column provides instructional information.

- The **Suffix** column lists the suffix of the data set containing the reference.

- The **Member** column contains the variable in question. There may be several instances of the same variable within a specific member.

Rendezvous Installation Checklist

Description	Suffix	Member
If you are transferring the Rendezvous installation files from a Server, obtain the IP address or Server name for the FTP process.	INSTALL	RVB1FTP
Obtain the data set and or member name where your site FTPDATA information resides.	INSTALL	RVB1FTP
Obtain the data set and or member name where your site TCPDATA information resides.	PROC	RVD, RVRD, RVSD, RVSRD, and RVCACHE

Uploading the Software

If you have acquired the Rendezvous software by downloading it from the TIBCO web site rather than via a cartridge, you will need to upload the Rendezvous software to the MVS host system using the FTP file transfer utility. Following is a sample JCL that details how to perform this file transfer:

Initial Installation

Sample JCL for an Upload

The sample JCL illustrates the process, using IBM's FTP to copy the Rendezvous installation file(s) from a LAN server or PC to the MVS host.

```
//<JOBNAME> JOB (&SYSUID),'RENDEZVOUS FTP',CLASS=A
//*
// SET FTPADDR=10.10.2.99   < IP Address or Host Name
//*
//FTP EXEC PGM=FTP,REGION=0M,PARM='&FTPADDR (TIMEOUT 20'
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//SYSTCPD DD DISP=SHR,DSN=TCPIP.PROFILE(FTPDATA)  < Host Member
//INPUT DD *
<lan-userid>
<lan-password>
cd /<lan-file-dir>
binary
LOCSITE RECFM=FB LRECL=80 BLKSIZE=27920 TR PRI=60 SEC=15
get #srvhlq.xmit.inst'  <USERHLQ>.XMIT.INST' (REPLACE
quit
/*
//
```

Receive the Initial Installation File

> **ⓘ**
> **Note**
> This step always needs to be performed during a Rendezvous installation on the MVS Host.

The `<USERHLQ>.XMIT.INST` file contains JCL that is required to receive other files and to create the appropriate libraries for Rendezvous. To complete this process, the TIBCO supplied variables must be substituted with the appropriate site-specific user values.

Follow these steps:

Procedure

1. From the ISPF Command Shell Panel, type the following command to receive a file containing the JCL required to begin a first-time installation:

    ```
    receive inda('<USERHLQ>.XMIT.INST')
    ```

    TSO will prompt with the following:

    ```
    INMR906A Enter restore parameters or 'DELETE' or 'END' +
    ```

2. Supply the following:

    ```
    DA('<USERHLQ>.INSTALL')
    ```

    After a successful receive, the following message is displayed:

    ```
    INMR001I Restore successful to dataset '<USERHLQ>.INSTALL'
    ```

The data set name supplied in Supply the following: is output and populated with the members that are required in order for you to continue with the installation.

Rendezvous MVS Installation Procedure

The following steps are required to install the MVS components for Rendezvous. All members reside in the `<USERHLQ>.INSTALL` dataset.

Edit the data sets and selected members listed in the steps below. Substitute the variables, follow any instructions that are contained in the member and then submit the JCL for execution.

STEP 1:	Make CLIST temporary file
	Change the *#JOBNAME*, *#USERHLQ*, *#USERVOL* statement variables according to your site requirements.
Member:	RVB$1MKT

STEP 2:	Execute File Tailoring clist for installation JCL.
	If an error is made during input, press the attention key (PA1) and start again at Step 1.
Member:	RVB$2EX1 (EX member)

File Tailoring Variables for Installation JCL

Variable	Default and structure	User Value
USERHLQ	TIBCO.RVB.VxRxMx	
VOLSER Volume Serial	TIB999	
JOBNAME	RVB87 First 5 characters are used	

Variable	Default and structure	User Value
JOBCLASS	A	
MSGCLASS	X	

STEP 3:	**Upload all installation files to z/OS MVS.**
Member:	For DFSMSdss sites, use RVB$3FT.
	For non DFSMSdss sites, use RVB$3FT@.

