Event Replicator for Adabas

Installation

Version 4.1.1

October 2022

ADABAS & NATURAL

This document applies to Event Replicator for Adabas Version 4.1.1 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: ARF-INSTALL-411-20221108

Table of Contents

Preface

This document describes the prerequisites and installation procedure for installing Event Replicator for Adabas.

The document is organized as follows:

System Requirements	Describes the system requirements of Event Replicator for Adabas.
About the Installation Tape	Describes the contents of the Event Replicator for Adabas installation tape and how to copy the tape contents.
Event Replicator for Adabas Installation Steps for z/OS Systems	Describes the steps you must perform to install Event Replicator for Adabas on z/OS systems.
Event Replicator for Adabas Installation Steps for z/VSE Systems	Describes the steps you must perform to install Event Replicator for Adabas on z/VSE systems.
Event Replicator for Adabas Installation Steps for BS2000 Systems	Describes the steps you must perform to install Event Replicator for Adabas on BS2000 systems.
Post-Installation Replication Implementation Steps (All Platforms)	Describes the general steps you must make to implement replication after Event Replicator for Adabas is installed on any platform.
Running in Verify Mode	Describes what happens when you run in verify (test) mode.
Security	Describes how to implement Trusted User ID for Replication.

1 Conventions

In the product documentation, the notation `vrs`, `vr`, or simply `v` is often used as a placeholder for the current product version, for example, in data set or module names.

Placeholder	Meaning	Definition
v	version	**Major Version** The first digit of the product version number indicates major architecture and functionality implementation or enhancement that adds value to the product.
r	release	**Minor Version** The second digit of the version number indicates functionality addition or enhancement that adds value to the product.
s	system maintenance level	**Correction Level** Correction levels contain error corrections only, without new functionality, including documentation of all modifications and repairs. In case it is necessary to include functional changes into a correction level, an exception handling process ensures that corresponding quality assurance activities are triggered. These functional changes are documented. The main target is to avoid impacts when you install such a correction level. The third number of an Adabas version denotes the system maintenance level. On certain platforms supported by Adabas, additional levels may exist, such as update package, patch level, service pack and hot fix.

2 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.softwareag.cloud**. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at **https://knowledge.softwareag.com**.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

3 System Requirements

This chapter describes the system requirements of Event Replicator for Adabas.

Supported Operating System Platforms

Software AG generally provides support for the operating system platform versions supported by their respective manufacturers; when an operating system platform provider stops supporting a version of an operating system, Software AG will stop supporting that version.

For information regarding Software AG product compatibility with IBM platforms and any IBM requirements for Software AG products, please review the *Product Compatibility for IBM Platforms* web page.

Before attempting to install this product, ensure that your host operating system is at the minimum required level. For information on the operating system platform versions supported by Software AG products, complete the following steps.

1. Access Software AG's Empower web site at *https://empower.softwareag.com*.
2. Log into Empower. Once you have logged in, you can expand **Products & Documentation** in the left menu of the web page and select **Product Version Availability** to access the Product Version Availability screen.
3. Use the fields on the top of this screen to filter its results for your Software AG product. When you click the **Search** button, the supported Software AG products that meet the filter criteria are listed in the table below the filter criteria.

   This list provides, by supported operating system platform:

   - the Software AG general availability (GA) date of the Software AG product;
   - the date the operating system platform is scheduled for retirement (OS Retirement);
   - the Software AG end-of-maintenance (EOM) date for the product; and
   - the Software AG end-of-sustained-support (EOSS) date for the product.

Note: Although it may be technically possible to run a new version of your Software AG product on an older operating system, Software AG cannot continue to support operating system versions that are no longer supported by the system's provider. If you have questions about support, or if you plan to install this product on a release, version, or type of operating system other than one listed on the Product Version Availability screen described above, consult Software AG technical support to determine whether support is possible, and under what circumstances.

Supported Hardware

For general information regarding Software AG product compatibility with other platforms and their requirements for Software AG products, visit Software AG's ***Hardware Supported*** web page.

Adabas Requirements

At this time, SAF-secured Replicator databases are not supported.

This section lists the versions of Adabas required by this version of Event Replicator for Adabas. It covers the following topics:

- z/OS Requirements
- ADARUN Requirements

z/OS Requirements

On z/OS systems, Event Replicator for Adabas requires Adabas Version 8.5 SP1 (z/OS) or later, with zaps applied from the ARF*vrs*.MVSZAPS data set and any subsequent ARF*vrs*.MVSZ*nnn* data sets (if they have been provided) for z/OS. Review the $README members of these data sets for details on the zaps.

ADARUN Requirements

There are also some required ADARUN parameter settings. For more information, read Step 6 of the ***Event Replicator for Adabas Installation Steps for z/OS Systems***, Step 8 of the ***Event Replicator for Adabas Installation Steps for VSE Systems***, or Step 5 of the ***Event Replicator for Adabas Installation Steps for BS2000 Systems*** elsewhere in this guide.

Adabas Online System (AOS) Requirements

A licensed copy of Adabas Online System (AOS) and the demo version of AOS are *not* required to support Event Replicator for Adabas.

However, AOS does print some useful statistics and help with the planning and administrative functions of Event Replicator for Adabas, so it is recommended. If you choose to install it, be sure to follow the installation instructions in the AOS manual for licensed versions.

If you only use a demo copy of AOS, note that:

- Only limited information concerning Event Replicator for Adabas is available to you with the demo copy.
- The same versions and maintenance levels of the AOS demo code are required as for a licensed copy of AOS.

Adabas SAF Security Requirements

Replication of password and cipher secured files is supported in conjunction with Adabas SAF Security Version 8.2.2 (and above) and RACF.

Adabas SAF Security fix AX822007 must be applied and, if using Adabas Version 8.3, the COR822.LX03 (or above) library must be made available, concatenated above the Adabas library.

Entire Net-Work Requirements

If you want to use the Data Mapping Tool to generate a global format buffers (GFB) and field tables (GFFT), you must have some Software AG middleware components installed.

1. The Software AG Directory Server is required. This product is commonly installed with Software AG open system products. If it is already installed at your site, you should not need to install it again. The Software AG Directory Server can be installed when you install Entire Net-Work Client, which is provided with the Entire Net-Work Administration software included with Event Replicator for Adabas.
2. Event Replicator for Adabas needs the ability to communicate with Adabas database and Event Replicator Servers on the mainframe. The recommended way to do this is to install Entire Net-Work Client (or Entire Net-Work for open systems) on the client side and Entire Net-Work Administration (or Entire Net-Work for mainframes 6.5 SP1 or higher) on the mainframe. Entire Net-Work Administration is a limited version of Entire Net-Work for mainframes and is shipped with Event Replicator for Adabas. It includes the Simple Connection Line Driver as well as Entire Net-Work Client. The combination of Entire Net-Work Client (or Entire Net-Work for open systems) and the Simple Connection Line Driver provides the mainframe communication required by Event Replicator for Adabas.

   Notes:

   a. When you use Entire Net-Work Administration (or Entire Net-Work for mainframes) and Entire Net-Work Client (or Entire Net-Work for open systems), the Event Replicator Servers and Adabas databases you maintain must be UES-enabled.

   b. As an alternative to running a separate Net-work session, the Event Replicator Server can run with ADATCP. For more information, see *Adding Targets* in the Entire Net-Work Administration documentation. For further details on ADATCP, refer to *Entire Net-Work for Mainframes > Point-to-Point Support for Adabas (ADATCP)*.

If the appropriate Entire Net-Work mainframe and client products are not already installed on your systems, install Entire Net-Work Administration on the mainframe and Entire Net-Work Client on the client side. For complete information on these products, read the *Entire Net-Work Administration documentation* in *Entire Net-Work Administration Installation Guide* and *Entire Net-Work Client Administration* in *Entire Net-Work Client Installation and Administration Guide*.

in *Entire Net-Work Client Installation and Administration Guide*.

Messaging System (webMethods EntireX or IBM WebSphere MQ) Requirements

A messaging system is not required for all Event Replicator functions. For example, one is not required for Adabas-to-Adabas replication or for Adabas-to-File replication. However, if your use of Event Replicator for Adabas requires a messaging system, the following messaging system requirements must be met:

- On z/OS platforms, webMethods EntireX version 10.3 or later (this is the preferred messaging system) or IBM WebSphere MQ version 7.1 or later are required.
- If you elect to use the BROKER module under z/OS, review the security considerations in the EntireX Broker documentation for administration of Broker stubs under z/OS.

