ⓢ software ᴬᴳ

Event Replicator for Adabas

Reference

Version 4.1.1

October 2022

ADABAS & NATURAL

Table of Contents

Preface

This documentation provides reference information for Adabas initialization parameters and Adabas utilities used by the Event Replicator for Adabas as well as the Adabas Online System screens pertinent to the Adabas Event Replicator Subsystem. It also describes the Event Replicator initialization parameters you can use in the Event Replicator Server startup job if you elect not to use the Adabas Event Replicator Subsystem to set up the definitions necessary for the Event Replicator for Adabas.

This document is organized as follows:

Pertinent Adabas Initialization (ADARUN) Parameters	Describes the Adabas initialization parameters required for the Event Replicator for Adabas. These parameters are specified in the ADARUN job for your Adabas database.
Utilities Used with Replication	Describes the utilities that are pertinent to replication, either for the Adabas database or for the Event Replicator Server.
Event Replicator Initialization Parameters	Describes the initialization parameters you must specify in the Event Replicator Server startup job if you are not using the online Adabas Event Replicator Subsystem.
C5 Command: Write User Data to Protection Log	Describes the Adabas C5 command, including how it can be used by the client application to submit messages to one or more Event Replicator Server destinations.

1 Conventions

This document covers the following topics:

- ADARPD Syntax
- Syntax Conventions
- Control Statement Rules
- Parameter Values

ADARPD Syntax

Utility control statements have the following format:

ADARPD *parameter subparameters*

where

ADARPD	is the name of the utility that specifies the Event Replicator initialization parameters.
parameter	is the name of the specific initialization parameter to be executed. For example, the following: ADARPD DBID ADARPD DESTINATION ADARPD SUBSCRIPTION
subparameters	is a list of subparameters pertinent to the specified ADARPD parameter. Some subparameters are required. Parameters in the list are almost always keywords with the format: *subparameter=value* A subparameter may have one or more operands. Most subparameters require that you select or otherwise specify an operand value. Some operands are positional (value1 , value2 ,..., value*x*), meaning that the values must be in a certain order as described in the text. All subparameters must be separated by commas. In the subparameter syntax descriptions in this documentation, subparameters are listed vertically (stacked) or are separated by vertical bars (	). Each list shows all possible subparameters, from which one or more can (or must) be specified. Although subparameters in the list must be separated by commas, these commas are omitted in the syntax statements when the subparameters are stacked.

Syntax Conventions

The following table describes the conventions used in syntax diagrams in this documentation.

Convention	Description
uppercase, bold	Syntax elements appearing in uppercase and bold font are Adabas keywords. When specified, these keywords must be entered exactly as shown.
lowercase, italic, normal font	Syntax elements appearing in lowercase and normal, italic font identify items that you must supply.
mixed case, normal font	Syntax elements appearing in mixed case and normal font (not bold or italic) identify values established by other Adabas control statements. This notation is usually used to identify how default values are determined for some parameters in Adabas syntax.
underlining	Underlining is used to identify default values, wherever appropriate.
vertical bars (|)	Vertical bars are used to separate mutually exclusive choices. **Note:** In more complex syntax involving the use of large brackets or braces, mutually exclusive choices are stacked instead.
brackets ([ ])	Brackets are used to identify optional elements. When multiple elements are stacked or separated by vertical bars within brackets, only one of the elements may be supplied.
braces ({ })	Braces may be used to identify required elements. When multiple elements are stacked or separated by vertical bars within brackets, one and only one of the elements must be supplied.
indentation	Indentation is used to identify subparameters of a parameter.
ellipsis (...)	Ellipses are used to identify elements that can be repeated. If other punctuation is included before a ellipsis, the punctuation must be used to separate repeated elements.
other punctuation and symbols	All other punctuation and symbols must be entered exactly as shown.

Control Statement Rules

The following rules apply for the construction of ADARPD control statements:

1. Each control statement must contain the letters "ADARPD" in positions 1-6.

2. The parameter name follows ADARPD, separated by at least one space.

3. Subparameter entries and multiple values within subparameter entries must be separated by commas.

4. No space is permitted before or after "=".

5. The comma following the last subparameter entry of a parameter is optional.

6. ADARPD control statement processing ends with position 72 or when a space is encountered after the beginning of the parameter list. Entries made in positions 73-80 are not processed.

7. A statement that contains an asterisk "*" in position 1 is read as a comment and is not processed.

8. ADARPD control statements are continued by specifying the extra parameters on a new statement following (and separated by at least one space from) the letters "ADARPD" in positions 1-6.

Parameter Values

Variable values actually specified following the equals "=" sign in parameters (represented by italicized labels in the preceding examples and elsewhere in this documentation) have the following syntax:

```
parameter = value
parameter = value-list
```

where *value* is as described in the following sections. *Value-list* is a variation of *value*, and is allowed either in place of or with *value*, depending on the individual parameter rules as described in the text.

value

Value may consist of a number or a string of alphanumeric or hexadecimal characters. In some optional subparameters, a default value is assumed if the subparameter is not specified.

Alphanumeric Values

Alphanumeric values are specified in one of the following ways:

If the value includes . . .	Single quotes (') around it are . . .
only upper- or lowercase letters, numeric digits and minus (-)	optional
any other characters, including a single quote itself (which must be entered twice)	required

Numeric Values

Numeric values are specified as follows:

If the value represents . . .	Specify . . .
a number of either blocks or cylinders	the letter B must immediately follow the value if blocks are being specified; otherwise, cylinders are assumed: SIZE=200B (200 blocks) SIZE=200 (200 cylinders)
an Adabas file	a one- to four-digit number (leading zeros permitted): FILE=3 FILE=03 FILE=162
a device type	a four-digit number corresponding to the model number of the device type to be used: DEVICE=3380
a field name or descriptor	a two-character field name corresponding to the field name or descriptor: FIELD1=NA

Hexadecimal Values

Hexadecimal values are accepted if this is specified in the parameter description. Hexadecimal values must be specified within single quotes (') following the indicator "X". For example:

```
X'0002DC9F'
```

value-list

'value,...' (alphanumeric values)

```
ADARPD   SFBAI='AA,BA,BB,BC,BD,CA,CB,CD,CE,CF.'
```

Positional Values

Subparameters may be positional if they relate to values specified on corresponding subparameters. For example:

```
ADARPD DAIFILE=97,DAIDBID=190,DATDBID=133,DATFILE=22
```

For input file 97, this example identifies the input file's database ID (190) and its associated target database ID (133) and file (22) for replication. Subparameters (DAIDBID, DATDBID, and DATFILE) must be specified for each occurrence of DAIFILE.

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://documenta-tion.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.software-ag.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.soft-wareag.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 Pertinent ADARUN Parameters

This chapter describes the ADARUN settings pertinent to Event Replicator for Adabas.

ASYTVS Parameter: Asynchronous Buffer Flush

Parameter	Specify . . .	Possible Values	Default
ASYTVS	whether to flush buffers asynchronously based on volume serial number.	YES | NO	YES

Specific Product Recommendations

Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

Example

The following example disables the asynchronous buffer flush based on volume.

```
ADARUN PROG=ADANUC,ASYTVS=NO
```

FMXIO Parameter: Parallel LFIOP I/O Operations

Parameter	Specify . . .	ASYTVS=	Minimum	Maximum	Default
FMXIO	the maximum number of parallel I/O operations by LFIOP flush processing.	YES	1	16	1
		NO	1	100	60

The LFIOP parameter enables asynchronous buffer flush operation and sets the I/O pool size. The FMXIO parameter sets the limit on the number of I/O operations that can be started in parallel by LFIOP flush processing.

Parameter Dependencies

The precise meaning of the FMXIO parameter depends on the setting of the ASYTVS parameter):

- When ASYTVS=YES (perform buffer flushes by volume), FMXIO specifies the number of I/Os to be started in parallel *on each volume*. The minimum and default number is 1; the maximum number is 16. If the specified number exceeds that, it is automatically reduced to 16.

- When ASYTVS=NO (perform buffer flushes in ascending RABN sequence, without regard to the distribution of the blocks over volumes), FMXIO specifies the number of I/Os to be started in parallel overall. The minimum number is 1; the maximum value is 100; and the default is 60.

Dynamic Modification

The setting of FMXIO can be modified dynamically using the command FMXIO=nnn from the operator console or the Modify Parameter function of Adabas Online System.

Specific Product Recommendations

Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

Example

The following example specifies that a maximum of 30 I/O operations for asynchronous buffer flushes can be started in parallel:

```
ADARUN PROG=ADANUC,ASYTVS=NO,FMXIO=30
```

LARGEPAGE Parameter: Large Page Support Option

Parameter	Specify . . .	Possible Values	Default
LARGEPAGE	whether or not the Adabas nucleus should use page-fixed one-megabyte (1M) large pages of real storage above the two-gigabyte (2G) bar (LARGEPAGE=L64), page-fixed 2G large pages above the 2G bar (LARGEPAGE=G64), or pageable one-megabyte (1M) large pages of 64-bit real storage (LARGEPAGE=P64). A value of "YES" is equivalent to specifying a value of *L64*. Large pages can only be used if your operating system supports them and 64-bit virtual storage.	G64 | L64 | P64 | NO | YES	NO

This parameter indicates whether or not the Adabas nucleus should use large pages. The default is NO.

> **Note:** At this time, this parameter is applicable only to z/OS systems.

Value	Meaning
G64	The Adabas nucleus will attempt to acquire virtual storage above the 2G bar backed by page-fixed 2G large pages. The virtual storage will be backed by large pages if the operating system supports them and if sufficient large pages are available in the system. If insufficient large pages are available, the virtual storage above the 2G bar will be backed by pageable four-kilobyte (4K) pages.

You can specify LARGEPAGE=G64 only if the ADARUN V64BIT parameter is set to "YES".

The LARGEPAGE=G64 setting does not apply to the flush I/O pool, Work and PLOG I/O buffers, or cast-out directory buffer (in a cluster environment). At this time, it applies only to ADARUN

Value	Meaning
	parameter LRPL (in an Adabas replication environment) in the Adabas nucleus (but not in the Event Replicator Server).
	Note: Allocation requests for storage to be backed by 2G large pages are rounded up to the nearest multiple of two gigabytes. Any difference between the specified size and the allocated size will remain unused. For this reason, we recommend that you specify a multiple of two gigabytes for any storage area to which the LARGEPAGE=G64 setting applies, to avoid leaving a potentially large amount of allocated storage unused. If LARGEPAGE=G64 is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.
L64 or YES	The Adabas nucleus will attempt to acquire virtual storage above the 2G bar backed by page-fixed 1M large pages. The virtual storage will be backed by large pages if the operating system supports them and if sufficient large pages are available in the system. If insufficient large pages are available, the virtual storage above the 2G bar will be backed by pageable 4K pages. A value of "YES" is equivalent to specifying a value of "L64". You can specify LARGEPAGE=L64 or LARGEPAGE=YES only if the ADARUN V64BIT parameter is set to "YES".
P64	Where appropriate, the Adabas nucleus will attempt to acquire virtual storage above or below the 2G bar backed by pageable 1M large pages in 64-bit real storage. The virtual storage will be backed by large pages if the operating system supports them and if sufficient large pages are available in the system. If insufficient large pages are available, the virtual storage above the 2G bar will be backed by pageable 4K pages. LARGEPAGE=P64 can be specified with the ADARUN V64BIT parameter set to "NO" or to "YES".
NO	Virtual storage above the 2G bar will be backed by 4K pages. This is the default value.

The amount of storage in your operating system available for page-fixed 1M or 2G large pages is set using the LFAREA parameter in PARMLIB member IEASYS*xx*. For complete information on the LFAREA parameter, refer to the *IBM z/OS Extended Addressability Guide*.

Example

The following example requests that the Adabas nucleus should make use of virtual storage above the 2G bar and should use 1M large pages.

```
ADARUN PROG=ADANUC,LARGEPAGE=YES,V64BIT=YES
```

LBP Parameter: Length of Buffer Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LBP	the maximum size of the Adabas buffer pool.	80 000	2147483647	350000

The maximum number of bytes to be used for the Adabas buffer pool during a session.

Adabas allocates the number of bytes specified to the Adabas buffer pool if sufficient memory is available. A minimum of 80000 bytes must be available to successfully initiate a session. For performance reasons, however, Software AG recommends that you specify at least 50000 bytes per thread. Ensure that sufficient memory is available to accommodate the LBP size specified. If asynchronous buffer flush management is used (that is, the LFIOP parameter specifies a nonzero value), the LBP value must be greater than that specified by LFIOP.

In environments running in 31-bit addressing mode, the buffer pool space is allocated above the 16-megabyte line.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

The buffer efficiency statistic can be used to determine the optimal size of the parameter.

Specific Product Recommendations

- Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example allocates a maximum of 800000 bytes to the Adabas buffer pool for this session.

```
ADARUN PROG=ADANUC,LBP=800000
```

LDEUQP Parameter: Length of Unique Descriptor Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LDEUQP	the size of the unique descriptor pool.	5000	2147483647	5000

The unique (UQ) descriptor pool comprises elements that are the length of a unique descriptor value plus 14 bytes. The UQ descriptor pool should be large enough to hold all unique descriptor values that are being changed by *all* transactions at any point in time.

Adabas will make an entry in this pool every time a unique descriptor value is deleted from or inserted into a file's index. A full update of a unique descriptor value produces two entries: one for the deletion of the old value and one for the insertion of the new value. The entry for an insertion is normally removed as soon as the index update has been performed. The entry for a deletion is removed when the current transaction ends. However, Adabas Cluster Services and Adabas Parallel Services nuclei always keep entries for insertions until the end of the transaction too.

Specific Product Recommendations

- Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

For example, assuming that the average length of unique descriptor values is about 16 bytes, the following Adabas session can process a maximum of about 1000 unique descriptors at one time.

```
ADARUN PROG=ADANUC,LDEUQP=30000
```

LFIOP Parameter: Asynchronous Buffer Pool Control

Parameter	Specify . . .	Minimum	Maximum	Default
LFIOP	whether to enable asynchronous buffer flush operation and set the I/O pool size.	0 | 1 | 80000	18446744073709551615 (although this is really limited by the available amount of virtual and real memory)	0

The asynchronous buffer flush I/O pool enables asynchronous Adabas buffer flush operation and defines the size of the related buffer. Specifying a non-zero value for LFIOP enables asynchronous buffer flush operation; the value specified determines the buffer size, in bytes.

If LFIOP is not specified or zero is specified (the default), synchronous buffer flushing is performed. Note that flushing is always synchronous in single-user mode (MODE=SINGLE).

Value	Meaning
0	No asynchronous buffer flush (the default); flushing is synchronous.
1	Asynchronous buffer flush operations occur. The flush pool is sized at 25% of the buffer pool (LBP=) size.
n (minimum of 80000)	Asynchronous buffer flush operations occur. The flush pool size is defined by the LFIOP setting (in bytes).

When LFIOP=1 or LFIOP=n and the specified or calculated flush pool size is less than 80000 bytes, Adabas uses a size of 80000 bytes.

In environments running in 31-bit addressing mode, the I/O pool is allocated above the 16-megabyte line.

In environments supporting 64-bit virtual storage, the flush I/O pool is allocated above the 2 gigabyte bar when the V64BIT ADARUN parameter is set to "YES".

Specific Product Recommendations

- Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server causing replication data to be written and read from the SLOG file.

- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1") consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example enables asynchronous buffer flush operation for this session and allocates an I/O pool of 200000 bytes (25% of the LBP size).

```
ADARUN PROG=ADANUC,LFIOP=1,LBP=800000
```

LFP Parameter: Length of Internal Format Buffer Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LFP	the size of the Adabas internal format buffer pool.	6000	2147483647	12 000

This pool stores internal format buffers used with Adabas read and update commands as described in the *Adabas Command Reference Guide*.

Each internal format buffer pool entry requires approximately 60 + (64 * *n*) bytes, where *n* equals the number of fields referenced in the command.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

In environments running in 31-bit addressing mode, the LFP pool is allocated above the 16-megabyte line.

Specific Product Recommendations

If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example allocates 8000 bytes for the Adabas internal format buffer pool for this session.

```
ADARUN PROG=ADANUC,LFP=8000
```

LP Parameter: Length of Data Protection Area (Work Part 1)

Parameter	Specify . . .	Minimum	Maximum	Default
LP	the number of blocks to allocate to Work part 1.	200	see text	1000

The LP parameter specifies the number of blocks to allocate to the Work data protection area, which is part 1 of the Work data set. This area must be large enough to accommodate the data protection information for the current transaction for all ET logic users. Details can be found in *Adabas Restart and Recovery* in the *Adabas Operations Manual*.

The LP parameter is no longer limited to 65535 blocks; however, it is still limited by the size of the Work data set and the sizes of Work parts 2, 3, and 4.

> **Note:** For any pending autorestart processing, Adabas retains and uses the value of the LP parameter from the previous session. If the LP parameter for the new session is specified with a different value than in the previous session, the nucleus will process any autorestart using the old value of LP, and establish the new LP size at the end of the autorestart.

Specific Product Recommendations

■ Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

■ If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example uses 2,000 blocks to store data protection information on the Adabas Work data set.

```
ADARUN PROG=ADANUC,LP=2000
```

LRPL Parameter: Adabas or Event Replicator Replication Pool Size

Parameter	Specify . . .	Possible Values	Default
LRPL	The size of the Adabas or Event Replicator replication pools.	minimum 20000	100000 (if REPLICATION=YES) or 0 (if REPLICATION=NO)

> **Notes:**

1. The LRPL parameter should be specified for an Adabas nucleus running with ADARUN parameter REPLICATION=YES. In addition, the replication pool will be allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters REPLICATION=YES and V64BIT=YES.

2. The LRPL parameter should be specified for an Event Replicator Server.

3. The LRPL parameter values in the Adabas nucleus and the Event Replicator Server may be different.

4. Software AG recommends setting LRPL to a relatively large value (e.g. LRPL=40M or greater). To determine what this setting should be for your site, you could run Adabas in a test environment with replication turned on, simulate the activity in the production environment on this test

system, and use the Adabas DRES operator command to view the current and high-water mark of replication pool usage.

5. If Event Replicator 3.4 SP1 or later is installed and if an Adabas nucleus with replication is run with V64BIT=YES and LARGEPAGE=G64, the allocation request for the replication pool will be rounded up to the nearest multiple of two gigabytes. Any difference between the specified size and the allocated size will remain unused. For this reason we recommend that if LARGE-PAGE=G64, you specify the LRPL value as a multiple of two gigabytes to avoid leaving a potentially large amount of allocated storage unused. If LARGEPAGE=G64 is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

Example:

```
ADARUN PROG=ADANUC,REPLICATION=YES,LRPL=50000K
```

LU Parameter: Length of Intermediate User Buffer Area

Parameter	Specify . . .	Minimum	Maximum	Default
LU	the size of the intermediate user buffer area.	none	none (see note 1)	65535 (see note 2)

The LU parameter is optional. The size specified must be large enough to accommodate all Adabas control information (204 bytes), all user buffers (format, record, search, value, ISN) that may be required for any given Adabas command, plus any user information passed from Adabas link routines to nucleus user exits. If the ADALNK has user exits, the addresses of the ACB and each buffer passed will be stored in this space. Also, if the ADALNK has declared LUSIZE and has user exits, this buffer size plus 64 bytes will also be stored in this space.

If the multifetch/prefetch option or a utility that needs large record/ISN buffers is to be used during the session, the setting of LU must be large enough to contain the buffers needed.

If you are using the ADARPD IQETBBROKERID NET communication (used with Event Replicator for Adabas), make sure that the webMethods EntireX Broker NET definition parameter IUBL is set to a value as large as the setting of this LU ADARUN parameter.

Notes:

1. An error occurs if the LU parameter specifies a value greater than the byte count implied by the NAB (number of attached buffers) parameter. On z/OS systems, LU cannot exceed a value greater than that produced by the following calculation: (`NABvalue x 4096`); on z/VSE and BS2000 systems, LU cannot exceed a value greater than that produced by the following calculation: (`NABvalue x 4096) - 256`. For more information about the NAB parameter, read *NAB: Number of Attached Buffers*, in *Adabas Operations Manual*.

2. Due to the length of the record buffer of the utilities that need the nucleus, e.g., ADAULD, the default value is set to 65,535. If the value of LU is less than 65,535 for an Adabas session, a response code will occur when such a utility is running.

The LU parameter syntax is:

```
LU={ n | 65535 }
```

Specific Product Recommendations

■ For Event Replicator Server databases running with Adabas 8, the LU parameter must be greater than or equal to 167,000.

   In addition, if data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work LU parameter must be greater than or equal to the LU parameter setting for the Event Replicator Server itself (greater than or equal to 167,000).

■ The ADACHK utility can use large record buffer lengths when making nucleus calls to verify spanned Data Storage records or an index structure with many levels. If this is the case, the settings of your LU and NAB ADARUN parameters may need to be increased.

■ For an Adabas nucleus running with the ADARUN REPLICATION=YES parameter, ensure the LU ADARUN parameter is set to a value greater than or equal to 70000.

■ For an Adabas Audit Server, the LU parameter must be greater than or equal to 167,000.

   In addition, if audit data is sent through Entire Net-Work from one or more Adabas nuclei to an Adabas Audit Server, the Entire Net-Work LU parameter must be greater than or equal to the LU parameter setting for the Adabas Audit Server itself (greater than or equal to 167,000).

Example

The following example runs the Adabas nucleus with an Adabas intermediate user buffer area of 20,000 bytes.

```
ADARUN PROG=ADANUC,LU=20000
```

LWP Parameter: Length of Adabas Work Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LWP	the size of the Adabas work pool.	80000 but not less than NT * 25000	2147483647	150000

> **Note:** The LWP (work pool length) parameter value must be equal to or greater than the LS (internal sort area length) value plus its minimum value (19968) multiplied by 2; that is, `(LS + 19968) * 2`.

The Adabas work pool area is used to store the following:

- descriptor value tables (DVT);

- intermediate user buffer areas (see parameter LU);

- sort work space areas (see parameter LS);

- work space during session and command execution.

In environments running in 31-bit addressing mode, the work pool space is allocated above the 16-megabyte line.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session. If expanded files are being used, the LWP value may need to be increased above the normal setting.

Specific Product Recommendations

If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example runs the Adabas nucleus with a work pool of 900000 bytes.

```
ADARUN PROG=ADANUC,LWP=900000
```

MLWTO Parameter: Multi-Line Console Message Control

📄 **Note:** At this time, this parameter only affects the appearance of replication-related and cluster-related messages (messages with prefixes ADAF*, ADAX*, PLI*, and PLX*).

Parameter	Specify . . .	Possible Values	Default
MLWTO	Whether multi-line messages that are normally written to the operator console as a series of single-line write-to-operator (WTO) messages should instead appear as multi-line WTO messages.	YES | NO	NO

To activate multi-line console messages, set MLWTO to YES in the nucleus startup job. When MLWTO is set to NO (or is allowed to default), no multi-line console messages will appear.

Multi-line console messages are only supported in z/OS and z/VSE environments. Specifying MLWTO=YES in BS2000 environments has no effect on the console messages.