STEP 4a:	**Rendezvous configuration files**
ⓘ Note	This member will allocate the Rendezvous daemons' configuration files.
Member:	RVB$4ACF

STEP 4b:	Define and allocate Rendezvous product libraries
ⓘ Note	
Member:	RVB$4ALL

STEP 5:	**Convert product installation files to appropriate datasets**
ⓘ Note	The installation can be restarted multiple times at this step if required.

Member:	For DFSMSdss sites, use `RVB$5RC`.
	For non DFSMSdss sites, use `RVB$5RC@`.

The following table defines the user substitution variables within JCL and parameter members that are required during the file tailoring process. Ensure that you have all the necessary information before executing the next step (Step 6).

File Tailoring Variables

Variable	Default and Structure	User Value
IBM Assembler Libraries HLQ	ASM *.SASMMAC2	
COBOL for MVS Libraries HLQ	IGY *.SIGYCOMP	

STEP 6:	**Execute File Tailoring clist to update the product supplied variables.**
🛈 **Note**	If an error is made during input, press the attention key (PA1) and execute the clist again.
Member:	`RVB$6EX2 (EX member)`

STEP 7:	**Populate the Rendezvous data sets with all the update and downloaded members.**
Member:	`RVB$7UPD`

Verifying the Rendezvous Installation

This section provides information that allows you to tests verify that Rendezvous components have been correctly installed and configured.

Step I1: Network daemon startup JCL

> **ℹ Note** Replace the Rendezvous variables with the appropriate user values. This member will start the Rendezvous daemon.

> Data Set: `<USERHLQ>.JCL`
>
> Member: `RVBIRVD`

> **⚠ Warning** After you have started the network daemon, you need to determine if it is operating correctly by performing a sample publish-and-subscribe operation.

You can use an http browser interface as a method for verifying that the daemon is running before continuing with the following steps.

For example, specify the following:

```
http://hostname_or_IP_address:7580/
```

Please refer to TIBCO Rendezvous Administration for a more complete description of the http browser interface.

Step I2: Reliable messaging subscriber startup JCL

> **ℹ Note** Replace the Rendezvous variables with the appropriate user values. This member will initiate a Rendezvous listener that receives messages from the publisher.

> Data Set: `<USERHLQ>.JCL`
>
> Member: `RVBILSN`

Step I3: Reliable messaging publisher startup JCL

ℹ️ **Note**   Replace the Rendezvous variables with the appropriate user values. This member will initiate a Rendezvous sender that publishes a message to an already executing listener.

Data Set: `<USERHLQ>.JCL`

Member: `RVBISND`

⚠️ **Warning**   After you have examined the output of the previous execution, you may consider testing certified messaging (RVCM) by performing the IVPs for RVCM publish and subscribe.

Step I4: Certified messaging (CM) ledger file(s) allocation JCL

ℹ️ **Note**   Replace the Rendezvous variables with the appropriate user values. This member will allocate the Rendezvous certified messaging ledger files.

Data Set: `<USERHLQ>.JCL`

Member: `RVGDEFLF`

Step I5: Certified messaging (CM) subscriber startup JCL

ℹ️ **Note**   Replace the Rendezvous variables with the appropriate user values. This member will initiate a Rendezvous certified messaging (CM) listener that receives messages from the certified messaging (CM) publisher. The listener sends the confirmation back to the publisher.

Data Set: `<USERHLQ>.JCL`

Member: `RVBICMLS`

Step I6: Certified messaging (CM) publisher startup JCL

ⓘ **Note**	Replace the Rendezvous variables with the appropriate user values. This member will initiate a Rendezvous certified messaging (CM) sender that publishes messages.

Data Set: `<USERHLQ>.JCL`

Member: `RVBICMSN`

Optional Installation

Rendezvous sample programs are supplied as source and executable modules as part of the installation. However, users may desire to change the source code to perform additional functions. JCLs and procedures have been supplied to compile the sample programs that normally have to be modified to be used within a user's environment.

Edit the data sets and selected members listed in the steps below. Substitute the variables, follow any instructions that are contained in the member and then submit it for execution.