Note: If you are using IBM WebSphere MQ Series definitions for your Event Replicator DESTINATION or IQUEUE definitions, a S0D3 abend can occur if you run it as a started task and specify the parameter REUSASID=YES. This is a documented IBM WebSphere MQ Series issue.

Refer to the webMethods EntireX documentation, the IBM WebSphere MQ documentation, and the section entitled *Messaging System Integration* (in the *Event Replicator for Adabas Administration and Operations Guide*) for details on how to configure the messaging system for use with the Event Replicator for Adabas.

Event Replicator Target Adapter Requirements

Support for Event Replicator for Adabas 4.1 SP1 is provided in Event Replicator Target Adapter 3.5 or above.

Natural Requirements

Natural 8.2 SP7 or later is required when using the Adabas Event Replicator Subsystem to set up replication definitions. Replication itself is independent of your version of Natural.

Predict Requirements

If you will be using the Event Replicator for Adabas feature that allows you to generate global format buffers and a field table using Predict, a **supported version** of Predict must also be installed. Otherwise, you will not be able to use this feature.

4 About the Installation Tape

This chapter describes the installation tape.

Data Sets Delivered

This section describes the data sets that are delivered with the Event Replicator for Adabas. In all data set names, *vrs* represents the version, release, and maintenance level numbers of the release of the Event Replicator for Adabas.

The following data sets are delivered with the Event Replicator for Adabas on z/OS platforms:

Data Set Name	Contains
ARF*vrs*.INPL	INPL of the online application
ARF*vrs*.JOBS	Sample JCL
ARF*vrs*.LOAD	Load modules
ARF*vrs*.MTSI	Sample input to ADAMTS
ARF*vrs*.SRCE	Sample source members
ARF*vrs*.ZAPS	Zaps for support of the Event Replicator for Adabas

The following data sets are delivered with the Event Replicator for Adabas on BS2000 platforms:

Data Set Name	Contains
ARF*vrs*.INPL	INPL of the online application
ARF*vrs*.MOD	Module library
ARF*vrs*.SRC	Sample JCL and sources
ARF*vrs*.ZAPS	Zaps for support of the Event Replicator for Adabas
ARF*vrs*.JOBS	Example installation jobs

On z/VSE platforms, the following data sets are delivered with the Event Replicator for Adabas:

Data Set Name	Contains
ARF*vrs*.INPL	INPL of the online application
ARF*vrs*.LIBR	Objects, phases, sample sources, jobs, and input to ADAMTS
ARF*vrs*.ZAPS	Zaps for support of the Event Replicator for Adabas

Copying the Tape Contents

This section describes how to copy the contents of the installation tape in different environments.

- z/OS Environments
- z/VSE Environments
- BS2000 Environments

z/OS Environments

Copy the data sets from the supplied installation medium to your disk before you perform the individual installation procedure for each component to be installed.

The way you copy the data sets depends on the installation method and the medium used:

- If you use System Maintenance Aid (SMA), refer to the copy job instructions provided in the *System Maintenance Aid* documentation.
- If you are not using SMA and want to copy the data sets from CD-ROM, refer to the README.TXT file on the CD-ROM.
- If you are not using SMA and want to copy the data sets from tape, follow the instructions in this section.

This section explains how to copy all data sets from tape to disk.

- Step 1: Copy Data Set COPY.JOB from Tape to Disk
- Step 2: Modify hilev.COPY.JOB on Your Disk
- Step 3: Submit COPY.JOB

Step 1: Copy Data Set COPY.JOB from Tape to Disk

- Modify the following sample job according to your requirements:

```
//SAGTAPE JOB SAG,CLASS=1,MSGCLASS=X
//* ---------------------------------
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,
// DISP=(OLD,PASS),
// UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=tape-volser),
// LABEL=(2,SL)
//SYSUT2 DD DSN=hilev.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=disk-volser,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
```

```
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

where:

tape-volser is the VOLSER of the tape, for example: `T12345`,
hilev is a valid high-level qualifier, and
disk-volser is the VOLSER of the disk.

- Execute the job to copy the data set `COPY.JOB` to your disk.

Step 2: Modify hilev.COPY.JOB on Your Disk

- Modify *hilev*`.COPY.JOB` according to your requirements:

  Set `EXPDT` to a valid expiration date, for example, `99365`.

  Set `HILEV` to a valid high-level qualifier, for example, `USERLIB`.

  Set `LOCATION` to a storage location, for example, `STORCLAS=ABC` or `UNIT=3390,VOL=SER=USR123`.

Step 3: Submit COPY.JOB

- Execute *hilev*`.COPY.JOB` to copy single, multiple, or all data sets to your disk.

z/VSE Environments

Copy the data sets from the supplied installation medium to your disk before you perform the individual installation procedure for each component to be installed.

The way you copy the data sets depends on the installation method and the medium used:

- If you use System Maintenance Aid (SMA), refer to the copy job instructions provided in the *System Maintenance Aid* documentation.
- If you are not using SMA and want to copy the data sets from CD-ROM, refer to the README.TXT file on the CD-ROM.
- If you are not using SMA and want to copy the data sets from tape, follow the instructions in this section.

This section explains how to copy the data sets `.LIBJ`, `.LIBR` and `.LICS` (if supplied) from tape to disk. All other data sets can be installed directly from the tape.

- Step 1: Copy Data Set COPYTAPE.JOB to Disk
- Step 2: Modify COPYTAPE.JOB on Your Disk

- Step 3: Submit COPYTAPE.JOB

Step 1: Copy Data Set COPYTAPE.JOB to Disk

- Modify the following sample job according to your requirements:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,                                                  +
* $$ DISP=D,LDEST=(*,UID),SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB LIBRCAT
* *****************************************
*     STORE COPYTAPE.JOB IN LIBRARY
* *****************************************
// ASSGN SYS004,nnn
// MTC REW,SYS004
// MTC FSF,SYS004,4
ASSGN SYSIPT,SYS004
// TLBL IJSYSIN,'COPYTAPE.JOB'
// EXEC LIBR,PARM='MSHP; ACC S=lib.sublib'
/*
// MTC REW,SYS004
ASSGN SYSIPT,FEC
/*
/&
* $$ EOJ
```

  where:

  nnn is the tape address, and
  lib.sublib is the library and sublibrary in which the data set `COPYTAPE.JOB` is to be stored.

- Execute the job to copy the data set `COPYTAPE.JOB` to disk.

  `COPYTAPE.JOB` contains the JCL required to copy the data sets `.LIBJ`, `.LIBR` and `.LICS` from tape to disk.

Step 2: Modify COPYTAPE.JOB on Your Disk

- Modify `COPYTAPE.JOB` according to your requirements and set the disk space parameters as appropriate.

Step 3: Submit COPYTAPE.JOB

- Execute COPYTAPE.JOB to copy the data sets .LIBJ, .LIBR and .LICS to your disk.

BS2000 Environments

Copy the files (data sets) from the supplied installation medium to your disk before you perform the individual installation procedure for each component to be installed.

The way you copy the files depends on the installation method and the medium used:

- If you want to copy the files from CD-ROM, refer to the *README.TXT* file on the CD-ROM.
- If you want to copy the files from tape, follow the instructions in this section.

This section explains how to copy all files from tape to disk.

- Step 1: Copy Library SRVvrs.LIB from Tape to Disk
- Step 2: Copy the Procedure COPY.PROC from Tape to Disk
- Step 3: Copy all Product Files from Tape to Disk

Step 1: Copy Library SRVvrs.LIB from Tape to Disk

This step is not necessary if you have already copied the library SRV*vrs*.LIB from another Software AG installation tape. For further information, refer to the element #READ-ME in this library. The library SRV*vrs*.LIB is stored on the tape as a sequential file named SRV*vrs*.LIBS containing LMS commands. The current version *vrs* can be obtained from the *Software AG Product Delivery Report*.