Example:

```
ADARUN PROG=ADANUC,MLWTO=YES
```

NAB Parameter: Number of Attached Buffers

Parameter	Specify . . .	Minimum	Maximum	Default
NAB	the number of attached buffers to be used.	1	varies, depending on the amount of available virtual storage	16

The NAB parameter defines the number of attached buffers to be used during the session. An attached buffer is an internal buffer used for interregion communication. It is required in all environments. Adabas allocates an attached buffer pool with a size equal to the value of NAB multiplied by 4096 bytes.

📄 **Note:** The allocation for buffers in the attached buffer pool is done in 256 byte slots; this means that each allocation is rounded to a multiple of 256. For example, if a size of 300 bytes is needed, the allocated space is 512 bytes.

You may specify as many attached buffers as fit into the available virtual storage.

In environments running in 31-bit addressing mode, the attached buffer pool space is allocated above the 16-MB line.

The NAB parameter syntax is:

```
NAB={ n | 16 }
```

Specific Product Recommendations

- For Event Replicator Server databases, set 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).

- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

- If data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work NAB parameter must also be set to a value greater than or equal to: `41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server`.

- Users of the Adabas Review hub should read *Storage Requirements* in the *Adabas Review Concepts Manual* for more information about the space requirements of the Command Queue for Adabas Review.

- The ADACHK utility can use large record buffer lengths when making nucleus calls to verify spanned Data Storage records or an index structure with many levels. If this is the case, the settings of your LU and NAB ADARUN parameters may need to be increased.

- For an Adabas Audit Server, set parameter NAB to a value greater than or equal to:

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

  For example, if one Adabas nucleus will be sending data to the Adabas Audit Server, set the NAB parameter greater than or equal to 410 (e.g. `NAB=420`).

  If data is sent through Entire Net-Work from one or more Adabas nuclei to an Adabas Audit Server, the Entire Net-Work NAB parameter must also be set to a value greater than or equal to:

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

Example

The following example runs the Adabas Review hub nucleus with 50 attached buffers.

```
ADARUN PROG=ADAREV,NAB=50
```

NC Parameter: Number of Command Queue Elements

Parameter	Specify . . .	Minimum	Maximum	Default
NC	the maximum number of command queue elements.	20	32767	200

The number of command queue elements (CQEs) established for the Adabas or Review hub session determines the maximum number of Adabas commands that may be queued or be in process at any one time during the session.

Each call from the Adabas nucleus is assigned a CQE. The CQE is released when the user receives the results of the command, the Adabas Review hub has processed the command, or the user has been timed out..

192 bytes are required for each CQE.

Software AG recommends that you set NC high enough to allow one command per active user for possible synchronization during execution of the online SAVE database function of the ADASAV utility.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

For more information about the space requirements of the Command Queue for Adabas Review, refer to *Storage Requirements* in the *Adabas Review Concepts Manual*.

Specific Product Recommendations

- For Event Replicator Server databases, set parameter `NC` to a value greater than or equal to: `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 `NC` parameter greater than or equal to 10 (for example NC=20).

- If data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work NC parameter must also be set to a value greater than or equal to: `10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server`.

- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

- For an Adabas Audit Server, set parameter NC to a value greater than or equal to:

```
10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
```

For example, if one Adabas nucleus will be sending data to the Adabas Audit Server, set the NC parameter greater than or equal to 10 (e.g. `NC=20`).

If data is sent through Entire Net-Work from one or more Adabas nuclei to an Adabas Audit Server, the Entire Net-Work NC parameter must also be set to a value greater than or equal to:

```
10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
```

Example:

Run the Adabas nucleus with a maximum of 500 elements in the command queue.

```
ADARUN PROG=ADANUC,NC=500
```

The following example runs the Adabas Review hub nucleus with a maximum of 500 elements in the command queue.

```
ADARUN PROG=ADAREV,NC=500
```

NH Parameter: Number of Hold Queue Elements

Parameter	Specify . . .	Minimum	Maximum	Default
NH	the maximum number of hold queue elements.	20	16777215	500

One hold queue element is required for each record (ISN) placed in hold status. A hold queue element is released when the record is released from hold status.

The hold queue is normally used to hold ISNs for the duration of a transaction. Therefore NH should be large enough to contain at least the average number of ISNs held per transaction multiplied by the maximum number of transactions running in parallel.

One hold queue element takes 52 bytes of storage. In environments running in 31-bit addressing mode, the hold queue is allocated above the 16-megabyte line.

Each shared hold request with a different command ID (CID), as well as a (shared or exclusive) hold request without a CID, is counted against the NH limit.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

Specific Product Recommendations

- Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example runs the Adabas nucleus with a maximum of 2000 elements in the hold queue.

```
ADARUN PROG=ADANUC,NH=2000
```

NISNHQ Parameter: Number of ISNs in Hold Queue per User

Parameter	Specify . . .	Minimum	Maximum	Default
NISNHQ	the maximum number of records that can be placed in hold status at the same time by one user.	1	16777215 (NH parameter maximum)	(NH / NU) * 2 or 20, whichever is larger

NISNHQ determines the maximum number of held ISNs the hold queue (see the NH parameter description) can contain from one user. The maximum value allowed for this parameter is the value set for the NH ADARUN parameter (which has a maximum value of 16777215). The default is the greater of either 20 or the number of ISNs the hold queue can contain divided by the number of users, multiplied by two.

A user who attempts to place more records in hold status than permitted receives response code 47 (ADARSP047), subcode 1.

Each shared hold request with a different command ID (CID), as well as a (shared or exclusive) hold request without a CID, is counted against the NISNHQ limit.

Specific Product Recommendations

Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

Overriding the Parameter Setting

You can override this parameter setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.

Examples

In the following example, the maximum number of records that may be in hold status for a single user is set at 100.

```
ADARUN PROG=ADANUC,NISNHQ=100,NH=450...
```

In the following example, NISNHQ is not specified. The maximum number of records that may be in hold status for a single user is 80 (the NISNHQ default of `(NH / NU) * 2`, which is greater than the alternative default of 20).

```
ADARUN PROG=ADANUC,NU=50,NH=2000...
```

NT Parameter: Number of Threads

Parameter	Specify . . .	Minimum	Maximum	Default
NT	the number of user threads to be used during the Adabas session.	4	250	5

A thread is used to execute a single command for any user (access or update). Each thread requires 8192 bytes.

In environments running in 31-bit addressing mode, the threads are allocated above the 16-megabyte line.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

For Event Replicator Server databases, this parameter must be greater than or equal to 15.

Specific Product Recommendations

- For Event Replicator Server databases, the NT parameter must be greater than or equal to 15.

- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

- For an Adabas Audit Server, the NT parameter must be greater than or equal to 15.

Example

The following example runs the Adabas nucleus with six (6) user threads.

```
ADARUN PROG=ADANUC,NT=6
```

NU Parameter: Number of User Queue Elements

Parameter	Specify . . .	Minimum	Maximum	Default
NU	the maximum number of user queue elements.	20	16777215	200

Adabas assigns a user queue element (UQE) to each active user when the user issues an OP command or (if the user did not issue an OP command) the first Adabas command.

Adabas releases the UQE when the user issues a CL command or when there is a timeout for an ET logic user who did not specify a restart ID in the control block (Additions 1 field) for the OP command.

Currently, 272 * (NU + 4) bytes are allocated to the user queue. The length of each UQE is 288 bytes.

In environments running in 31-bit addressing mode, the user queue is allocated above the 16-megabyte line.

Because each user's file list can be a different size, the number of users who can be active at the same time may be less than the NU value. Therefore, you may need to specify an NU value that is greater than the maximum number of users desired.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

Specific Product Recommendations

If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example runs the Adabas nucleus with a maximum of 500 elements in the user queue.

```
ADARUN PROG=ADANUC,NU=500
```

REPLICATION Parameter: Replication Control

Parameter	Specify . . .	Possible Values	Default
`REPLICATION`	Whether to activate replication for the Adabas nucleus.	YES | NO	NO

Valid values are described in the following table.

Value	Meaning
YES	Activate replication for the Adabas nucleus.
NO	The default setting. Do not activate replication for the Adabas nucleus.

When REPLICATION=NO is specified and there are replicated files on the database, replication will be turned off for all replicated files. A message is issued during nucleus initialization indicating which files were defined with replication.

The `REPLICATION` parameter should not be specified for the Event Replicator Server. REPLICATION=NO must be explicitly specified if:

- You want to start the Adabas nucleus without replication active, *and*
- `REPLICATION=YES` was specified in the previous Adabas nucleus session, *and*
- Replication has been defined for one or more files on the database.

`REPLICATION=NO` must be explicitly specified for utility execution if:

- A utility requiring an active nucleus is executed with `MODE=SINGLE`, *and*
- `REPLICATION=YES` was specified in the previous Adabas nucleus session, *and*
- Replication has been defined for one or more files on the database.

> **Note:** The replication pool will be allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters REPLICATION=YES and V64BIT=YES.

Example

The following example activates replication for an Adabas session.

```
ADARUN PROG=ADANUC,REPLICATION=YES
```

RPLCONNECTCOUNT Parameter: Replication Connection Attempt Count

Parameter	Specify . . .	Possible Values	Default
RPLCONNECTCOUNT	the number of connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails with an unexpected response code.	0 - 2147483647	0

A value of zero indicates that no connection attempts should occur; a value of zero makes the most sense in situations where the Adabas database and the Event Replicator Server execute together on the same logical partition (LPAR). If the Adabas database and the Event Replicator Server execute on different LPARs, however, setting a real value using this command helps avoid errors that might arise if network problems occur because the network is not started or a network connection between the Adabas database and the Event Replicator Server is lost.

RPLCONNECTINTERVAL Parameter: Replication Connection Attempt Interval

Parameter	Specify . . .	Possible Values	Default
RPLCONNECTINTERVAL	the interval (in seconds) between connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails with an unexpected response code.	0 - 2147483647	0

A value of zero indicates that no connection attempts should occur; a value of zero makes the most sense in situations where the Adabas database and the Event Replicator Server execute together on the same logical partition (LPAR). If the Adabas database and the Event Replicator Server execute on different LPARs, however, setting a real value using this command helps avoid errors that might arise if network problems occur because the network is not started or a network connection between the Adabas database and the Event Replicator Server is lost.

RPLPARMS Parameter: Replication Parameter Location

Parameter	Specify . . .	Possible Values	Default
RPLPARMS	Where your replication definitions (initialization parameters) should be read from.	BOTH	FILE |NONE	PARMS	See below.

This parameter is only used when starting the Event Replicator database.

If the RPLPARMS parameter is not specified, the following logic is used to determine where to read the replication definitions:

- If the Replicator system file is loaded on the Event Replicator database, the replication definitions are read from that file.

- If the Replicator system file is not loaded on the Event Replicator database, the replication definitions are read from DDKARTE.

Valid values are described in the following table.

Value	Meaning
BOTH	The replication definitions are read first from the Replicator system file and secondly from DDKARTE. Separate entries are built for each and errors are generated if there are duplicate entries.
FILE	The replication definitions are read from the Replicator system file, which is maintained using the Adabas Event Replicator Subsystem in Adabas Online System (AOS).
NONE	No replication definitions are read. The Event Replicator Server will establish itself with a null set of replication definitions and will not communicate or handshake with related Adabas nuclei. This mode allows maintenance to occur to the Replicator system file, without the Event Replicator Server communicating with related Adabas nuclei.
PARMS	The replication definitions are read from DDKARTE.

RPLSORT Parameter: Event Replicator for Adabas Transaction Data Sorting Control

Parameter	Specify . . .	Possible Values	Default
RPLSORT	Whether Event Replicator for Adabas transaction data sorting should occur or not.	YES | NO	YES

This parameter is only used for the Adabas database from which transactions will be replicated. It should not be used for the Event Replicator Server database.

By default (RPLSORT=YES), Event Replicator for Adabas sorts transaction data by a key consisting of the file number, ISN, and relative update number. If you do not want this sort to occur, specify RPLSORT=NO.

It is important to note that the RPLSORT setting will affect how the Event Replicator consolidates transaction modifications and the order in which they are replicated. When RPLSORT=YES, all modifications for a given ISN in a file are consolidated before replication occurs. For complete information on regular modification consolidation, read *Detailed Adabas Nucleus Processing* in the *Event Replicator for Adabas Concepts*. When RPLSORT=NO is specified, the modifications are still consolidated, but their referential integrity is maintained. In other words, the chronological order of the updates is maintained for each ISN in a file.

> **Note:** If one or more source files with unique descriptors are replicated to an Adabas target file also with unique descriptors, you may need to run with RPLSORT set to "NO" to prevent an invalid response code198 (ADARSP198) from occurring when replicating the data.

Example

For example, assume the following modifications occur, in the following order:

```
Insert File 77 ISN 80 Seq 1
Update File 77 ISN 80 Seq 2
Insert File 77 ISN 1010 Seq 3
Update File 77 ISN 80 Seq 4
Update File 77 ISN 1010 Seq 5
```

If `RPLSORT=YES`, these modifications are consolidated and replicated as follows:

```
Insert File 77 ISN 80 Seq 1 + Seq 2 + Seq 4
Insert File 77 ISN 1010 Seq 3 + Seq 5
```

If `RPLSORT=NO`, these modifications are consolidated and replicated as follows:

```
Insert File 77 ISN 80 Seq 1 + Seq 2
Insert File 77 ISN 1010 Seq 3
Update File 77 ISN 80 Seq 4
Update File 77 ISN 1010 Seq 5
```

The end result is the same, regardless of whether RPLSORT is set to YES or NO, but multiple modifications to the same ISN in a file may be necessary when RPLSORT is set to NO.

RPWARNINCREMENT Parameter: Replication Pool Warning Message Interval

Parameter	Specify . . .	Possible Values	Default
`RPWARNINCREMENT`	The interval (in percent of `LRPL` usage) at which warning messages should be sent that indicate that the replication pool usage has exceeded the threshold set by the `RPWARNPERCENT` **parameter**.	1-99	10

This ADARUN parameter applies to both the Adabas and Event Replicator Server replication pools.

RPWARNINTERVAL Parameter: Replication Pool Usage Interval

Parameter	Specify . . .	Possible Values	Default
RPWARNINTERVAL	The interval, in seconds, during which replication pool usage warning messages are suppressed on the console.	1-2147483647	60

This ADARUN parameter applies to both the Adabas and Event Replicator Server replication pools.

When the number of replication pool usage warning messages sent to the console exceeds the limit specified by the RPWARNMESSAGELIMIT parameter, all such messages are suppressed for the interval of time specified by this parameter. When the interval specified by the RPWARNINTERVAL parameter has passed, warning messages resume.

This parameter, along with the RPWARNMESSAGELIMIT parameter, is provided to avoid flooding the console with replication pool usage warning messages.

RPWARNMESSAGELIMIT Parameter: Replication Pool Usage Warning Message Count

Parameter	Specify . . .	Possible Values	Default
RPWARNMESSAGELIMIT	The number of replication pool usage warning messages that can be issued to the console before message suppression begins.	1-2147483647	5

This ADARUN parameter applies to both the Adabas and Event Replicator Server replication pools.

When the threshold specified by this parameter is reached, all replication pool usage warning messages are suppressed until the message interval specified by the RPWARNINTERVAL parameter has passed.

This parameter, along with the RPWARNINTERVAL parameter, is provided to avoid flooding the console with replication pool usage warning messages.

RPWARNPERCENT Parameter: Replication Pool Usage Threshold

Parameter	Specify . . .	Possible Values	Default
`RPWARNPERCENT`	The threshold for replication pool usage (expressed as a percentage of `LRPL`) at which warning messages should be sent. If "0" is specified or taken as the default there are no warning messages	0-99	0

This ADARUN parameter applies to both the Adabas and Event Replicator Server replication pools.

The usage level should be expressed as a percentage. Warning messages will be printed when usage of the replication pool exceeds this percentage (threshold).

V64BIT Parameter: 64-Bit Virtual Storage Option

Note: This parameter is applicable only to z/OS and z/VSE systems.

Parameter	Specify . . .	Possible Values	Default
`V64bit`	whether or not the Adabas nucleus should use virtual storage above the two gigabyte bar (64-bit virtual storage). In order for the nucleus to use 64-bit virtual storage, the operating system must also support 64-bit virtual storage.	YES | NO	NO

This parameter indicates whether or not the Adabas nucleus should use 64-bit virtual storage. The default is NO.

Currently, the following storage areas will be allocated above the two gigabyte bar in an Adabas nucleus:

- The flush I/O pool (see ADARUN parameter `LFIOP`)

- The PLOG protection I/O buffers (see ADARUN parameter `NPLOGBUFFERS`)

- The Work part 1 protection I/O buffers (see ADARUN parameter `NWORK1BUFFERS`)

- The cast-out directory buffer, in cluster environments (automatically derived from the attributes of the global cache space for Adabas Cluster or Adabas Parallel Services).

The use of 64-bit virtual storage will help alleviate virtual storage constraints below the two gigabyte bar in installations that use large amounts of virtual storage in Adabas nucleus sessions (for example, nucleus sessions with ADARUN parameter settings of LBP=1200M and LFIOP=200M).

Value	Meaning
YES	The Adabas nucleus will use 64-bit virtual storage (storage above the two gigabyte bar).
NO	The default setting. The Adabas nucleus will not use 64-bit virtual storage.

Users in z/OS environments should note that virtual storage above the two gigabyte bar (64-bit virtual storage) is shown in dumps to SVCDUMP or SYSMDUMP but not in dumps to MPMDUMP, SYSABEND, or SYSUDUMP. Therefore, we recommend that you use the SVCDUMP in the JCL of an Adabas nucleus running with ADARUN V64BIT=YES to ensure that 64-bit virtual storage is shown in any SVC dumps that are produced. If, for some reason, you do not want to produce SVC dumps, we recommend that you use SYSMDUMP to ensure 64-bit virtual storage is shown in a dump of the Adabas nucleus address space.

Specific Product Recommendations

- The replication pool will be allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters `REPLICATION=YES` and `V64BIT=YES`.

- The replication pool will be allocated in 64-bit virtual storage when the Event Replicator Server runs with ADARUN `V64BIT=YES` parameter.

- The Auditing pool (LAP) will be allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters `AUDITING=YES` and `V64BIT=YES`.

Example

The following example indicates that the Adabas nucleus will make use of virtual storage above the two-gigabyte bar and should use 1M large pages.

```
ADARUN PROG=ADANUC,LARGEPAGE=YES,V64BIT=YES
```

ZIIP Parameter: Activate Usage of Adabas for zIIP

This parameter is valid for z/OS operating environments only.

The ZIIP parameter controls whether Adabas activates Adabas for zIIP (AZPAD).

> **Note:** Adabas for zIIP requires its own license file (AZPAD), in addition to the standard Adabas license file. If the AZPAD license file is missing or found invalid, Adabas will start but will run as if ZIIP=NO had been specified.

Parameter	Specify . . .	Possible Values	Default
ZIIP	whether or not to activate Adabas for zIIP.	YES | NO	NO

Value	Meaning
YES	Usage of Adabas for zIIP is enabled. Adabas offloads CPU time from the general processors to System z Integrated Information Processors (zIIP). The purpose of this configuration is to reduce the CPU consumption on the general processors.
NO	The default setting. Usage of Adabas for zIIP is disabled. Adabas runs without the option to offload CPU time to zIIPs.

Dynamic Modification

If Adabas was started with ZIIP=YES, the setting of the ZIIP parameter can be changed at runtime - that is, set to NO and subsequently back to YES - using the from the operator console, the ZIIP function, or the Modify Parameters menu of Adabas Online System.

If Adabas was started with ZIIP=NO, the setting of the ZIIP parameter cannot be changed to YES later in this session.

ZIIP may be set differently for different nuclei in a cluster (it is a local, modifiable parameter).

Specific Product Recommendations

- Event Replicator Server databases and Adabas nuclei running with ADARUN `REPLICATION=YES` support the use of Adabas for zIIP subject to the availability of the appropriate license file (AZPAD).

- The Adabas Audit Server and Adabas nuclei running with `ADARUN AUDITING=YES` support the use of Adabas for zIIP subject to the availability of the appropriate license file (AZPAD).

Example

The following example activates the use of Adabas for zIIP.

```
ADARUN PROG=ADANUC,ZIIP=YES
```

Other ADARUN Settings

In addition to the new ADARUN parameters described above, the following ADARUN parameter settings are required in the Event Replicator Server to support the Event Replicator for Adabas:

- When the Event Replicator Server runs with Adabas 8, the LU parameter must be greater than or equal to 167,000.

- Set 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).