In addition, this section describes an alternate UNIX installation.

Compiling C Programs

Step 1a: C language options file

> **ⓘ Note** This member is the Options File used for Rendezvous C compiles that needs to include members from the system C and Rendezvous libraries. Substitute the appropriate values and installation variables that are located within this member.

Data Set: `<USERHLQ>.CNTL`

Member: `OPTFRVB`

Step 1b: Rendezvous C language compilation JCL

> **ⓘ Note** The JCL listed below uses a procedure, which may have to be modified so that it conforms to your site's requirements and executes correctly.

Data Set: `<USERHLQ>.JCL`

Member: `RV$3CPL`

Compiling COBOL Programs

Step 1c: Rendezvous COBOL compilation JCL

ℹ
Note

The JCL listed below uses a procedure, which may have to be modified so that it conforms to your site's requirements and executes correctly. This member will compile COBOL supplied sample program(s).

Data Set: `<USERHLQ>.JCL`

Member: `RV$CCPL`

Alternate UNIX installation

You can install the standard UNIX-based z/OS install package as an alternative to the standard Rendezvous for z/OS software. However, this method installs software only for use in a UNIX System Services environment.

Installation Files for z/OS USS Version 1.6 and Later

Obtain this installation package:

```
TIB_rv_8.8.0_zos16uss_zSeries.zip
```

The installation procedure is identical to other UNIX installations. For details, see the UNIX installation sections in TIBCO Rendezvous Installation.

Rendezvous Java Libraries

Rendezvous Java libraries are available for the IBM z/OS operating system in a UNIX System Services environment. These libraries require IBM Java 1.5.0. To check the Java support level on your host computer, execute the following command:

```
../J5.0/bin/java -version
```

For z/OS platforms, compare the command output with the following requirements:

```
java version "1.5.0"
```

```
Java(TM) 2 Runtime Environment, Standard Edition (build pmz31devifx-
20071025 (SR6b))
```

```
IBM J9 VM (build 2.3, J2RE 1.5.0 IBM J9 2.3 z/OS s390-31 j9vmmz3123-
20071007 (JIT enabled)
```

```
J9VM - 20071004_14218_bHdSMr
```

```
JIT - 20070820_1846ifx1_r8
```

```
GC - 200708_10)
```

z/OS MVS

This section describes administration and use of Rendezvous software using MVS procedures.

Before doing any of these administrative procedures, you must have completed the installation of the MVS version of Rendezvous software, and executed the IVP steps successfully, as described in Verifying the Rendezvous Installation.

Starting Daemons and Applications

User Profile Requirements

Any *userid* used to run the Rendezvous application must have a valid RACF OMVS segment defined.

- If you are starting Rendezvous as a batch job, verify that the submitter's *userid* has a valid OMVS segment defined.

- If you are starting Rendezvous as a started task, verify that the *userid* assigned by the RACF STARTED class or ICHRIN03 started procedures table has a valid OMVS segment. Verification can be done by using the **RACF LISTUSER** command.

```
TSO LISTUSER TIBCO01 OMVS NORACF
USER=TIBCO01
OMVS INFORMATION
UID= 0000012345
HOME= /tibco/tibco01
PROGRAM= /bin/sh
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMAX= NONE
PROCUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE
***
```

TIBCO Software Inc. recommends running Rendezvous daemons with a *userid* without resource limits specified in the OMVS segment, as shown in the example `userid TIBCO01`. This allows Rendezvous to run within the limits defined to the overall system in BPXPRMxx. There are no special requirements for UID, HOME, or PROGRAM in order to run Rendezvous.

If an existing *userid* does not have an OMVS segment, you can define one by using the **RACF ALTUSER** command. For more information regarding the use of the various RACF commands and the setup of an OMVS segment, refer to "*SecureWay Security Server RACF Command Language Reference*" or equivalent documentation published by IBM.

If the installation is using the FACILITY class profile BPX.DEFAULT.USER instead of defining OMVS segments for every userid, the *userid* defined in the FACILITY class profile BPX.DEFAULT.USER should be examined to ensure that it does not have resource limits defined if the user submits RV jobs. For more information about managing the FACILITY class profile BPX.DEFAULT.USER, see "*SecureWay Security Server RACF Security Administrator's Guide*" documentation published by IBM.