- Execute the following commands to convert SRV*vrs*.LIBS into an LMS library:

```
/IMPORT-FILE  SUPPORT=*TAPE(FILE-NAME=SRVvrs.LIBS,-
/   VOLUME=volser, DEV-TYPE=tape-device)
/ADD-FILE-LINK  LINK-NAME=EDTSAM, FILE-NAME=SRVvrs.LIBS,-
/   SUPPORT=*TAPE(FILE-SEQ=3), ACC-METH=*BY-CAT,-
/   BUF-LEN=*BY-CAT, REC-FORM=*BY-CAT, REC-SIZE=*BY-CAT
/START-EDT
@READ  '/'
@SYSTEM 'REMOVE-FILE-LINK  EDTSAM'
@SYSTEM 'EXPORT-FILE  FILE-NAME=SRVvrs.LIBS'
@WRITE  'SRVvrs.LIBS'
@HALT
/ASS-SYSDTA  SRVvrs.LIBS
/MOD-JOB-SW  ON=1
/START-PROG  $LMS
/MOD-JOB-SW  OFF=1
/ASS-SYSDTA  *PRIMARY
```

where:

`tape-device` is the device type of the tape, for example, `TAPE-C4`, and
`volser` is the VOLSER of the tape (see the *Software AG Product Delivery Report*).

Step 2: Copy the Procedure COPY.PROC from Tape to Disk

- Call the procedure `P.COPYTAPE` in the library `SRVvrs.LIB` to copy the procedure `COPY.PROC` to disk:

  ```
  /CALL-PROCEDURE  (SRVvrs.LIB,P.COPYTAPE), -
  /   (VSNT=volser, DEVT=tape-device)
  ```

  If you use a TAPE-C4 device, you can omit the parameter `DEVT`.

Step 3: Copy all Product Files from Tape to Disk

- Enter the procedure `COPY.PROC` to copy all product files to disk:

  ```
  /ENTER-PROCEDURE  COPY.PROC, DEVT=tape-device
  ```

  If you use a TAPE-C4 device, you can omit the parameter `DEVT`.

  The result of this procedure is written to the file `L.REPORT.SRV`.

5 Event Replicator Installation Steps for z/OS Systems

Be sure your system meets the requirements described in ***System Requirements***, earlier in this chapter.

To install Event Replicator for Adabas on z/OS systems, complete the steps described in this chapter.

Step 1. Install the Replication Load Modules

Install the replication load modules by concatenating the delivered load library in sequence or copying the modules to the library used for running the Adabas nucleus and utilities. The Replication load modules can be found in ARF*vrs*.LOAD.

Step 2. Check, Prepare, and Install the Product License File

You must install a valid license file on all mainframe platforms in which your Software AG mainframe product is installed. The license file is provided as an XML document (encoding is US-ASCII) and must remain in that format, even on the mainframe. It must not be modified. Any modification of the license file will invalidate the digital signature and the license check will fail. In the event of a check failure, please contact your Software AG technical support representative.

For a full product list of license file names, load modules and DD/Link names, refer to the Adabas installation documentation.

Note: Thirty days before the license expires, license check failure messages are produced. Your software product will still function, but these messages warn you that it is time to obtain a new license.

In the following steps, you will prepare the license file and then install it:

- Preparing the Product License File
- Installing the Product License File

Preparing the Product License File

The product license file is supplied on the individual customer installation tape or separately via an e-mail attachment. Before you can install the license, you must transfer it from e-mail or the installation tape and store it on a z/OS system. This section describes how to do this for a license distributed either by e-mail or on the installation tape.

Note: ARF411 now requires a unique product license (ARFLIC) along with the ARF zIIP license (AZPRPLIC) if you have a zIIP product license. For distribution purposes, these will be known as ARF*vrs*.LICS and ARF*vrs*.LICZ. For more information, see the section ***Special Considerations for zIIP support***.

To prepare the license file from an e-mail attachment, complete the following steps:

1. Transfer the license to z/OS, as described in the Adabas documentation *Software AG Mainframe Product Licensing > Transferring a License File from PC to a z/OS Host Using FTP*.
2. Verify that the transferred license file is stored in an Adabas source library (with RECFM=F or FB and LRECL=80), taking care to preserve its format as ASCII.

To prepare the license file from the installation tape, complete the following step:

- Verify that the license file is stored from the tape into an Adabas source library (with RECFM=F or FB and LRECL=80), taking care to preserve its format as ASCII.

Installing the Product License File

Once the license file has been prepared, you can install it in one of two ways:

- You can convert the license to a load module (ARFLIC) that is then loaded by the Adabas Reptor startup job.
- You can reference the license file in the Adabas Reptor startup job by DD statement.

This section describes both methods.

To convert the license file to a load module, complete the following steps:

1. Review and modify sample job ASMLICAM in the ADA*vrs*.JOBS library, as follows:

   Set the variable MLCLOAD to point to the license load library (MLC*vrs*.LOAD).

   Set the variable USRLOAD to point to an appropriate user load library.

   Note: This user load library must also be included in the STEPLIB concatenation for Adabas Reptor servers.

   Set the LICFILE parameter to point to the dataset containing the Adabas Replication license file you transferred to z/OS earlier.

   Set the LMOD parameter to the load module name ARFLIC.
2. Submit sample job ASMLICAM. This job runs the MAKE function of the LICUTIL utility to convert the license text file to an assembler source module. ASMLICAM then links and assembles the assembler source to generate a load module called ARFLIC, which is stored in the specified user load library (USRLOAD). For more information about the LICUTIL utility,

refer to the Adabas documentation's *Software AG Mainframe Product Licensing > Using the License Utility: LICUTIL*.

3 Update your Adabas Reptor jobs to reference the user load library so ARFLIC will be loaded by the Adabas Reptor server at startup.

To reference the license file in Adabas Reptor jobs, complete the following steps:

1 Make sure any previously created ARFLIC load module is inaccessible to your Reptor jobs. The Reptor server first tries to load ARFLIC and, if unsuccessful, it reads from a dataset defined to the DD statement DDLARF.

2 Update your Adabas Reptor jobs to reference the license.

Step 3. Apply Necessary Zaps

Apply any necessary zaps, as described in the $README file on the ARF*vrs*.MVSZAPS library and in any subsequent ARF*vrs*.MVSZ*nnn* data sets (if they have been provided).

Step 4. (Optional) Create the ADAMQS Load Module for WebSphere MQ Support

If WebSphere MQ will be used, execute a job for linking ADAMQS. To create the required Event Replicator Server load module called ADAMQS, use JCL similar to this:

```
//LINK      EXEC PGM=IEWL,PARM='LIST,XREF,REUS,RENT'
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//ADALIB   DD DISP=SHR,DSN="ADABAS-Event-Replicator-Load-Library"
//MQSLIB   DD DISP=SHR,DSN="MQ-Series-SCSQLOAD-Library"
//SYSLMOD  DD DISP=SHR,DSN="User-Load-Library"
//SYSLIN   DD *
 INCLUDE ADALIB(ADAMQT)
 INCLUDE MQSLIB(CSQBSTUB)
 ENTRY ADAMQS
 NAME  ADAMQS(R)
/*
```

Note: If you are using IBM WebSphere MQ Series definitions for your DESTINATION or IQUEUE definitions, a S0D3 abend can occur if you run it as a started task and specify the parameter REUSASID=YES. This is a documented IBM WebSphere MQ Series issue.

Step 5. Create the Event Replicator Server

Create a database for use as your Event Replicator Server. Read the ADADEF documentation in *Event Replicator for Adabas Reference Guide* for the new parameters used when defining an Event Replicator Server.

If you want to perform transaction logging or use TLOG destination definitions, be sure to create and activate a CLOG (command log). For more information, refer to your *Adabas DBA Tasks* documentation.

Note: User application files should not be loaded on the Event Replicator Server.

Step 6. (Optional) Load a Replicator System File

To use Event Replicator for Adabas and customize its processing, you must supply various replication definitions. You can maintain these definitions in one of two ways:

- You can specify Event Replicator initialization parameters for the replication definitions in DDKARTE in the Event Replicator Server startup job. For more information about the initialization parameters you can specify in the Event Replicator Server startup job, read *Event Replicator Server Initialization Parameters* in *Event Replicator for Adabas Reference Guide*. This method of defining replication definitions does not require the use of a Replicator system file because the definitions are read from the initialization parameters in the startup job.
- You can maintain your replication definitions in the Replicator system file using the Adabas Event Replicator Subsystem. The Adabas Event Replicator Subsystem is an online interface that must be loaded into Natural before you can use it. Access to the Adabas Event Replicator Subsystem is then available through Natural or from Adabas Online System. For more information, read *Using the Adabas Event Replicator Subsystem* in *Adabas Event Replicator Subsystem User's Guide*.