■ Set parameter NC to a value greater than or equal to:

```
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 NC parameter greater than or equal to 10 (for example NC=20).

■ The NT parameter must be greater than or equal to 15.

■ When the SLOG file is defined for use on the Event Replicator Server, replication data may be written and read from the SLOG file. With this in mind, consider the values set for the following ADARUN performance and tuning parameters: ASYTVS, FMXIO, LBP, LDEUQP, LFIOP, LP, NH, NISNHQ.

> **Notes:**

1. If data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work parameters LU , NAB , and NC must also be set as described above.

2. If NPADACALLS is set to a value greater than "1", you should make sure that the following ADARUN parameters are set appropriately for the target Adabas nucleus, ensuring the target nucleus can handle updates by multiple concurrent users: LBP , LDEUQP , LFIOP , LFP , LP , LWP , NAB , NC , NH , NT , and NU .

4 Utilities Used with Replication

Various Adabas utilities and specific utility functions are used in replication. Some of these utilities and functions apply only to the Adabas database from which data is being replicated, some of them apply only to the Event Replicator Server processing the replicated data, and some of them apply to both the Adabas database and the Event Replicator Server.

The following utilities and utility functions apply specifically to the Adabas database:

- *ADADBS MODFCB Parameters*
- *ADADBS REPLICATION Function*
- *ADADEF DEFINE REPTOR Function*
- *ADADEF MODIFY REPTOR Function*
- *ADALOD LOAD Parameters*
- *ADALOD UPDATE Parameters*
- *ADAORD REORASSO Parameters*
- *ADAREP Report Description*
- *ADARES Utility Replication Functions*
- *ADASAV Utility Replication Functions*
- *ADASCR Utility Replication Functions*

The following utilities and utility functions apply specifically to the Event Replicator Server:

- *ADADBS OPERCOM Function*
- *ADADBS REPTOR Function*
- *ADADEF DEFINE REPTOR Function*
- *ADADEF MODIFY REPTOR Function*
- *ADALOD LOAD Parameters*

- *ADAORD STORE Parameters*
- *ADAREP Report Description*
- *ADARIS Utility: Initial-State*
- *ADARPE Utility: Extract TLOG Records*
- *ADARPL Utility: PLOG Replication Replay*
- *ADARPP Utility: Print TLOG Records*
- *RPULD and RPLOD Utilities*

In addition to the utilities described in this chapter, Event Replicator for Adabas provides two additional utilities, ADAMTR and ADAMTS, that you can use to test your webMethods EntireX and WebSphere MQ queues. For more information, read *Testing webMethods EntireX and WebSphere MQ Queues*, in *Event Replicator for Adabas Application Programmer's Reference Guide*.

5   ADADBS Utility Replication Functions

Two separate ADADBS utility functions impact replication processing for an Adabas database or for an Event Replicator Server:

- The **REPLICATION function** controls whether replication is activated for an Adabas database file. This function should be used for Adabas database files, and *not* with Event Replicator Server files.

- The **REPTOR function** to activate and deactivate specific DBID, destination, and subscription definitions in the Replicator system file (Event Replicator Server) and to open or close destination and IQUEUE definitions in the Replicator system file.

In addition, the **ADADBS MODFCB function** includes a parameter that can be specified to indicate whether an Adabas database file may be updated only by the Event Replicator Server as part of Adabas-to-Adabas replication or by other means as well.

ADADBS MODFCB Parameters

The `RPLUPDATEONLY` parameter can be used in the ADADBS MODFCB function to indicate whether an Adabas database file may be updated only by the Event Replicator Server as part of Adabas-to-Adabas replication or by other means as well. This parameter is optional. Valid values are "YES" or "NO". A value of "YES" indicates that the file can only be updated via Event Replicator processing; a value of NO indicates that the file can be updated by any normal means, including Event Replicator processing. There is no default; if no value is specified for the `RPLUPDATEONLY` parameter in the ADADBS MODFCB function, the value used previously for the file is used.

ADADBS OPERCOM Parameters

This section covers the following topics:

- DRPLSTAT
- RPLCHECK

DRPLSTAT

Use `DRPLSTAT` to display the replication-related statistics for an Adabas database (with replication turned on) or for an Event Replicator Server. When issued against an Adabas database (with replication turned on), the statistics listed include:

- The total number of replication transactions completely processed.

- The current number of pending replicated transactions (transactions that have been committed, but not yet processed)

- The current number of incomplete transactions that will be replicated (but are not yet committed).

When issued against an Event Replicator Server, the statistics related to destinations, global values, and subscriptions in the database are listed. Replay Utility (ADARPL) statistics are also included.

RPLCHECK

The `RPLCHECK` command can be used in the ADADBS OPERCOM function to perform the replication cross-check function for all active databases known (defined in one or more subscriptions) to the Event Replicator Server. When this command is run using the ADADBS OPERCOM function, the information about the cross-check function is printed to the ADADBS DDDRUCK data set. The information printed by ADADBS is the same as the information printed by the Event Replicator Server during the cross-check process initiated by the RPLCHECK operator command.

> **Note:** Not valid for use with an Adabas nucleus; only valid for use against an Event Replicator Server. If this command is issued against a database that is not an Event Replicator Server, error messages result.

ADADBS REPLICATION Function

The ADADBS REPLICATION function activates or deactivates replication for an Adabas database file. This function should be used with Adabas databases; it is not valid with Event Replicator Servers.

This section covers the following topics:

- Required Parameters
- Optional Parameters
- Examples

Required Parameters

FILE: Database File

    `FILE` is the file number of the database file for which replication is to be activated or deactivated. This parameter is required; there is no default.

> **Note:** Replication may not be turned on for an Adabas system file or for a ciphered file.

ON | OFF | MODIFY: Set Replication

    One of the parameters `ON`, `OFF`, or `MODIFY` must be set. There is no default. Each parameter is described below:

- `ON` turns replication on for the specified file.

> **Note:** Replication may not be turned on for a file that allows spanned data storage records (i.e. a file compressed with the ADACMP COMPRESS option SPAN).

- `OFF` turns replication off for the specified file.

- `MODIFY` allows you to modify one or more of the replication-related values (KEY, TARGET, DSBI) for a replicated file. The `ACTIVATE` and `DEACTIVATE` parameters allow you to change the replication status for a replicated file.

Optional Parameters

ACTIVATE: Activate Replication

The `ACTIVATE` parameter is only allowed when MODIFY is specified. It will activate replication for a file that has replication defined. Replication must already be inactive for the file.

DEACTIVATE: Deactivate Replication

The `DEACTIVATE` parameter is only allowed when MODIFY is specified. It will deactivate replication for a file that has replication defined. Replication must already be active for the file.

DSBI: Define Before Image

The `DSBI` parameter defines whether or not before images of data storage are collected for replication during the update of a record on a file. Following are rules for setting the DSBI parameter:

- Possible values: "ON", "OFF"

- The parameter is optional when "ON" or "MODIFY" is specified.

- The parameter has a default value of "ON" when `ADADBS REPLICATION ON` is specified.

- The parameter is not allowed when `ADADBS REPLICATION OFF` is specified.

For more information about how this parameter is used in Adabas database processing during replication, read *Detailed Adabas Nucleus Processing* in *Event Replicator for Adabas Concepts*.

KEY: Define Primary Key

The `KEY` parameter defines the primary key for replication. Following are rules for setting the `KEY` parameter:

- The parameter is optional when "ON" or "MODIFY" is specified.

- The parameter is not allowed when "OFF" is specified.

- The parameter may *not* be specified when "OFF" or `NOKEY` is specified.

> **Note:** The field name specified must be a descriptor on the file. Note that "descriptor" in this case is used generically, as the field may be a descriptor, subdescriptor, superdescriptor, etc.

NOKEY: Remove Primary Key Setting

The `NOKEY` parameter removes the primary key setting when MODIFY is specified. Following are rules for setting the `NOKEY` parameter:

- The parameter is not allowed when "OFF" is specified.

- The parameter is the default setting when "ON" is specified and neither `NOKEY` nor `KEY` is specified.
- The parameter may *not* be specified when `KEY` is specified.

TARGET: Event Replicator Target ID

`TARGET` defines the Event Replicator target database ID. This parameter is required when "ON" is specified, is optional when "MODIFY" is specified, and is not allowed when "OFF" is specified.

Examples

```
ADADBS REPLICATION FILE=33,ON,TARGET=206,KEY=AA
```

or

```
ADADBS REPLICATION FILE=70,OFF
```

or

```
ADADBS REPLICATION FILE=71,MODIFY
ADADBS ACTIVATE
```

or

```
ADADBS REPLICATION FILE=72,MODIFY
ADADBS DEACTIVATE
```

ADADBS REPTOR Function

The ADADBS REPTOR function provides activation and deactivation control of Event Replicator resources. This function should be used with Event Replicator Servers; it is not valid with other Adabas databases.

This section covers the following topics:

- Required Parameters
- Optional Parameters

- Examples

Required Parameters

ACTIVATE | DEACTIVATE | OPEN | CLOSE: Event Replicator Server request

One of the parameters, `ACTIVATE`, `DEACTIVATE`, `OPEN`, or `CLOSE` must be set. There is no default. Each parameter is described below:

- `ACTIVATE` requests activation for the specified file, database ID, destination, or subscription.

- `DEACTIVATE` requests deactivation for the specified file, database ID, destination, or subscription.

- `OPEN` requests that an unavailable destination or input queue be opened. When this parameter is specified, either the `DESTINATION` or `IQUEUE` parameter must be specified.

- `CLOSE` requests that an available destination or input queue be closed. When this parameter is specified, either the `DESTINATION` or `IQUEUE` parameter must be specified.

Optional Parameters

DBID: Replicated DBID

The `DBID` parameter, when specified without the `FILE` parameter, will activate replication for any inactive files or deactivate replication for any active files for a given DBID. When specified with the `FILE` parameter, the `DBID` parameter identifies the database in which the file specified by the `FILE` parameter resides.

The `DBID` parameter is mutually exclusive with the `DESTINATION`, `IQUEUE`, and `SUBSCRIPTION` parameters.

DESTINATION: Replication Destination

The `DESTINATION` parameter can only be specified when the `ACTIVATE`, `DEACTIVATE`, `OPEN`, or `CLOSE` parameters are specified. The `DESTINATION` parameter supplies the name of the destination that should be activated, deactivated, opened, or closed. The destination specified must be defined to the Event Replicator Server.

If "ACTIVATE" is specified, the destination must already be inactive; if "DEACTIVATE" is specified, the destination must already be activated.

If "OPEN" is specified, the destination must be in an unavailable state; if "CLOSE" is specified, the destination must be in an available state.

The `DESTINATION` parameter is mutually exclusive with the `DBID`, `FILE`, `IQUEUE`, and `SUBSCRIPTION` parameters.

FILE: Replicated File

The `FILE` parameter will activate or deactivate replication for a single file in a specific database. A corresponding `DBID` parameter must also be specified.

If "ACTIVATE" is specified, the file must already be deactivated; if "DEACTIVATE" is specified, the file must already be activated.

The `FILE` parameter is mutually exclusive with the `DESTINATION`, `IQUEUE`, and `SUBSCRIPTION` parameters.

IQUEUE: Replication Input Queue

The `IQUEUE` parameter can only be specified when the `OPEN` or `CLOSE` parameters are specified. It supplies the name of the input queue (IQUEUE) that should be opened or closed and must be defined to the Event Replicator Server.

If "OPEN" is specified, the input queue must be in an unavailable state; if "CLOSE" is specified, the input queue must be in an available state.

The `IQUEUE` parameter is mutually exclusive with the `DBID`, `DESTINATION`, `FILE`, and `SUBSCRIPTION` parameters.

SUBSCRIPTION: Replication Subscription

The subscription specified for the `SUBSCRIPTION` parameter must be defined to the Event Replicator Server. It specifies the name of the subscription definition to use.

If "ACTIVATE" is specified, the subscription must already be deactivated. If "DEACTIVATE" is specified, the subscription must already be activated.

The `SUBSCRIPTION` parameter is mutually exclusive with the `DBID`, `FILE`, `IQUEUE`, and `DESTINATION` parameters.

Examples

The following example requests that the Event Replicator Server activate all inactive files for database 232.

```
ADADBS REPTOR ACTIVATE,DBID=232
```

The following example requests that the Event Replicator Server activate file 2 on database 232. The file is currently inactive.

```
ADADBS REPTOR ACTIVATE,DBID=232,FILE=2
```

The following example requests that the Event Replicator Server activate the destination defined by the DEST0001 destination. The destination is currently inactive.

```
ADADBS REPTOR ACTIVATE,DESTINATION=DEST0001
```

The following example requests that the Event Replicator Server activate the subscription defined by the SUBS0001 subscription. The subscription is currently inactive.

```
ADADBS REPTOR ACTIVATE,SUBSCRIPTION=SUBS0001
```

The following example requests that the Event Replicator Server deactivate all active files for database 232.

```
ADADBS REPTOR DEACTIVATE,DBID=232
```

The following example requests that the Event Replicator Server deactivate file 2 on database 232. The file is currently active.

```
ADADBS REPTOR DEACTIVATE,DBID=232,FILE=2
```

The following example requests that the Event Replicator Server deactivate the destination defined by the DEST0001 destination. The destination is currently active.

```
ADADBS REPTOR DEACTIVATE,DESTINATION=DEST0001
```

The following example requests that the Event Replicator Server deactivate the subscription defined by the SUBS0001 subscription. The subscription is currently active.

```
ADADBS REPTOR DEACTIVATE,SUBSCRIPTION=SUBS0001
```

The following example requests that theEvent Replicator Server open (restart) destination DEST0001. Each output task will be asked to process the destination if that output task does not already have the destination open for processing.

```
ADADBS REPTOR OPEN,DESTINATION=DEST0001
```

The following example requests that the Event Replicator Server close destination DEST0001. Each output task will be asked to close the destination if that output task does not already have the destination closed for processing.

```
ADADBS REPTOR CLOSE,DESTINATION=DEST0001
```

The following example requests that the Event Replicator Server open (restart) input queue IQUEUE01. Each input task will be asked to open the input queue if that input task does not already have the input queue open for processing.

```
ADADBS REPTOR OPEN,IQUEUE=IQUEUE01
```

The following example requests that the Event Replicator Server close input queue IQUEUE01. Each input task will be asked to close the input queue if that input task does not already have the input queue closed for processing.

```
ADADBS REPTOR CLOSE,IQUEUE=IQUEUE01
```

6 ADADEF Utility Replication Functions

Two separate ADADEF utility functions impact replication processing for an Adabas database or for an Event Replicator Server:

- The **DEFINE REPTOR function** controls whether the database you are defining is an Adabas database or an Event Replicator Server.

- The **MODIFY REPTOR function** specifies whether the database that is running is an Adabas database or an Event Replicator Server.

ADADEF DEFINE REPTOR Function

The `REPTOR` parameter must be specified for the ADADEF DEFINE function when using Replication.

REPTOR: Set the Event Replicator Server

The `REPTOR` parameter is used with ADADEF DEFINE to specify whether a normal Adabas database is being defined, or whether an Event Replicator Server is being defined. Possible values are "YES" or "NO" (default). The format is `ADADEF DEFINE REPTOR=YES`.

Specify `REPTOR=NO` for a normal Adabas database. Specify `REPTOR=YES` for an Event Replicator Server.

ADADEF MODIFY REPTOR Function

The `REPTOR` parameter is used with ADADEF MODIFY function when using Replication.

REPTOR: Set the Event Replicator Server

The `REPTOR` parameter is used with ADADEF MODIFY to indicate whether the database that is running is an Adabas database or an Event Replicator Server. Possible values are "YES" or "NO" (default). The format is `ADADEF MODIFY REPTOR=YES`.

For a normal Adabas database, the `REPTOR` parameter must always be set to "NO". When an Event Replicator Server is running, the `REPTOR` parameter must be set to "YES".

7 ADALOD Utility Replication Functions

Several ADALOD LOAD parameters can be specified for the Adabas database and for the Event Replicator Server when using replication. These are described in this chapter. In addition, the ADALOD UPDATE RPLLOAD parameter can be specified for the Adabas database files when using replication.

ADALOD LOAD Parameters for the Adabas Database

Several ADALOD LOAD parameters can be specified for the Adabas database files when using replication. The replication-related parameters are only allowed for ADALOD LOAD if replication is already turned on for a database.

- RPLDSBI: Before Image for Data Storage
- RPLERRORDEACTFILE: Deactivate Replication for File on Error
- RPLINITERROR: Replication Control on Error
- RPLKEY: Primary Key for Replication
- RPLLOAD: Replicate Load Data
- RPLTARGETID: Replication Target ID
- RPLUPDATEONLY: Allow Only Event Replicator Processing Updates
- Examples

> **Note:** The version of Event Replicator specified in an ADALOD LOAD or UPDATE utility job that specifies RPLLOAD=YES must match the version of Event Replicator used by the Event Replicator Server. In addition, the Adabas version used by the Event Replicator Server must be greater than or equal to the Adabas version used in an ADALOD LOAD or UPDATE utility job that specifies RPLLOAD=YES.

RPLDSBI: Before Image for Data Storage

The RPLDSBI parameter turns on the collection of before images of data storage during an update command to the file. Parameter RPLDSBI may only be specified if the RPLTARGETID parameter is also specified. Specify RPLDSBI to turn on the collection of data storage before images during an update. For more information about how this setting is used in Adabas database processing during replication, read *Adabas Nucleus Replication Setup* and *Detailed Adabas Nucleus Processing* in *Event Replicator for Adabas Concepts*.

The RPLDSBI parameter may not be specified in the same ADALOD LOAD run as the REPLICATOR parameter.

RPLERRORDEACTFILE: Deactivate Replication for File on Error

RPLERRORDEACTFILE = [NO | YES]

Use the RPLERRORDEACTFILE parameter to deactivate replication for the file for which replication errors occurred during ADALOD LOAD processing. RPLERRODEACTFILE controls ADALOD LOAD behavior for replication initialization errors as well as other replication errors (in contrast with the RPLINITERROR parameter which affects ADALOD LOAD behavior only for replication initialization errors).

> **Note:** The RPLERRORDEACTFILE parameter can only be specified if the RPLLOAD parameter is set to "YES" or "FILE".

Valid settings for RPLERRORDEACTFILE are "YES" and "NO":

- Specifying "YES" indicates that replication for the file should be deactivated when a replication error occurs.
- Specifying "NO" (the default) indicates that replication for the file should not be deactivated.

> **Caution:** If you elect to have ADALOD LOAD continue its processing after a replication error, you are also responsible for recovering your environment appropriately.

Once replication is deactivated for a file it can only be reactivated using Adabas Online System (AOS) or the ADADBS utility.

RPLINITERROR: Replication Control on Error

RPLINITERROR = [<u>FAIL</u> | CONTINUE]

Use the RPLINITERROR parameter to indicate whether ADALOD processing should continue if replication initialization errors occur.

> **Note:** The RPLINITERROR parameter can only be specified if the RPLLOAD parameter is set to "YES" or "FILE".

Valid settings for RPLINITERROR are "FAIL" and "CONTINUE":

- Specifying "FAIL" (the default) indicates that ADALOD processing should fail if a replication initialization error occurs. This is the how ADALOD LOAD works now, before the enhancement described in this enhancement preview is applied.

- Specifying "CONTINUE" indicates that ADALOD LOAD processing should continue if a replication initialization error occurs.

> **Caution:** If you elect to have ADALOD LOAD continue its processing after a replication error, you are also responsible for recovering your environment appropriately.

RPLKEY: Primary Key for Replication

The `RPLKEY` parameter specifies the primary key for replication. This parameter may only be specified if the `RPLTARGETID` parameter is also specified. The format is RPLKEY='*primary-key-for-replication*'. For more information about how this primary key is used in Adabas database processing during replication, read *Adabas Nucleus Replication Setup* and *Detailed Adabas Nucleus Processing* in *Event Replicator for Adabas Concepts*.

The `RPLKEY` parameter may not be specified in the same ADALOD LOAD run as the `REPLICATOR` parameter.

RPLLOAD: Replicate Load Data

The ADALOD LOAD RPLLOAD parameter can be specified for the Adabas database files when using replication. It is only allowed for ADALOD LOAD if replication is already turned on for a database.

> **Note:** The version of Event Replicator specified in an ADALOD LOAD utility job that specifies RPLLOAD=YES must match the version of Event Replicator used by the Event Replicator Server. In addition, the Adabas version used by the Event Replicator Server must be greater

than or equal to the Adabas version used in an ADALOD LOAD utility job that specifies RPLLOAD=YES.

The `RPLLOAD` parameter indicates whether or not data, and possibly the FCB/FDT, loaded to the Adabas database via the ADALOD LOAD utility will be replicated to the Event Replicator Server. Valid values are "YES", "FILE", and "NO"; the default is "NO":

- When RPLLOAD=YES is specified, ADALOD LOAD replicates data to the Event Replicator Server that it loads to the Adabas database.

- When RPLLOAD=FILE is specified, ADALOD LOAD replicates the FCB/FDT and data to the Event Replicator Server. When the data is replicated to an Adabas destination, the file is first allocated on the target database. The DESTINATION parameters DREPLICATEUTI for the target destination and DAREPLICATEUTI for the target file must be set to YES to use this option.

- When RPLLOAD=NO is specified, ADALOD LOAD does not replicate its load data to the Event Replicator Server.

> **Caution:** If ADALOD LOAD ends abnormally (due to insufficient space, for example), updates made to the file before the abnormal ending cannot be backed out; there is no automated recovery for the updated data or for the replicated data. Software AG therefore recommends that you perform an ADASAV SAVE on the file before you run ADALOD LOAD.

The ADALOD LOAD RPLLOAD parameter may not be set for a file with spanned DS record support.

RPLTARGETID: Replication Target ID

The `RPLTARGETID` parameter specifies the Event Replicator target ID used when the Adabas file data is replicated. The format is RPLTARGETID='*reptor-target-id*'.

The `RPLTARGETID` parameter may not be specified in the same ADALOD LOAD as the `REPLICATOR` parameter. For more information about how this target ID is used in Adabas database processing during replication, read *Adabas Nucleus Replication Setup* in *Event Replicator for Adabas Concepts*.

> **Note:** Replication may not be turned on for an Adabas system file or a ciphered file.

RPLUPDATEONLY: Allow Only Event Replicator Processing Updates

The `RPLUPDATEONLY` parameter can be used in the ADALOD LOAD function to indicate whether an Adabas database file may be updated only by the Event Replicator Server as part of Adabas-to-Adabas replication or by other means as well. This parameter is optional. Valid values are "YES" or "NO". A value of "YES" indicates that the file can only be updated via Event Replicator processing; a value of NO indicates that the file can be updated by any normal means, including Event Replicator processing.

If the file is a new file, the default for this parameter is "NO".

However, if the file specified in the ADALOD LOAD function is an existing file, there is no default for this parameter. If no value is specified for the `RPLUPDATEONLY` parameter in the ADALOD LOAD function for an existing file, the value used previously for the file is used.

Examples

```
ADALOD LOAD FILE=60,RPLTARGETID=13,RPLKEY=AA,RPLDSBI
```

ADALOD LOAD Parameters for the Event Replicator Server

Several ADALOD LOAD parameters can be specified for the Event Replicator Server when using replication. These are described in this section:

- REPLICATOR: Load an Adabas Replicator System File
- SLOG: Load an Adabas SLOG File
- Examples

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

REPLICATOR: Load an Adabas Replicator System File

The `REPLICATOR` parameter is a file type parameter. Use this parameter to load the Replicator system file into the Event Replicator Server.

The Replicator system file stores the Event Replicator initialization parameters. When it is loaded into the Event Replicator Server, it can be read during Event Replicator Server startup. You can modify the initialization parameters in the Replicator system file using the Adabas Event Replicator Subsystem. If the Replicator system file cannot be found at startup, the Event Replicator initialization parameters are read from the DDKARTE statements of the Event Replicator Server startup job.

The `REPLICATOR` parameter may not be specified when loading a file on a database that is not an Event Replicator Server. The contents of DD/EBAND are ignored when loading a Replicator system file. For more information about Adabas system files, read about the FILE parameter of the ADALOD LOAD function in your Adabas utilities documentation (the *Adabas Utilities Manual*).

The `REPLICATOR` parameter may not be specified in the same ADALOD LOAD as the `RPLTARGETID`, `RPLKEY`, and, `RPLDSBI` parameters.

SLOG: Load an Adabas SLOG File

The `SLOG` parameter is a file type parameter. Use this parameter to load the SLOG system file into the Event Replicator Server.

An SLOG file is an Adabas system file used only by the Event Replicator Server to store subscription logging data. The `SLOG` parameter may not be specified when loading a file on a database that is not an Event Replicator Server. The contents of DD/EBAND are ignored when loading an SLOG file.

Examples

