Rocket software

Data Replicate and Sync tcAgent Installation and Startup V7.0

Notices

Copyright

Trademarks

Examples

This information might contain examples of data and reports. The examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

License agreement

Corporate information

Rocket Software, Inc. develops enterprise infrastructure products in four key areas: storage, networks, and compliance; database servers and tools; business information and analytics; and application development, integration, and modernization.

Website: www.rocketsoftware.com

Rocket Global Headquarters

77 4th Avenue, Suite 100

Waltham, MA 02451-1468

USA

To contact Rocket Software by telephone for any reason, including obtaining pre-sales information and technical support, use one of the following telephone numbers.

Country and Toll-free telephone number

- United States: 1-855-577-4323
- Australia: 1-800-823-405
- Belgium: 0800-266-65
- Canada: 1-855-577-4323
- China: 400-120-9242
- France: 08-05-08-05-62
- Germany: 0800-180-0882
- Italy: 800-878-295
- Japan: 0800-170-5464
- Netherlands: 0-800-022-2961
- New Zealand: 0800-003210
- South Africa: 0-800-980-818
- United Kingdom: 0800-520-0439

Contacting Technical Support

The Rocket Community is the primary method of obtaining support. If you have current support and maintenance agreements with Rocket Software, you can access the Rocket Community and report a problem, download an update, or read answers to FAQs. To log in to the Rocket Community or to request a Rocket Community account, go to www.rocketsoftware.com/support. In addition to using the Rocket Community to obtain support, you can use one of the telephone numbers that are listed above or send an email to support@rocketsoftware.com.

Table of contents

General information

The Data Replicate and Sync component – Data Replicate and Sync Agent – runs on Microsoft Windows, various Unix derivatives, z/OS. It has the following functions:

Provision of resources for

- data processing
- monitoring
- the control board or the dashboard for definition and control of data replication
- scheduling functions

A running instance of the Data Replicate and Sync Agent is needed on every workstation on which Data Replicate and Sync functions are executed. Therefore, the software can be installed as a service in a LUW (Linux, Unix, Windows) environment. In a z/OS environment, the software is started as job or STC.

Various instances of the Data Replicate and Sync Agent communicate with each other via the Agent network. Thus, cross-platform data replication can be managed and monitored by the respective Agents.

The following manual contains all information needed for installation and administration.

The manual is structured as follows:

- Data Replicate and Sync LUW installation
- Data Replicate and Sync z/OS installation
- tcAgent start parameters

LUW installation

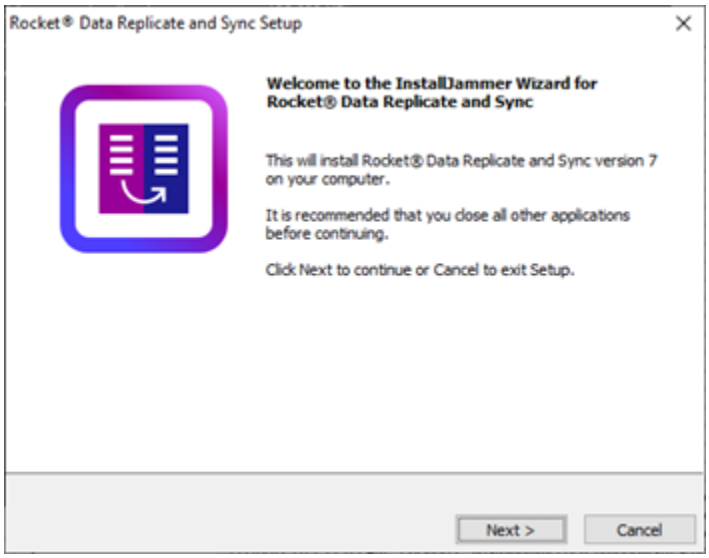
Windows installation

The installation is delivered as zip file. After unpacking, start the setup.exe with a double click. The installation wizard will guide you through the setup and prompt you to enter the necessary information.

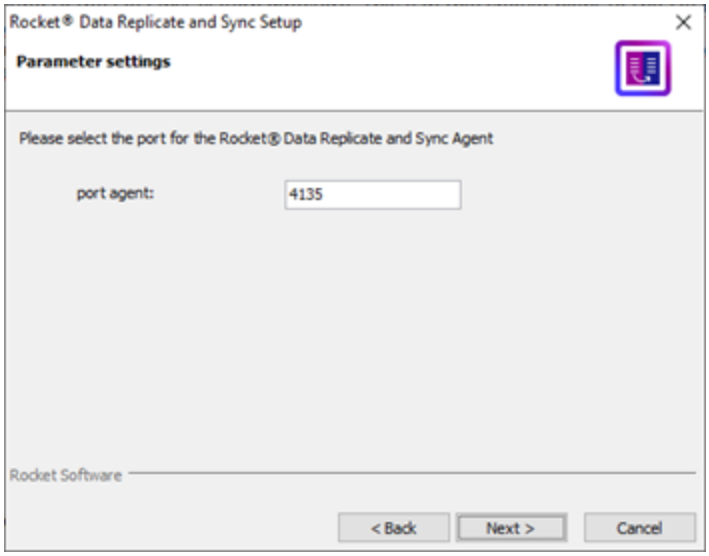
Before you begin:

- Please make sure that Java version 7 or higher is installed and ideally a **JAVA_HOME** environment variable is set pointing to the java installation directory. This variable should point to the java installation directory normally named `jdk...` or `jre...` not the bin directory inside the `jre` or `jdk` directory. If this variable is not specified you will see a dialog asking you to select this directory. Here, the same rules apply.
- About 150 megabytes of disk space is required for the Data Replicate and Sync installation.

1. After unpacking, start the `setup.exe` with a double click.



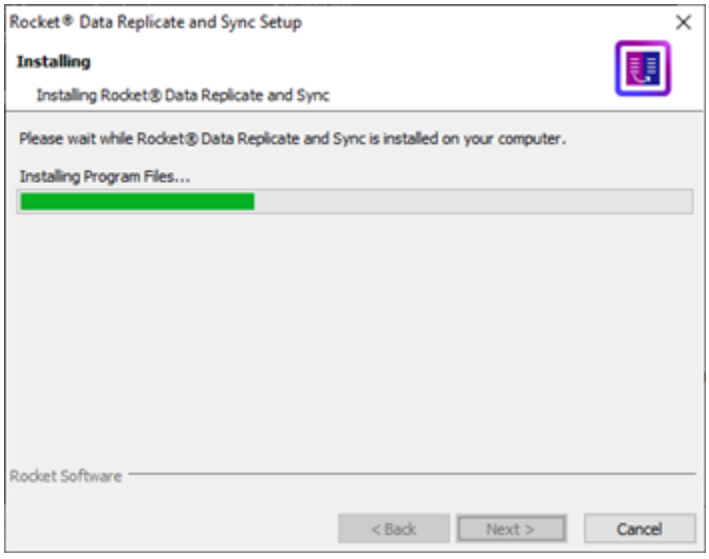
   The **Next** button displays the next page. The installation can be terminated any time.
2. Select the **Next** button.
   The **Parameter settings** pane is displayed.



3. On this page, enter the port on which the Data Replicate and Sync Agent is running.
   Ensure that the communication on this port is allowed by the firewall and that the selected port is not used by other applications. The port can be changed in file `tcAgent.ini` in the installation directory later.
4. Select the **Next** button.
   The **Choose Destination Location** pane is displayed.

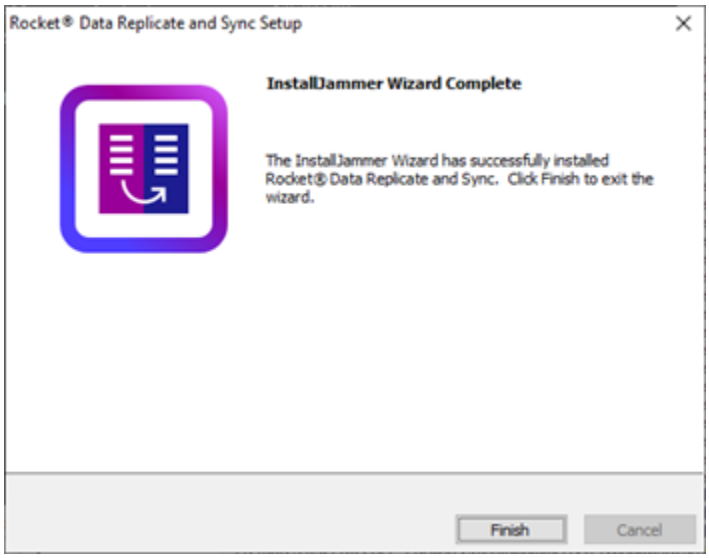
   On this page, the directory in which the Data Replicate and Sync files should be stored is selected. Ensure sufficient permissions since files are changed and allocated during operation of an Agent. With the button **Browse** a new directory can be selected in which the Data Replicate and Sync files and subdirectories will be stored during the course of the installation.