JCL and Procedures

After installation, Rendezvous daemons and example programs reside in the `<USERHLQ>.LOAD` load library. Rendezvous daemons can be started from JCL or catalogued procedures that have been supplied during the installation processes. JCL has been provided to execute the example programs.

Refer to MVS Batch Examples for a list example programs and JCL members.

We recommend that you maintain catalogued JCL procedures to run the Rendezvous daemons, and define them as started tasks, so the system operator at the master console can start and monitor them.

The following steps are required to define daemons as started tasks.

Procedure

1. First, you need to copy the RVD / RVRD procedures from `<USERHLQ>.PROC` into a procedure library that is defined in the JES2 startup.

2. Modify the variables `<USERHLQ>` and the data set name that contains the TCPDATA member.

3. The daemon procedures can now be started from the MVS console.

The daemon procedures are long-running applications. The Rendezvous daemons support operational commands to terminate them. For the syntax and usage of these commands, refer to z/*OS MVS System Commands*.

Examples:

```
F < [jobname.]identifier >,SHUT
P < [jobname.]identifier >
(Stop an active daemon)
```

When you start a daemon, the following messages are written to the system log:

```
RVB1000I Rendezvous Daemon Started.
RVB1001I Rendezvous Daemon ports. Listen:7500 Http:7580
```

When you issue the command to stop a daemon, you will see the following message in the system log:

```
RVB1009I Rendezvous daemon shutdown in progress
```

Daemon Tasks (Non-Swappable and No CPU Limit)

We recommend that you run the Rendezvous daemon programs set as non-swappable and no CPU time limit, especially in a production environment. For a Rendezvous daemon program to be made non-swappable and no CPU time limit, you need to add the following to the system parameter library.

```
'SYS.PARMLIB(SCHEDxx)'
PPT PGMNAME(daemon_program_name)
    NOSWAP
    SYST
```

Additionally, the Rendezvous daemon programs load module must reside in an APF authorized library. When the Rendezvous daemon program has been restarted it should be running as NON-SWAPPABLE (NS).

Sites with Multiple Stacks

When the system uses multiple TCP/IP stacks, insert the following JCL statement before the Rendezvous application execution steps:

```
//TCAFF EXEC PROC=TIBTCAFF,STACK=TCPIP-STACK-NAME
```

This statement identifies the stack that the application will use.

Time Zone

The daemon receives timestamps from the z/OS LE environment. If you have not configured the LE environment with the correct time zone, then messages from the daemon report time in GMT format.

You can dynamically set the time zone by setting the `TZ` environment variable in the daemon's startup `JCL` or `PROC`. Use either of the following examples as a model (adjusting for your actual time zone):

```
//RVD EXEC PGM=RVD,PARM='ENVAR("TZ=GMT-7"),/&TIBPARM'
//RVD EXEC PGM=RVD,PARM='ENVAR("TZ=PST8PDT"),/&TIBPARM'
```

The first example calculates the time zone as an offset from GMT (either + or −). The second example uses the time zone name.

You can find the appropriate time zone values in *UNIX System Services Command Reference*, Setting the Local Time Zone with the TZ Environment Variable.

Case Sensitivity in Input Parameters

Rendezvous program parameters are case sensitive. Be sure to turn `CAPS OFF` when editing JCL and parameter members that are edited and used during execution of Rendezvous daemons and applications.

For example, the Rendezvous daemon (rvd) accepts the **-listen** parameter. The parameter must be in lower case letters.

ℹ️
Note

Any application program that uses Rendezvous API calls must have the correct case when specifying input to the API call.