```
ADALOD LOAD FILE=89,REPLICATOR
```

or

```
ADALOD LOAD FILE=40,SLOG
```

ADALOD UPDATE Parameters

The ADALOD UPDATE RPLLOAD parameter can be specified for the Adabas database files when using replication. It is only allowed for ADALOD UPDATE if replication is already turned on for a database.

> **Note:** The version of Event Replicator specified in an ADALOD UPDATE utility job that specifies RPLLOAD=YES must match the version of Event Replicator used by the Event Replicator Server. In addition, the Adabas version used by the Event Replicator Server must be greater than or equal to the Adabas version used in an ADALOD UPDATE utility job that specifies RPLLOAD=YES.

The `RPLLOAD` parameter indicates whether or not inserts and deletes to the Adabas database via the ADALOD UPDATE utility will be replicated to the Event Replicator Server. Valid values are "YES" and "NO"; the default is "NO".

When RPLLOAD=YES is specified, ADALOD UPDATE inserts and deletes to the Adabas database will be replicated to the Event Replicator Server. When RPLLOAD=NO is specified, ADALOD UPDATE inserts and deletes are *not* replicated.

> **Caution:** If ADALOD UPDATE ends abnormally (due to insufficient space, for example), updates made to the file before the abnormal ending cannot be backed out; there is no automated recovery for the updated data or for the replicated data. Software AG therefore recommends that you perform an ADASAV SAVE on the file before you run ADALOD UPDATE.

8 ADAORD Utility Replication Functions

Several ADAORD utility function parameters can be specified for the Adabas database and for the Event Replicator Server when using replication. These are described in this chapter.

ADAORD REORASSO Parameters for the Adabas Database

The `RPLUPDATEONLY` parameter can be used in the ADAORD REORASSO function to indicate whether an Adabas database file may be updated only by the Event Replicator Server as part of Adabas-to-Adabas replication or by other means as well. This parameter is optional. Valid values are "YES" or "NO". A value of "YES" indicates that the file can only be updated via Event Replicator processing; a value of NO indicates that the file can be updated by any normal means, including Event Replicator processing. There is no default; if no value is specified for the `RPLUPDATEONLY` parameter in the ADAORD REORASSO function, the value used previously for the file is used.

ADAORD STORE Parameters for the Adabas Database

Several ADAORD STORE parameters can be specified for the Adabas database files when using replication. The replication-related parameters are only allowed for ADAORD STORE if replication is already turned on for a database. These are described in this section:

The replication-related parameters for ADAORD STORE are described below:

- RPLDSBI: Before Image for Data Storage
- RPLKEY: Primary Key for Replication
- RPLTARGETID: Replication Target ID
- RPLUPDATEONLY: Allow Only Event Replicator Processing Updates
- Examples

RPLDSBI: Before Image for Data Storage

The `RPLDSBI` parameter indicates whether the collection of before images of data storage during an update command to the file should be done if replication is turned on for the file.

> **Note:** This parameter can only be specified if replication is turned on for the file.

If RPLDSBI is not specified, the default value depends on the target database ID (DBID) of the RESTORE and its replication state; if the ADAORD STORE DBID is the same as the original RE-STRUCTURE DBID and REPLICATION=YES in the target DBID, the original RPLDSBI value is used. Otherwise, RPLDSBI is set to NO.

For more information about how this setting is used in Adabas database processing during replication, read *Adabas Nucleus Replication Setup* and *Detailed Adabas Nucleus Processing* in *Event Replicator for Adabas Concepts*.

RPLKEY: Primary Key for Replication

The `RPLKEY` parameter specifies the primary key for replication. The format is RPLKEY='*primary-key-for-replication*'. When a valid descriptor name is specified for RPLKEY, the specified descriptor is used as the primary replication key for the corresponding file. When OFF is specified for RPLKEY, no primary key is used for replication (RPLKEY is turned off) for the corresponding file.

> **Note:** This parameter can only be specified if replication is turned on for the file, and if the provided descriptor name is a valid Adabas descriptor.

If RPLKEY is not specified, the default value depends on the target database ID (DBID) of the RESTORE and its replication state; if the ADAORD STORE DBID is the same as the original RE-STRUCTURE DBID and REPLICATION=YES in the target DBID, the original RPLKEY value is used. Otherwise, RPLDSBI is set to OFF.

For more information about how this primary key is used in Adabas database processing during replication, read *Adabas Nucleus Replication Setup* and *Detailed Adabas Nucleus Processing* in *Event Replicator for Adabas Concepts*.

RPLTARGETID: Replication Target ID

The `RPLTARGETID` parameter specifies the Event Replicator target ID used when the Adabas file data is replicated. The format is RPLTARGETID='*reptor-target-id*'. When a valid target ID is specified for RPLTARGETID, the specified target is used as the target for replicated data for the corresponding file. When OFF or 0 (zero) are specified for RPLTARGETID, no target is used for replication for the corresponding file.

If RPLTARGETID is not specified, the default value depends on the target database ID (DBID) of the RESTORE and its replication state; if the ADAORD STORE DBID is the same as the original RESTRUCTURE DBID and REPLICATION=YES in the target DBID, the original RPLTARGETID value is used. Otherwise, RPLTARGETID is set to OFF.

For more information about how this target ID is used in Adabas database processing during replication, read *Adabas Nucleus Replication Setup* in *Event Replicator for Adabas Concepts*.

> **Note:** Replication may not be turned on for an Adabas system file, a ciphered file, or a file that allows spanned data storage records (i.e. a file compressed with the ADACMP COM-PRESS option SPAN).

RPLUPDATEONLY: Allow Only Event Replicator Processing Updates

The `RPLUPDATEONLY` parameter can be used in the ADAORD STORE function to indicate whether an Adabas database file may be updated only by the Event Replicator Server as part of Adabas-to-Adabas replication or by other means as well. This parameter is optional.

Valid values are "YES" or "NO". A value of "YES" indicates that the file can only be updated via Event Replicator processing; a value of NO indicates that the file can be updated by any normal means, including Event Replicator processing.

If the file is a new file, the default for this parameter is "NO".

However, if the file specified in the ADAORD STORE function is an existing file, there is no default for this parameter. If no value is specified for the `RPLUPDATEONLY` parameter in the ADAORD STORE function for an existing file, the value used previously for the file is used.

Examples

```
ADAORD STORE FILE=60,RPLTARGETID=13,RPLKEY=AA,RPLDSBI
```

ADAORD STORE Parameters for the Event Replicator Server

The `REPLICATOR` and `SLOG` parameters are used when the ADAORD STORE processes Replicator system files.

The replication-related parameters for ADAORD STORE are described below:

- REPLICATOR: Store the Replicator System File
- SLOG: Store the SLOG System File
- Examples

REPLICATOR: Store the Replicator System File

Use this parameter to store the Replicator system file from the DDFILEA/FILEA tape as the new Replicator system file for the database. The new Replicator system file must have the same file number as the old Replicator system file.

When the `REPLICATOR` parameter is not specified, the Replicator system file on the DDFILEA/FILEA tape is not stored in the database, even if it is specified by a `FILE` or `ALLFILES` parameter.

SLOG: Store the SLOG System File

Use this parameter to store the SLOG system file from the DDFILEA/FILEA tape as the new SLOG file for the database. The new SLOG file must have the same file number as the old SLOG file.

When the SLOG parameter is not specified, the SLOG system file on the DDFILEA/FILEA tape is not stored in the database, even if it is specified by a FILE or ALLFILES parameter.

Examples

```
ADAORD STORE ALLFILES,SLOG
```

or

```
ADAORD STORE FILE=89,REPLICATOR,OVERWRITE
```

9 ADAREP Report Description

The ADAREP utility displays information about the status of replication for the database, files, and the Event Replicator Server. This chapter describes the status information for the Event Replicator Server appearing on the report.

General Information Section

The general information section includes the following when replication is active:

```
Replication            = Yes
```

It includes the following when replication is *not* active:

```
Replication            = No
```

When a Replicator system file is loaded, the general information section includes the following:

```
Replicator File        = 89
```

When an SLOG system file is loaded, the general information section includes the following:

```
Reptor SLOG File       =        40
```

File Information Section

The file information section includes the following when a file is replicated with a primary key defined:

```
File is replicated:
  Replicator target ID            =     206
  Collect before images of updates   = Yes
  Primary key of replicated records  = AA     ↵
```

It includes the following when a file is replicated *without* a primary key defined:

```
File is replicated:
  Replicator target ID            =     206
  Collect before images of updates   = Yes     ↵
```

Checkpoint Information Section

Two new replication-related checkpoints are provided if the `CPLIST` or `CPEXLIST` parameters are specified:

Type	Name	Originator	Description
5E	SYNS	ADADBS	ADADBS REPTOR function
6E	SYNS	ADADBS	ADADBS REPLICATION function

10 ADASAV Utility Replication Functions

The following ADASAV RESTONL and RESTORE utility functions can be specified for the Adabas database and for the Event Replicator Server when using replication: FILES, FMOVE, and GCB. The parameters provided for use in replicating restored data using these utility functions are:

- RPLACTIVE is an optional parameter that specifies the inactive flag setting for a file during restore processing. Valid inactive flag settings are "YES", "NO", or no setting at all. A setting of "YES" turns *off* the replication inactive flag for a file. A setting of "NO" turns *on* the replication inactive flag.

  If no setting is specified, the default value is used. The default depends on the target database ID of the restore processing and its replication state. If *any of the following conditions* are met, the default is "YES"; otherwise the default is "NO":

  - The original replication is turned off.
  - The restore DBID is not the same as the original saved DBID.
  - The original replication target ID has been changed.
  - The original replication-before-image has been changed.
  - The replication primary key has changed.
  - The original replication is turned off.

  When restoring more than one file, the correct RPLACTIVE settings must be specified as positional values corresponding to the file numbers' positions in the FILES parameter file list.

- RPLDATA is an optional parameter that indicates whether the data in a file should be replicated to the replication target ID (RPLTARGETID parameter).

  Valid replication settings are "YES", "NO", "CREATE", or no setting at all. A setting of "YES" causes the restore function to replicate the file data to the replication target during restore processing. A setting of "NO" will not replicate the data to the replication target during restore processing. A setting of CREATE causes the restore function to replicate the file data to the

replication target during restore processing, but also sends a "create file" transaction to the replication target. If no setting is specified, the default "NO" is used.

> **Note:** Values of "YES" or "CREATE" can only be specified if replication is turned on for the corresponding file.

When restoring more than one file in the FILE file list, the RPLDATA settings must be specified as positional values corresponding to the file numbers' positions in the FILES parameter file list.

ADASAV RESTORE with the RPLDATA parameter will not replicate data for a file with spanned DS record support.

■ RPLDSBI is an optional parameter that indicates whether the collection of before images of data storage should occur for an update command to a file.

Valid RPLDSBI settings are "YES", "NO", or no setting at all. A setting of "YES" indicates that the collection of before images of data storage will occur for the file during restore processing. A setting of "NO" indicates that the collection of before images of data storage will *not* occur for the file during restore processing.

If no setting is specified, the default value is used. The default depends on the target database ID of the restore processing and its replication state. If the restore DBID is the same as the originally saved DBID and REPLICATION=YES in the target DBID, the default is "YES"; otherwise the default is "NO".

> **Note:** A values of "YES" can only be specified if replication is turned on for the corresponding file.

When restoring more than one file in the FILE file list, the RPLDSBI settings must be specified as positional values corresponding to the file numbers' positions in the FILES parameter file list.

■ RPLKEY is an optional parameter that specifies the primary key for replication.

Valid RPLKEY settings are a two-character field name, "OFF", or no setting at all. Specifying a field name identifies that field as the primary key for replication. A setting of "OFF" indicates that no primary key should be used for replication.

If no setting is specified, the default value is used. The default depends on the target database ID of the restore processing and its replication state. If the restore DBID is the same as the originally saved DBID and REPLICATION=YES in the target DBID, the original RPLKEY value for the file is used; otherwise the default is "OFF".

> **Note:** A primary key can only be set if replication is turned on for the file and if the field name is a valid Adabas field according to the field definition table (FDT) for the file. When a new RPLKEY is specified it will not be confirmed as a valid Adabas field until the end

of the ADASAV run. At that time, if any RPLKEY is found to be invalid, a warning message is issued, the RPLKEY is set to "OFF", and condition code 8 is returned.

When restoring more than one file in the FILE file list, the RPLKEY settings must be specified as positional values corresponding to the file numbers' positions in the FILES parameter file list.

- RPLTARGETID is an optional parameter that specifies the target ID of the Event Replicator Server to which the restored transactions should be sent. .

  Valid RPLTARGETID settings are a valid target ID, "OFF", or no setting at all. Specifying a target ID identifies that as the target for replication. A setting of "OFF" or "0" indicates that no replication target should be used for replication.

  If no setting is specified, the default value is used. The default depends on the target database ID of the restore processing and its replication state. If the restore DBID is the same as the originally saved DBID and REPLICATION=YES in the target DBID, the original RPLTARGETID value for the file is used; otherwise the default is "OFF".

  > **Note:** A replication target ID can only be specified if replication is turned on for the file.

  When restoring more than one file in the FILE file list, the RPLTARGETID settings must be specified as positional values corresponding to the file numbers' positions in the FILES parameter file list.

- The `RPLUPDATEONLY` parameter can be used in the ADASAV RESTONL function to indicate whether an Adabas database file may be updated only by the Event Replicator Server as part of Adabas-to-Adabas replication or by other means as well. This parameter is optional.

  Valid values are "YES" or "NO". A value of "YES" indicates that the file can only be updated via Event Replicator processing; a value of NO indicates that the file can be updated by any normal means, including Event Replicator processing.

  If no value is specified, the default RPLUPDATEONLY setting of the file at the time of the corresponding SAVE operation is used.

11 ADARES Utility Replication Functions

The ADARES REGENERATE and ADARES BACKOUT functions (including both the BACKOUT SIBA and BACKOUT DPLOG or MPLOG functions) support the `RPLDATA` parameter for your Adabas databases only. This parameter cannot be specified for your Event Replicator Servers.

The optional `RPLDATA` parameter can be used to cause the Adabas nucleus to replicate the protection log data sent to it. Valid values for this parameter are "YES" and "NO"; the default is "NO".

12 ADARIS Utility: Run Initial-State Process

The ADARIS utility, also known as the *Initial-State utility*, can be used to run an initial-state process for a single Adabas file and transfer the resulting initial-state data to a sequential file. This utility reads the data from the Adabas file using the L1 or L4 command, processes the data as requested by the Event Replicator subscription definition specified in the ADARIS utility job, and writes the initial-state data to a sequential file.

The following processing steps occur:

1. The ADARIS utility reads through all records (if no ISNLIST is specified) or only the records specified by the ISNLIST parameter.

2. The selected records are processed according to the rules in the subscription, and the override parameters provided in the ADARIS utility job.

3. The result of the processing is written to the sequential file specified by the DDOUT job control statement.

If the ADARIS utility parameter DCLASS is set to a value of "SAGTARG" for XML, or "SAGTRGB" for Binary raw data format. The resulting sequential output file can be processed by the Event Replicator Target Adapter.

If `DCLASS=SAGTARG` or `DCLASS=SAGTRGB` are *not* specified, the file cannot be used by the Event Replicator Target Adapter. The layout of the data written to a record in the sequential file is the same as the layout of the data written to a message for a webMethods EntireX Broker or WebSphere MQ replication destination. A description of this layout can be found, elsewhere in this chapter.

Support for new Mapping Tool Optimized Global Format Buffers

This release of the Event Replicator Target Adapter Data Mapping Tool introduces a new option that allows the user to optimize a Global Format buffer containing MU's, PE's, and MU's within PE's.

Following restrictions apply:

- The DCLASS specified for this new initial-state subscription must be SAGTARG in the Event Replicator Server system file definition. This can be overridden in the ADARIS utility to specify `DCLASS=SAGTRGB` to produce output in binary raw data format

- The destination must be an webMethods EntireX Broker, MQ Series, or Null destination. In the case of the ADARIS utility the sequential DDOUT file is also allowed.

This chapter describes the ADARIS utility in the following sections:

ADARIS Installation and Prerequisites

This section describes how to install the ADARIS utility, including any prerequisites required before you can install and use the utility. It also describes the prerequisites for using the sequential output file produced by ADARIS with the Event Replicator Target Adapter.

- Prerequisites
- Installation Steps

Prerequisites

The following prerequisites must be met before attempting to use the ADARIS utility:

1. The ADARIS utility requires at least Event Replicator for Adabas Version 3.5.2.

2. The ADARIS utility requires a version of Adabas that is in maintenance support.

3. The sequential output data set produced by the ADARIS utility must have `RECFM=VB`. If you specify the DDOUT control statement in your JCL without any data set attributes, the sequential output data set will be produced using the following attributes:

   - Record format: VB

   - Record length: 32756

   - Block size: 32760

4. You must have the **Show File Source** configuration file preference selected. For more information, refer to the section *Setting Configuration File and Target Adapter Preferences* in the *Event Replicator Target Adapter Administration* documentation.

Installation Steps

To install the ADARIS utility software, copy the contents of the installation medium to disk (as described in the *Event Replicator for Adabas Installation* documentation) and concatenate the load library in your ADARIS JCL.

Running the ADARIS Utility

≫ **To run the ADARIS utility in batch mode:**

1   Set up the ADARIS job. A sample JCL is provided in *Sample JCL*, elsewhere in this section.

      ■ Verify that the following ADARUN parameters are specified in the ADARIS JCL:

```
ADARUN PROG=ADARIS, DBID=dbid, SVC=svc, MODE=MULTI
```

where *dbid* is the Adabas database ID on which the file that is being replicated resides and *svc* is the SVC number to be used for communications with Adabas and the Event Replicator Server. Running this utility with MODE=SINGLE will default to MODE=MULTI.

■ If universal encoding support (UES) is used by the Event Replicator Server and you are using the International Components for Unicode for Software AG (ICS) library for universal encoding service (UES) support, you must include the following additional library in the STEPLIB:

```
//STEPLIB DD DISP=SHR,DSN=SAG.ICSvrs.Lnnn
```

where *nnn* is the load library level.

ICS provides UES support with Adabas Version 8.5 and above.

■ If universal encoding support (UES) is used by the Event Replicator Server and you are using the Software AG internal product library (APS) for universal encoding service (UES) support, you must include the following additional libraries in the STEPLIB:

```
//STEPLIB DD DISP=SHR,DSN=SAG.APSvrs.LDnn
//       DD DISP=SHR,DSN=SAG.APSvrs.LD00
```

where *nn* is the load library level.

You must also add the following additional JCL related to internal product libraries:

```
//DDECSOJ DD DISP=SHR,DSN=ADABAS.ADAvrs.EC00
//SYSPARM DD *
SYSTEM_ID=ADAAPS
ABEND_RECOVERY=NO
THREAD_ABEND_RECOVERY=NO
```

APS provides the default UES support with Adabas Version 8.4 and below.

■ Specify valid values for the parameters of the ADARIS utility. Values for the FILE and SUBSCRIPTION parameter are required; all other parameter values are optional. For more information, read *ADARIS Syntax and Parameters*, elsewhere in this section

If you are going to use the sequential output dataset produced by the ADARIS utility as input to the Event Replicator Target Adapter, be sure to specify DCLASS=SAGTARG or DCLASS=SAGTRGB.

2   Both the Adabas database and the corresponding Event Replicator Server must be active.

3   Run the ADARIS job.

A sequential output file is produced.

Considerations when running the ADARIS utility

When ADARIS is used, its processing of data is done asynchronously to that of the Event Replicator. The data is then delivered to the Event Replicator Target Adapter via a file. To prevent any inconsistencies from occurring, with processing by the Event Replicator, it is important to close the Event Replicator destination(s) that send data to the Event Replicator Target Adapter. The destination(s) should be closed the entire time that the ADARIS utility runs *and* until the output data is processed by the Event Replicator Target Adapter. Once done, the destination(s) can be re-opened.

> **Note:** If the ADARIS utility is canceled via an operator command or an abnormal termination, an operator command `STOPU=job-name` should be issued to stop and delete the user queue element assigned to this job. This will help ensure any ISNs still on hold for this job are freed.

The reason for closing the Event Replicator destination(s) is to ensure that data is processed by the Event Replicator Target Adapter in the correct sequence. Transactions will result in data being accessed randomly, but ADARIS will process data sequentially by descriptor or ISN.

> **Important:** If data, for the same file, is being modified when ADARIS is processing the data, it is important to use the SLOG option for the destination(s). This way no data should be lost for the duration of the ADARIS initial-state.

ADARIS Syntax and Parameters

This is the syntax of the ADARIS utility:

```
ADARIS FILE = fn1, SUBSCRIPTION = sname
              [ DARC = architecture-key ]
              [ DCLASS = name ]
              [ DCLASSPARM = parmdata ]
              [ DEXIT = exitname ]
              [ DEXITWORKSIZE = size ]
              [ DMAXOUTPUTSIZE = n [K] ]
              [ ISNLIST = isn-list ]
              [ HOLD = YES | NO ]
              [ LRPL = poolsize ]
              [ READOPTION = BLANK | I ]
              [ STARTDATE = yyyymmdd [ STARTTIME = hhmmss ] ]
```

Each parameter is described below. Only the FILE and the SUBSCRIPTION parameters are required. All other parameters are optional.

DARC

The DARC parameter is an optional parameter. Use it to define the data architecture for fields in the URB* control structures sent to the destination. Valid values are the integer sum of the byte order and encoding family:

Control Structure Element	Values	Description
Byte order (b)	0	High-order byte first.
	1	Low-order byte first.
Encoding family (e)	0	ASCII encoding family.
	2	EBCDIC encoding family.

The default is 2 (byte order "0" plus encoding family "2").

DCLASS

Use the DCLASS parameter to specify the class type for the destination definition. There are 2 possible values for this parameter: `SAGTARG` or `SAGTRGB`. There is no default.

If you want to use the sequential output data set produced by ADARIS with the Event Replicator Target Adapter, this parameter should be set to `SAGTARG` or `SAGTRGB`.

When set to `SAGTARG` the output produced is in an XML format that can be processed by the Event Replicator Target Adapter. If it is set to `SAGTRGB` the output is in a binary raw data format that can be processed by the Event Replicator Target Adapter.

> **Note:** Only `DCLASS=SAGTARG` can be specified in subscriptions using Optimized Global Format buffers defined in the Event Replicator Server system file. This value can be overridden with `DCLASS=SAGTRGB` for the ADARIS utility executions.

DCLASSPARM

The DCLASSPARM parameter is an optional parameter. Use it to specify up to 120 bytes of character data to be passed to the optional destination output user exit. Any character data is valid for this parameter and there are no defaults.

If either `DCLASS=SAGTARG` or `DCLASS=SAGTRGB` is specified to invoke support for the Event Replicator Target Adapter, you may want to specify one or more of the following keyword parameters.

> **Note:** These parameter keywords must be specified in uppercase and enclosed by single quotes.

ADA

The ADA keyword is available for use with the `SAGTARG` DCLASS exit.

Specify the ADA keyword in the DCLASSPARM parameter if the target of the destination requires the Adabas 2-byte short name be included with the schema for all fields.

NOSPRE

The NOSPRE keyword is available for use with both `SAGTARG` and `SAGTRGB` DCLASS exits.

Specify the NOSPRE keyword in the DCLASSPARM parameter if you do not want the subscription name to prefix the names of the tables produced by the Event Replicator Target Adapter. When NOSPRE is specified, the schema file name (Predict view name) alone is used for the table names; when NOSPRE is not specified, the subscription name prefixes the schema file name in the table names.

> **Note:** Oracle identifiers are limited to 30 characters. If NOSPRE is not specified and an Oracle RDBMS is used by the Event Replicator Target Adapter, the identifier names may exceed 30 characters and errors may occur. We recommend using NO-SPRE if an Oracle RDBMS is also used.

SPRE

Specify the SPRE keyword in the DCLASSPARM parameter if you do want the subscription name to prefix the names of the tables produced by the Event Replicator Target Adapter.

The notation is `SPRE=xxxxxxxx`. Where *xxxxxxxx* can be a subscription prefix to be used instead of the subscription name used in the job execution. The prefix can be 1 to 8 characters in size.

The SPRE keyword is available for use with both `SAGTARG` and `SAGTRGB` DCLASS exits.

OPTIONS

The OPTIONS keyword parameter can be used to specify options for the destination. Specify the OPTIONS keyword parameter, using the syntax `OPTIONS=nnnn`. Possible values of the OPTIONS (*nnnn*) are listed in the following table. However, if you want to combine options, add their values together and enter the total value.

For example, if you want to combine option 32 (to send the full image on an update) and option 64 (to set the XML transaction committed time value to local time instead of GMT/UTC), specify `OPTIONS=96` (with 96 being the sum of 32 and 64).

Option Value	Description
1	This option is no longer supported. If specified, it will be ignored.
2	Specify `OPTIONS=2` to indicate that long names should be used. This option will cause long names to be sent in place of the default short names used for various elements and attributes. Short names are the default and save on the amount of data being transferred. Long names make for better readability. For example, the short name <F> would appear as <Field> using long names.
4	Specify `OPTIONS=4` to ensure that invalid XML characters found in alphanumeric fields are not translated to spaces.

Option Value	Description
8	Specify `OPTIONS=8` to ensure that trailing blanks in alphanumeric fields are not removed.
16	Specify `OPTIONS=16` to ensure that characters used by XML are not replaced automatically with predefined entity references. For example, if `OPTIONS=16` is set, the ampersand (&) character would not be replaced with the literal "&".
32	Specify `OPTIONS=32` to send the full image on an update and insert. The full before image (if available) and after image of all fields are sent for an update and insert, even if the field values were not changed or are null. **Note:** This option is mutually exclusive with option 128. If both options are specified, then `OPTIONS=32` will be set and `OPTIONS=128` ignored.
64	Specify `OPTIONS=64` to set the XML transaction committed time value to local time instead of GMT/UTC.
128	Specify `OPTIONS=128` if the before image is not to be included when a field is updated/changed if the before image data storage records are included. Primary field before images will still be included because the key is needed to find the record that is to be changed. **Note:** This option is mutually exclusive with option 32. If both options are specified, then `OPTIONS=32` will be set and `OPTIONS=128` ignored.
256	Specify `OPTIONS=256` to suppress ARMXL004I information messages.

TRACE

The TRACE keyword parameter can be used to specify the contents of the trace. Specify the TRACE keyword parameter, using the syntax `TRACE=`*nnnn*. Possible values of the TRACE (*nnnn*) are listed in the following table. However, if you want to trace multiple control blocks, add their trace values together and enter the total value. For example, to trace the before and after images of the URBD control blocks, you would specify `TRACE=24` because the sum of 8 (URBD control block before image) and 16 (URBD control block after image) is 24.

> **Note:** If tracing is enabled using this keyword parameter, be sure to include the following JCL statement in the ADARIS JCL: `//DDTRACE1 DD SYSOUT=X`

Settings for DCLASS=SAGTARG

Option Value	Description
1	Trace the URBS control block.
2	Trace the URBT control block.
4	Trace the URBR control block.
8	Trace the URBD control block before image.
16	Trace the URBD control block after image.
32	Trace the URBF/URBG control block before image.

Option Value	Description
64	Trace the URBF/URBG control block after image.
128	Trace the send buffer.
256	Trace the URBY control block.
512	Trace the URBO control block.
1024	Trace the output parameters.
2048	Trace the subscription table.

Settings for DCLASS=SAGTRGB

Option Value	Description
1	Trace the output buffer.
2	Trace the URBD control block after image.
4	Trace the URBR control block.
8	Trace the URBT control block.
16	Trace the URBH control block.
32	Trace the URBF/URBG control block before image.
64	Trace the URBS control block.
128	Trace the URBY control block.
256	Trace the URBO control block.
512	Trace the URBD AI control blocks
1024	Trace the URBF/G AI control blocks
2048	Trace the subscription table.

DEXIT

The DEXIT parameter is an optional parameter. This parameter is not available for general use. At the direction of a Software AG representative, use the DEXIT parameter to specify the name of the destination output user exit. There is no default.

DEXITWORKSIZE

The DEXITWORKSIZE parameter is an optional parameter. This parameter is not available for general use. At the direction of a Software AG representative, use the DEXIT parameter to specify the size of the destination output user exit work area. There is no default.

DMAXOUTPUTSIZE

The DMAXOUTPUTSIZE parameter is an optional parameter. Use this parameter to specify the maximum output size (in bytes) for the destination. Valid values are 0 (zero) or an integer in the range from 4096 through 32752. If no value is specified and the LRECL and BLKSIZE parameters are *not* included in the JCL DD statement for DDOUT, the default value is 32752.

> 📄 **Note:** You can specify the value for this parameter in a purely numeric form or use the K at the end of the number to specify kilobytes. For example, `DMAXOUTPUTSIZE=4K` is the same as `DMAXOUTPUTSIZE=4096`.

The value for this parameter will be used if it is less than or equal to the record length (LRECL) of the output data set.

FILE

The FILE parameter is a required parameter. Use it to specify the number of an input file associated with the subscription identified by the **SUBSCRIPTION** parameter. Valid values are any valid file number. There is no default.

Records from the specified file are replicated according to the specifications in the SUBSCRIPTION.

HOLD

Use this optional parameter to indicate whether the record being replicated should be placed in hold status while it is being replicated. Valid values are "YES" or "NO". The default is "YES".

If ADARIS is executed while users are modifying the related file, run ADARIS with HOLD=YES. If ADARIS is executed while no modifications are being done, ADARIS may be run with HOLD=NO.

See also the explanation of the READOPTION parameter below.

ISNLIST

Use this optional parameter to specify a list of the ISNs or the records to be processed. Valid values include a list of valid ISNs (from 1 - 4294967295), which can also be a range of ISNs. A value of 0 is not allowed. There is no default.

Separate ISN numbers with commas (no blanks) in the list. Use dashes to indicate a range of ISNs forming a range group.

A maximum of 32 such range groups can be specified, where individual ISN's are considered one range group each. For example, `ISNLIST=1-10,32,100-120,150` is considered as 4 range groups where ISN 32 and 150 are considered as one range group each.

LRPL

Use this optional parameter to specify the size of the replication pools. Valid values range from 300,000 bytes to 2,147,483,647 bytes. The default is 2048K.

> 📄 **Note:** You can specify the value for this parameter in a purely numeric form or use the K at the end of the number to specify kilobytes.

READOPTION

This optional parameter has valid values of "BLANK" or "I".

The default is "BLANK", and will cause ADARIS to issue a separate read command for each possible ISN.

Specifying the "I" value will cause ADARIS to set the ACB Command Option 2 to 'I', when reading through a range of ISNs. This may be used with either the HOLD=NO (L1 command) or HOLD=YES (L4 command).

Specifying "I" may improve performance by reducing the number of read commands sent to the nucleus. Reading in ISN sequence, rather than issuing a read command for each specific ISN is generally better, if there are many deleted records or gaps in the ISN ranges for that file.

READOPTION interacts with the HOLD parameter and may affect the records selected if the specified file is being modified while ADARIS is running. The following should be considered when deciding what options to use:

- If no modifications are being applied to the file when ADARIS is executing, you may specify HOLD=NO and READOPTION=I.

- Specify HOLD=YES and READOPTION=BLANK if the file modifications include records being deleted.

- Specify HOLD=YES and READOPTION=I if the file is being modified and no records are being deleted.

STARTDATE

This optional parameter is used in conjunction with STARTTIME to provide information for the Event Replicator Target Adapter about the date the destination used for sending update transactions for a DBID/FNR was closed and the transactions written to SLOG.

The date must be entered in the form `YYYYMMDD` with the value for `YYYY` being equal or greater than 2016.

STARTTIME

This optional parameter is used in conjunction with STARTDATE to provide information for the Event Replicator Target Adapter about the time the destination used for sending update transactions for a DBID/FNR was closed and the transactions written to SLOG.

The time must be entered in the form `HHMMSS`. Valid values range from `000000` to `235959`.

SUBSCRIPTION

The SUBSCRIPTION parameter is a required parameter. Use it to specify the name of the subscription to which the initial-state data is to be replicated. Valid values are any alphanumeric subscription names (1 - 8 characters long). The subscription must have been previously-defined in the Replicator system file. There is no default.

When an Optimized Global format buffer is used, a separate subscription for initial-state only should be used, that specifies the Optimized Global format buffer.

Examples

This section provides examples of ADARIS parameter usage.

Example 1: Using an ISN List (ISNLIST)

In the following example, records in file 2 with ISNs specified in the ISNLIST parameter will be read from the database and replicated to the subscription named SUBS. Then the records will be written to the sequential file.

```
ADARIS FILE=2
       SUBSCRIPTION=SUBS
       ISNLIST=1-100,142,200-300,1000-100000
```

The records with ISNs of 142 or in the following ranges are processed:

1-100
200-300
1000-100000

Example 2: Producing Output for the Event Replicator Target Adapter

In the following example, all the ISNs of file 2 will be read from the database and replicated to the subscription named SUBS. The sequential output file produced for the run can be used as input to the Event Replicator Target Adapter.

```
ADARIS FILE=2
       SUBSCRIPTION=SUBS
       DCLASS=SAGTARG
```

Sample JCL

For z/OS environments

The following sample JCL could be used to run the ADARIS utility in z/OS environments:

```
//ADARIS    JOB
//*
//*
//RIS      EXEC PGM=ADARUN
//STEPLIB  DD DISP=SHR,DSN=ARF.Vvrs.LOAD   <=== Reptor load lib
//         DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD  <=== Adabas load lib
//DDOUT    DD DISP=(NEW,CATLG,CATLG)
//         SPACE=(TRK,(4500,1500,RLSE)UNIT=SYSDA,
//         DSN=<dataset.name>
//DDDRUCK  DD SYSOUT=X
//DDPRINT  DD SYSOUT=X
```

```
//SYSUDUMP DD SYSOUT=X
//DDCARD   DD *
ADARUN PROGRAM=ADARIS,DBID=dbid,SVC=svc,MODE=MULTI,DEVICE=3390
//DDKARTE  DD *
ADARIS SUBSCRIPTION=subscription-name,FILE=fnr
//                                              ↵
```

Use the DDOUT control statement to identify the name of the sequential output data set in which the replicated initial-state data should be stored.

The sequential output data set produced by the ADARIS utility must have `RECFM=VB`. If you specify the DDOUT control statement in your JCL without any data set attributes, the sequential output data set will be produced using the following attributes:

- Record format: VB

- Record length: 32756

- Block size: 32760

For BS2000 environments

In a BS2000 environment the following should be taken into consideration:

- The file identified by the DDOUT link name should have a variable blocked record format:

```
/   CREATE-FILE    F-NAME = ADARIS.OUT, SPACE=(480,48)
/   ADD-FILE-LINK  LINK = DDOUT,   F-NAME = ADARIS.out,        -
/                  RECORD-FORMAT=*VARIABLE,                    -
/                  RECORD-SIZE  = 32750 ,                      -
/                  BUFFER-LENGTH=*STD, SIZE=16
```

If universal encoding support (UES) is used by the Event Replicator Server, you have to do the following:

- BLSLIB access to the APS module libraries is required, therefore these extras statements are required:

```
/SET-FILE-LINK APSLIB,$SAG.APSvrs.LIB
/SET-FILE-LINK BLSLIBnn,$SAG.APSvrs.LIB
```

> **Note:** Newer APS module libraries are updates of the original. We advise you to copy the original, merge the new library into it, and reference the merged library in the JCL.

- The job has to contain the procedure call below to access the DDECSOJ object library:

```
CALL-PROCEDURE ($SAG.APSvrs.LIB,LMSLINKLIB),(LNK-NAME=BLSLIBn1)
```

where `BLSLIBn1` is last in the BLSLIB sequence.

- The DDECSOJ SET-FILE-LINK statement should point to the ECS encoding objects `$SAG.ADAvrs.ECnn` library.

- No special Software AG internal product software (APS) parameters have to be set.

- The setting of extra options on the START-PROGRAM statement can be made as follows:

```
/START-PROGRAM ($SAG.ADAvrs.MOD,ADARUN),-
/ RUN-MODE=*ADVANCED(ALT-LIB=YES,LOAD-INF=*REF,UNRES=*DELAY,-
/ MESSAGE=*ERROR)
```

Pertinent Operator Commands

DSTAT Command

Use this command to display the ADARIS statistics on request.

When issued against a running ADARIS job, the current ADARIS statistics are displayed in ADAU7B messages.

The syntax for DSTAT is described under *DSTAT Command* in section *Pertinent Operator Commands*, in the *Event Replicator for Adabas Administration and Operations Guide*.

Using the Sequential File with Event Replicator Target Adapter

≫ **To process the sequential output file in the Event Replicator Target Adapter:**

1   Edit the configuration file preferences in the Administration Tool and verify that the **Show File Source** configuration file preference is selected. For more information, refer to the section *Setting Configuration File and Target Adapter Preferences* in *About the Event Replicator Target Adapter Administration Tool* in the *Event Replicator Target Adapter Administration* documentation.

2   Create a new source file definition, as follows:

   1. Start up the Administration tool.

      The Administration tool window is displayed.

   2. On left side of the Administration tool window, expand the **Configuration File** node.

   3. Expand the **Sources** node.

4. Right-click on **File** and select **New** in the resulting context menu.

Or:

In the **File** menu select **New > File Source**.

A blank file source definition tab is displayed in the Administration tool window.

5. Specify field values for the file source definition. Each field and option is described in the following table.

Entry Name	Description
Description	Specify a description for your File source definition.
Directory	Specify the path of the directory in which the sequential output file is (or will be) stored.
Target	From the associated drop-down menu, select the target definition to which replicated data received from this source should be sent.
Autostart	Select this option to indicate that this File source definition should be automatically started when the Event Replicator Target Adapter is started. By default, this option is not selected.

6. Once all fields have been specified, click on the disk symbol on the toolbar to save the file source definition.

Or:

In the **File** menu select **Save**.

A **Save As** pop-up window is displayed.

7. Specify the name of the new file source definition in the **Save As** pop-up window and click **OK**.

The definition is saved in the context file for the Event Replicator Target Adapter.

3 Using FTP in Binary Mode, transfer the sequential output file to the Event Replicator Target Adapter machine. You can do this in one of two ways:

■ You can transfer it using FTP from the mainframe to the Event Replicator Target Adapter machine. In this case, the Event Replicator Target Adapter machine must be an FTP server.

■ You can issue an `ftp get` command from the Event Replicator Target Adapter machine to the mainframe.

In all cases, you should transfer the file via FTP to the directory you identified in the file source definition in Step 3.

> **Notes:**

1. If **Autostart** is selected for the file source definition or if the file source definition is otherwise started, the Event Replicator Target Adapter will immediately start to process files it finds in the file source directory, regardless of whether or not the FTP transfer has completed - unless the file name in the file source starts with the prefix "temp_".

2. You may run into problems if you are transferring the sequential output file via FTP across a slow network connection or if the Event Replicator Target Adapter is processing the data faster than it can be transferred using FTP. If this is the case, you should transfer the sequential output file to a file with a name starting with the prefix "temp_". This will ensure that the Event Replicator Target Adapter does not start to populate your relational database with data before the FTP transfer is complete. Once the FTP transfer is complete, you should then manually rename the file to something that does not start with the prefix "temp_" to start Event Replicator Target Adapter processing of the file.

3. Alternatively, you can verify that the file source is stopped and that **Autostart** is not selected while the FTP transfer is running. Once the FTP transfer is complete, you can start the file source to start Event Replicator Target Adapter processing of the sequential file data.

Using the STARTDATE and STARTTIME Parameters

When performing an Initial State with ADARIS, the results of the execution would be the inserting of data, from the Adabas mainframe database file, on a RDBMS Table located on Linux, UNIX or Windows. Under some circumstances, after the user has activated the Subscription, modifications may start being replicated before the Initial State starts.

In this case, either the Destination, to which the modifications to the file is being sent, or the Target Adapter Source that corresponds to the Event Replicator Destination configured for Normal data (updates) must be closed/stopped, respectively.

Consider the following circumstances:

▪ Updates to the Adabas file, on the mainframe, are being replicated. During this process, various inserts, updates and deletes will be sent to the Event Replicator Server. They will not be applied to the RDBMS until the queue to the Event Replicator Target Adapter is opened.

▪ ADARIS is started and the start time is registered and part of the information that will be passed to the Event Replicator Target Adapter. ADARIS will then read the relevant records and, at the end of processing, ADARIS will pass the end of processing information to the Event Replicator Target Adapter.

▪ Then the Event Replicator Target Adapter, processes the information from ADARIS, it records the start time of the initial state and then starts applying all the ADARIS data to the RDBMS table as inserts. At the end, it captures the end time of the ADARIS processing.

- After the initial state completes the Event Replication Destination for modifications and/or the Event Replicator Target Adapter source should be opened to allow the flow of the data to the Event Replicator Target Adapter. During this time, based on the start and end time of the ADARIS processing time, the `Error Continue` option is activated. During this period, if any rows already exist on the RDBMS table, the `Delete before Insert` option, will result in the data to be applied. Likewise, any updates and deletes will be processed in the normal way and applied to the RDBMS table.

- Once the Event Replicator Target Adapter finishes processing all modifications made between the start and end of the ADARIS processing, the `Error Continue` option will be deactivated, unless this option is set to be active during the Event Replicator Target Adapter session.

- If the file modifications occur before the ADARIS Initial State starts, the Event Replicator Target Adapter will not have set the `Error Continue` option on and the `Delete before Insert` would not be applied and this could result in an SQL error and the Event Replicator Target Adapter processing stop. To prevent this, the STARTDATE and STARTTIME parameters should be used and set these parameters to the oldest modification not yet processed by the Event Replicator Target Adapter. Once this is done, the `Error Continue` option will be activated as of the STARTDATE and STARTTIME setting, as opposed to the start of the ADARIS job.

- Another possible anomaly, which may happen, is if an update was issued between the STARTDATE / STARTTIME and the ADARIS start time, and the record was deleted from the Adabas file while the ADARIS Initial state was running. So, when the modifications started being processed, the update would have not found the row on the RDBMS table. This anomaly will be captured and ignored because the `Error Continue` option was active. Likewise, when the Delete is processed, it will also be ignored; however, the event will be recorded on the SQL error log.

13 ADARPE Utility: Extract TLOG Records

When File-type destination definitions are used (DTYPE=FILE), the records written to these destinations are written as TLOG records to the command log (CLOG). The ADARPE utility allows you to extract and process data from selected TLOG records in an Adabas command log (CLOG) file, sending the output to a user-selected output file. Records can be selected based on various selection criteria, including subscription name, destination name, and modification type. An output user exit can also be invoked to allow you to reject records prior to writing them to the output file. Finally, you can also specify selection criteria that allow you to control which data from the selected records is written to the output file.

This chapter describes the ADARPE utility.

ADARPE Prerequisites

The input CLOG data set to an ADARPE run must be identified on the DDCLOG JCL statement for the run. The input data set provided must be a sequential CLOG data set created in one of the following ways:

- Directly by Adabas when writing a sequential CLOG.
- By the ADARES CLCOPY utility. For more information about this utility, read your Adabas utilities documentation.

The ADARPE output data sets must be defined with RECFM=FB or RECFM=VB.

ADARPE Syntax and Parameters

This is the syntax of the ADARPE utility:

```
ADARPE EXTRACT NAME = name
                OUTPUTFILE = [ dd-dlblname ]
                [DBID = dbid [FILE = [ fn1 [, fn2 ] ... [, fn100 ] ] ] ...
                [DESTINATIONS = [ destdef1 [, destdefn ] ]
                [HEADER = [ YES | NO ]
                [IMAGE = [AI | BI | BOTH ]
                [MODIFICATIONTYPE = [INSERT | UPDATE | DELETE | ISTATE | ALL ]
                [OUTPUTEXIT = [ exitname ]
                [PAYLOADDATA = [ YES | NO ]
                [PAYLOADPREFIX = [ YES | NO ]
                [RECORDINFO = [ YES | NO ]
                [RECORDPREFIX = [ YES | NO ]
                [STATE = [ YES | NO ]
                [SUBSCRIPTIONS = [ subdef1 [, subtdefn ] ]
                [TEST]
                [TRANSACTIONINFO = [ YES | NO ]
                [TRANSACTIONLEVEL = [TRAN | RECORD | NONE ]
```

Each parameter is described below. These parameters can be used to filter the TLOG records that are extracted from the CLOG. Only records that meet all of the filter criteria set by these parameters are extracted.

EXTRACT

The EXTRACT parameter is a required parameter. It is the main keyword that denotes the start of a set of extraction keywords and must be the first parameter specified for ADARPE. Each subsequent specification of the `EXTRACT` parameter will result in the definition of a new set of extract criteria.

DBID

Optionally, you can specify a database from which records should be selected for extraction. All records for the specified database ID will be extracted, unless only specific files are requested using the `FILE` parameter.

Up to 100 DBID parameters can be specified in a single run of the ADARPE utility.

DESTINATIONS

Optionally, you can specify one or more destination definition names to be used in filtering which records are extracted. Only records written to the destination names specified in this parameter will be extracted. Valid destination definition names are one to eight alphanumeric characters long. They must be unique within a given set of ADARPE extract parameters.

FILE

This optional parameter can only be specified if the `DBID` parameter has previously been specified. Using this parameter, you can specify one to 100 file numbers within the database listed in the most recent `DBID` parameter. All records for these files will be extracted, unless other parameters set additional filter criteria for them.

File numbers specified must be unique within a given set of ADARPE extract parameters.

HEADER

Use this optional parameter to indicate whether a header record will be written to the output file. Valid values are "YES" (write the header record) and "NO" (do not write the header record). The default is "YES".

The contents of the header (when HEADER=YES) is specified are stored in a URBU record.

IMAGE

Use this optional parameter to identify the types of record images that should be written to the output file for the records that are selected. Valid values are described in the table below:

Value	Description
AI	Only after images of selected records are written to the output file. If no after image exists for a selected record, no payload data for that record is written.
BI	Only before images of selected records are written to the output file. If no before image exists for a selected record, no payload data for that record is written.
BOTH	Both before and after images of selected records are written to the output file. This is the default.

MODIFICATIONTYPE

Optionally, use this parameter to identify the type of modifications you want to extract. Valid values for this parameter are described in the table below.

Value	Description
ALL	Use this value to indicate that records of all modification types should be written to the output file. This is the default.
DELETE	Use this value to indicate that only records of deletions should be written to the output file.
INSERT	Use this value to indicate that only records of insertions should be written to the output file.
ISTATE	Use this value to indicate that only records of initial-state operations should be written to the output file.
UPDATE	Use this value to indicate that only records of update operations should be written to the output file.

NAME

This is a required parameter. Use this parameter to specify a name for a set of ADARPE extract parameters. Valid names can be up to eight characters long.

OUTPUTEXIT

Optionally, use this parameter to specify the name of a user exit you want run to further filter the TLOG records in the CLOG. The ADARPE utility will load this user exit and call after all other parameter filter criteria have been applied to a record, but prior to writing the record to the output file.

> **Note:** The user exit must be a program you have written and supplied in the JCL for the run.

User exit names must be one to eight alphanumeric characters long.

For more information about the output user exit, read *ADARPE Output User Exits*, elsewhere in this section/

OUTPUTFILE

This is a required parameter. Specify the output DD name or DLBL name of the output data set as listed in the JCL for the run. DD and DLBL names must be one to eight alphanumeric characters long and must be unique within a given ADARPE run. Selected TLOG records from the CLOG are extracted to the file identified by this parameter.

PAYLOADDATA

Optionally, use this parameter to indicate whether the payload data is written to the output file when a record is extracted. Valid values for this parameter are "YES" (extract the payload) and "NO" (do not extract the payload). The default is "YES".

PAYLOADPREFIX

Optionally, use this parameter to indicate whether the URBD (data record) control information element is written to the output file when a record is extracted. Valid values for this parameter are "YES" (extract the URBD) and "NO" (do not extract the URBD). The default is "YES".

RECORDINFO

Optionally, use this parameter to indicate whether the URBR (record) control information element is written to the output file when a record is extracted. Valid values for this parameter are "YES" (extract the URBR) and "NO" (do not extract the URBR). The default is "YES".

RECORDPREFIX

Optionally, use this parameter to indicate whether the URBL (transaction log record) control information element is written to the output file when a record is extracted. Valid values for this parameter are "YES" (extract the URBL) and "NO" (do not extract the URBL). The default is "YES".

STATE

Optionally, use this parameter to indicate whether the URBS (status/response) control information element is written to the output file when a record is extracted. Valid values for this parameter are "YES" (extract the URBS) and "NO" (do not extract the URBS). The default is "YES".

SUBSCRIPTIONS

Optionally, you can specify one or more subscription definition names to be used in filtering which records are extracted. Only records covered by (assigned to) the subscriptions named in this parameter will be extracted. Valid subscription definition names are one to eight alphanumeric characters long. They must be unique within a given set of ADARPE extract parameters.

TEST

The TEST parameter tests the operation syntax without actually performing the operation. Only the syntax of the specified parameters can be tested; not the validity of values and variables.

TRANSACTIONINFO

Optionally, use this parameter to indicate whether the URBT (transaction) control information element is written to the output file when a record is extracted. Valid values for this parameter are "YES" (extract the URBT) and "NO" (do not extract the URBT). The default is "YES".

TRANSACTIONLEVEL

Optionally, use this parameter to indicate the level of consistency the ADARPE utility maintains across sequential CLOGs from different runs of the utility. Valid values for this parameter are "NONE", "RECORD", or "TRAN".

ADARPE Output User Exits

When an output user exit is specified for the run, it is loaded prior to processing the input sequential CLOGs. If a specified exit does not exist, the ADARPE utility will terminate with an error. Otherwise, the exit will be called prior to the record being written to an output file after the selection processing. The exit will be passed the parameters in a URBZ parameter block. For more information, read about the URBZ DSECT, in *Event Replicator for Adabas Application Programmer's Reference Guide*.

The ADARPE user exit call is structured as described by the URBQ DSECT, in *Event Replicator for Adabas Application Programmer's Reference Guide*.

Sample JCL

The following sample JCL could be used to run ADARPE. In this example, three output files are produced, identified by the DD names OUTRPE1, OUTRPE2, and OUTRPE3. A sample job, EXTTLOG, is provided in the MVSJOBS data set.