5. Select the **Next** button.
   The **Installing** pane is displayed.

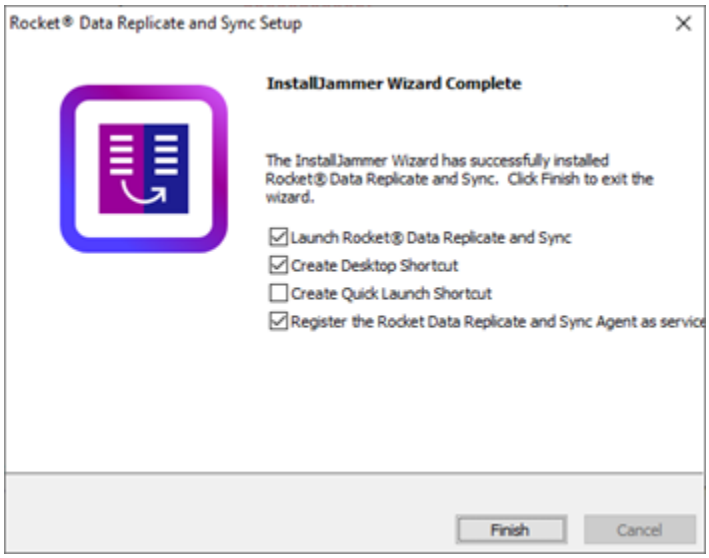


   Copying the files can take some time depending on the performance of the computer. About 150 megabytes of disk space is required for the Data Replicate and Sync installation.

   This page is displayed if the installation was successful. The Data Replicate and Sync Agent can be registered as a service now.



If a **JAVA_HOME** environment variable was specified or a valid JAVA installation was found during the installation the option of creating shortcuts will be available on this page and the Dashbaord can be launched from the setup.



Otherwise, these shortcuts can not be created and just the agent will be started on finishing the installation.

The Dashboard can be started manually using the shortcuts or by double clicking on the `Dashboard.jar` in the installation directory.

6. Select the **Finish** button.

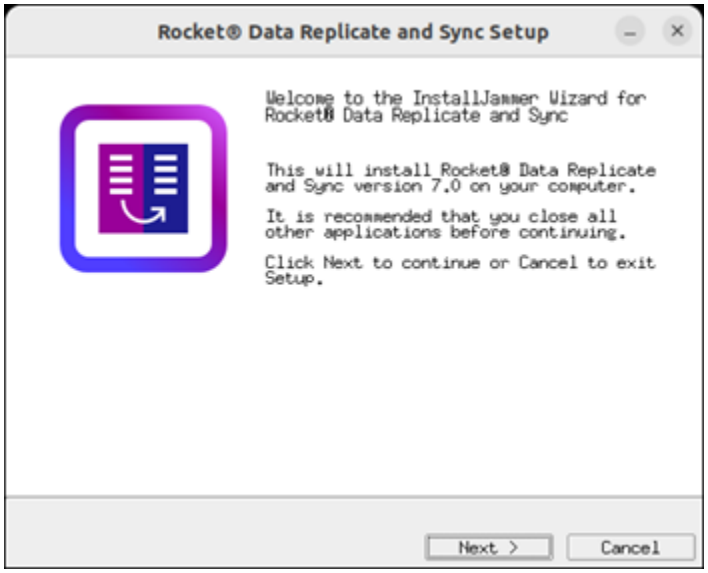
Linux installation

The Linux installation is distributed as `*.setup` file. It is available in the download area.

Before you begin:

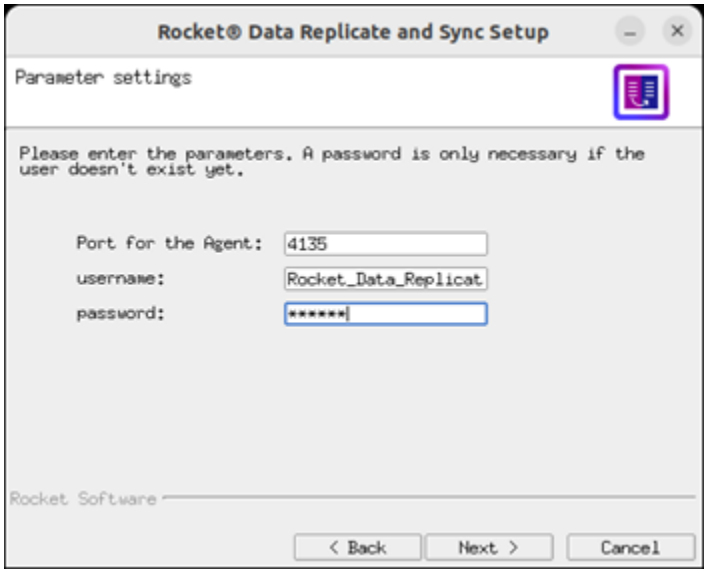
- UnixODBC version 2.1.14 or higher and Java version 1.7.0 or higher are required for Data Replicate and Sync. If the required versions are not installed, the installation will be terminated with a corresponding error message.
- If the Data Replicate and Sync Agent should be registered as service or if the setup should create a user for the Data Replicate and Sync installation, the assistant has to be started with `sudo` or as `root` user.

1. Execute the `.setup` file.
   The welcome pane of the installation wizard is displayed.



   The installation can be terminated at any time.

2. Select the **Next** button.
   The **Parameter settings** pane is displayed.

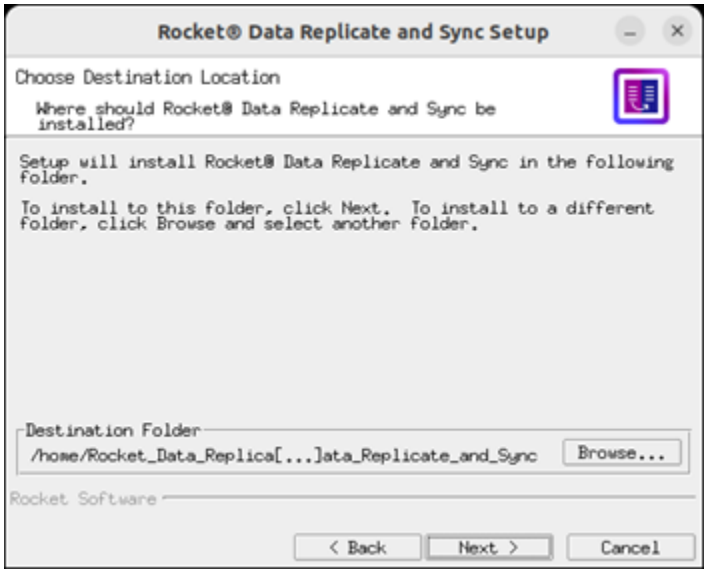


   The installation for Linux/UNIX installs the components for one user. This user owns the installation directory and runs the service if the Agent is registered as service. The user running Data Replicate and Sync should not be the root user and the agent not be started with root permissions.

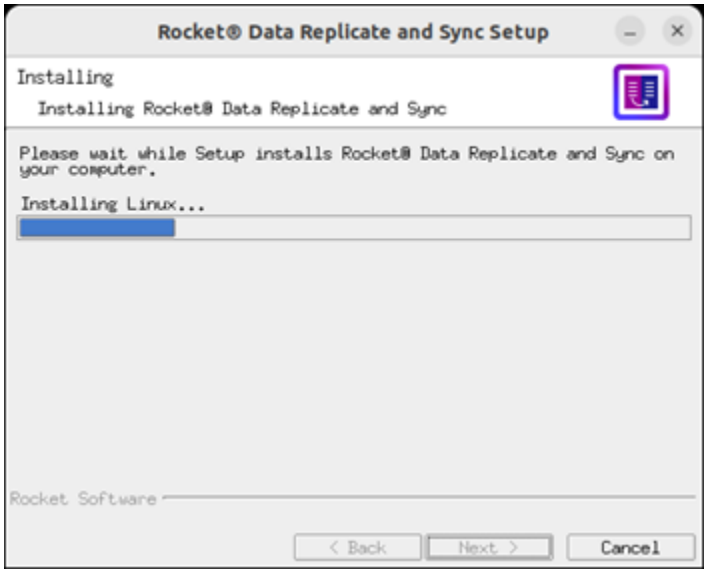
   If the selected user does not exist yet, the initial password for the new user has to be specified in the

password field. If the user already exists, you can ignore the **password** field.