JCL Statements for Daemons

The following models illustrate the general form of the JCL EXEC statement for any Rendezvous daemon (rvd, rvsd, rvrd, rvsrd, or rvcache):

```
//RVD EXEC PGM=RVD,PARM='-no-multicast'
//STEPLIB DD DISP=SHR,DSN=<USERHLQ>.LOAD
//TIXFILE DD DISP=SHR,DSN=<USERHLQ>.CNTL
```

or:

```
//RVRD EXEC PGM=RVRD,PARM='-store "DD:RVRDCFG"'
//STEPLIB DD DISP=SHR,DSN=<USERHLQ>.LOAD
//TIXFILE DD DISP=SHR,DSN=<USERHLQ>.CNTL
//RVRDCFG DD DISP=SHR,DSN=<USERHLQ>.CFG.RVRD
```

The case-sensitive parameter **-no-multicast** specifies that the daemon disables multicast (and broadcast) communication. It changes daemon behavior in the following ways:

- When a client sends a message to a public subject, the daemon does not multicast it (nor broadcast it) to the network.

- When a routing daemon receives multicast or broadcast messages from the network, it does not forward them to other daemons within the local network.

When multicast communication is disabled, the daemon disables all UDP traffic to and from the affected daemon (multicast, broadcast and point-to-point). Daemons continue to operate as follows:

- All messages (including public subjects) flow among all the clients of the daemon.

- All messages (including public subjects) flow in both directions between local clients of a routing daemon and the daemon's neighbors.

The `TIXFILE DD` statement specifies the location of the z/OS Rendezvous license ticket file. Note that license ticket files are no longer required.

The `STEPLIB DD` statement specifies the location of the Rendezvous daemon.

Daemons other than 'RVD' can take the '-store' parameter to store configuration data. The installation allocates the configuration files for the daemons, (for example, `"<USERHLQ>.CFG.RVRD"`). The configuration file must reside on a single DASD volume.

The daemons configuration files are opened internally with a C function call, `fopen()`. The qualifier(s) referenced in the '-store' parameter does not require that the standard "//" be before the `DD:RVBCFG`.

Example:

```
    PARM='-store "DD:RVRDCFG" '
```

Adding a Certificate

The following models illustrate the general form for specifying a certificate file when entering the certificate through the HTTP interface:

- To specify as a z/OS dataset: `//'HLQ.SLQ(MEMB)'`

- To specify as a DDNAME: `DD:DDNAME`

- To specify as a USS file: `/dir/dir/file.ext`

Certificate files have the following data set restrictions:

- z/OS PEM certificate files must be EBCDIC and in 80 record length data sets.

- z/OS PK12 certificate files must be in a dataset with a record length greater than 80.

- USS PEM certificate files can be ASCII or EBCDIC.

JCL Statements for Applications

This model illustrates the general form of JCL for running any Rendezvous program (that is, any program that calls Rendezvous API functions).

Optional

When the system uses multiple TCP/IP stacks, insert the following JCL statement, which identifies the stack that the application will use:

```
//SYSTCPD DD DISP=SHR,DSN=your.TCPDATA.file
```

General Form

```
//RVPGM EXEC PGM=<RVPGM>,PARM='<RV Transport parameters>'
//STEPLIB DD DISP=SHR,DSN=your.user.loadlib
```

Alternate Form

Consider this example from `<USERHLQ>.JCL(RVBCLSN)`:

```
//RVLISTEN EXEC PGM=RVBCLSN,PARM='rv.test.subject'
//STEPLIB DD DISP=SHR,DSN=your.user.loadlib
```

The program uses the default transport settings and the user-supplied parameter *rv.test.subject* is a substitution variable.

JCL Statements for Certified Messaging (RVCM)

CM Ledger File - Allocation characteristics

When executing a RVCM program that uses the disk resident ledger file, you may use the RVGDEFLF JCL member as an example to allocate one or more ledger files. The ledger file must reside on a single DASD volume, due to Language Environment restrictions in the support of multi-volume data sets.

If you allocate a ledger file using another method, refer to this member for the correct DCB information.

ⓘ

Note

Except for the DSNTYPE, RECFM and LRECL parameters all other values can be changed to suit your applications RVCM ledger requirements.

For MVS batch, a ledger file that exists on disk must be defined as a PDS. Failure to adhere to this requirement will result in an error condition when executing Rendezvous example certified messaging programs.