If you determine that you want to maintain your replication definitions in the Replicator system file, you must load the file on the Event Replicator Server.

To load a Replicator system file on the Event Replicator Server:

1 Use the ADALOD utility. A sample ADALOD utility job is provided in member ADALODRP of the ARF*vrs*.JOBS data set. The only ADALOD utility parameter you should specify in this member is the REPLICATOR parameter. The others are not valid when loading the Replicator system file. See the ADALOD documentation in *Event Replicator for Adabas Reference Guide* for more information.

2 Do either of the following in Natural to identify the physical database file to be associated with the system file:

- Run a Natural NTLFILE macro for the file. For complete information, refer to your Natural LFILE parameter documentation, found in the Natural for Mainframe documentation on Software AG's **Empower** web site.
- When you start the Adabas Event Replicator Subsystem, be sure to set the Natural LFILE parameter as described in *Accessing the Adabas Event Replicator Subsystem*, in the *Adabas Event Replicator Subsystem User's Guide*. The LFILE parameter can be specified either as a dynamic parameter or inside a Natural SYSPARM profile.

Step 7. (Optional) Load an SLOG System File

If you intend to use the subscription logging (SLOG) facility, you will need an `SLOG system file`, which is an Adabas system file on the Event Replicator Server. To set this up, please read *Setting Up Subscription Logging*, in the *Event Replicator for Adabas Administration and Operations Guide*. For complete information about the SLOG facility, read *Using the Subscription Logging Facility*, in the *Event Replicator for Adabas Administration and Operations Guide*.

Step 8. (Optional) Load the Adabas Event Replicator Subsystem Application Into Natural

If you intend to use the Adabas Event Replicator Subsystem to maintain replication definitions in the Replicator system file (see Step 4), it must be loaded into Natural. To do this, use Natural's INPL utility to load the ARF`vrs`.INPL data set into the Natural system file libraries.

Step 9. Set Up the Event Replicator Server Startup JCL

Set up the startup JCL and parameters for the *Event Replicator Server* job. Use the JCL for an existing Adabas nucleus as a starting point for creating the Event Replicator Server JCL. For complete information on the ADARUN parameters pertinent to the Event Replicator Server, read *Pertinent ADARUN Parameters* in *Event Replicator for Adabas Reference Guide*.

1. Specify an initial setting for the ADARUN parameter `RPLPARMS` to "NONE". The `RPLPARMS` parameter identifies where the replication definitions (initialization parameters) should be read from. A setting of "NONE" allows you to run the Event Replicator Server when you have no definitions set up, providing you an opportunity to specify them.
2. Set ADARUN parameter `LRPL` to the size of the Event Replicator Server replication buffer. For performance reasons, Software AG recommends setting LRPL to a relatively large value (e.g. LRPL=40M).

3. Set ADARUN parameter `LU` to 167,000 or greater.
4. Set ADARUN parameter `NAB` to a value greater than or equal to:

   ```
   41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server
   ```

   For example, if one Adabas nucleus will be sending data to the Event Replicator Server, set the NAB parameter greater than or equal to 410 (for example NAB=420).
5. Set ADARUN parameter `NT` to a value greater than or equal to 15.

Step 10. Start the Event Replicator Server Job

Run the Event Replicator Server job you set up in the previous step.

Note: If you are using IBM WebSphere MQ Series definitions for your DESTINATION or IQUEUE definitions, a S0D3 abend can occur if you run it as a started task and specify the parameter REUSASID=YES. This is a documented IBM WebSphere MQ Series issue.

Step 11. Implement Replication

The previous steps in this section allowed you to install Event Replicator for Adabas, but they did not implement or start replication. How replication is implemented varies by site, but general implementation steps are described in ***Post-Installation Replication Implementation Steps (All Platforms)***, elsewhere in this guide.

Special Considerations for zIIP Support

This section contains the following topics:

- Prerequisites
- Libraries
- License

- Adabas Online System

Prerequisites

Prerequisite for the zIIP support in the Event Replicator for Adabas is a z13 or z14 mainframe with one or more zIIP engines, running z/OS 2.1 or above. The Event Replicator Server needs one zIIP engine to perform effectively.

For zIIP support, the Event Replicator Server must be based on Adabas Version 8.5.

For online administration of the zIIP-related functions and statistics of the Event Replicator Server, the version of the Adabas Online System provided with Adabas Version 8.5 must be installed.

If one or more destinations defined in the Event Replicator Server use EntireX Broker as the messaging system (DTYPE=ETBROKER), EntireX Broker version 10.1 patch level 11 or higher must be used.

Libraries

For the zIIP license check in Event Replicator for Adabas (see **License** below), use the Mainframe License Check version 1.3.3 from the following library:

- MLC136.LOAD

License

An Event Replicator Server that is to run with zIIP support requires an associated license file (AZPRP) in addition to the Adabas license file. If the AZPRP license is not provided or erroneous, the Event Replicator Server will run, but without zIIP support (`ZIIP=NO`).

The AZPRP license can be transferred to the mainframe in the same way as the Adabas license. The license can be made available to the Event Replicator Server as a load module with the name ‘AZPRPLIC’. This is similar to the handling of the Adabas license load module ADALIC described in the Adabas documentation under *Installing Adabas for z/OS*.

Alternatively or in addition, the license file can be referred to by a ‘DDLAZPRP’ DD statement in the nucleus job/started task. This is a fallback for the case that the AZPRPLIC module cannot be loaded.

Adabas Online System

If the ADA85 INPL dataset for Adabas 8.5 is installed, it will be possible to use Adabas Online System (AOS) to review the zIIP statistics of the Event Replicator Server and to view and modify the `ZIIP` parameter, even if you do not have an AOS product license.

6 Event Replicator Installation Steps for z/VSE Systems

Be sure your system meets the requirements described in ***System Requirements***, earlier in this chapter.

To install Event Replicator for Adabas on z/VSE systems, complete the steps described in this chapter.

Step 1. Set Up the Event Replicator Server Library and Data Set Definition Procedure

Set up a VSE procedure for the Event Replicator Server by tailoring the ARFPROC.X JCL supplied in the ARF*vrs* library member.

This procedure defines the libraries and files to be used for the Event Replicator Server.

Step 2. Create the Event Replicator Server

Tailor and run the ADAFRMR.X and ADADEFR1.X JCL members in the ARF*vrs* library to define the Event Replicator Server nucleus. For more information about the new ADADEF parameters used when defining an Event Replicator Server, read the ADADEF documentation in *Event Replicator for Adabas Reference Guide*.

If you want to perform transaction logging or use TLOG destination definitions, be sure to create and activate a CLOG (command log).

Note: User application files should not be loaded on the Event Replicator Server.

For more information about transaction logging, read *Transaction Logging* in *Event Replicator for Adabas Concepts*. For more information about TLOG destination definitions, contact your Software AG customer support representative. Finally, for more information about creating and activating a CLOG, refer to your *Adabas DBA Tasks* documentation.

Step 3. (Optional) Load a Replicator System File

To use Event Replicator for Adabas and customize its processing, you must supply various replication definitions. You can maintain these definitions in one of two ways:

- You can specify Event Replicator initialization parameters for the replication definitions in SYSIPT in the Event Replicator Server startup job. This is placed behind the ADARUN statements as shown in the sample JCL found in the ADANUCA1.X member of the ARF*vrs* library. For more information about the initialization parameters you can specify, read *Event Replicator Server*

Initialization Parameters in *Event Replicator for Adabas Reference Guide*. This method of defining replication definitions does not require the use of a Replicator system file because the definitions are read from the initialization parameters in the startup job.

- You can maintain your replication definitions in the Replicator system file using the Adabas Event Replicator Subsystem. The Adabas Event Replicator Subsystem is an online interface that must be loaded into Natural before you can use it. Access to the Adabas Event Replicator Subsystem is then available through Natural or from Adabas Online System. For more information, read *Using the Adabas Event Replicator Subsystem* in *Adabas Event Replicator Subsystem User's Guide*.