3. Make sure that the specified port is not used by another program and that it is open on the firewall.
   After the installation, the port can be customized in file `tcAgent.ini`.
4. Select the **Next** button.
   The user is created and the Choose Destination Location pane is displayed.

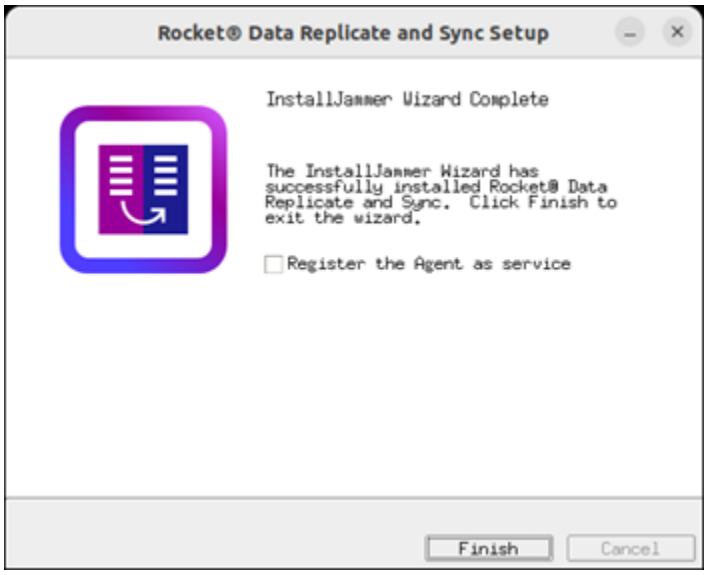


5. On this page the directory in which the Data Replicate and Sync files should be stored is specified. The default directory is `Data Replicate and Sync` in the `home` directory of the selected user. A new directory can be selected with the **Browse** button.
6. Select the **Next** button.
   The **Installing** pane is displayed.



Copying the files can take some time depending on the performance of the computer. About 150 megabytes of disk space is required for the Data Replicate and Sync installation.

When the installation is finished, the **InstallJammer Wizard Complete** pane is displayed.



The Data Replicate and Sync Agent can be registered as a service now. In this case, it is started in the background.

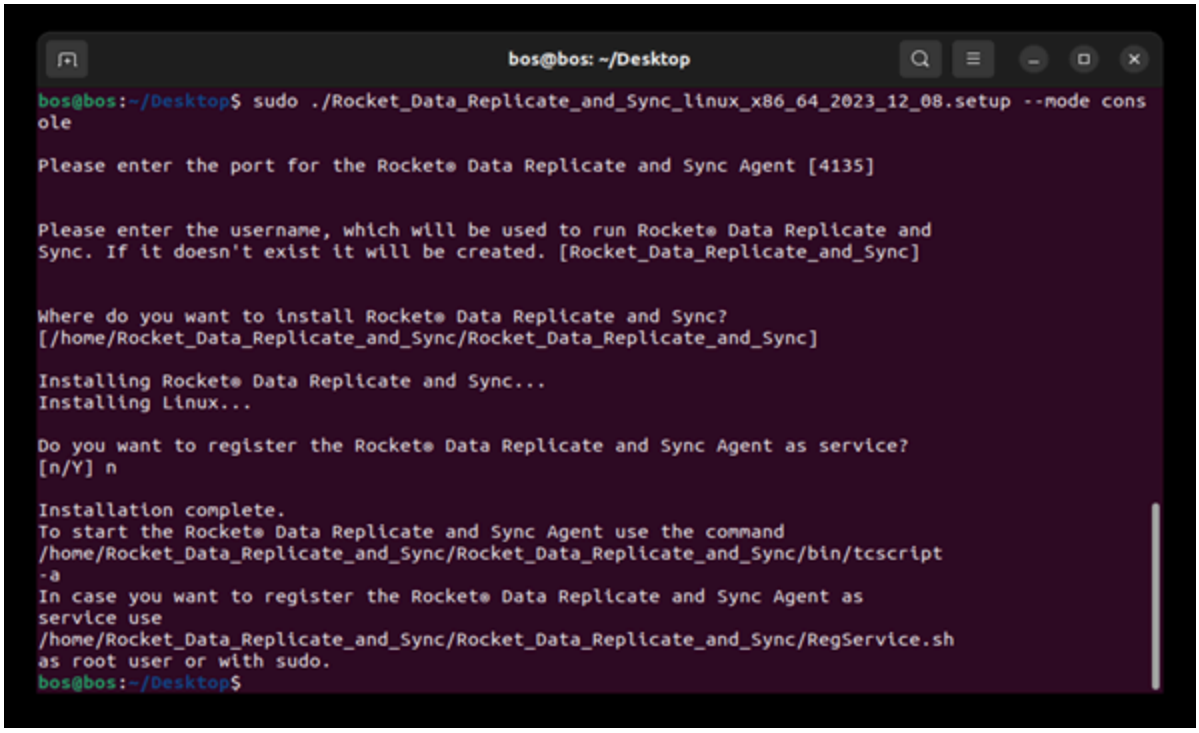
If the Data Replicate and Sync Agent is not registered as a service, the Data Replicate and Sync Agent can be started with `tcscript -a` .

Console installation for Linux

If a graphical output is not possible, the installation is performed in console mode. Performing the setup with sudo or as root user allows to register the agent as a service, otherwise this part has to performed manually at a later time. Setup has to be called as `root` user.

The console mode can be enforced with `setup --mode console` even if a graphical output is possible.

The following shows an example installation in console mode. The user Data Replicate and Sync runs as should not be the root user and the agent not be started with root permissions.



The installation in console mode can be terminated at all times with `Ctrl` + `c` . Please note that the user is created after the user prompt if the user does not exist yet. This user will not be removed if the installation is terminated.

There are default values for all necessary entries except for the password for creating a new user. If they are correct, only the enter key has to be pressed. For Yes/No questions the capitalized letter is the default value.

In the example above the default values (press enter key) were used. Only the passwords for the new user was entered. This entry is not displayed on the console for security reasons.

Data Replicate and Sync z/OS installation

> **Important:**
>
> The small and bold portions of the following JCL examples have to be adjusted to the environment.
>
> It is highly recommended to always use the examples from the installed Data Replicate and Sync files since they are always kept up-to-date. The following JCL examples are only up-to-date as of the creation of this document.

The Data Replicate and Sync Host Installation is distributed in the form of binary PC files.

Loading the installation files

The following files are required by the Data Replicate and Sync installation:

- `MVSINST.BIN` The Data Replicate and Sync INSTLIB in binary format
- `MVSLOAD.BIN` The Data Replicate and Sync LOADLIB in binary format
- `MVSMAC.BIN` The Data Replicate and Sync MACLIB in binary format

On the host, three sequential files (PS format) in which the libraries are transferred via a PC file transfer or FTP are required.

- `Data Replicate and Sync.MVSINST.BIN`
- `Data Replicate and Sync.MVSLOAD.BIN`
- `Data Replicate and Sync.MVSMAC.BIN`

The format of every file:

```
Organization  . . . : PS
Record format . . . : FB
Record length . . . : 80
Block size  . . . . : e.g. 27920
1st extent cyls . . : 10
Secondary cyls  . . : 5
```

> **Important:** The three files `MVSINST.BIN`, `MVSLOAD.BIN`, and `MVSMAC.BIN` have to be transferred in binary format to those sequential files. They must not be transferred with the option CRLF ASCII. This can be done with a terminal file transfer program or via FTP.

After the transfer, the data can be implemented into the Data Replicate and Sync libraries with the following job:

```
//RESTORE  JOB CLASS=A,MSGCLASS=F
//*
//RESTINST EXEC PGM=IKJEFT01
//IN       DD DISP=SHR,DSN=Data Replicate and Sync.MVSINST.BIN
```

```
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDDNAME(IN)
DSN('Data Replicate and Sync.INSTLIB') UNIT(unit) VOLUME(volser) DIRECTORY(30)
/*
//RESTLOAD EXEC PGM=IKJEFT01
//IN       DD DISP=SHR,DSN=Data Replicate and Sync.MVSLOAD.BIN
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDDNAME(IN)
DSN('Data Replicate and Sync.LOADLIB') UNIT(unit) VOLUME(volser) DIRECTORY(100)
/*
//RESTMAC  EXEC PGM=IKJEFT01
//IN       DD DISP=SHR,DSN=Data Replicate and Sync.MVSMAC.BIN
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDDNAME(IN)
DSN('Data Replicate and Sync.MACLIB') UNIT(unit) VOLUME(volser) DIRECTORY(30)
/*
```

The Data Replicate and Sync libraries are created by the `RECEIVE` command in the correct format and the corresponding size, and then cataloged.

APF authorization of the Data Replicate and Sync loadlib

The loadlib has to be APF authorized prior to the first start of the Data Replicate and Sync host component if the connection to Db2 should be done via RRSAF or if security checks should be done via SAF systems (e.g. RACF).