From <USERHLQ>.JCL(RVGDEFLF):

```
//* ALLOCATE THE TIBCO RVCM SENDER'S LEDGER DATA SET
//*
//ALLOCAT1 EXEC PGM=IKJEFT01,DYNAMNBR=300
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  ALLOCATE                              -
      DSNAME('<USERHLQ>.CMLEDGER.SENDER01') -
      NEW CATALOG                       -
      RECFM(F)                          -
      LRECL(512)                        -
      BLKSIZE(6144)                     -
      DSORG(PS)                         -
```

```
       NEW CATALOG                     -
       SPACE(5,1) CYL                  -
       DIR(0)                          -
       UNIT(3390)
 /*
 //* ALLOCATE THE TIBCO RVCM LISTENER'S LEDGER DATA SET
 //*
 //ALLOCAT2 EXEC PGM=IKJEFT01,DYNAMNBR=300
 //SYSTSPRT DD SYSOUT=*
 //SYSTSIN  DD *
   ALLOCATE                           -
       DSNAME('<USERHLQ>.CMLEDGER.LISTEN01') -
       NEW CATALOG                     -
       RECFM(F)                        -
       LRECL(512)                      -
       BLKSIZE(6144)                   -
       DSORG(PS)                       -
       SPACE(5,2) CYL                  -
       DIR(0)                          -
       UNIT(3390)
 //*
```

CM Ledger File - JCL `ddname` specification

When using a disk file ledger, an application input parameter with the following format is required:

```
[-ledger "//DD:ddname"]
```

where *ddname* is the name of an existing `DDNAME` file referenced in your JCL jobstream.

The default is NULL. A NULL value causes a non-persistent, memory-based ledger file to be used.

The following ledger file example shows the use of the *ddname* for Listeners and Senders:

```
 //*
 //** CM Ledger file (for Listener) example
 //*
 //RUNCCMLC EXEC PGM=RVBCCMLC,
 //         PARM='-ledger "//DD:RVCMLFL1" rvcm-mvs RVCM-MVS'
 //*
 //STEPLIB   DD DISP=SHR,DSN=&USERHLQ..LOAD
 //RVCMLFL1  DD DISP=SHR,DSN=&USERHLQ..CMLEDGER.LISTEN01
 //*
```

```
//*-------------- START OF NEXT EXAMPLE -----------------------
//*
//** CM Ledger file (for Sender) example
//*
//RUNCCMSN EXEC PGM=RVBCCMSN,
//   PARM='-ledger "//DD:RVCMLFS1" RVCM-MVS "RVCM Test Message "'
//*
//STEPLIB  DD DISP=SHR,DSN=&USERHLQ..LOAD
//RVCMLFS1 DD DISP=SHR,DSN=&USERHLQ..CMLEDGER.SENDER01
//*
```

CM Examples-Listener and Sender programs

To initiate the RVCM Listener and Sender example applications, submit JCL members.

- For COBOL, use `RVBCCMLC / RVBCCMLS`, then `RVBCCMSN`.

- For C, use `RVB3CMLS`, then `RVB3CMSN`.

Programmer's Checklist

All application programmers in an MVS environment should make the following checks.

- User options file

  A Language Environment user options object is supplied as part of the installation. When compiling Rendezvous enabled programs, it is required to include the object during the link-edited step. The installation- supplied procedures already cater for this.

  Including the supplied user options object ensures that `POSIX(ON)` is set at run-time and memory allocations are optimized for applications using the Rendezvous API.

- Execution

  The Rendezvous API uses the TCP/IP socket interface. Most installations require a SYSTCPD DD statement in the execution JCL. Contact your TCP specialist for details.

C

C application programmers in an MVS environment can use this checklist.

- Code

    Include the appropriate Rendezvous C header files. (See the section, Include These Header Files, TIBCO Rendezvous C Reference.)

- Compile/Pre-link/Link

  Sample JCL for compiling C programs is member `RV$3CPL`.

  — Application source files normally reside in the.C dataset.

  — Requirements are specified in the applicable compile procedure.

COBOL

COBOL application programmers in an MVS environment can use this checklist.