If you determine that you want to maintain your replication definitions in the Replicator system file, you must load the file on the Event Replicator Server.

To load a Replicator system file on the Event Replicator Server:

1 Use the ADALOD utility. A sample ADALOD utility job is provided in member ADALODRP.X of the ARF*vrs* library. The only ADALOD utility parameter you should specify in this member is the `REPLICATOR` parameter. The others are not valid when loading the Replicator system file. See the ADALOD documentation in *Event Replicator for Adabas Reference Guide* for more information.

2 Do either of the following in Natural to identify the physical database file to be associated with the system file:

   - Run a Natural NTLFILE macro for the file. For complete information, refer to your Natural LFILE parameter documentation, found in the Natural for Mainframe documentation on Software AG's **Empower** web site.
   - When you start the Adabas Event Replicator Subsystem, be sure to set the Natural LFILE parameter as described in *Accessing the Adabas Event Replicator Subsystem*, in the *Adabas Event Replicator Subsystem User's Guide*. The LFILE parameter can be specified either as a dynamic parameter or inside a Natural SYSPARM profile.

Step 4. (Optional) Load an SLOG System File

If you intend to use the subscription logging (SLOG) facility, you will need an *`SLOG system file`*, which is an Adabas system file on the Event Replicator Server. To set this up, please read *Setting Up Subscription Logging*, in the *Event Replicator for Adabas Administration and Operations Guide*. For complete information about the SLOG facility, read *Using the Subscription Logging Facility*, in the *Event Replicator for Adabas Administration and Operations Guide*.

Step 5. (Optional) Load the Adabas Event Replicator Subsystem Application Into Natural

If you intend to use the Adabas Event Replicator Subsystem to maintain replication definitions in the Replicator system file (see Step 4), it must be loaded into Natural. To do this, use Natural's INPL utility to load the ARF*vrs*.INPL data set into the Natural system file libraries.

If you want to use the Adabas Event Replicator Subsystem, access it as described in *Accessing the Adabas Event Replicator Subsystem*, in *Adabas Event Replicator Subsystem User's Guide*. Be sure also to identify the Replicator system file you want to update as soon as you access the Adabas Event Replicator Subsystem. For information on identifying, loading, and unloading the Replicator system file using the Adabas Event Replicator Subsystem, read *Identifying, Loading, and Unloading the Replicator System File*, in *Adabas Event Replicator Subsystem User's Guide*.

Step 6. Set Up the Event Replicator Server Startup JCL

Set up the startup JCL and parameters for the *Event Replicator Server* job. Use the JCL for an existing Adabas nucleus as a starting point for creating the Event Replicator Server JCL. For complete information on the ADARUN parameters pertinent to the Event Replicator Server, read *Pertinent ADARUN Parameters* in *Event Replicator for Adabas Reference Guide*.

1. Specify an initial setting for the ADARUN parameter `RPLPARMS` to "NONE". The `RPLPARMS` parameter identifies where the replication definitions (initialization parameters) should be read from. A setting of "NONE" allows you to run the Event Replicator Server when you have no definitions set up, providing you an opportunity to specify them.
2. Set ADARUN parameter `LRPL` to the size of the Event Replicator Server replication buffer. For performance reasons, Software AG recommends setting LRPL to a relatively large value (e.g. LRPL=40M).
3. Set ADARUN parameter `LU` to 167,000 or greater.
4. Set ADARUN parameter `NAB` to a value greater than or equal to:

   ```
   41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server
   ```

   For example, if one Adabas nucleus will be sending data to the Event Replicator Server, set the NAB parameter greater than or equal to 410 (for example NAB=420).
5. Set ADARUN parameter `NT` to a value greater than or equal to 15.

Step 7. Start the Event Replicator Server Job

Run the Event Replicator Server job you set up in the previous step.

Step 8. Implement Replication

The previous steps in this section allowed you to install Event Replicator for Adabas, but they did not implement or start replication. How replication is implemented varies by site, but general implementation steps are described in ***Post-Installation Replication Implementation Steps (All Platforms)***, elsewhere in this guide.

7 Event Replicator Installation Steps for BS2000 Systems

Be sure your system meets the requirements described in ***System Requirements***, earlier in this chapter.

To install Event Replicator for Adabas on BS2000 systems, complete the steps described in this chapter.

Step 1. Install the Replication Load Modules

Install the replication load modules by concatenating the delivered load library in sequence or copying the modules to the library used for running the Adabas nucleus and utilities. The Replication load modules can be found in ARF*vrs*.MOD.

Step 2. Create the Event Replicator Server

Create a database for use in the Event Replicator Server startup job. See the ADADEF documentation in *Event Replicator for Adabas Reference Guide* for the new parameters used when defining an Event Replicator Server.

If you want to perform transaction logging or use TLOG destination definitions, be sure to create and activate a CLOG (command log). For more information, refer to your *Adabas DBA Tasks* documentation.

Note: User application files should not be loaded on the Event Replicator Server.

Step 3. (Optional) Load a Replicator System File

To use Event Replicator for Adabas and customize its processing, you must supply various replication definitions. You can maintain these definitions in one of two ways:

- You can specify Event Replicator initialization parameters for the replication definitions in DDKARTE in the Event Replicator Server startup job. For more information about the initialization parameters you can specify, read *Event Replicator Server Initialization Parameters* in *Event Replicator for Adabas Reference Guide*.
- You can maintain your replication definitions in the Replicator system file using the Adabas Event Replicator Subsystem. The Adabas Event Replicator Subsystem is an online interface that must be loaded into Natural before you can use it. Access to the Adabas Event Replicator Subsystem is then available through Natural or from Adabas Online System. For more information, read *Using the Adabas Event Replicator Subsystem* in *Adabas Event Replicator Subsystem User's Guide*.

If you determine that you want to maintain your replication definitions in the Replicator system file, you must load the file on the Event Replicator Server.

To load a Replicator system file on the Event Replicator Server:

1. Load a Replicator system file on the Event Replicator Server (defined in the previous step) using the ADALOD utility. A sample ADALOD utility job is provided in member ADALODRP of the ARF`vrs`.SRC library. The only ADALOD utility parameter you should specify in this member is the REPLICATOR parameter. The others are not valid when loading the Replicator system file. See the ADALOD documentation in *Event Replicator for Adabas Reference Guide* for more information.
2. Do either of the following in Natural to identify the physical database file to be associated with the system file:

   - Run a Natural NTLFILE macro for the file. For complete information, refer to your Natural LFILE parameter documentation, found in the Natural for Mainframe documentation on Software AG's **Empower** web site.
   - When you start the Adabas Event Replicator Subsystem, be sure to set the Natural LFILE parameter as described in *Accessing the Adabas Event Replicator Subsystem*, in the *Adabas Event Replicator Subsystem User's Guide*. The LFILE parameter can be specified either as a dynamic parameter or inside a Natural SYSPARM profile.

Step 4. (Optional) Load an SLOG System File

If you intend to use the subscription logging (SLOG) facility, you will need an `SLOG system file`, which is an Adabas system file on the Event Replicator Server. To set this up, please read *Setting Up Subscription Logging*, in the *Event Replicator for Adabas Administration and Operations Guide*. For complete information about the SLOG facility, read *Using the Subscription Logging Facility*, in the *Event Replicator for Adabas Administration and Operations Guide*.

Step 5. (Optional) Load the Adabas Event Replicator Subsystem Application Into Natural

If you intend to use the Adabas Event Replicator Subsystem to maintain replication definitions in the Replicator system file (see Step 4), it must be loaded into Natural. To do this, use Natural's INPL utility to load the ARF`vrs`.INPL data set (file) into the Natural system file libraries.

If you want to use the Adabas Event Replicator Subsystem, access it as described in *Accessing the Adabas Event Replicator Subsystem*, in *Adabas Event Replicator Subsystem User's Guide*. Be sure also to identify the Replicator system file you want to update as soon as you access the Adabas Event Replicator Subsystem. For information on identifying, loading, and unloading the Replicator system file using the Adabas Event Replicator Subsystem, read *Identifying, Loading, and Unloading the Replicator System File*, in *Adabas Event Replicator Subsystem User's Guide*.