Please register the loadlib in the respective member of SYS1.PARMLIb.

The following console command is used for the APF authorization in the running system:

```
SETPROG APF,ADD,DSNAME=Data Replicate and Sync.LOADLIB,VOLUME=volser
```

The following console command is used if the Data Replicate and Sync loadlib resides on a SMS managed volume:

```
SETPROG APF,ADD,DSNAME=Data Replicate and Sync.LOADLIB,SMS
```

Creation and initialization of the Data Replicate and Sync configuration file

Data Replicate and Sync requires information at run-time that can be maintained by administrators using the Data Replicate and Sync Control Board or the Data Replicate and Sync Dashboard. This information is stored in a VSAM/RRDS file with an internal data file system. Run-time information is also saved to this file.

The following job control allocates the file. A sample of the job is available as member DEFDISK in the INSTLIB.

```
//DEFDISK JOB  ,'DEF TVSM DISK',CLASS=A,MSGLEVEL=(1,1),MSGCLASS=A
//*
//*****************************************************************
//*      DEFINE RDRS CONTROL DISK                                 *
//*****************************************************************
```

```
//*
//IDCAMS   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
         DEFINE CLUSTER                          -
         (      NAME(Data Replicate and Sync.DISK.RRDS) -
                NUMBERED                         -
                SHR(2)                           -
                VOL(volser)                      -
                RECSZ(32760 32760)               -
                REC(100 100))                    -
         DATA(NAME(Data Replicate and Sync.DISK.DATA))
/*
```

The following job control initializes the file. A sample of the job is available as member INITDISK in the INSTLIB.

```
//INITDISK JOB  ,'INIT TVSM DISK',CLASS=A,MSGLEVEL=(1,1),MSGCLASS=A
//*
//*****************************************************************
//*      INITIALIZE RDRS CONTROL DISK                             *
//*****************************************************************
//*
//INIT     EXEC PGM=TVSVDIUP,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=Data Replicate and Sync.LOADLIB
//TVSMDSK  DD DISP=SHR,DSN=Data Replicate and Sync.DISK.RRDS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
FILE TVSMDSK
PART CONFIG
MODE U

FDISK FORMAT SIZE=5MB

MD SCRIPTS
MD SECURITY
MD PROJECTS
MD CODETABLES
/*
```

Installing Db2 access (optional)

The following will explain how to install the database access. All requirements regarding the repository can be found in manual *Data Replicate and Sync – the Repository*.

The job control for the Bind is part of member BINDDB2 (member BINDDB12 for Db2 as of version 12) in the INSTLIB.

```
//BINDDB2  JOB  ,'BIND DB2',CLASS=A,MSGLEVEL=(1,1),MSGCLASS=A
//*****************************************************************
//* BIND THE PLAN FOR RDRS AGENT ACCESS TO DB2                    *
//*****************************+***********************************
//BINDUNL EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
```

```
//STEPLIB  DD DSN=db2.SDSNLOAD,DISP=SHR
//         DD DSN=db2.RUNLIB.LOAD,DISP=SHR
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(xxxx)
  BIND PLAN(TVS70DBM) MEM(TVS70DBM) +
       CURRENTDATA(NO) ACT(REP) ISO(CS) ENCODING(EBCDIC) +
       LIB('rdrs.MACLIB')
  BIND PLAN(TVS70DBU) MEM(TVS70DBU) +
       CURRENTDATA(NO) ACT(REP) ISO(CS) +
       LIB('rdrs.MACLIB')
```

Starting the Data Replicate and Sync Host Agent

Before you begin:

The Data Replicate and Sync Host Agent can be run as job or started task. A sample for the job is available as member RUNAGNT in the INSTLIB.

```
//RUNAGNT  JOB  ,'RUN RDRS',CLASS=A,MSGLEVEL=(1,1),MSGCLASS=A
//*
//*****************************************************************
//*      EXECUTE RDRS HOST AGENT                                  *
//*****************************************************************
//*
//RUN      EXEC PGM=TCSCPZUM,REGION=0M,PARM='PREF=SYS'
//STEPLIB  DD DISP=SHR,DSN=Data Replicate and Sync.LOADLIB
//TVSMDSK  DD DISP=SHR,DSN=Data Replicate and Sync.DISK.RRDS
//SYSENV   DD DISP=SHR,DSN=Data Replicate and Sync.MACLIB(SYSENV)
//STDENV   DD DISP=SHR,DSN=Data Replicate and Sync.MACLIB(STDENV)
//SYSOUT   DD SYSOUT=*
//SYSERR   DD SYSOUT=*
//SYSTRC   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDTRC   DD SYSOUT=*
//SYSRDR   DD SYSOUT=(A,INTRDR)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPARM  DD *
$Agent
Sysid=ZOS-Agent
Language=DE
TCPIPName=TCPIP
Port=4120
ProtocolDir=(JES)
/*
```

Note: Please note that the TCP/IP region specified in the Data Replicate and Sync job has to be available in case that components requiring TCP/IP are activated during the start of the Data Replicate and Sync Agent.

tcAgent start parameters

The start parameters define the methods of the Data Replicate and Sync Agents and control the initialization of Data Replicate and Sync. Not all parameters are required or available on every platform, and not all parameters have to be specified. Default values are assigned to various parameters and will be underlined in the following documentation.

- In a LUW (Linux/Unix/Windows) environment the start parameters are read from the file `tcAGENT.ini`.
- In a z/OS environment the start parameters are read from the file `SYSPARM`.

General settings

Sysid: System identification in the Agent's network

Format:	`Sysid=`*`sysid`*
Use:	All platforms
Required:	Yes Each Data Replicate and Sync Agent must have a unique ID. If this ID is missing, the startup is terminated with the error message TVS004. The SYSID is an ID with a maximum of 20 characters divided into an optional group name and the agent name. Both names are separated by a slash.
Examples:	`Sysid=ZOS-AGENT` `Sysid=WIN-AGENT`

Language: message language

Format:	Language=[DE|EN|AM]
Use:	All platforms
Required:	No This parameter specifies the language of the messages.
Examples:	Language=DE German Language=AM American English Language=EN British English

MessageSuppressConsole: suppress output of messages

Format:	MessageSuppressConsole=[n|n-m]
Use:	All platforms
Required:	No This parameter specifies whether certain messages should also be displayed on the console/stdout and the system log.

	Default is that all messages are displayed on the console/stdout.
Examples:	`MessageSuppressConsole=46` `MessageSuppressConsole=200-204`

CCSID: local codepages

Format:	`CCSID=...`
Use:	All platforms
Required:	No This parameter specifies the default codepage number for metadata imports. If this parameter is not specified, 1252 is the default for LUW agents and 37 is the default for z/OS.
Examples:	`CCSID=1141 Austria, Germany EBCDIC with euro sign` `CCSID=1047 Latin-1 / Open Systems EBCDIC`

ConfDirectory: location for configuration information

Format:	`ConfDirectory=...`
Use:	All platforms
Required:	• Linux/Unix/Windows: Yes • z/OS: No Configuration information is saved to this location at runtime (e.g. position of the repository). **LUW Agent** Specification of a fully qualified path to save files. **z/OS Agent** Specification of a path in the Data Replicate and Sync configuration file. If this is missing, "TVSMDSK:CONFIG:/" is assumed.

WorkDir: location for temporary files

Format:	`WorkDir=...`
Use:	All platforms

<table>
<tr><td>Required:</td><td>No

This parameter specifies the settings for the temporary files (e.g. restart files).

Windows Agent:
Specification of a fully-qualified path to save files. If this is missing, <code>%temp%</code> is assumed (the temporary directory defined in Widnows).

Unix/Linux Agent:
Specification of a fully-qualified path to save files. If this is missing, <code>/tmp</code> is assumed (the user's temporary directory defined in Linux/Unix).

z/OS Agent:
Datasets (z/OS) are created. The given name is used as prefix for the name creation. Required specifications for the allocation of files (UNIT=,VOLSER=etc.) can be given in the following specification:
<table><tr><td>RECFM=</td><td>Record format ['F', 'V', 'FB', 'VB']</td></tr><tr><td>LRECL=</td><td>Logical record size</td></tr><tr><td>BLKSIZE=</td><td>Block size</td></tr><tr><td>SPACE=</td><td>Space type [TRK, CYL, BLK]</td></tr><tr><td>PRI=</td><td>Primary space</td></tr><tr><td>SEC=</td><td>Secondary space</td></tr><tr><td>UNIT=</td><td>UNIT</td></tr><tr><td>VOLSER=</td><td>VOLSER</td></tr><tr><td>RLSE=</td><td>RLSE=1 Relase unused space at close</td></tr></table></td></tr>
</table>

```
WorkDir=TCVWORK;RECFM=VB;LRECL=32000;BLKSIZE=32004;SPACE=CYL;PRI=10;UNIT=3390;VOLSER=VOLUME
```

If the definition is longer than 80 characters, the remaining characters are given in the next line without an additional keyword in column 1.