- Code

  Include the appropriate Rendezvous COBOL copybooks.

- Compile/Pre-link/Link

  Sample JCL for compiling COBOL programs is member `RV$CCPL`.

    Application source files normally reside in the `.COBOL` dataset.

    Requirements are specified in the applicable compile procedure.

MVS Batch Examples

This appendix describes members required to build and run MVS Batch transactions that use the Rendezvous Reliable API.

Sample Programs

The following are members related specifically to MVS Batch.

ⓘ
Note
These Assembler examples are not ready for compilation and they are not executable

---

Rendezvous Assembler Basis Code Segment Examples

Suffix	Member	Description
ASM	RVBACMLS	Sample code segments of RV CM Listener
	RVBACMSN	Sample code segments of RV CM Sender
	RVBALSN	Sample code segments of RV Basic Listener
	RVBASND	Sample code segments of RV Basic Sender

Rendezvous C Examples

Suffix	Member	Description
C	RVB3CLNT	Sample of RV RPC client-side
	RVB3CMLS	Sample of RV CM Listener
	RVB3CMSN	Sample of RV CM Sender
	RVB3DISP	C program of RV multiple dispatcher threads
	RVB3DQLS	C program of generic DQ RV subscriber

Suffix	Member	Description
	RVB3FTMN	Sample of RV FT Group Monitor.
	RVB3FTTM	Sample of RV FT Timestamp example
	RVB3INIV	C program get initial values out of rvcache
	RVB3LSN	Sample of RV Basic Listener
	RVB3PRTY	C program of queues, groups and priorities
	RVB3SND	Sample of RV Basic Sender
	RVB3SNMF	Sample of RV multi-field message publisher
	RVB3SRVR	Sample C RV RPC server-side
	RVB3USRT	C program of RV custom message types
	RVPERFM	Sample of RV performance tester - master side
	RVPERFS	Sample of RV performance tester - slave side
H	TIBRV	This header file for RV C API
	CM	This header file for the RV certified message delivery and distributed queue C API.
	FT	This header file for the RV fault tolerance C API.
JCL	RVB3CMLS	JCL to execute RVCM Listener

Suffix	Member	Description
	RVB3CMSN	JCL to execute RV CM Sender
	RVB3LSN	JCL to execute RV Basic Listener
	RVBCLSMF	JCL to execute RV multi-field Listener
	RVB3SND	JCL to execute RV Basic Sender
	RVB3SNMF	JCL to execute RV multi-field Sender

Rendezvous COBOL Examples

Suffix	Member	Description
COBOL	RVBCCMLC	RV CM Listener using automated call-back function
	RVBCCMLS	Sample of RV CM Listener
	RVBCCMSN	Sample of RV CM Sender
	RVBCFTMN	Sample of RV FT Group Monitor.
	RVBCFTTM	Sample of RV FT Timestamp example
	RVBCLSC	RV Basic Listener using automated call-back function
	RVBCLSN	Sample of RV Basic Listener
	RVBCLSMF	COBOL program of RV multi-field message listener
	RVBCSND	Sample of RV Basic Sender
	RVBCSNMF	Sample of RV multi-field message publisher

Suffix	Member	Description
COPYBOOK	TIBRV	Copybook for RV enumerated types
	TIBRVCM	Copybook for RV CM definitions and constants
	TIBRVFT	Copybook for RV FT definitions and constants
	TIBRVTEQ	Copybook for RV enumerated constants
JCL	RVBCCMLC, RVBCCMLS	JCL to execute RV CM Listener
	RVBCCMSN	JCL to execute RV CM Sender
	RVBCLSC, RVBCLSN	JCL to execute RV Basic Listener
	RVBCLSMF	JCL to execute RV multi-field Listener
	RVBCSND	JCL to execute RV Basic Sender
	RVBCSNMF	JCL to execute RV multi-field Sender

Rendezvous General Examples

Suffix	Member	Description
C	RVB3LGRR	Sample Rendezvous CM ledger file review
JCL	RVGCACHE	JCL to execute RV Cache daemon
	RVGDEFLF	JCL to allocate CM Ledger File
	RVGRVD	JCL to execute RV daemon
	RVGRVRD	JCL to execute RV routing daemon