Step 6. Set Up the Event Replicator Server Startup JCL

Set up the startup JCL and parameters for the *Event Replicator Server* job. Use the JCL for an existing Adabas nucleus as a starting point for creating the Event Replicator Server JCL. For complete information on the ADARUN parameters pertinent to the Event Replicator Server, read *Pertinent ADARUN Parameters* in *Event Replicator for Adabas Reference Guide*.

1. Specify an initial setting for the ADARUN parameter `RPLPARMS` to "NONE". The `RPLPARMS` parameter identifies where the replication definitions (initialization parameters) should be read from. A setting of "NONE" allows you to run the Event Replicator Server when you have no definitions set up, providing you an opportunity to specify them.
2. Set ADARUN parameter `LRPL` to the size of the Event Replicator Server replication buffer. For performance reasons, Software AG recommends setting LRPL to a relatively large value (e.g. LRPL=40M).
3. In Adabas, set the `SUBMPSZ` (say 50,000,000) parameter large enough to accommodate the `LRPL` parameter above.
4. Set ADARUN parameter `LU` to 167,000 or greater.

   If you are using webMethods Broker and NET communication, make sure that the DEFAULTS=NET section of the file identified by the ETBFILE link parameter file sets the IUBL parameter to the same size as the LU ADARUN parameter setting.
5. Set ADARUN parameter `NAB` to a value greater than or equal to:

   ```
   41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server
   ```

   For example, if one Adabas nucleus will be sending data to the Event Replicator Server, set the NAB parameter greater than or equal to 410 (for example NAB=420).
6. Set ADARUN parameter `NT` to a value greater than or equal to 15.
7. Set ADARUN parameter `CMDQMODE` to "ABOVE". Large `NC` and `NAB` values are also required here in the Entire Net-Work and Event Replicator Server. These need to be placed above the 16MB area where there is more space.

Notes:

1. If you are using webMethods EntireX NET communication, make sure that the parameters in the DEFAULTS=NET section of the file identified by the ETBFILE link name include the following settings: EXTENDED-ACB-SUPPORT=YES and NABS=1000 (or more).

2. If you are using webMethods EntireX TCP communication, make sure that the parameters in the DEFAULTS=NET section of the file identified by the ETBFILE link name include the following setting: TRANSPORT=TCP or TRANSPORT=TCP-NET.

Step 7. Set up the ADALNK Communication File

If the default router IDTNAME ADABAS5B is not being used, assign a permanent file to the DDLNKPAR link name as follows:

```
/SET-FILE-LINK  DDLNKPAR,filename
```

The file identified by the DDLNKPAR link name should contain an ADALNK IDTNAME parameter setting (where *xxxxx* is the desired suffix):

```
ADALNK IDTNAME=ADAxxxxx
```

Step 8. (Optional) Assign Permanent Files to BS2000 Link Names

Where there is no access to the BS2000 console, the Adabas nucleus and Event Replicator Server jobs should assign permanent files to the following link names:

```
/SET-FILE-LINK  DDPRINT,ddprint_permanent_filename
```

```
/SET-FILE-LINK  DDDRUCK,dddruck_permanent_filename
```

These files can be accessed when the jobs are running by specifying:

```
SHOW-FILE ddprint/dddruck_permanent_filename
```

This will not show the actual file data until the next file block is filled and flushed.

Step 9. Start the Event Replicator Server Job

Run the Event Replicator Server job you set up in Step 5.

Step 10. Set up Subtasking with EntireX Broker Access (Optional)

If an output queue with ETBROKERNAME=BROKER or an input queue with IQTYPE=ETBBROKER has been defined (read *Event Replicator Initialization Parameters*, in the *Event Replicator for Adabas Reference Guide*), then the BS2000 Event Replicator Server will run subtasks. In BS2000 environment, these are separate jobs that communicate via common memory.

The number of output subtasks is determined by the SUBTASKS parameter setting. The number of input subtasks is determined by the number of IQUEUE definitions.

If all is correctly defined, this will show:

```
ADAF90 00073 001 output subtasks to be attached
 ADAI5X 00073   Subtask ADABSP   started with TSN 7245
 ADAF93 00073 001 input  subtasks to be attached
 ADAI5X 00073   Subtask ADABSP   started with TSN 7246
 ADAF92 00073 Output subtasks have been initialized
 ADAF92 00073 Input  subtasks have been initialized
 ↩
```

Here, 73 is the node number of the Event Replicator Server, 7245 is the job number of the output subtask, and 7246 is that of the input subtask. The job name is always ADABSP (this can be seen by the SHOW-USER-STATUS command).

If there is a problem with the definitions or connections, the following may occur:

```
ADAF90 00073 001 output subtasks to be attached
 ADAI5X 00073   Subtask ADABSP   started with TSN 7245
 ADAF93 00073 001 input  subtasks to be attached
 ADAI5X 00073   Subtask ADABSP   started with TSN 7246
 ADAI68 ADAIO2 DISABLING SUBROUTINE  7246 ADABSP
 ADAF95 00073 All input subtasks could not be initialized
 ↩
```

If this happens, review the output protocol of the subtask. This is a sequential file with the naming convention:

```
L.O.<replicator-nucleus-task-number>.ADABSP##.<timestamp>
```

Within this file, a message in the following format may be seen:

```
ADAF87 00073 AMI R=CONNECT  Q=ETB0901: RC=008/060/F0F2F1F5/F0F1F4F8
```

In this example, the communication type is NET to a Broker with a Broker ID of "901". The last two fields can be interpreted as 0215/0148. These are the EntireX Broker error message code number and message number (for more information, refer to the *EntireX Broker Messages and Codes*).

Note: When an input subtask failure occurs, attempts are made to reconnect as many times as defined by IQRETRYCOUNT and intervals of IQRETRYINTERVAL.

The subtask jobs are automatically terminated when the Event Replicator Server terminates.

Step 11. Verify Zaps Have Been Applied

The ARF*vrs*.ZAPS library contains the element ZAPV*vrs*(J). This contains a list of Adabas released zaps that need to be applied for the Event Replicator to work. Please ensure that these zaps have been applied before starting the Event Replicator.

Step 12. Implement Replication

The previous steps in this section allowed you to install Event Replicator for Adabas, but they did not implement or start replication. How replication is implemented varies by site, but general implementation steps are described in ***Post-Installation Replication Implementation Steps (All Platforms)***, elsewhere in this guide.

8 Post-Installation Replication Implementation Steps (All Platforms)

Once the Event Replicator for Adabas has been installed, replication can be implemented and started. How replication is implemented varies greatly by site and situation, but this chapter describes some general implementation steps.

Step 1. Supply Event Replicator Definitions

Replication definitions are used to customize the replication process. These definitions are specified in initialization parameters read from DDKARTE in the Event Replicator Server startup job or in the Replicator system file.

- If you want to use Event Replicator initialization parameters to specify the replication definitions that will be read from DDKARTE in the Event Replicator Server startup job, read *Event Replicator Server Initialization Parameters* in *Event Replicator for Adabas Reference Guide*.
- If you want to use the Adabas Event Replicator Subsystem to maintain the replication definitions in the Replicator system file, access it as described in *Accessing the Adabas Event Replicator Subsystem*, in *Adabas Event Replicator Subsystem User's Guide*. Be sure also to identify the Replicator system file you want to update as soon as you access the Adabas Event Replicator Subsystem. For information on identifying the Replicator system file using the Adabas Event Replicator Subsystem, read *Identifying, Loading, and Unloading the Replicator System File*, in *Adabas Event Replicator Subsystem User's Guide*. For information on maintaining definitions using the Adabas Event Replicator Subsystem, read *Using the Adabas Event Replicator Subsystem* in *Adabas Event Replicator Subsystem User's Guide*

Once the Event Replicator for Adabas in installed, its replication processing is driven by definitions you specify. These definitions are described in the following table in order of importance to replication (required definitions are listed first).

Note: You can run Event Replicator for Adabas in verify (test) mode, by turning on verification in the VERIFYMODE replication definition. This is useful if you want to test the definitions you have specified before you start using Event Replicator for Adabas in production mode. For more information, read *Running in Verify Mode*, in *Event Replicator for Adabas Installation Guide*.