ProtocolDir: protocol output

Format:	`ProtocolDir=...`
Use:	All platforms
Required:	Yes This parameter defines the location for protocol output for the Agent and the data processes running on the Agent. LUW Agent Specification of a fully-qualified pathf ro file output

	z/OS Agent (JES) for the output in JES or file specification as for WorkDir=

MaxLogFileAge: automatic deletion of protocol files

Format:	`MaxLogFileAge=n[m|h|d|w|y]`
Use:	LUW
Required:	Maximum age of the protocol files, all files in the protocol directory are deleted once the age has been reached. The value can be supplemented by `h` (hours), `d` (days), `w` (weeks) or `y` (years). Times less than two minutes are not taken into account. Specifying `MaxLogFileAge=0` disables the mechanism. Default: 90d

TraceLevel: activating tracing

Format:	`TraceLevel=[0|1|2|3]`
Use:	All platforms
Required:	No To analyze a problem, the support team may need a trace of the agent run. This parameter controls whether and to what extent data is output for tracing. This information is written to the directory which was specified with the parameter `ProtocolDir=`. If `0` is specified, no data is written, for `1`-`3` more detailed information is written.

Security: activating security checks

Format:	`Security=[N|U|S|I|E|L]`
Use:	All platforms
Required:	No This parameter defines the security settings. When the security settings are active, Data Replicate and Sync checks the user ID and the password of incoming connections as well as administrative procedures in the specified security system. The Data Replicate and Sync access checks are performed on a logical level. It can only be checked whether the resource is accessible to the user based on the Data Replicate and Sync security settings. Security settings that have been defined on an operating system level are not affected or overridden by it. A user can only access operating system resources that are also available to Data Replicate and Sync. If

	Data Replicate and Sync is not allowed to access a resource, the user will also not be able to access it, even when the Data Replicate and Sync access check has been successful.
Possible values:	**N (No)** No security checks (default) **I (Internal)** The internal security of Data Replicate and Sync will be used **E (External)** Security checks will be passed to another Agent. The name of this Agent is defined with the parameter `SecurityAgent=`. **S (SAF, only z/OS Agent):** The security of the Security Authorization Facility SAF will be used. Define SAF if you plan to use RACF, ACF/2, TOP/SECRET, or any other SAF product. The name of a security class can be defined with `SAFClass=`, the default is `$TVSCLAS`. **U (Userexit)** Specify a user program that performs security checks. The name of the module is defined with the parameter `SecurityUserExit=`. **L (LDAP, only LUW Agents)** The Security of the Lightweight Directory Access Protocol will be used. The name of the Directory Server is defined with the parameter `SecurityServer=`.

TCP/IP settings

TCPIPName: name of the TCP/IP component

Format:	`TCPIPName=[name]`
Use:	z/OS
Required:	No This parameter specifies the name of the TCP/IP component on a mainframe. z/OS: The name of the TCP/IP region (default TCPIP)

IPAddress: address for the bind of the TCP/IP listener

Format:	`IPAddress=[0.0.0.0]`
Use:	All platforms
Required:	No This parameter specifies to which TCP/IP address the Listener should be bound for making contact to

	the Control Board / Dashboard and Agents on other platforms. If this parameter is missing, it will be bound to the default address 0.0.0.0.

Port: port for the bind of the TCP/IP listener

Format:	`Port=[4120]`
Use:	All platforms
Required:	No The parameter specifies to which TCP/IP port the Listener should be bound for making contact to the Control Board / Dashboard and Agents on other platforms. If this parameter is missing, it will be bound to the default port 4120.

SSLIPAddress: address for the bind of the SSL TCP/IP listener

Format:	`SSLIPAddress=`
Use:	All platforms
Required:	No This parameter controls to which SSL TCP/IP address the Listener should be bound for making contact to the Control Board / Dashboard and Agents on other platforms. If this parameter is missing, it will be bound to the default address 0.0.0.0.

SSLPort: port for the bind of the SSL TCP/IP listener

Format:	`SSLPort=`
Use:	All platforms
Required:	No This parameter specifies to which TCP/IP port the SSL Listener should be bound for making contact to the Control Board / Dashboard and Agents on other platforms. If this parameter is missing, no SSL Listener is started.

ServerSSLCertDef: definition of a file with information about certificates to be used for the SSL TCP/IP listener

Format:	`ServerSSLCertDef=`

Use:	z/OS, Linux/Unix/Windows
Required:	No The parameter specifiesthename of a definition with information about the certificates to be used when accepting SSL/TLS secured TCP/IP connections (ACCEPT on*SSLPort*). A certificate definition is created using the Dashboard via another TCP/IP communication paths stored on the Agents local file system.

ProcessSSLCertDef: definition of a file with information about certificates to be used for process-agent-communication

Format:	`ProcessSSLCertDef=`
Use:	z/OS, Linux/Unix/Windows
Required:	No This directive defines the name of a definition with information about certificates to be used when processes communicate to their local Agent with SSL/TLS secured TCP/IP connections. As soon as this definition becomes active, processes only communicate with their local Agent using SSL/TLS secured TCP/IP connections. A certificate definition is created using the Dashboard via another TCP/IP communication paths stored on the Agents local file system.

MailServer: connect to a mail server

Format:	`MailServer=`
Use:	All platforms
Required:	No Status messages of data and monitoring processes are sent through this mail server if desired.
Examples:	`MailSever=192.168.0.10:25` `MailServer=intern.net.mailserver`

IPUseGetNameInfo: Control of name resolution for incoming connections

This parameter determines whether an attempt should be made to determine the IP name for incoming connections (default Yes).

Format:	IPUseGetNameInfo=[Yes|No]
Use:	All platforms
Required:	No

RemoteAgentIdleTimeout: Disconnection of agent connections

This parameter defines the time period in seconds after which connections to external agents should be automatically terminated if there is no activity (default: no timeout).

Format	RemoteAgentIdleTimeout=[seconds]
Use:	All platforms
Required:	No

Database-related settings

DB2Attach: connection method for Db2 under z/OS

Format:	`DB2Attach=[C|R]`
Use:	z/OS
Required:	No The parameter specifies which method is used to connect to a Db2 subsystem under z/OS. **R (RRSAF)** Use the Recoverable Resource Manager Services Attachment Facility **C (CAF)** Use the Call Attach Facility **Note:** The current user has to have at least write permission to the Data Replicate and Sync Repository and the catalog tables. The user of a real-time script (IFI_306) has to have the `MONITOR2` permission. Different users can be assigned for the tasks.

DB2SubSystems: List of the supported Db2 instances

Format:	`DB2SubSystems=...`
Use:	z/OS
Required:	Only if the systems should be displayed to the user. A list of Db2 subsystems (z/OS) which should be displayed to the control board / dashboard user by

	Data Replicate and Sync.
Examples:	`DB2SubSystems=DSN1` `DB2SubSystems=DSN1;DSN2`

IMSIDs: list of the supported IMS instances

Format:	`IMSIDs=...`
Use:	z/OS
Required:	Only if the systems should be displayed to the user. A list of IMS systems which should be displayed to the control board / dashboard user by Data Replicate and Sync.
Examples:	`IMSIDs=IVP1` `IMSIDs=IVP1;IVP2`

AdabasSVCs: list of the supported Adabas systems

Format:	`AdabasSVCs=...`
Use:	z/OS
Required:	Only if the systems should be displayed to the user. A list of Adabas systems which are identified through the SVC nr. and should be displayed to the control board / dashboard user by Data Replicate and Sync.
Examples:	`AdabasSVCs=249` `AdabasSVCs=249;250`

DATACOM/DB: Database ID for dictionary

The id of the database containing the DATACOM/DB data dictionary for the meta data import. The default is 2.

Format:	`DatacomDictDB=...`

Use:	z/OS
Required:	No
Examples:	`DatacomDictDB=3`

DATACOM/DB: Status of components to work with

The status of the objects accessed in the DATACOM/DB data dictionary for the meta data import. The default is P.