Suffix	Member	Description
	RVGRVSD	JCL to execute RV secure daemon
	RVGRVSRD	JCL to execute RV secure routing daemon
	RVUINIFT	JCL to execute "INIFTST" (card)
	RVULDGRR	JCL to execute RV CM ledger review
	RVUPERFM	JCL to execute RV performance tester - master side
	RVUPERFS	JCL to execute RV performance tester - slave side
	RV$CCPL	JCL to compile COBOL RV API applications
	RV$3CPL	JCL to compile C RV API applications

Rendezvous General Procedures

Member	Description
TIBTCAFF	Procedure to set TCPIP stack affinity
RVCACHE	Procedure to execute RV Cache daemon
RVD	Procedure to execute RV daemon
RVRD	Procedure to execute RV routing daemon
RVSD	Procedure to execute RV secure daemon
RVSRD	Procedure to execute RV secure routing daemon
TIBCNCPL	Procedure to compile prelink & link COBOL application

Member	Description
TIB3NCPL	Procedure to compile prelink & link C application

Rendezvous Installed Data Sets

This appendix lists all of the installed data sets that are associated with Rendezvous.

Installed Data Sets

Data Set	Space (1st/2nd)	DSORG	RECFM	LRECL	BLK SIZE
<USERHLQ>.ASM	CYL (3/1)	PO	FB	80	27920
<USERHLQ>.C	CYL (5/1)	PO	VB	100	27998
<USERHLQ>.CFG.RVCACHE	CYL (2/1)	PS	F	512	6144
<USERHLQ>.CFG.RVPERFM	CYL (1/1)	PS	F	512	6144
<USERHLQ>.CFG.RVPERFS	CYL (1/1)	PS	F	512	6144
<USERHLQ>.CFG.RVRD	CYL (2/1)	PS	F	512	6144
<USERHLQ>.CFG.RVSD	CYL (2/1)	PS	F	512	6144
<USERHLQ>.CFG.RVSRD	CYL (2/1)	PS	F	512	6144
<USERHLQ>.CMLEDGER. LISTEN01	CYL (5/2)	PS	FB	512	6144
<USERHLQ>.CMLEDGER. SENDER01	CYL (5/1)	PS	FB	512	6144
<USERHLQ>.CNTL	CYL (5/1)	PO	FB	80	27920
<USERHLQ>.CNTLSAMP	CYL (2/1)	PO	FB	80	27920
<USERHLQ>.COB	CYL (3/1)	PO	FB	80	27920

Data Set	Space (1st/2nd)	DSORG	RECFM	LRECL	BLK SIZE
<USERHLQ>.COPY	CYL (2/1)	PO	FB	80	27920
<USERHLQ>.H	CYL (2/1)	PO	VB	100	27998
<USERHLQ!>.INSTALL	CYL (8/2)	PO	FB	80	27920
<USERHLQ>.JCL	CYL (5/1)	PO	FB	80	27920
<USERHLQ>.JCLSAMP	CYL (3/1)	PO	FB	80	27920
<USERHLQ>.LIBRARY	CYL (25/5)	PO	FB	80	3120
<USERHLQ>.LOAD	CYL (150/0)	PO	U	0	27920
<USERHLQ>.OBJ	CYL (3/0)	PO	FB	80	27920
<USERHLQ>.PROC	CYL (3/1)	PO	FB	80	27920
<USERHLQ>.PROCSAMP	CYL (1/1)	PO	FB	80	27920
<USERHLQ>.SKEL	CYL (1/1)	PO	FB	80	27920
<USERHLQ>.SSL.LIBRARY	CYL (20/5)	PO	FB	80	3120
<USERHLQ>.XMIT.INST	CYL (3/1)	PS	FB	80	27920
<USERHLQ>.XMIT.MAIN	CYL (71/5)	PS	FB	80	27920

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO Rendezvous® Product Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.

- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

for the availability of a specific version of Cloud SG software on a specific operating system platform.