Definition Type	Defines	How many definitions are required?
destination	The destination of the replicated data. Destination definitions can be created for Adabas, File, webMethods EntireX, WebSphere MQ, and Null destinations. To maintain destination definitions using DDKARTE statements of the Event Replicator Server startup job, read *Destination Parameter* in *Event Replicator for Adabas Reference Guide*. To maintain destination definitions using the Adabas Event	Required. At least one destination definition is required for data replication to occur. Create one definition for every Event Replicator for Adabas destination you intend to use.

Definition Type	Defines	How many definitions are required?
	Replicator Subsystem, read *Maintaining Destination Definitions* in *Adabas Event Replicator Subsystem User’s Guide*.	
subscription	A set of specifications to be applied to the replication of the data. These include (but are not limited to): ■ the identification of the Adabas files that should be replicated and how they should be replicated (SFILE definitions that should be processed as part of the subscription) ■ architecture key, output alpha and wide-character keys that should be used ■ the name of the resend buffer definition that should be used for replication, if any ■ various settings relating to the availability of the subscription in specific circumstances Subscription definitions identify SFILE definitions and resend buffer definitions that should be used. At least one SFILE definition is required. To maintain subscription definitions using DDKARTE statements of the Event Replicator Server startup job, read *SUBSCRIPTION Parameter* in *Event Replicator for Adabas Reference Guide*. To maintain subscription definitions using the Adabas Event Replicator Subsystem, read *Maintaining Subscription Definitions* in *Adabas Event Replicator Subsystem User’s Guide*.	Required. At least one subscription definition is required for data replication to occur.
SFILE	An Adabas file to be replicated and the replication processing that should occur for that file. SFILE definitions are sometimes referred to as *subscription file definitions* and are referenced by subscription definitions. An SFILE definition identifies (among other things): ■ the Adabas database ID and file number that should be replicated ■ the transaction filter definitions that should be used to filter the data in the Adabas file during replication (if any) ■ the subscription user exit that should be processed during replication (if any) ■ whether insert, delete, and update transactions should be replicated ■ the file’s alpha character encoding, if any ■ the GFB definitions that should be used for replication, if any, or the specific format buffer definitions that should be used instead.	Required. At least one SFILE definition is required for data replication to occur.

Definition Type	Defines	How many definitions are required?
	To maintain SFILE definitions using DDKARTE statements of the Event Replicator Server startup job, read *SUBSCRIPTION Parameter* in *Event Replicator for Adabas Reference Guide*. To maintain SFILE definitions using the Adabas Event Replicator Subsystem, read *Maintaining SFILE Definitions* in *Adabas Event Replicator Subsystem User's Guide*.	
initial-state	An initial-state request for data from the target application. Initial-state definitions identify the subscription, destination, and specific Adabas files to use in an Event Replicator for Adabas initial-state run. To maintain initial-state definitions using DDKARTE statements of the Event Replicator Server startup job, read *INITIALSTATE Parameter* in *Event Replicator for Adabas Reference Guide*. To maintain initial-state definitions using the Adabas Event Replicator Subsystem, read *Maintaining Initial-State Definitions* in *Adabas Event Replicator Subsystem User's Guide*.	Not required. If you want initial-state data produced in an Event Replicator for Adabas run, only one initial-state definition is required. Otherwise, no initial-state data definition is required.
IQUEUE	The input queue on which Event Replicator for Adabas should listen for requests from webMethods EntireX and WebSphere MQ targets. To maintain IQUEUE definitions using DDKARTE statements of the Event Replicator Server startup job, read *IQUEUE Parameter* in *Event Replicator for Adabas Reference Guide*. To maintain IQUEUE definitions using the Adabas Event Replicator Subsystem, read *Maintaining Input Queue (IQUEUE) Definitions* in *Adabas Event Replicator Subsystem User's Guide*.	Not required. At least one IQUEUE definition is required for every EntireX Communicator or WebSphere MQ target you intend to use. If webMethods EntireX or WebSphere MQ are not used, no IQUEUE definition is required.
GFB	A global format buffer (GFB) definition stored separately for use in SFILE definitions. You can specify GFBs manually or generate them using Predict file definitions. When you generate them, a field table is also generated. While a format buffer specification is required in a subscription's SFILE definition, a stored GFB definition does not need to be used. The SFILE definition could simply include the format buffer specifications it needs. To maintain GFB definitions using DDKARTE statements of the Event Replicator Server startup job, read *GFORMAT Parameter* in *Event Replicator for Adabas Reference Guide*. To maintain GFB definitions using the Adabas Event Replicator Subsystem, read *Maintaining GFB Definitions* in *Adabas Event Replicator Subsystem User's Guide*.	Not required. No GFB definition is required. If a global format buffer is needed, at least one GFB definition is required.

Definition Type	Defines	How many definitions are required?
resend buffer	A resend buffer that can be used by any subscription to expedite the retransmission of a transaction. To maintain resend buffer definitions using DDKARTE statements of the Event Replicator Server startup job, read *RESENDBUFFER Parameter* in *Event Replicator for Adabas Reference Guide*. To maintain resend buffer definitions using the Adabas Event Replicator Subsystem, read *Maintaining Resend Buffer Definitions* in *Adabas Event Replicator Subsystem User's Guide*.	Not required. No resend buffer definition is required. If you elect to retransmit a transaction, at least one resend buffer definition is required.
transaction filter	A filter definition that can be used to filter the records used for replication based on the values of fields in those records. To maintain transaction filter definitions using DDKARTE statements of the Event Replicator Server startup job, read *FILTER Parameter* in *Event Replicator for Adabas Reference Guide*. To maintain transaction filter definitions using the Adabas Event Replicator Subsystem, read *Maintaining Transaction Filter Definitions* in *Adabas Event Replicator Subsystem User's Guide*.	Not required. No transaction filter definition is required. If you want to use a transaction filter to filter records used in replication, at least one transaction filter definition is required.

The applicable definitions and the sequence in which they should be set up varies, depending on the destination. Five destinations are supported in the Event Replicator for Adabas: Adabas, webMethods EntireX, IBM WebSphere MQ, File, and Null. The following table describes each of these destination types and lists the definitions that apply to the destination in the order in which they should be defined.

Destination Type	Description	Definition List and Order of Creation
Adabas	Data is replicated to one or more Adabas files.	1. destination definitions, as necessary (referenced by subscription definitions) 2. global format buffer definitions, if needed (can be referenced by the SFILE definitions) 3. subscription definition, including at least one SFILE definition 4. one or more SFILE definitions (included in the subscription definition) 5. initial-state definition
webMethods EntireX	Replicated data is written to an output queue via webMethods EntireX.	1. destination definitions, as necessary (referenced by subscription definitions) 2. IQUEUE definition 3. global format buffer definitions, if needed (can be referenced by the SFILE definitions) 4. subscription definition, including at least one SFILE definition

Destination Type	Description	Definition List and Order of Creation
		5. one or more SFILE definitions (included in the subscription definition) 6. initial-state definition
File	Replicated data is written to the CLOG, using TLOG URBLTDOD records.	1. destination definitions, as necessary (referenced by subscription definitions) 2. global format buffer definitions, if needed (can be referenced by the SFILE definitions) 3. subscription definition, including at least one SFILE definition 4. one or more SFILE definitions (included in the subscription definition) 5. initial-state definition
WebSphere MQ	Replicated data is written to an output queue via IBM WebSphere MQ.	1. destination definitions, as necessary (referenced by subscription definitions) 2. IQUEUE definition 3. global format buffer definitions, if needed (can be referenced by the SFILE definitions) 4. subscription definition, including at least one SFILE definition 5. one or more SFILE definitions (included in the subscription definition) 6. initial-state definition
Null	Data replication is tested without actually sending the data to a destination.	1. destination definitions, as necessary (referenced by subscription definitions) 2. global format buffer definitions, if needed (can be referenced by the SFILE definitions) 3. subscription definition, including at least one SFILE definition 4. one or more SFILE definitions (included in the subscription definition) 5. initial-state definition

Step 2. Customize the Event Replicator Server Startup JCL

Once the replication definitions have been defined for your Event Replicator Server, customize the startup JCL to identify where the replication definitions should be found for the Event Replicator Server run.