Format:	`DatacomDictStatus=...`
Use:	z/OS
Required:	No
Examples:	`DatacomDictStatus=T`

CICSSystems: list of the supported CICS systems

Format:	`CICSSystems=...`
Use:	z/OS
Required:	Only if the systems should be displayed to the user. A list of CICS systems which should be displayed to the control board / dashboard user by Data Replicate and Sync.
Examples:	`CICSSystems=A06CICS1` `CICSSystems=A06CICS1;A06CICS2`

IDMSCVs: list of the supported IDMS systems

Format:	`IDMSCVs=...`
Use:	z/OS
Required:	Only if the systems should be displayed to the user A list of IDMS systems which should be displayed to the control board / dashboard user by Data Replicate and Sync. The display of an IDMS system has the following format: cvn[(main_program[,dictionary-database])]

	cvn CV number of the IDMS system that should be visible. **main_program** Main program which can be used to access this IDMS (optional, default "IDMS"). **dictionary-database** Database (optional, no default) in which the definitions of the schema compiler are stored (e.g. APPLDICT)
Examples:	`IDMSCVs=1(IDMS,APPLDICT)` `IDMSCVs=1(IDMS,APPLDICT);2(IDMS,APPLDICT)`

Note:

- IDMS provides access modules for batch programs. For IDMS, the Data Replicate and Sync Host Agent is such a batch program. The IDMS batch access module that should be used by the Data Replicate and Sync host agent has to be specified with the start parameter *accessmodule*, because a variety of batch access modules can be generated in an IDMS system.
- Regarding the IDMS access, the Data Replicate and Sync Host Agent is an IDMS batch program. This is why all statements and parameters have to be specified in the startup job of the Data Replicate and Sync Host Agent that are required in the respective environment for an IDMS batch program (e.g. specification of the IDMS libary/-ies, DD statements for SYSIDMS and/or SYSCTL).

SVC for DBMS extensions

Format:	`SVC=n[,SET|,RESET|,DELETE]`
Use:	z/OS
Required:	Only if VSAM CDC data is to be captured via Data Replicate and Sync DBMS extension. The Data Replicate and Sync SVC is required for DBMS extensions that run in an unauthorized environment. A TYPE 3 SVC is installed. **n:** A number between 200 and 255 can be defined. **SET:** The SVC will be installed if not already done. **RESET:** The SVC is renewed (e.g. after a Data Replicate and Sync system maintenance). **DELETE:** The SVC is uninstalled.
Examples:	`SVC=241`

DBMS Extensions for VSAM-CDC

To determine VSAM changed data on z/OS systems without an installed IBM CICS® VSAM Recovery (CICS VR) function, Data Replicate and Sync can capture the changes to selected VSAM files using so-called DBMS extensions. They communicate directly with the application programs and forward the data in their own format into z/OS Logstream(s).

A correspondingly defined process for Active Logs/Realtime CDC can then process this data.

A DBMS extension for VSAM-CDC is set up directly in the JCL of the batch job or in the transaction manager. No changes to the application programs are required to set up a DBMS extension. None of the actions performed by a DBMS extension are visible to the application program.

Note: The Data Replicate and Sync DBMS extensions for VSAM-CDC use services that are only available from z/OS version 1.3.

A collector definition is used to determine which VSAM files are to be monitored and to which z/OS log stream the captured change data is to be written. For each collector used, the Data Replicate and Sync host agent allocates a z/OS system-wide memory area in subpool 241. This memory area is retained even if the Data Replicate and Sync host agent is shut down, so that changes can continue to be captured and forwarded to the z/OS log streams.

DBMS extension for VSAM files in CICS-TS

The Data Replicate and Sync DBMS extension for VSAM files in CICS-TS is available for z/OS from version 1.3.

The Data Replicate and Sync CICS/VSAM DBMS extension itself consists of the module TVSCICSEX. This is located in the Data Replicate and Sync loadlib after the Data Replicate and Sync host agent has been installed.

The Data Replicate and Sync loadlib must be included in the DFHRPL concatenation of the CICS startup JCL.

The following CICS CSD definitions are required. A job for the batch update of the CICS CSD exists as a member DEFCICS in the INSTLIB.

```
DEFINE PROGRAM(TVSCICEX) GROUP(Data Replicate and Sync) LANGUAGE(ASSEMBLER)
                   DATALOCATION(ANY) EXECKEY(CICS) CEDF(NO)
                   CONCURRENCY(THREADSAFE)
DEFINE PROGRAM(TVSDLICX) GROUP(Data Replicate and Sync) LANGUAGE(ASSEMBLER)
                   DATALOCATION(ANY) EXECKEY(CICS)
DEFINE PROGRAM(TVSCDCPR) GROUP(Data Replicate and Sync) LANGUAGE(ASSEMBLER)
                   DATALOCATION(ANY) EXECKEY(CICS)
DEFINE PROGRAM(TVSMSGEN) GROUP(Data Replicate and Sync) LANGUAGE(ASSEMBLER)
                   DATALOCATION(ANY) EXECKEY(CICS)
DEFINE PROGRAM(TVSMSGDE) GROUP(Data Replicate and Sync) LANGUAGE(ASSEMBLER)
                   DATALOCATION(ANY) EXECKEY(CICS)
DEFINE PROGRAM(TVSMSGAM) GROUP(Data Replicate and Sync) LANGUAGE(ASSEMBLER)
                   DATALOCATION(ANY) EXECKEY(CICS)
DEFINE PROGRAM(TVSMSGFR) GROUP(Data Replicate and Sync) LANGUAGE(ASSEMBLER)
                   DATALOCATION(ANY) EXECKEY(CICS)
DEFINE PROGRAM(TVSMSGIT) GROUP(Data Replicate and Sync) LANGUAGE(ASSEMBLER)
                   DATALOCATION(ANY) EXECKEY(CICS)
DEFINE PROGRAM(TVSCICGF) GROUP(Data Replicate and Sync) LANGUAGE(ASSEMBLER)
```

```
                    DATALOCATION(ANY)
*
DEFINE TRANSACTION(TVSV) GROUP(Data Replicate and Sync) PROGRAM(TVSCICEX)
DEFINE TRANSACTION(TVSD) GROUP(Data Replicate and Sync) PROGRAM(TVSDLICX)
*
ADD GROUP(Data Replicate and Sync) LIST(XYZLIST)
```

The group name `Data Replicate and Sync` is optional and can be adapted to local naming conventions. The transaction `TVSV` is used to manually switch the Data Replicate and Sync CICS/VSAM DBMS extension on and off and should only be accessible to system administrators.

The Data Replicate and Sync CICS/VSAM DBMS extension is automatically activated when the CICS starts by entering the TVSCICEX program in the CICS PLTPI.

```
DFHPLT TYPE=ENTRY,PROGRAM=TVSCICEX
```

Data Replicate and Sync CICS/VSAM DBMS extensions connect to Data Replicate and Sync collectors of type `VSME`. The identification of the collector definition represents the CICS Applid (e.g. A01CICS01).

To manage the VSME collectors through the Data Replicate and Sync Dashboard, the Data Replicate and Sync Host Agent requires information about the CICS systems (see CICSSystems: list of the supported CICS systems) and a connection to the CICS systems. During administration, it is determined for which CICS systems collectors are to be created and which files from the CICS FCT are to be monitored in these systems.

In order to connect to CICS systems, the CICS loadlib `cics.SDFHEXCI` must be included in the STEPLIB concatenation of the Data Replicate and Sync host agent.

`cics.SDFHEXCI` is mostly not APF authorized. Please authorize this loadlib before restarting the Data Replicate and Sync host agent.

Data Replicate and Sync uses the *CICS External Interface* to communicate with CICS systems. If this is not yet installed, a session/connection must be defined and installed in the CICS in addition to the Data Replicate and Sync CICS definitions. A job for the batch update of the CICS CSD exists as member `DEFEXCI` in the `INSTLIB`.

```
DEFINE CONNECTION(EXCI) GROUP (EXCI) ACCESSMETHOD(IRC)
                        PROTOCOL(EXCI) CONNTYPE(GENERIC)
DEFINE SESSION(EXCI) GROUP (EXCI) CONNECTION(EXCI)
                        PROTOCOL(EXCI) RECEIVEPFX(<) RECEIVECOUNT(5)
```

Additional information on this can be found in the IBM literature *CICS External Interfaces Guide*.

When communicating with the CICS, the program TVSCICGF is called there in order to obtain information about the VSAM files defined in the CICS/FCT.

The Data Replicate and Sync DBMS extension for VSAM can be activated and deactivated manually with transaction TVSV.

Possible calls:

TVSV START	Activating the DBMS Extension.
TVSV STOP	Deactivating the DBMS Extension. The deactivation is rejected if LUWs are still being monitored.
TVSV FSTOP	Deactivating the DBMS Extension. The deactivation is not rejected if LUWs are still being monitored.

TVSV STATE	View the status of the DBMS Extension.

Peculiarities

If applications read VSAM data records with a data record length that is too short in the `READ FOR UPDATE` command only to delete them afterwards, the activation of the so-called before exit is required. Without this additional function, Data Replicate and Sync may not receive enough information about the key values during replication. This feature is not enabled by default.

Activation on manual start:

`TVSV STARTB` Activate DBMS extension with before exit.

Activation when starting via `PLTPI`:

The value (`TVSCICEX='B'`) must be added to the CICS SIT parameter **INITPARM**.

DBMS extension for VSAM files in z/OS batch

The Data Replicate and Sync VSAM/BATCH DBMS extension itself consists of the module `TVSVSMBX`. This is located in the Data Replicate and Sync loadlib after the Data Replicate and Sync host agent has been installed.

The Data Replicate and Sync VSAM/BATCH DBMS extension `TVSVSMBX` is entered in the JCL of the batch job instead of the application program to be called. The name of the actual program is passed in the `JCL PARM=` statement. After the initialization of the VSAM monitoring, control is then passed to the original application program.

The Data Replicate and Sync Loadlib must be included in the `STEPLIB` concatenation of the batch job.

Necessary changes to the JCL

Original JCL:

```
//BATCHJOB JOB ...
//*
//STEP1 EXEC PGM=programm
//STEPLIB DD DSN=user.application.loadlib,DISP=SHR
//VSAMFILE DD DSN=user.vsam.dataset,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
...
```

Adapted JCL:

```
//BATCHJOB JOB ...
//*
//STEP1 EXEC PGM=TVSVSMBX,PARM=(collinfo,programm)
//STEPLIB DD DSN=user.application.loadlib,DISP=SHR
// DD DSN=Data Replicate and Sync.loadlib,DISP=SHR
//VSAMFILE DD DSN=user.vsam.dataset,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
...
```

Description of the parameters for PARM=

Description of the parameters for `PARM=`:

collinfo:
Identification of the Data Replicate and Sync collector in which the CDC data is to be saved.

programm:
Original application program of the batch job.

If data was originally also transferred to the application program with the `PARM=` instruction, these are appended to the new parameters, separated by a comma:

- Before: `PARM=(data1,data2,...)`
- After: `PARM=(collinfo,programm,data1,data2,...)`

After the initialization of the VSAM monitoring, the additional data is transparently transmitted to the application program.

Data Replicate and Sync VSAM/BATCH DBMS extensions connect to Data Replicate and Sync collectors of type `VSMB`. The identification of the collector to be used results from the information `collinfo` of the `PARM=` instruction.

Installing the VSAM update server in z/OS

The maintenance of VSAM files can optionally be done via an update server under CICS. This is necessary when the VSAM files that should be maintained are allocated for online operation so that the Data Replicate and Sync Host Agent cannot allocate them for change accesses.

The Data Replicate and Sync VSAM update server under CICS is accessed by the processes via TCP/IP. It is irrelevant whether the Apply process takes place on the host or on a workstation.

Installation CICS VSAM update server z/OS

The following CICS CSD definitions are necessary. For the batch update of the CICS CSD there is a Job containing all the necessary definitions as member with the name `DEFCICS` in the `INSTLIB`.

```
DEFINE PROGRAM(TVSCIVSM) GROUP(RDRS) LANGUAGE(ASSEMBLER)
               DATALOCATION(ANY)
*
DEFINE TRANSACTION(TVSC) GROUP(Data Replicate and Sync) PROGRAM(TVSCIVSM)
*
ADD GROUP(Data Replicate and Sync) LIST(XYZLIST)
```

The group name *RDRS* is optional and can be adjusted to the local naming conventions.

For the integration of the Data Replicate and Sync VSAM update server into z/OS TCP/IP, the IBM CICS TCP/IP socket interface has to be installed. The IBM standard listener CSKL is used.

The following values are recommended for the IBM Listener CSKL (statement in EZAC DIS LISTENER):

```
APPLID ===> xxxxxxxx APPLID of CICS System
TRANID ===> CSKL Transaction Name of Listener
PORT ===> 04180 Port Number of Listener
AF ===> INET Listener Address Family
IMMEDIATE ===> YES Immediate Startup Yes!No
BACKLOG ===> 020 Backlog Value for Listener
NUMSOCK ===> 050 Number of Sockets in Listener
ACCTIME ===> 060 Timeout Value for ACCEPT
GIVTIME ===> 010 Timeout Value for GIVESOCKET
REATIME ===> 000 Timeout Value for READ
```

Console interface under z/OS

The Data Replicate and Sync Agent under z/OS has a console interface that can output statistical values and enables active intervention with the processing.

An automation system can react to the messages of the Data Replicate and Sync Agent and can issue more commands. More detailed information about the Data Replicate and Sync console messages can be found in the manual *Data Replicate and Sync Messages*.

All commands described in this manual are passed to the Agent via the console command `MODIFY` (z/OS).

z/OS

/F jobname,**command**

jobname is the name of the Data Replicate and Sync Agent Job or the name of the Started Task.

The command DISPLAY

The command `DISPLAY` displays various information about the Data Replicate and Sync Agent.

The command `DISPLAY` can be abbreviated with `DIS`. The parameters can also be abbreviated. The abbreviation is marked with an underscore.

Format:

z/OS:

```
/MODIFY jobname,DISPLAY parameter
```

or

```
/F jobname,DISPLAY parameter
```

The following table contains an overview of the possible parameters.

Parameter	Description
`USER ALL`	Displays all connected users. Example: `F TVSMTEST,DIS USER ALL` `TVS020I Console command received: DIS USER ALL` `TVS403I List of connected user (1 entries):` `TVS404I Name: John (192.168.0.68:55494)`
`USER` *`name`*	Displays a certain user. Example:

Parameter	Description
	`` F TVSMTEST,DIS USER JOHNDOE TVS020I Console command received: DIS USER JOHNDOE TVS406I Username.............: JOHNDOE TVS407I IP Address..........: 192.168.000.232:27837 TVS409I Signed on since.....: 2007.07.04 06:21:33.690190 TVS410I Last activity at....: 2007.07.04 06:44:38.234620 ``` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.
`AGENT ALL`	Displays all external Agents. Example: ``` F TVSMTEST,DIS AGENT ALL TVS020I Console command received: DIS AGENT ALL TVS431I List of known Agents (2 entries): TVS432I Name: W10-Agent-II (192.168.0.68:4120) TVS432I Name: UNX-Agent-II (192.168.0.45:4120) ```
`AGENT` *`name`*	Displays a certain Agent. Example: ``` F TVSMTEST,DIS AGENT W10-Agent-II TVS020I Console command received: DIS AGENT W10-Agent-II TVS434I Agent...............: W10-Agent-II TVS435I IP Address..........: 192.168.0.68:4120 TVS436I Connected...........: Yes TVS437I Connected since.....: 2019-02-28-17.07.56.555252 TVS438I Last activity.......: 2019-02-28-17.35.49.489640 TVS439I Host version........: 7.0.0 TVS441I Operating system....: MSWIN TVS442I Computer Date/Time..: 2019-02-28-16.35.50.335735 ``` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.
`PROCESS ALL`	Displays all active processes. Example: ``` F TVSMTEST,DIS PROCESS ALL TVS020I Console command received : DIS PROCESS ALL ```

Parameter	Description
	`` TVS450I List of active process tasks (1 entries): TVS451I Name: IMSLOG Reader, TCB=1FE5E5E4 ```
PROCESS *name*	Displays a certain active process. Example: ``` TVS020I Console command received: DIS PROCESS 'IMSLOG Reader' TVS453I Process-Name.............: IMSLOG Reader TVS454I Process origin...........: TVS455I Process-class...........: A TVS456I Process-Task TCB.........: 008BD620 TVS457I Start Date/Time..........: 2019-02-28-15.08.05.271414 TVS458I State of process task....: TVS459I Last state-change........: 2019-02-28-15.08.05.525235 TVS466I Input: Control records.: 0 TVS467I Data blocks.....: 0 TVS468I Data records....: 0 TVS469I Output: Control records.: 0 TVS470I Data blocks.....: 0 TVS471I Data records....: 0 ``` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.
CLASS ALL	Displays all process runtime classes. Example: ``` F TVSMTEST,DIS CLASS ALL TVS020I Console command received: DIS CLASS ALL TVS420I Class: A, maximum: 99999, active: 00001, maximum reached: 00000 TVS420I Class: B, maximum: 00010, active: 00000, maximum reached: 00000 TVS420I Class: C, maximum: 00010, active: 00000, maximum reached: 00000 TVS420I Class: D, maximum: 00010, active: 00000, maximum reached: 00000 TVS420I Class: E, maximum: 00010, active: 00000, maximum reached: 00000 TVS420I Class: F, maximum: 00010, active: 00000, maximum reached: 00000 ```
CLASS *classid*	Displays a certain process runtime class. Example: ``` F TVSMTEST,DIS CLASS A TVS020I Console command received: DIS CLASS A ```