1. Change the setting for the ADARUN parameter `RPLPARMS` to "BOTH", "FILE", or "PARMS". The `RPLPARMS` parameter identifies where the replication definitions (initialization parameters) should be read from:

   - If you have specified and intend to maintain your replication definitions solely in DDKARTE statements of the Event Replicator Server startup job, specify the value of `RPLPARMS` as "PARMS".
   - If you have specified and intend to maintain your replication definitions solely in a Replicator system file associated with this Event Replicator Server, specify the value of `RPLPARMS` as "FILE".
   - If you have specified and intend to maintain your replication definitions in both DDKARTE statements of the Event Replicator Server startup job and a Replicator system file associated with the Event Replicator Server, specify the value of `RPLPARMS` as "BOTH".

     Note: In this case, the DDKARTE statements in the Event Replicator Server startup job are read after the definitions in the Replicator system file. Therefore the DDKARTE initialization parameter settings will override any replication definitions in the Replicator system file.

   Read *Adabas Initialization (ADARUN) Parameters* in *Event Replicator for Adabas Reference Guide* for further details.

2. If one or more WebSphere MQ queues will be used by the Event Replicator Server, ensure all load libraries in the Event Replicator Server STEPLIB or JOBLIB concatenation are APF-authorized.

3. If tracing is enabled for a destination class or user exit (using the TRACE keyword parameter of the `DCLASSPARM` parameter), be sure to include the following JCL statement in the startup JCL of the Event Replicator Server:

   ```
   //DDTRACE1 DD   SYSOUT=X
   ```

   For more information about the `DCLASSPARM` parameter, which is valid only for webMethods EntireX or WebSphere MQ destinations, read `DCLASSPARM`, in *Adabas Event Replicator Subsystem User's Guide*.

4. If you choose to automate Replay Utility (ADARPL) processing by the Event Replicator Server for specific subscriptions, you need to add the DDJCLIN and DDJCLOUT JCL statements in the Event Replicator Server startup JCL. For complete information on the Event Replicator

Server startup JCL updates required for automated ADARPL processing, read *Automating Replay Processing*, in the *Event Replicator for Adabas Administration and Operations Guide*.

Step 3. (Optional) Modify the Sample Subscription User Exit

If appropriate, modify the sample subscription user exit located in the ARF*vrs*.SRCE library and assemble. Ensure the subscription user exits are in the Event Replicator Server job STEPLIB concatenation. See ARF*vrs*.JOBS (z/OS), ARF*vrs*.LIBR (z/VSE), or ARF*vrs*.SRC (BS2000) for sample job ASMUSERX that can be used to assemble the subscription user exit. Refer to the section entitled *Using the Event Replicator Server Subscription User Exit* (in the *Event Replicator for Adabas Administration and Operations Guide*) for detailed information on using the exit.

Once the user exit is modified and assembled, it can be called during replication runs. For each combination of subscription and file, the user exit to be called can be specified using the SFSEXIT initialization parameter (file-related parameter of the subscription) in the Event Replicator Server startup job.

Step 4. Restart the Event Replicator Server

Stop and restart the Event Replicator Server to pick up the replication definitions and any subscription user exit specifications you have added.

Note: If you use a Replicator system file to store your replication definitions, you can use the RPLREFRESH command to refresh resource definitions in your Event Replicator Server configuration while the Event Replicator Server is running. For more information, read *RPLREFRESH Command*, in *Event Replicator for Adabas Administration and Operations Guide*.

Step 5. Modify the JCL for the Adabas Database and Restart It

Add the ADARUN parameter `REPLICATION=YES` to the Adabas nucleus job containing the files available for replication. Add the `LRPL` parameter for setting the size of the Adabas nucleus replication pool. The replication pool in the Adabas nucleus does not necessarily need to be the same size as the replication pool in the Event Replicator Server. For performance, Software AG recommends setting `LRPL` to a relatively large value (e.g. LRPL=30M). For more information about these parameters, read *Pertinent ADARUN Parameters* in *Event Replicator for Adabas Reference Guide*.

For BS2000 installations, if the database is an Adabas version 8 database, set the ADARUN `SUBMPSZ` parameter large enough to accommodate the `LRPL` parameter above. Be sure also to add the ARF*vrs*.MOD library to the BLSLIB or to the Adabas nucleus to be replicated.

Ensure the LU ADARUN parameter is set to a value greater than or equal to 70000.

When all modifications have been made, restart the Adabas database.

Step 6. Start Replication Processing

Start replication processing in any of the following ways, as needed for your organization:

- If you need to populate the target database with an initial copy of the data in your Adabas database prior to starting replication, run an Event Replicator initial-state request. For more information about initial-state data, read *Replicating an Initial Version of All Your Data* in *Event Replicator for Adabas Concepts*.
- If you just want to start replicating changed data in the Adabas database, turn replication on for various Adabas files. You can do this in an ADADBS utility run or using Adabas Online System, if it is installed. For more information, read *ADADBS REPLICATION Function* in *Event Replicator for Adabas Reference Guide* or *Controlling Replication of an Adabas Database File* in *Event Replicator for Adabas Administration and Operations Guide*.

9 Running in Verify Mode

You can test the replication definitions used to customize the replication process by running Event Replicator for Adabas in verify (test) mode. To do this, you must turn on verify mode using one of the following settings:

- Use the `VERIFYMODE` parameter in the DDKARTE statements of the Event Replicator Server startup job.
- Use the **Verify Mode** global value setting in the Replicator system file. This value is set using the Adabas Event Replicator Subsystem.

When verify mode is turned on:

- All of the non-null destinations specified in the Replicator system file or as DDKARTE statements in the Event Replicator Server startup job are first changed to null destinations to ensure that data is not actually written to the destinations. A message is sent to the job log for each changed destination.
- Event Replicator for Adabas runs, testing all of its definitions as it does so, without actually writing any data to the destinations.
- When errors occur during subscription decompression, a related message is sent to the job log for the first occurrence of the error for a given subscription-SFILE-image definition combination. (Decompression failures are always noted in the DRPLPARM output, regardless of the setting of the `VERIFYMODE` parameter.)
- If no errors occur during subscription decompression, a related message is sent to the job log for the first successful decompression of a given subscription-SFILE-image definition combination. In addition, successful decompression attempts are noted in the DRPLPARM output when `VERIFYMODE=YES`.

10 Security

These are the steps to perform to implement Trusted User ID in Event Replicator for Adabas.

1. Set `SECURITY=YES` in the `DEFAULTS=BROKER` section of the Broker Attribute File.
2. Set `TRUSTED-USERID=YES` in the `DEFAULTS=SECURITY` section of the Broker Attribute File.
3. Set `BKTUID=Y` in member WAL*vrs*.SRCE(SAFPARMS).
4. Assemble/Linkedit load module SAGCFG using job WAL*vrs*.JOBS(SAFI010), e.g. into WAL*vrs*.LOAD, or a Loadlib of your choice
5. Linkedit module BROKER, which is used by the Reptor nucleus, to include the following:

   ```
   EXXvrs.LOAD(NA2PETS)
   WALvrs.LOAD(SAFCFG)
   ```

   Example:

   ```
   //<jobname> JOB <account>
   //EXXJ109 EXEC PGM=IEWL,
   // PARM='LET,LIST,MAP,NORENT,NOREUS,SIZE=(512K,48K)'
   //SYSTERM DD SYSOUT=*
   //SYSPRINT DD SYSOUT=*
   //SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(999,99))
   //EXXLIB DD DISP=SHR,DSN=<exxlib>
   //WALLIB DD DISP=SHR,DSN=<wallib>
   //SYSLMOD DD DISP=SHR,DSN=<outlib>
   //SYSLIN DD *
   INCLUDE EXXLIB(BROKER)
   INCLUDE EXXLIB(NA2PETS)
   INCLUDE WALLIB(SAFCFG)
   ENTRY BROKER
   NAME BROKER(R)
   /*
   //
   ```

6. Set `SAF=YES` in member WAL*vrs*.SRCE(LNKGBLS). /* Batch/TSO *//
7. Assemble/Linkedit the Adabas Link Routine Globals Module LNKGBLS (Batch/TSO), using job WAL*vrs*.JOBS(ASMGBLS).
8. Restart Broker.
9. Restart Reptor.

Index