Parameter	Description
	`TVS420I Class: A, maximum: 99999, active: 00001, maximum reached: 00000` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.
`JOB ALL`	Displays all active Jobs. Example: `F TVSMTEST,DIS JOB ALL` `TVS020I Console command received: DIS JOB ALL` `TVS488I List of jobs (0007 entries):` `TVS489I ID: EVT00074, Process=ADA_BULK.TSF` `TVS489I ID: EVT00073, Process=ADA_BULK.TSF` `TVS489I ID: EVT00072, Process=ADA_BULK.TSF` `TVS489I ID: EVT00071, Process=ADA_BULK.TSF` `TVS489I ID: EVT00069, Process=ADA_BULK.TSF` `TVS489I ID: EVT00068, Process=ADA_BULK.TSF` `TVS489I ID: EVT00066, Process=ADA_BULK.TSF`
`JOB` *`jobid`*	Displays a certain Job. Example: `F TVSMTEST,DIS JOB EVT00074` `TVS020I Console command received: DIS JOB EVT00074` `TVS491I Job-ID.....................: EVT00074` `TVS492I Process-Name.............: ADA_BULK.TSF` `TVS493I Date/Time of scheduled start: 2004.12.23-01:30:32.000000` `TVS495I Date/Time of next start.....: 0000.00.00-00:00:00.000000` `TVS496I Active process.............: N` `TVS497I Last run Start.............: 2004.12.23-01:30:32.000000` `TVS498I Last run End...............: 2004.12.23-01:30:40.000000` `TVS499I Last run Returncode.........: 00000000` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.

The command START

The command `DISPLAY` displays various information about the Data Replicate and Sync Agent.

The command `DISPLAY` can be abbreviated with `DIS`. The parameters can also be abbreviated.

Format:

z/OS:

```
/MODIFY jobname,DISPLAY parameter
```

or

```
/F jobname,DISPLAY parameter
```

The following table contains an overview of the possible parameters.

Parameter	Description
`USER ALL`	Displays all connected users. Example: `F TVSMTEST,DIS USER ALL` `TVS020I Console command received: DIS USER ALL` `TVS403I List of connected user (1 entries):` `TVS404I Name: John (192.168.0.68:55494)`
`USER` *name*	Displays a certain user. Example: `F TVSMTEST,DIS USER JOHNDOE` `TVS020I Console command received: DIS USER JOHNDOE` `TVS406I Username.....: JOHNDOE` `TVS407I IP Address.....: 192.168.000.232:27837` `TVS409I Signed on since.....: 2007.07.04 06:21:33.690190` `TVS410I Last activity at....: 2007.07.04 06:44:38.234620` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.
`AGENT ALL`	Displays all external Agents. Example: `F TVSMTEST,DIS AGENT ALL` `TVS020I Console command received: DIS AGENT ALL` `TVS431I List of known Agents (2 entries):` `TVS432I Name: W10-Agent-II (192.168.0.68:4120)` `TVS432I Name: UNX-Agent-II (192.168.0.45:4120)`
`AGENT` *name*	Displays a certain Agent. Example:

Parameter	Description
	`` F TVSMTEST,DIS AGENT W10-Agent-II TVS020I Console command received: DIS AGENT W10-Agent-II TVS434I Agent.............: W10-Agent-II TVS435I IP Address.........: 192.168.0.68:4120 TVS436I Connected..........: Yes TVS437I Connected since.....: 2019-02-28-17.07.56.555252 TVS438I Last activity.......: 2019-02-28-17.35.49.489640 TVS439I Host version........: 7.0.0 TVS441I Operating system....: MSWIN TVS442I Computer Date/Time..: 2019-02-28-16.35.50.335735 ``` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.
`PROCESS ALL`	Displays all active processes. Example: ``` F TVSMTEST,DIS PROCESS ALL TVS020I Console command received : DIS PROCESS ALL TVS450I List of active process tasks (1 entries): TVS451I Name: IMSLOG Reader, TCB=1FE5E5E4 ```
`PROCESS name`	Displays a certain active process. Example: ``` TVS020I Console command received: DIS PROCESS 'IMSLOG Reader' TVS453I Process-Name.............: IMSLOG Reader TVS454I Process origin..........: TVS455I Process-class...........: A TVS456I Process-Task TCB.........: 008BD620 TVS457I Start Date/Time..........: 2019-02-28-15.08.05.271414 TVS458I State of process task....: TVS459I Last state-change........: 2019-02-28-15.08.05.525235 TVS466I Input: Control records.: 0 TVS467I Data blocks.....: 0 TVS468I Data records....: 0 TVS469I Output: Control records.: 0 TVS470I Data blocks.....: 0 TVS471I Data records....: 0 ``` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.

Parameter	Description
`CLASS ALL`	Displays all process runtime classes. Example: `F TVSMTEST,DIS CLASS ALL` `TVS020I Console command received: DIS CLASS ALL` `TVS420I Class: A, maximum: 99999, active: 00001, maximum reached: 00000` `TVS420I Class: B, maximum: 00010, active: 00000, maximum reached: 00000` `TVS420I Class: C, maximum: 00010, active: 00000, maximum reached: 00000` `TVS420I Class: D, maximum: 00010, active: 00000, maximum reached: 00000` `TVS420I Class: E, maximum: 00010, active: 00000, maximum reached: 00000` `TVS420I Class: F, maximum: 00010, active: 00000, maximum reached: 00000`
`CLASS` *`classid`*	Displays a certain process runtime class. Example: `F TVSMTEST,DIS CLASS A` `TVS020I Console command received: DIS CLASS A` `TVS420I Class: A, maximum: 99999, active: 00001, maximum reached: 00000` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.
`JOB ALL`	Displays all active Jobs. Example: `F TVSMTEST,DIS JOB ALL` `TVS020I Console command received: DIS JOB ALL` `TVS488I List of jobs (0007 entries):` `TVS489I ID: EVT00074, Process=ADA_BULK.TSF` `TVS489I ID: EVT00073, Process=ADA_BULK.TSF` `TVS489I ID: EVT00072, Process=ADA_BULK.TSF` `TVS489I ID: EVT00071, Process=ADA_BULK.TSF` `TVS489I ID: EVT00069, Process=ADA_BULK.TSF` `TVS489I ID: EVT00068, Process=ADA_BULK.TSF` `TVS489I ID: EVT00066, Process=ADA_BULK.TSF`
`JOB` *`jobid`*	Displays a certain Job. Example:

Parameter	Description
	`` F TVSMTEST,DIS JOB EVT00074 TVS020I Console command received: DIS JOB EVT00074 TVS491I Job-ID.....................: EVT00074 TVS492I Process-Name.................: ADA_BULK.TSF TVS493I Date/Time of scheduled start: 2004.12.23-01:30:32.000000 TVS495I Date/Time of next start.....: 0000.00.00-00:00:00.000000 TVS496I Active process..............: N TVS497I Last run Start..............: 2004.12.23-01:30:32.000000 TVS498I Last run End................: 2004.12.23-01:30:40.000000 TVS499I Last run Returncode.........: 00000000 ``` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.

The command STOP

The command `STOP` can be used to stop a running process.

The command `STOP` can be abbreviated with `STO`. The parameters can also be abbreviated.

Format:

z/OS:

```
/MODIFY jobname,STOP parameter
```

or

```
/F jobname,STOP parameter
```

The following table contains an overview of the possible parameters.

Parameter	Description
`PROCESSS` *name*	The process specified by the parameter is stopped. Example: ``` F TVSMTEST,STO PROCESS 'IMSLOG Reader' TVS020I Console command received: STO PROCESS 'IMSLOG Reader' TVS179I User FRANK terminates process task hvp.tsf, TCB=008D0058 TVS900I Vprocess terminated with Returncode 0501000A ``` A description of the displayed values can be found in the manual *Data Replicate and Sync Messages*.

The command SHUTDOWN

The command `SHUTDOWN` is used to shut down the Data Replicate and Sync Host Agent.

Format:

z/OS:

```
/MODIFY jobname,SHUTDOWN
```

or

```
/F jobname,SHUTDOWN
```

or

```
/P jobname
```

The command TRACE

To analyze a problem, the support team may need a trace of the agent run. The command `TRACE` is used to control writing of that trace. This information is written to the directory which was specified with the startup parameter `ProtocolDir=`.

Format:

z/OS:

```
/MODIFY jobname,TRACE [LEVEL n|ON|OFF]
```

or

```
/F jobname,TRACE [LEVEL n|ON|OFF]
```

LEVEL *n*
: If `0` is specified, no data is written, for `1` - `3` more detailed information is written.

ON
: Corresponds to specifying level 2.

OFF
: Corresponds to specifying level `0` , thus deactivating the tracewriting.

Copyright

License information on third-party components used can be found in Copyright