```
//ADARPE    JOB
//*
//ADARPE    EXEC PGM=ADARUN
//STEPLIB  DD DISP=SHR,DSN=ARFvrs.LOAD        <=== ARF library
//DDCLOG   DD DISP=SHR,DSN=ADABAS.REPTOR.CLOGCOPY
//MERGIN1  DD DISP=SHR,DSN=EXAMPLE.MERGIN1
//MERGIN2  DD DISP=SHR,DSN=EXAMPLE.MERGIN2
//OUTRPE1  DD DSN=EXAMPLE.OUTPUT1,
//           DISP=(,CATLG),UNIT=DISK,SPACE=(TRK,(1,1))
//OUTRPE2  DD DSN=EXAMPLE.OUTPUT2,
//           DISP=(,CATLG),UNIT=DISK,SPACE=(TRK,(1,1))
```

```
//OUTRPE3  DD DSN=EXAMPLE.OUTPUT3,
//          DISP=(,CATLG),UNIT=DISK,SPACE=(TRK,(1,1))
//DDDRUCK  DD SYSOUT=X
//DDPRINT  DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD   DD *
ADARUN PROGRAM=ADARPE,SVC=xxx,DEVICE=dddd,DB=yyyyy
/*
//DDKARTE  DD *
ADARPE EXTRACT NAME=OUT1
ADARPE  OUTPUTFILE=OUTRPE1
ADARPE  DESTINATIONS='AA,BB'
*
ADARPE EXTRACT NAME=OUT2
ADARPE  OUTPUTFILE=OUTRPE2
ADARPE  DBID=dddd,FILE=ffff
*
ADARPE EXTRACT NAME=OUT3
ADARPE  OUTPUTFILE=OUTRPE3
ADARPE  SUBSCRIPTIONS='CC,DD'
*
/*                                  ↵
```

The following table describes the DD statements in this JCL in more detail:

DD Statement	Storage	Identifies
DDCLOG	disk	The input TLOG data from the sequential command log (CLOG), created by ADARES CLCOPY or output from a previous ADARPE run.
MERGIN1/MERGIN2	disk	Intermediate files from and to which intermediate results may be read or written.
DDDRUCK	printer	The ADARPE report, messages, and parameters.
DDPRINT	printer	Pertinent ADARUN messages.
DDCARD	reader	Pertinent ADARUN parameters.
DDKARTE	reader	The ADARPE selection parameters.
OUTRPE*n*	disk	The sequential output file specified on the OUTPUTFILE parameter.

ADARPE Parameter Examples

The following example is for extracts named OUT6, OUT4, and OUT3. This run of ADARPE produces three output files OUTRPE1, OUTRPE2, and OUTRPE3.

```
ADARPE EXTRACT NAME=OUT6
ADARPE DBID=1955,FILE=222
ADARPE OUTPUTFILES='OUTRPE1'
ADARPE EXTRACT NAME=OUT4
ADARPE DBID=1955,FILE=224
ADARPE OUTPUTFILES='OUTRPE2'
ADARPE EXTRACT NAME=OUT3
ADARPE DBID=1955,FILE=229
ADARPE OUTPUTFILES='OUTRPE3'
```

The following parameters would only test the parameters set for an ADARPE utility run; the ADARPE utility would not actually run. In addition, these series of parameters request that an output user exit named EXIT1 be called during the ADARPE run.

```
ADARPE TEST
ADARPE EXTRACT NAME=OUT6
ADARPE DBID=1955,FILE=222
ADARPE OUTPUTFILES='OUTRPE1'
ADARPE OUTPUTEXIT=EXIT1
ADARPE TRANSACTIONINFO=NO
ADARPE EXTRACT NAME=OUT4
ADARPE DBID=1955,FILE=224
ADARPE OUTPUTFILES='OUTRPE2'
ADARPE TRANSACTIONINFO=NO
ADARPE OUTPUTEXIT=EXIT2
ADARPE EXTRACT NAME=OUT3
ADARPE DBID=1955,FILE=229
ADARPE TRANSACTIONINFO=NO
ADARPE OUTPUTFILES='OUTRPE3'
ADARPE OUTPUTEXIT=EXIT1
```

ADARPE Output Examples

Here is a sample of the output produced by an ADARPE run. The input file mentioned in the output is a reference to the CLOG file that was read as input to ADARPE.

```
A D A R P E   V3.2  SM1   DBID = 01954  Started         2007-07-13  08:25:14

Parameters:
-----------

ADARPE EXTRACT NAME=OUT5
ADARPE  OUTPUTFILE=OUTRPE1
ADARPE  TRANSACTIONLEVEL=RECORD




A D A R P E   V3.2  TLOG Extract Utility Summary       2007-07-13  08:43:11
```

```
Input file statistics
          862 CLOG records were read
          862 TLOG records were read
          277 Destination output data records were read
           88 URBT records were read
           88 URBR records were read
          101 URBD records were read
            0 URBS records were read

Intermediate input file statistics
            4 records were read from intermediate input file MERGIN1
            1 URBT records were read
            1 URBR records were read
            0 URBD records were read
            0 URBS records were read
            2 header/trailer records were read

Intermediate output file statistics
            2 records were written to intermediate output file MERGIN2
            0 URBT records were written
            0 URBR records were written
            0 URBD records were written
            0 URBS records were written
            2 header/trailer records were written

EXTRACT=OUT5
          279 records selected
           89 URBT records were selected
           89 URBR records were selected
          101 URBD records were selected
            0 URBS records were selected

          279 records written
           88 URBT records were written
           89 URBR records were written
          101 URBD records were written
            0 URBS records were written
            1 URBU records were written


A D A R P E  Terminated normally                 2007-07-13  08:43:11
  ↵
```

The intermediate input and output messages are provided *only* if TRANSACTIONLEVEL=RECORD or TRAN is specified.

If a user exit (OUTPUTEXIT parameter) supplied in the run, an additional line is printed indicating if any records have been skipped. For example:

```
A D A R P E   V3.2  SM1   DBID = 01954  Started            2007-07-13  11:33:40


Parameters:
-----------


ADARPE EXTRACT NAME=OUT3
ADARPE  OUTPUTFILE=OUTRPE1


ADARPE EXTRACT NAME=OUT5
ADARPE  OUTPUTFILE=OUTRPE2
ADARPE  OUTPUTEXIT=UEXIT2


A D A R P E   V3.2  TLOG Extract Utility Summary          2007-07-13  11:33:40


Input file statistics
          1286 CLOG records were read
          1286 TLOG records were read
           385 Destination output data records were read
           124 URBT records were read
           124 URBR records were read
           136 URBD records were read
             1 URBS records were read



EXTRACT=OUT3
           385 records selected
           124 URBT records were selected
           124 URBR records were selected
           136 URBD records were selected
             1 URBS records were selected

           386 records written
           124 URBT records were written
           124 URBR records were written
           136 URBD records were written
             1 URBS records were written
             1 URBU records were written


EXTRACT=OUT5
           385 records selected
           124 URBT records were selected
           124 URBR records were selected
           136 URBD records were selected
             1 URBS records were selected

           101 records written
           124 URBT records were written
           124 URBR records were written
           136 URBD records were written
             1 URBS records were written
             1 URBU records were written
```

```
          285 records skipped by userexit UEXIT2


A D A R P E   Terminated normally                        2007-07-13  11:33:40
```

14 ADARPL Utility: PLOG Replication Replay

The ADARPL utility, also known as the Replay Utility, provides a mechanism by which you can read an Adabas PLOG and resend replicated data to one or more Event Replicator Servers. This utility reads the sequential (merged) PLOG of an Adabas database and, based on the parameters you specify, sends related data to one or more Event Replicator Servers.

> **Note:** The version of Event Replicator specified in an ADARPL utility job must match the version of Event Replicator used by the Event Replicator Server. In addition, the Adabas version used by the Event Replicator Server must be greater than or equal to the Adabas version used in an ADARPL utility job.

Replay processing in Event Replicator for Adabas can be run in any of three modes: *synchronized*, *unsynchronized*, and *replay-only*. When you initiate a replay process as an ADARPL batch job without specifying a replay token generated using the Adabas Event Replicator Subsystem, the replay process is run in *unsynchronized* mode. You can run ADARPL in *synchronized* and *replay-only* modes if you specify the replay token for a replay request generated using the Adabas Event Replicator Subsystem or the standalone replay API.

In addition, you can automate ADARPL processing. When you request ADARPL automation, the Event Replicator Server nucleus will automatically generate an ADARPL job stream using parameters specified using the Adabas Event Replicator Subsystem or a standalone replay API. For complete information on automating ADARPL processing, read *Automating Replay Processing*, in the *Event Replicator for Adabas Administration and Operations Guide*.

For complete information on when to use replay processing and replay processing modes, read *Understanding Replay Modes*, in *Event Replicator for Adabas Administration and Operations Guide*. For complete information on how to generate a replay token using the Adabas Event Replicator Subsystem, read *Initiating a Replay Request Using the Adabas Event Replicator Subsystem* in *Adabas Event Replicator Subsystem User's Guide*. For complete information on how to generate a replay token using the standalone replay API, read *Initiating a Replay Request Using the Standalone API*, in *Event Replicator for Adabas Administration and Operations Guide*.

Functional Overview

The Replay Utility can recover replication data from the sequential PLOG data sets (after copying and merging them) for the time over which replication processing was interrupted. You must be sure to supply the correct:

- PLOG information
- date and time settings
- file number, subscription, and destination settings
- target Event Replicator Server information

New data replication and processing for the same database files can be occurring simultaneously, although replication does not need to be active to use the Replay Utility.

When replication has been activated for a database file and the stream of replication data coming into the Event Replicator Server for that file is interrupted for any reason, you can use the Replay Utility. The following processing occurs once the Replay Utility is started:

1. The Replay Utility reads through all transactions on the specified sequential PLOG, starting with records with the specified start (*from*) date and time, and ending with records with the stop (*to*) date and time. Start and stop times are specified in the Replay Utility run.

2. As records are read, only those for fully completed transactions on the specified database files are processed. However, ADARPL cannot determine the exact time of every update recorded on the PLOG. It tries to accurately process only the data requested by the `FROMTIME` and `TOTIME` parameters, subject to the rule that it never processes less protection data than requested, as follows:

   ■ Replication data is built for all transactions read from the PLOG.

   ■ Replication data is built only for transactions where both the transaction start and the transaction end (ET) are found on the input PLOG. This ensures that no partial transactions will be sent to the Event Replicator Server during the replay.

   ■ Replication data for transactions that end (ET) before the `FROMDATE/FROMTIME` specification are discarded and not sent to the Reptor. This selection is based on the PLOG record timestamp.

   ■ If `FROMDATE/FROMTIME` is not specified all replication data is sent to the Event Replicator Server.

   ■ If `TODATE/TOTIME` is specified replay processing stops when a PLOG record is read that has a timestamp higher than or equal to `TOTIME`. At this time partial transactions (any replication data built but not yet committed) are discarded and will not be sent to the Event Replicator Server.

   ■ If `TODATE/TOTIME` is not specified ADARPL processing stops after reading the last block of the PLOG. At this time partial transactions (any replication data built but not yet committed) are discarded and will not be sent to the Event Replicator Server.

3. Transactions selected for processing are sent to the target Event Replicator Servers selected for the run. The target Event Replicator Servers may differ from those specified originally for replication.

4. If a transaction was sent successfully to the Event Replicator Server before the original failure and its Replay Utility *resend flag* is turned *on*, messages are sent to the target application indicating that duplicate data may have been replicated. If a transaction was sent successfully to the Event Replicator Server before the failure and its Replay Utility resend flag is turned *off* for the record, no warning messages are sent, although duplicate data may still have been replicated. The resend flags are set based on the *resend* date and times set for Replay Utility run. For more information, read about the **Replay Utility syntax**, later in this section.

5. Specification of `RESENDDATE` and `RESENDTIME` are not supported when using a token. When using a token, ADARPL will list the "ADARPL start date/time" which was used to select the PLOG datasets to be allocated to the replay. This information is only for your reference - it does not affect the above transaction selection.

All date and time parameters are specified in local time. `FROMDATE`, `FROMTIME`, `TODATE` and `TOTIME` are internally converted to UTC since the time stamps on the PLOG are written in UTC. This conversion is always done based on the *current* time difference between local time and UTC.

The ADARPL Replay utility will not replicate data for a file with spanned DS record support turned on.

ADARPL Prerequisites

The following prerequisites must be met to effectively use the Replay Utility in batch mode:

■ Verify that the following ADARUN parameters are specified in ADARPL JCL:

```
ADARUN PROG=ADARPL,DBID=dbid,SVC=svc,MODE=MULTI
```

where *dbid* is the Adabas database ID on which the files that are being replicated reside and *svc* is the SVC number to be used for communications with Adabas and the Event Replicator Server. Running the Replay Utility with MODE=SINGLE will default to MODE=MULTI.

■ Verify that the correct PLOG is used for the run and that it is a sequential PLOG, not a dual PLOG. The PLOG is specified with the DD name DDSIIN.

■ Specify valid values for ADARPL parameters, as appropriate. The following information is needed.

   ■ Determine what the last successful transmission time was for replication. You can obtain this timestamp from the deactivation messages that occur when replication stops.

   ■ Determine the file numbers, subscriptions, or destinations for which replication data was not received. At least one file number, subscription definition name, or destination definition name must be specified. Multiple files, subscriptions, and destinations may be specified simultaneously.

   ■ Determine what the resend dates and times for the data should be so that the target Event Replicator Server is warned when duplicate transactions may have been sent.

   ■ Determine the IDs of the Event Replicator Servers to which you want to send the data. These target IDs may differ from those specified originally for replication.

■ Verify that the target application can handle duplicate records.

■ Either the Adabas database must be active or the DDASSO DD statement must be specified in the JCL, identifying the ASSO data set for the run. The Replay Utility will attempt to issue a call

to Adabas to obtain the GCB, FCBs, and FDTs from the nucleus. If this call fails, it will attempt to read this information itself using the ASSO data set specified in the Replay Utility run.

> **Note:** When running ADARPL job using the TOKEN parameter, the Adabas database must be active.

ADARPL Syntax and Parameters

The syntax and parameters vary, depending on whether you want to initiate ADARPL using a token generated by the Adabas Event Replicator Subsystem.

Syntax for Initiating ADARPL Without A Token

The syntax of the ADARPL utility to initiate replay processing without a token is:

```
ADARPL REPLAY { [FILES=fn1[,fn2]...]  [SUBSCRIPTION=sname[,sname]...]  ↵
[DESTINATION=dname[,dname]...]  }
            [FROMDATE=yyyymmdd [FROMTIME=hhmmss]]
            [LRPL=poolsize]
            [NU=numusers]
            [ORIGINDBID=dbid]
            [ORIGINFILES=fn1[, fn2]...]
            [PLOGDBID=dbid]
            [PLOGFILES=fn1[, fn2]...]
            [RESENDDATE=yyyymmdd [RESENDTIME=hhmmss]]
            [RPLDSBI={Y | N}[,{Y | N}]...]
            [RPLKEY=primary-key[,primary-key]...]
            [RPLSORT={YES | NO}
            [RPLTARGETID=id | RPLTARGETS=id1[,id2]...]
            [STATINTERVAL=interval]
            [TODATE=yyyymmdd [TOTIME=hhmmss]]
```

Syntax for Initiating ADARPL With A Token (Synchronized and Replay-only Replay Modes)

The syntax of the ADARPL utility to initiate replay processing using a token generated from the Adabas Event Replicator Subsystem:

```
ADARPL REPLAY TOKEN=num
            RPLTARGETID=id
            [LRPL=poolsize]
            [NU=numusers]
```

> **Note:** When running ADARPL with the TOKEN parameter, Adabas must be active.

ADARPL Parameters

Each parameter is described in the following table. Not all parameters can be used in the same run.

Parameter	Description	Required?	Default
DESTINATION	Specifies one or more valid Event Replicator for Adabas destination definition names. **Note:** The maximum number of values you can specify for this parameter is 1,000.	No -- although at least one DESTINATION. FILE, or SUBSCRIPTION parameter must be specified.	none
FILES	Specifies one or more valid Adabas file numbers for which replication was turned on. The PLOG records for these files will be replayed and replicated, based on other settings in the Replay Utility run. **Note:** The maximum number of values you can specify for this parameter is 1,000.	No -- although at least one DESTINATION. FILE, or SUBSCRIPTION parameter must be specified.	none
FROMDATE	Specifies a start date in $yyyymmdd$ format. Replay processing will include transactions in the PLOG that ended at or after this date.	No	Replay Utility processing starts at the beginning of the PLOG and includes all completed transactions.
FROMTIME	Specifies a start time in $hhmmss$ format. Replay processing will include transactions in the PLOG that ended at or after this time. This parameter cannot be specified unless the FROMDATE parameter is specified also.	No	Replay Utility processing starts at the beginning of the PLOG records for the date specified by the FROMDATE parameter, if any, and includes all completed transactions.
LRPL	Specifies the size of the replication pool to be used by ADARPL for keeping replication data. Values can also be specified in kilobytes by adding a "K" after the value (for example, 1000K).	No	100,000 bytes
NU	Specifies the number of User Queue Elements that will be generated during a replay process. This number must be greater than or equal to the maximum number of users that have open transactions simultaneously.	No	1000

Parameter	Description	Required?	Default
ORIGINDBID	Specifies the Adabas database ID (the URBTDBID) that should be listed in the replicated PLOG transactions sent to the Event Replicator Server during replay. This parameter is particularly useful when the Event Replicator Server receiving the replayed data uses a different series of database numbers in its subscriptions than the Adabas database associated with the PLOG being replayed. For example, if an Adabas database has a DBID of 10, but the equivalent Event Replicator Server has a DBID of 50 and corresponding subscriptions for DBID 50, you would specify 50 as the value of the ORIGINDBID in the replay run. This would ensure that the replay transactions from database 10 are processed by the subscription for database 50.	No	The database ID specified by the ADARUN DBID parameter for the ADARPL run.
ORIGINFILES	This parameter can only be specified if the FILES parameter is also specified. It allows you to specify the Adabas file number that should be listed in the replicated PLOG transactions sent to the Event Replicator Server during replay processing. This parameter is particularly useful when the Event Replicator Server receiving the replayed data uses a different series of file numbers in its subscriptions than the Adabas database associated with the PLOG being replayed. **Note:** The maximum number of values you can specify for this parameter is 1,000.	No	none
PLOGDBID	Specifies an alternate Adabas database ID. The records from the PLOG data set associated with this alternate Adabas database are replayed to the Event Replicator Server using the GCBs and FCB/FDTs from the database specified by the ADARUN DBID parameter. In other words, using this parameter you can run ADARPL with a PLOG from one database and the ADARUN DBID pointing to a different database. For example, you might want to run ADARPL using a production PLOG in a test environment. In this case, the ADARUN DBID would point to the test database and the PLOGDBID parameter would be set to the production database ID.	No	The database ID specified by the ADARUN DBID parameter for the ADARPL run.
PLOGFILES	This parameter can only be specified if the FILES parameter is also specified. It allows you to specify alternate PLOG file numbers for the run. The records	No	none

Parameter	Description	Required?	Default
	from the alternate PLOG files are replayed to the Event Replicator Server. **Note:** 1. This parameter cannot be specified if any of the files listed in the FILES parameter is for a file in an expanded file chain. 2. The maximum number of values you can specify for this parameter is 1,000.		
REPLAY	A required keyword for the Replay Utility.	Yes	none
RESENDDATE	Specifies a PLOG record date in *yyyymmdd* format at which the Replay Utility should turn off its resend flags. Records with dates after the specified RESENDDATE will have their resend flags turned off.	No	Replay Utility send all transactions with the resend flag turned on.
RESENDTIME	Specifies a PLOG record time in *hhmmss* format at which the Replay Utility should turn off its resend flags. This parameter cannot be specified unless the RESENDDATE parameter is specified also. Records with times after the specified RESENDTIME will have their resend flags turned off.	No	Replay Utility send all transactions for records prior to the date specified by the RESENDDATE parameter (if any) with the resend flag turned on.
RPLDSBI	Specifies whether the collection of before images of data storage should occur for an update command to the file. This parameter can only be specified if either the RPLTARGETID or RPLTARGETS parameter is specified. Valid values for this parameter are "Y" (the collection of before images of data storage will occur for the file) or "N" (this information is obtained from the FCB). If this parameter is specified, one setting must be specified for each file in the FILES parameter. Values should be listed in single quotes and separated by commas. To specify the default, simply type a comma for that file's value. In the following example, the second file is using the default (N): RPLDSBI='Y,,Y,Y'	No	N

Parameter	Description	Required?	Default
	This parameter is only allowed if `FILES` parameter is specified. **Note:** The maximum number of values you can specify for this parameter is 1,000.		
RPLKEY	Specifies the primary key for replication. This parameter can only be specified if either the `RPLTARGETID` or `RPLTARGETS` parameter is specified. If this parameter is specified, one setting must be specified for each file in the `FILES` parameter. Values should be listed in single quotes and separated by commas. To specify a blank `RPLKEY` in a list, simply type a comma for that file's value. In the following example, the third file in the `FILES` list has not `RPLKEY`: `RPLKEY='AA,AB,,AG'` This parameter is only allowed if `FILES` parameter is specified. **Note:** The maximum number of values you can specify for this parameter is 1,000.	No	Replay Utility assumes no primary key.
RPLSORT	Specifies whether Adabas transaction data sorting should occur or not. Valid values for this parameter are "YES" or *NO*. For complete information on the affects of using `RPLSORT`, read about the ADARUN *RPLSORT Parameter*, elsewhere in this guide.	No	YES
RPLTARGETID	Specifies the target ID of the Event Replicator Server to which the Replay Utility transactions should be sent. This parameter is mutually exclusive with the `RPLTARGETS` parameter; only one of them should be specified in a Replay Utility run. If more than one file is specified in the `FILES` parameter, all transactions from these files will be sent to the same Event Replicator Server target. Only one Event Replicator target ID can be specified per file.	Yes, if the `SUBSCRIPTION`, , or `DESTINATION` parameters are specified. Otherwise, this parameter is *not* required.	Event Replicator target ID stored in the FCB is used.
RPLTARGETS	Specifies the target IDs of the Event Replicator Servers to which Replay Utility transactions should be sent. This parameter is mutually exclusive with the `RPLTARGETID` parameter; only one of them should be specified in a Replay Utility run. Only one Event Replicator target ID can be specified per file. One setting must be specified for each file in the `FILES` parameter. Values should be separated by	No	Event Replicator target ID stored in the FCB is used.

Parameter	Description	Required?	Default
	commas. To use the default, simply type a comma for that file's value. In the following example, the target ID for the third file is obtained from the FCB: `RPLTARGETS=23,24,,25` This parameter can be specified only if the `SUBSCRIPTION` and `DESTINATION` parameters are *not* specified. **Note:** The maximum number of values you can specify for this parameter is 1,000.		
STATINTERVAL	Specifies the interval, in seconds, at which ADARPL should print out DSTAT statistics. Valid values are "0" (zero) or an integer in the range from "60" to "2,147,483,647". A value of "0" indicates that statistics should not be printed.	No	0
SUBSCRIPTION	Specifies one or more valid Event Replicator for Adabas subscription definition names. **Note:** The maximum number of values you can specify for this parameter is 1,000.	No -- although at least one DESTINATION. FILE, or SUBSCRIPTION parameter must be specified.	none
TODATE	Specifies an end date in *yyyymmdd* format. Replay processing will stop with transactions in the PLOG that ended at or after this date.	No	Replay Utility processing stops at the end of the PLOG and includes all completed transactions.
TOKEN	Specifies the token number of the replay request generated using the Adabas Event Replicator Subsystem. This token number can be found in the Event Replicator Server job log or the message (or other returned data) that was issued by the Adabas Event Replicator Subsystem or the standalone replay API when the replay request was initiated. For complete information on how to generate a replay token using the Adabas Event Replicator Subsystem, read *Initiating a Replay Request Using the Adabas Event Replicator Subsystem* in *Adabas Event Replicator Subsystem User's Guide*. For complete information on how to generate a replay token using the standalone replay API, read *Initiating a Replay Request Using the Standalone API*, in *Event Replicator for Adabas Administration and Operations Guide*.	Yes, if you want to initiate replay processing using a replay token	none

Parameter	Description	Required?	Default
	Valid values are positive integers between 1 and 32767.		
TOTIME	Specifies an end time in *hhmmss* format. Replay processing will stop with transactions in the PLOG that ended at or after this time. This parameter cannot be specified unless the TODATE parameter is specified also.	No	Replay Utility processing stops at the end of the PLOG records for the date specified by the TODATE parameter, if any, and includes all completed transactions.

Commands Useful to ADARPL

The DSTAT command prints statistics about ADARPL processing on request. The interval at which the statistics are printed is governed by the setting of the STATINTERVAL parameter specified when ADARPL was started.

For more information about this command, read *DSTAT Command* in *Event Replicator for Adabas Administration and Operations Guide*.

ADARPL Examples

This section provides three examples of ADARPL parameter usage.

Example 1 -- Using FILES and RPLTARGETID

```
ADARPL REPLAY FILES=1,2,3,4
              RPLDSBI='Y,Y,Y,N'
              RPLKEY='AA,AB,,AG'
              RPLTARGETID=25
```

In this example:

- File 1: the collection of before images of data storage will occur for this file during replay processing (RPLDSBI=Y), the primary key AA will be used, and the replicated data will be sent to the Event Replicator Server target 25.

- File 2: the collection of before images of data storage will occur for this file during replay processing (RPLDSBI=Y), the primary key AB will be used, and the replicated data will be sent to the Event Replicator Server target 25.

- File 3: the collection of before images of data storage will occur for this file during replay processing (RPLDSBI=Y) and the replicated data will be sent to the Event Replicator Server target 25. Its primary key will be blank.

- File 4: the collection of before images of data storage will *not* occur for this file during replay processing (RPLDSBI=N), the primary key AG will be used, and the replicated data will be sent to the Event Replicator Server target 25.

Example 2 -- Using FILES and RPLTARGETS

```
ADARPL REPLAY FILES=1,2,3,4
               RPLTARGETS=23,24,25,26
               RPLDSBI=',,,,'
               RPLKEY=',,,,'
```

- File 1: the collection of before images of data storage will *not* occur for this file during replay processing (RPLDSBI=N is the default) and the replicated data will be sent to the Event Replicator Server target 23. Its primary key will be blank.

- File 2: the collection of before images of data storage will *not* occur for this file during replay processing (RPLDSBI=N is the default) and the replicated data will be sent to the Event Replicator Server target 24. Its primary key will be blank.

- File 3: the collection of before images of data storage will *not* occur for this file during replay processing (RPLDSBI=N is the default) and the replicated data will be sent to the Event Replicator Server target 25. Its primary key will be blank.

- File 4: the collection of before images of data storage will *not* occur for this file during replay processing (RPLDSBI=N is the default) and the replicated data will be sent to the Event Replicator Server target 26. Its primary key will be blank.

Example 3 -- Using FILES Alone

```
ADARPL REPLAY FILES=31
```

Replicated data will be recovered for file 31. The target Event Replicator Server IDs and RPLDSBI settings will be obtained from the FCBs. No primary key will be used for the file.

Example 4 -- Using FILES With SUBSCRIPTION and DESTINATION

```
ADARPL REPLAY FILES=31
     RPLTARGETID=23
     SUBSCRIPTION='SUB01,SUB02'
     DESTINATION='DEST02'
```

Replicated data will be recovered for file 31 and sent to the Event Replicator Server 23. The RPLDSBI settings will be obtained from the FCBs. No primary key will be used for the file. Event Replicator for Adabas processing will process transactions to destination DEST02 and as defined in subscriptions SUB01 and SUB02.

Sample JCL

The following sample JCL could be used to run ADARPL. In this sample, PLOG records from files 1, 4, and 6 of the Adabas database are replayed.

> **Note:** The ASSO data set is required in the JCL if the Adabas database is inactive.

```
//ADARPL    JOB
//*
//*  ADARPL: Sample JCL to invoke ADARPL to process completed
//*          transactions starting at the beginning of the PLOG and
//*          ending at the end of the PLOG for files 1, 4, and 6.
//*          The Reptor target IDs, the replication keys and whether
//*          before images of data storage should occur for update
//*          commands to the file, will be read from the FCBs.
//*
//RPL       EXEC  PGM=ADARUN
//STEPLIB  DD  DISP=SHR,DSN=ADABAS.Vvrs.LOAD  <=== Adabas load lib
//DDASSOR1 DD  DISP=SHR,DSN=EXAMPLE.DBdbid.ASSOR1 <=== Adabas ASSO
//DDSIIN   DD  DISP=SHR,DSN=EXAMPLE.PLOG101   <=== Sequential PLOGs
//         DD  DISP=SHR,DSN=EXAMPLE.PLOG102   <=== (concatenated)
//         DD  DISP=SHR,DSN=EXAMPLE.PLOG103
//DDDRUCK  DD  SYSOUT=X
//DDPRINT  DD  SYSOUT=X
//SYSUDUMP DD  SYSOUT=X
//DDCARD   DD  *
ADARUN PROG=ADARPL,DBID=dbid,SVC=svc,MODE=MULTI,DEVICE=3390
//DDKARTE  DD  *
ADARPL REPLAY FILES=1,4,6
//
```

15 ADARPP Utility: Print TLOG Records

The ADARPP utility prints selected TLOG records from an Adabas command log (CLOG) file. Records can be selected based on date and time, destination, and subscription. The level of printed output can be controlled for each type of TLOG data. The printed output can be formatted, produced in dump format, or both. Finally, the printed dates and times of the TLOG records can be adjusted for specific time zones.

Prerequisites

Before it can be processed by the ADARPP utility, the Adabas command log must first be copied using the CLCOPY function of the ADARES utility. For more information about this function, refer to your Adabas utilities documentation.

Syntax and Parameters

The syntax of the ADARPP utility is as follows:

```
ADARPP [DESTINATION=destination-list]
       [DTLASSIGN={0|1|2|3}]
       [DTLCOMP={0|1|2|3}]
       [DTLSLOGREAD={0|1|2|3}]
       [DTLSLOGWRITE={0|1|2|3}]
       [DTLTOUTPUT={0|1|2|3}]
       [FROMDATE=yyyymmdd [FROMTIME=hhmmssth]]
       [PAGESIZE={linecount|58}]
       [PRINT={DUMP|FORMAT|BOTH}]
       [STATE={YES|NO}]
       [STLFILTER={0|1|2|3}]
       [STLIGNORE={0|1}]
       [STLINPUT={0|1|2|3}]
       [STLMATCH={0|1}]
       [STLNOMATCH={0|1}]
       [STLOUTPUT={0|1|2|3}]
       [SUBSCRIPTION= subscription-list]
       [TIMEZONE={LOCAL|MACHINE|nn}]
       [TLCOMP={0|1}]
       [TLINPUT={0|1|2|3}]
       [TLISTATE={0|1|2|3}]
       [TLISTATECOMP={0|1|2|3}]
       [TLNOSUB={0|1|2|3}]
       [TLQCOMP={0|1|2|3}]
       [TLREQERR={0|1|2|3}]
       [TLREQRECV={0|1|2}]
       [TLREQREJECT={0|1|2}]
       [TLRETRANS={0|1|2|3}]
```

```
[TLSTATUS={0|1|2|3}]
[TODATE=yyyymmdd [TOTIME=hhmmssth]]
```

Each parameter is described in the following table.

Parameter	Description	Valid Values	Default
DESTINATION	Specifies the name of one or more destination definitions for which TLOG records should be eligible for printing. Other TLOG records for other destinations would NOT be eligible for printing. **Note:** Records not eligible because they are not listed with this parameter may be selected if the SUBSCRIPTION parameter is also specified in the same ADARPP run. The SUBSCRIPTION and DESTINATION parameters are processed by ADARPP using OR logic; that is, if both SUBSCRIPTION and DESTINATION are specified, a TLOG record need only meet the requirement of one of them to be eligible for printing. Up to 32 destination names may be specified. If more than one is specified, they must be specified together in single quotes and separated by commas. For example: `ADARPP DESTINATION='DEST01,DEST02'`	Any valid destination definition names	All records for all destinations are eligible for printing
DTLASSIGN	Sets the print level for TLOG records logged when a transaction was assigned to a destination. Values have the following meanings: ■ 0: Print no output for the event. ■ 1: Print event records and output transaction data. ■ 2: Print event records, output transaction data, and file/ISN information. ■ 3: Print event records, output transaction data, file/ISN information, and the payload.	0-3	3
DTLCOMP	Sets the print level for TLOG records logged when a transaction is successfully output to a destination. Values have the following meanings: ■ 0: Print no output for the event. ■ 1: Print event records and output transaction data. ■ 2: Print event records, output transaction data, and file/ISN information. ■ 3: Print event records, output transaction data, file/ISN information, and the payload.	0-3	3

Parameter	Description	Valid Values	Default
DTLSLOGREAD	Sets the print level for TLOG records logged when a transaction is read from the SLOG. Values have the following meanings: ■ 0: Print no output for the event. ■ 1: Print event records and output transaction data. ■ 2: Print event records, output transaction data, and file/ISN information. ■ 3: Print event records, output transaction data, file/ISN information, and the payload.	0-3	3
DTLSLOGWRITE	Sets the print level for TLOG records logged when a transaction is written to the SLOG. Values have the following meanings: ■ 0: Print no output for the event. ■ 1: Print event records and output transaction data. ■ 2: Print event records, output transaction data, and file/ISN information. ■ 3: Print event records, output transaction data, file/ISN information, and the payload.	0-3	3
DTLTOUTPUT	Sets the print level for TLOG URBLTDOD records in the CLOG. Values have the following meanings: ■ 0: Print no output for the event. ■ 1: Print event records and output transaction data. ■ 2: Print event records, output transaction data, and file/ISN information. ■ 3: Print event records, output transaction data, file/ISN information, and the payload.	0-3	3
FROMDATE	Specifies a date in *yyyymmdd* format. TLOG records with this date or later will be eligible for printing. When no date is specified, records from the beginning of the TLOG file or later are eligible for printing. FROMDATE values must be less than or equal to any values specified for TODATE.	---	none
FROMTIME	Specifies a time in *hhmmssth* format. This parameter cannot be specified unless the FROMDATE parameter is specified also. TLOG records with this date and time or later will be eligible for printing. When no time is specified, records in the TLOG file from the beginning of the day specified by the FROMDATE parameter or later are eligible for printing.	---	none

Parameter	Description	Valid Values	Default
	`FROMDATE` and `FROMTIME` values must be less than or equal to any values specified for `TODATE` and `TOTIME`.		
`PAGESIZE`	Adjusts the number of lines per page in the DDDRUCK file.	A valid integer greater than or equal to 20.	58
`PRINT`	Identifies the print output format. Values have the following meanings: ■ "DUMP": Print a simple hexadecimal/character printout. ■ "FORMAT": Print the formatted block syntax. ■ "BOTH": Print both the DUMP and FORMAT formats of the data.	DUMP, FORMAT, or BOTH	FORMAT
`STATE`	Indicates whether state change TLOG records should be printed. Values have the following meanings: ■ "YES": Print state change TLOG records. ■ "NO": Don't print state change TLOG records.	YES or NO	YES
`STLFILTER`	Sets the print level for TLOG records logged when a transaction is excluded from replication due to filter specifications or a user exit. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records, filter reason, and output transaction data. ■ "2": Print event records, filter reason, output transaction data, and file/ISN information for the record that was rejected. ■ "3": Print event records, filter reason, output transaction data, file/ISN information, and the payload for the record that was rejected.	0-3	3
`STLIGNORE`	Set the print level for the event when a filter is ignored for an output transaction. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records containing filter details and source and target field details and values.	0-1	1

Parameter	Description	Valid Values	Default
STLINPUT	Sets the print level for TLOG records logged when a transaction is assigned to a subscription. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records and input transaction data. ■ "2": Print event records, input transaction data, and file/ISN information. ■ "3": Print event records, input transaction data, file/ISN information, and the payload.	0-3	3
STLMATCH	Set the print level for the event when a filter is matched by an output transaction. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records containing filter details and source and target field details and values.	0-1	1
STLNOMATCH	Set the print level for the event when a filter is not matched by an output transaction. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records containing filter details and source and target field details and values.	0-1	1
STLOUTPUT	Sets the print level for TLOG records logged when a transaction is output for a subscription. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records and output transaction data. ■ "2": Print event records, output transaction data, and file/ISN information. ■ "3": Print event records, output transaction data, file/ISN information, and the payload.	0-3	3
SUBSCRIPTION	Specifies the name of one or more subscription definitions for which TLOG records should be eligible for printing. Other TLOG records for other subscriptions would NOT be eligible for printing. **Note:** Records not eligible because they are not listed with this parameter may be selected if the DESTINATION parameter is also specified in the same ADARPP run. The SUBSCRIPTION and DESTINATION parameters are processed by ADARPP	Any valid subscription definition names	All records for all subscriptions are eligible for printing

Parameter	Description	Valid Values	Default
	using OR logic; that is, if both `SUBSCRIPTION` and `DESTINATION` are specified, a TLOG record need only meet the requirement of one of them to be eligible for printing. Up to 32 subscription names may be specified. If more than one is specified, they must be specified together in single quotes and separated by commas. For example: `ADARPP SUBSCRIPTION='SUBS01,SUBS02'`		
TIMEZONE	Adjusts the time zone prior to converting time values for the report. Values have the following meanings: ■ "LOCAL": Times for the report are adjusted by the local adjustment value found on the machine. ■ "MACHINE": Times are printed as they are found in the TLOG record. ■ `'[+|-]nn'` Times are adjusted backwards or forwards by the number of hours specified. Valid values for nn range from 0 through 23. Note that these values are listed in single quotes. For example: `ADARPP TIMEZONE='+6'`	LOCAL, MACHINE, or-23 to +23	LOCAL
TLCOMP	Sets the print level for TLOG records logged when a transaction has been completely processed by the Event Replicator Server. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records and input transaction data.	0-1	1
TLINPUT	Sets the print level for TLOG records logged for transaction assignment queue events. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records and input transaction data. ■ "2": Print event records, input transaction data, and file/ISN information. ■ "3": Print event records, input transaction data, file/ISN information, and the payload.	0-3	3

Parameter	Description	Valid Values	Default
TLISTATE	Sets the print level for TLOG records logged when an initial-state request is started. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event and URBS records. ■ "2": Print event and URBI records. ■ "3": Print event, URBS, and URBI records.	0-3	3
TLISTATECOMP	Sets the print level for TLOG records logged when an initial-state request is completed by the Event Replicator Server. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event and URBS records. ■ "2": Print event and URBI records. ■ "3": Print event, URBS, and URBI records.	0-3	3
TLNOSUB	Sets the print level for TLOG records logged when a transaction is not assigned to any subscription. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records and input transaction data. ■ "2": Print event records, input transaction data, and file/ISN information. ■ "3": Print event records, input transaction data, file/ISN information, and the payload.	0-3	3
TLQCOMP	Sets the print level for TLOG records logged prior to a transaction being assigned to the completion queue. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records and input transaction data. ■ "2": Print event records, input transaction data, and file/ISN information. ■ "3": Print event records and all input information available for the event.	0-3	3
TLREQERR	Sets the print level for TLOG records logged when a user request is rejected. Values have the following meanings: ■ "0": Print no output for the event.	0-3	3

Parameter	Description	Valid Values	Default
	■ "1": Print event and URBS records. ■ "2": Print event and URBI records. ■ "3": Print event, URBS, and URBI records.		
TLREQRECV	Sets the print level for TLOG records logged when a user request has been received. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records but no data. ■ "2": Print event records and input buffer, if available.	0-2	2
TLREQREJECT	Sets the print level for TLOG records logged when a user request is rejected due to an error. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event records but no data. ■ "2": Print event records and input buffer, if available.	0-2	2
TLRETRANS	Sets the print level for TLOG records logged when a retransmit (resend) request is received from the user. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event and URBS records. ■ "2": Print event and URBI records. ■ "3": Print event, URBS, and URBI records.	0-3	3
TLSTATUS	Sets the print level for TLOG records logged when a status request is received from the user. Values have the following meanings: ■ "0": Print no output for the event. ■ "1": Print event and URBS records. ■ "2": Print event and URBI records. ■ "3": Print event, URBS, and URBI records.	0-3	3
TODATE	Specifies a date in *yyyymmdd* format. TLOG records with this date or earlier will be eligible for printing. When no date is specified, records to the end of the TLOG file or earlier are eligible for printing. FROMDATE values must be less than or equal to any values specified for TODATE.	---	none

Parameter	Description	Valid Values	Default
TOTIME	Specifies a time in *hhmmssth* format. This parameter cannot be specified unless the TODATE parameter is specified also. TLOG records with this date and time or earlier will be eligible for printing. When no time is specified, records in the TLOG file from the end of the day specified by the TODATE parameter or earlier are eligible for printing. FROMDATE and FROMTIME values must be less than or equal to any values specified for TODATE and TOTIME.	---	none

JCL Requirements and Examples

The following job control information must be specified to run the ADARPP utility on z/OS systems:

Item	DD Name	Storage	Description
TLOG file, created by ADARES CLCOPY	DDCLOG	disk	Copied form of the command log (CLOG) file.
ADARPP report and messages	DDDRUCK	printer	Parameters, messages, and printed report.
ADARUN messages	DDPRINT	printer	Messages and codes
ADARUN parameters	DDCARD	reader	Operations
ADARPP parameters	DDKARTE	reader	ADARPP selection and print criteria.

A sample job, PRITLOG, is provided in the MVSJOBS data set.

16   RPULD and RPLOD Utilities

The RPULD (unload) and RPLOD (load) utilities can be used to unload and load Replicator system file definitions to and from Natural workfile 1.

■ The RPULD utility can be used to unload Replicator system file definitions to Natural workfile 1.

The resulting file can be read in as DDKARTE statements in an Event Replicator Server startup job. There are two situations in which this will not work. If the value of either the `SFBAI` or `SFBBI` parameters in an SFILE definition are longer than 40 bytes, you cannot read the RPULD utility output as DDKARTE input to an Event Replicator Server startup job. To resolve this, you must define a global format buffer, using either the `SGFORMATAI` or `SGFORMATBI` parameters, to replace the corresponding `SFBAI` or `SFBBI` parameters.

■ The RPLOD utility can be used to load definitions in Natural workfile 1 to a Replicator system file.

The input definitions are a replacement of the definitions defined on the file, and it is not a "merge" process. For example, if you specify a subscription name, you should specify all the sub-parameters necessary to define it, or if you specify any of the global values as input to your RPLOD, you should specify all of them, as any missing global values will have "default values" assigned to them, which may not be your desired settings.

These utilities share a common syntax, with the exception of a couple of parameters. In addition, both utilities can be run as batch jobs or using the Adabas Event Replicator Subsystem.

This chapter provides instructions for the batch method of running these utilities. For information on running them using the Adabas Event Replicator Subsystem, read *Identifying, Loading, and Unloading the Replicator System File* in *Adabas Event Replicator Subsystem User's Guide*.

JCL Requirements

The RPULD and RPLOD utilities can run as Natural batch jobs using Natural workfile 1. The following JCL requirements must be met by the Natural batch job:

■ Specifications for Natural workfile 1 must be included in the JCL for the job. The DD name for Natural workfile 1 is CMWKF01, with a record format of F or FB and a record length of 80. The BS2000/OSD link name is W01.

■ The SAG editor is used internally by the RPLOD and RPULD utilities. Consequently, the `EDPSIZE` parameter must be specified in the JCL. For complete information about the `EDPSIZE` parameter, refer to your Natural documentation.

Sample JCL for the RPULD and RPLOD utilities is provided in sample members RPLOD and RPULD of the MVSJOBS data set. You will need to modify these jobs for your use.

Syntax

The syntax of the RPULD and RPLOD utilities is almost identical, with the exception of the RELATED and REPLACE parameters. The RELATED parameter is only available when running the RPULD parameter; the REPLACE parameter is only available when running the RPLOD utility.

The following diagram depicts the syntax of the RPULD and RPLOD utilities:

```
RPLOD | RPULD  [ ENTITY ={ * | GF | DE | SB | TF | IQ | RB | RP | IS } ]
               [ NAME = definition-name ]
               DBID = dbid
               FNR = file-number
               [ PASSWORD = password ]
               [ CIPHER = cipher-code ]
               [ RELATED ={ Y | N } ]*
               [ REPLACE ={ Y | N } ]**
               [ STATUS ={ C | S } ]


*  Parameter available only for the RPULD utility
** Parameter available only for the RPLOD utility
```

> **Note:** When specifying parameters for RPLOD or RPULD, be sure you use the Natural input delimiter character identified in Natural's ID parameter to separate the parameters specified for the RPLOD or RPULD run. The default Natural input delimiter character is a comma, but if this has been overridden at your site using the ID parameter, you will need to use the one specified in the ID parameter. For more information, refer to your Natural documentation.

Parameters are described in the next section. You can specify them using their keywords or as positional parameters. If you elect to specify them as positional parameters, they must be specified in the order shown above. Examples are provided in *Examples* elsewhere in this section.

Parameters

The following table describes each of the RPULD and RPLOD utility parameters in alphabetic order.

Parameter	Description	Required?	Default
CIPHER	The cipher code necessary to access the Replicator system file identified by the DBID and FNR parameters.	No	No cipher code is used to access the Replicator system file.
DBID	The database ID of the database to which the Replicator system file applies. When running the RPULD utility, this is the DBID of the database with the Replicator system file containing definitions you want to unload to Natural workfile 1. When running the RPLOD utility, this is the DBID of the database with the Replicator system file to which you definitions loaded from Natural workfile 1.	Yes	No default.
ENTITY	Identifies a specific definition type to load or unload. Valid values are: ■ blank: all definitions are selected ■ "*": all definitions are selected ■ "DE": destination definitions are selected ■ "GF": global format buffer definitions are selected ■ "IQ": input queue definitions are selected ■ "IS": initial-state definitions are selected ■ "RB": resend buffer definitions are selected ■ "RP": Replicator parameters / global values are selected ■ "SB": subscription definitions and their associated SFILE definitions are selected ■ "TF": transaction filter definitions are selected	No	All definitions with names that match the NAME parameter specifications are selected for utility processing.
FNR	The file number of the Replicator system file with the database identified by the DBID parameter. When running the RPULD utility, this is the file number of the Replicator system file from which definitions should be unloaded to Natural workfile 1. When running the RPLOD utility, this is the file number of the Replicator system file to which definitions should be loaded from Natural workfile 1.	Yes	No default.
NAME	The name of the definitions to select. Caution should be exercised before using any special characters as part of the name of the definition, as there are the wildcards "*", ">", and "<" that can be used to limit selections.	No	All definitions of the types specified by the ENTITY parameter are

Parameter	Description	Required?	Default
	For example, `AB4*` selects all definitions with names starting with the letters "AB4". As another example, `AB4>` selects all definitions with names that sort (alphanumerically) after and including names starting with the letters "AB4". If you have specified entries containing the wildcard characters ("*", ">", "<"), and you are trying to use the wildcards for selection, or to select an exact named definition, your results may not be as expected. This may also occur with the "," (comma), or "'" (single quote) characters. For example, `A>B*` would not return all definitions with names starting with "A>B", but everything beginning with A and greater. And if a name of the definition has a "'" (single quote) at both the beginning and end of the name, then it will require doubling those single quote entries at the beginning and end of the name to properly work with RPULD and RPLOD. For example, `AB'C` works, it will be accepted as "AB'C", however `'ABC'` would have to be entered as `''ABC''` (2 single quotes at beginning and end) for it to be accepted and loaded as "'ABC'".		selected for utility processing.
PASSWORD	The password necessary to access the Replicator system file identified by the `DBID` and `FNR` parameters.	No	No password is used to access the Replicator system file.
RELATED	Whether or not related definitions called by a selected definition should also be unloaded. This parameter is available only for RPULD runs. Valid values are "Y" (all related definitions should also be unloaded) or "N" (all related definitions should not be unloaded unless they are selected directly through other RPULD parameter settings).	No	Y
REPLACE	Whether or not existing definitions should be replaced with the newly loaded definitions. This parameter is available only for RPLOD runs. Valid values are "Y" (existing definitions should be replaced) or "N" (existing definitions should not be replaced).	No	N
STATUS	The version of the selected definitions that should be loaded or unloaded. Valid values are "C" (current versions) or "S" (scheduled versions).	No	All versions are selected for utility processing.

Examples

The following examples are shown using both parameter keywords and positional parameters. If you elect to use positional parameters, they must be specified in the order shown in the **syntax** diagram, elsewhere in this section.

- RPULD Example
- RPLOD Example

Note: The examples in this section use a comma to delimit the RPULD and RPLOD parameters. This is the Natural default input delimiter character. If your site has overridden this default using Natural's ID parameter, you will need to use the delimiter character specified by the ID parameter instead of a comma. For more information, refer to your Natural documentation.

RPULD Example

The following RPULD example will unload all definitions from the specified Replicator system file with names starting with the characters "ABC". Definitions will be unloaded from Replicator system file (file "3") on database "18013". Definitions related to the selected definitions (those with names starting with "ABC") will also be unloaded. No password or cipher code are necessary when accessing the Replicator system file .

- Using Parameter Keywords
- Using Positional Parameters

**Using Parameter Keywords**

```
RPULD NAME=ABC*,DBID=18013,FNR=3,RELATED=Y
```

Using Positional Parameters

```
RPULD ,ABC*,18013,3,,,Y,
```

The RPULD utility specifications given in this sample are diagramed below:
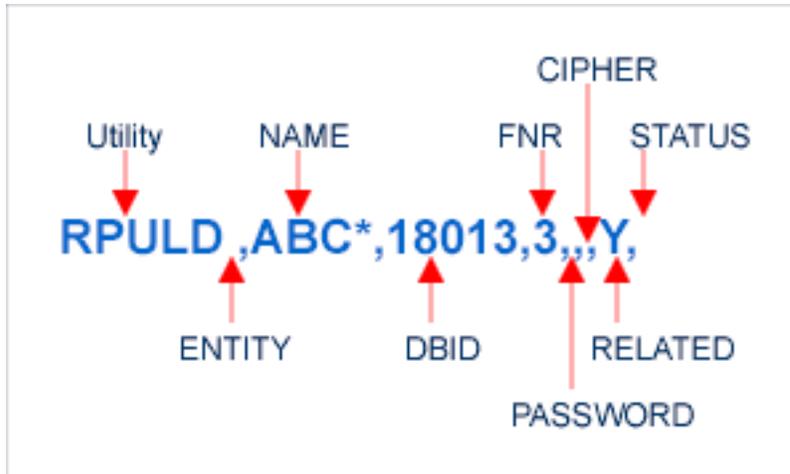
RPLOD Example

The following RPLOD example will load all definitions with names starting with the characters "ABC". The selected definitions will be loaded into the Replicator system file (file "3") on database "18013", replacing existing records in the file, if necessary. No password or cipher code are necessary when accessing the Replicator system file.

- Using Parameter Keywords
- Using Positional Parameters

Using Parameter Keywords

```
RPLOD NAME=ABC*,DBID=18013,FNR=3,REPLACE=Y
```

Using Positional Parameters

```
RPLOD ,ABC*,18013,3,,,Y,
```

The RPLOD utility specifications given in this sample are diagramed below:
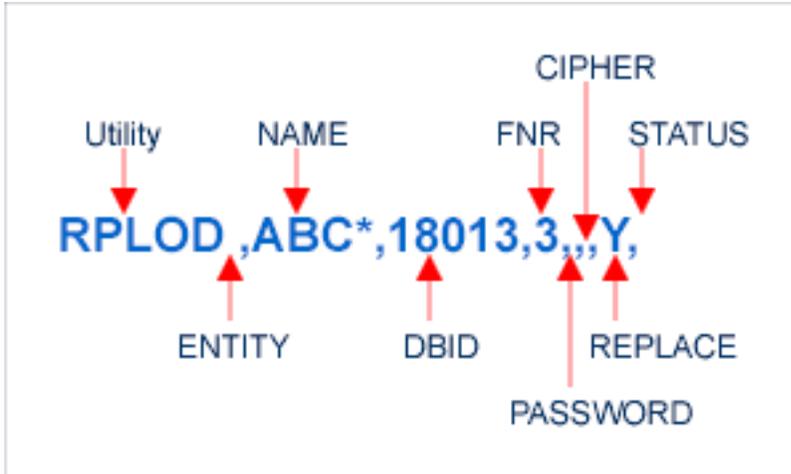
Sample Natural Batch Jobs

The following are sample Natural batch jobs that run the RPLOD and RPULD utilities. They are provided in members RPLOD and RPULD of the MVSJOBS data set. You will need to modify these jobs for your use.

- RPLOD Sample
- RPULD Sample

RPLOD Sample

```
//RPLOD    JOB
//*
//*
//*
//* LOAD DEFINITIONS FROM WORK FILE 1
//* INTO REPTOR SYSTEM FILE
//*
//*
//NATBAT    EXEC PGM=NATvrsBA,
// PARM=('IM=D,ID='','',MADIO=0,MAXCL=0,MT=0,',
//       'EDPSIZE=64,INTENS=1')
//STEPLIB   DD DSN=NATURAL.Vvrs.LOAD,DISP=SHR      <=== NATURAL LOAD
//*
//DDPRINT   DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//DDCARD    DD *
ADARUN DB=YYYYY,DE=3390,SVC=XXX,MODE=MULTI
//CMPRINT   DD SYSOUT=*
//CMWKF01   DD DSN=EXAMPLE.REPTOR.PARMS,
//             DISP=SHR
//CMSYNIN   DD *
```

```
LOGON SYSRPTR
RPLOD NAME=*,DBID=YYYYY,FNR=FFFFF,REPLACE=Y
/*                                                                ↵
```

RPULD Sample

```
//RPULD    JOB
//*
//*
//*
//* UNLOAD DEFINITIONS FROM REPTOR SYSTEM FILE
//* TO WORK FILE 1
//*
//*
//NATBAT    EXEC PGM=NATvrsBA,
// PARM=('IM=D,ID='','',MADIO=0,MAXCL=0,MT=0,',
//       'EDPSIZE=64,INTENS=1')
//STEPLIB   DD DSN=NATURAL.Vvrs.LOAD,DISP=SHR     <=== NATURAL LOAD
//*
//DDPRINT   DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//DDCARD    DD *
ADARUN DB=YYYYY,SVC=XXX,MODE=MULTI
//CMPRINT   DD SYSOUT=*
//CMWKF01   DD DSN=EXAMPLE.REPTOR.PARMS,
//            DCB=(RECFM=FB,LRECL=80),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE)
//CMSYNIN  DD *
LOGON SYSRPTR
RPULD NAME=*,DBID=YYYYY,FNR=FFFFF,RELATED=Y
/*                                                                ↵
```

17 Event Replicator Initialization Parameters

To use Event Replicator for Adabas and customize its processing, you must supply various replication definitions. These definitions can be specified in two ways:

- Specified as initialization parameters, which are read from the DDKARTE statements of the Event Replicator Server startup job. For more information, read this chapter.

- Read from the Replicator system file at Event Replicator Server startup. The definitions are maintained in the Replicator system file using the Adabas Event Replicator Subsystem, the provided online menu-driven interface. For more information, read *Using the Adabas Event Replicator Subsystem* in the *Adabas Event Replicator Subsystem User's Guide*.

  > **Note:** If you use a Replicator system file to store your replication definitions, you can also use the RPLREFRESH command to refresh resource definitions in your Event Replicator Server configuration while the Event Replicator Server is still running. For more information, read *RPLREFRESH Command*, elsewhere in this guide.

You can elect to use one or both methods, depending on what works best for you. The method used is controlled by ADARUN parameter `RPLPARMS`.

This chapter describes the initialization parameters you can specify in the DDKARTE statements of the Event Replicator Server startup job. Each input record read from DDKARTE must start with the characters ADARPD. In addition, be sure to follow the conventions described in *Conventions*, elsewhere in this guide, when using these initialization parameters.

DATABASE Parameter

By default, Event Replicator for Adabas attempts to connect with any Adabas database encountered during an Event Replicator Server session. You may, however, prefer to control these connection attempts using an Adabas database connection definition. This section describes the syntax and parameters of database connection definitions.

Syntax:

```
DATABASE ID=dbid
     [DBCONNECT={YES|NO}]
```

This section covers the following DATABASE parameters and subparameters:

- ID Parameter
- DBCONNECT Parameter

- Examples

ID Parameter

Parameter	Specify . . .	Possible Values	Default
`ID`	the database ID of an Adabas database for which you want to control connection attempts.	numeric; 1-65,535	none

By default, the Event Replicator Server attempts to connect to every Adabas database it encounters. If you would like to control these connection attempts, use this parameter and a corresponding `DBCONNECT` parameter.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Database ID` parameter in *Controlling Connections to Adabas Databases*, in the *Adabas Event Replicator Subsystem User's Guide*.

DBCONNECT Parameter

Parameter	Specify . . .	Possible Values	Default
`DBCONNECT`	whether or not the Event Replicator Server specified in the preceding `DATABASE` parameter should initiate an attempt to connect to the Adabas database at Event Replicator Server startup.	"YES" or "NO"	YES

If you specify `DBCONNECT=YES`, the Event Replicator Server will initiate an attempt to connect to the Adabas database identified by the preceding `DATABASE` parameter after the Event Replicator Server starts.

If you specify `DBCONNECT=NO`, the Event Replicator Server will not initiate an attempt to connect to the Adabas database identified by the preceding `DATABASE` parameter. The database may connect with the Event Replicator Server at some point, but the Event Replicator Server will not initiate the connection at Event Replicator Server startup.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Connect` parameter in *Controlling Connections to Adabas Databases*, in the *Adabas Event Replicator Subsystem User's Guide*.

Examples

In the following example, the Event Replicator Server will *not* initiate a connection to Adabas database 203.

```
ADARPD DATABASE ID=203
ADARPD DBCONNECT=NO
```

In the following example, the Event Replicator Server will initiate a connection to Adabas database 203, but not to database 204.

```
ADARPD DATABASE ID=203,DBCONNECT=YES
ADARPD DATABASE ID=204,DBCONNECT=NO
```

DBID (Default Database ID) Parameter

Syntax:

```
[DBID=dbid]
```

Parameter	Specify . . .	Possible Values	Default
DBID	a default database ID.	numeric; 1-65,535	none

A default database ID may be specified. The default value is used when subscription-related parameter SFDBID is *not* specified for a combination of subscription and file.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Replicator Server DBID` field in *Changing the Event Replicator Server and Replicator System File*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
[DBID=database-ID]
```

DESTINATION Parameter

The Event Replicator Server supports the following destination types: Adabas, EntireX Broker, File, WebSphere MQ, or Null. Different subparameters are used with the NAME parameter, depending on the destination type.

Syntax:

```
DESTINATION NAME=destination-name
     [DACTIVE={YES|NO}]
```

```
    [DCLASS=name]
    [DCLASSPARM=parmdata]
    [DCOMMITTHRESHOLD=5]
    [DEXIT=name]
    [DEXITWORKSIZE=size]
    [DLOG={YES|NO}]
    [DMAXOUTPUTSIZE|\=n[K]]
    [DOPEN={YES|NO|GLOBAL}]
    [DREPLICATEUTI={YES|NO}]
    [DRETRYCOUNT={n|GLOBAL}]
    [DRETRYINTERVAL={n|GLOBAL}]
    [DTLASSIGN={0|1|2|3 }]
    [DTLCOMP={0|1|2|3 }]
    [DTLSLOGREAD={0|1|2|3 }]
    [DTLSLOGWRITE={0|1|2|3 }]
    DTYPE={ADABAS|ETBROKER|FILE|MQSERIES|NULL}
```

The following additional parameter can be used or all destination types supported by Event Replicator for Adabas except Adabas destinations (DTYPE=ADABAS):

```
[DSTATLOG={YES|NO}]
```

The following additional parameters are used when DTYPE=ADABAS is used:

```
[DTLADABAS={0|1|2|3}]
[DAERROR={BACKOUT|ALTACTION|CLOSE}]
DAIFILE=input-file-number
    DAIDBID=input-dbid
    DATDBID=target-dbid
    DATFILE=target-file-number
    [DAREPLICATEUTI={YES|NO}]
    [DATMETHOD={ISN|KEY}]
    [DATKEYSB=search-buffer]
    [DATKEYAIO=after-image-key-offset]
    [DATKEYAIL=after-image-key-length]
    [DATKEYBIO=before-image-key-offset]
    [DATKEYBIL=before-image-key-length]
    [DATKEYKYO=key-buffer-key-offset]
    [DATKEYKYL=key-buffer-key-length]
  ↩
```

The following additional parameters are used when DTYPE=ETBROKER is used:

```
[DARC=architecture-key]
[DETBBROKERID=entire-broker-id]
[DETBSERVICE=entire-broker-service]
[DETBSERVICENAME=entire-broker-service-name]
[DETBSERVICECLASS=entire-broker-service-class]
[DETBSCONV={YES|NO}]
[DETBTOKEN=entire-broker-token]
[DETBUSERID=entire-broker-userid]
[DEVENTLOG={YES|NO}]
```

```
[DHBINTERVAL=n]
[DQFULLDELAY={n|GLOBAL}]
```

The following additional parameters are used when DTYPE=MQSERIES is used:

```
[DARC=architecture-key]
[DEVENTLOG={YES|NO}]
[DHBINTERVAL=n]
[DMQQMGRNAME=mqseries-queue-manager-name]
[DMQQNAME=mqseries-queue-name]
[DMQDYNQNAME=mqseries-dynamic-queue-name]
[DMQFORMAT=format]
[DMQCCSID=ccsid]
[DQFULLDELAY={n|GLOBAL}]
```

> **Note:** If you are running on z/OS 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.

The following additional parameters are used when DTYPE=FILE is used:

```
[DEVENTLOG={YES|NO}]
```

The following additional parameters are used when DTYPE=NULL is used:

```
[DARC=architecture-key
[DEVENTLOG={YES|NO}]
[DHBINTERVAL=n]
```

This section covers the following topics:

- NAME Parameter
- DACTIVE Parameter
- DAERROR Parameter
- DAIDBID Parameter
- DAIFILE Parameter
- DARC Parameter
- DAREPLICATEUTI Parameter
- DASUBSCRIPTION Parameter
- DATDBID Parameter
- DATFILE Parameter
- DATKEYAIL Parameter
- DATKEYAIO Parameter
- DATKEYBIL Parameter
- DATKEYBIO Parameter
- DATKEYKYL Parameter
- DATKEYKYO Parameter
- DATKEYSB Parameter

- Special Information about Destinations with DTYPE=ADABAS

NAME Parameter

Parameter	Specify . . .	Possible Values	Default
NAME	the name of the output destination definition.	alphanumeric; 1-8 characters	none

Parameter NAME is required and is used to uniquely identify an output destination definition. This value is only referenced within the Event Replicator Server. The value may be used when the Event Replicator Server writes a related message.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Destination Name` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. All destination definitions require a name, regardless of destination type.

DACTIVE Parameter

Parameter	Specify . . .	Possible Values	Default
DACTIVE	whether the destination should be activated at Event Replicator Server startup.	"YES" or "NO"	YES

If "YES" is specified, the destination is marked as active at Event Replicator Server startup; if "NO" is specified, the destination is marked as inactive at Event Replicator Server startup (it is inactive).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Destination Active` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DAERROR Parameter

Parameter	Specify . . .	Possible Values	Default
DAERROR	the action to be taken when an error occurs during replication to an Adabas destination.	"BACKOUT", "ALTACTION", or "CLOSE"	ALTACTION

This parameter is valid only when the **destination type** is Adabas (DTYPE=ADABAS).

Additional actions are taken, based on the value of this DAERROR parameter. These actions are described in the following table:

Value	Processing
ALTACTION	Processing continues with the next update that is part of the same transaction. Some special processing occurs when DAERROR=ALTACTION: ■ If an insert is processed and the record already exists, the record is updated. ■ If an update is processed and the record does not exist, the record is inserted. ■ If a delete is processed and the record does not exist, processing continues with the next record. For other errors, the record is skipped.
BACKOUT	A message is issued indicating that the transaction will be backed out and then ignored. The current transaction is backed out and processing continues with the next transaction to be replicated.
CLOSE	A message is issued indicating that the transaction will be backed out and the destination will be closed. The current transaction is backed out and transaction logging (to the SLOG file) will begin, if defined for the destination.

In all cases (ALTACTION, BACKOUT, and CLOSE), if response 148 is returned and the SLOG system file is available, the destination is closed.

If an insert, update, or delete operation fails because a replicated record already exists or does not exist, an appropriate message is issued. If the transaction fails because of an error, a message containing the two-character Adabas command, the database ID, the file number, the response code and the subcode is written. If the `DATMETHOD` parameter is set to ISN, the text "ISN" will be appended to this message as well as the ISN value. If the `DATMETHOD` parameter is set to "KEY", the record key will be written in both hexadecimal and readable formats in a separate message.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Error Action` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DAIDBID Parameter

Parameter	Specify . . .	Possible Values	Default
DAIDBID	an input database ID associated with a DAIFILE parameter.	numeric; 1-65,535	none

For each combination of parameters DAIDBID and DAIFILE, the related subscription must have the same combination of SFDBID and SFILE.

Combinations of parameters DAIDBID and DAIFILE must be unique within a single Adabas destination definition. This implies that the input from a database and file combination can only have a single Adabas target within a specific Adabas destination definition. If you want the input from a database and file combination to go to more than one Adabas target, define multiple Adabas destination definitions.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Input DBID` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DAIFILE Parameter

Parameter	Specify . . .	Possible Values	Default
DAIFILE	the number of an input file for an Adabas destination definition.	numeric file number	none

Parameter DAIFILE must be specified at least once when the destination has DTYPE=ADABAS. Parameter DAIFILE may be specified one or more times for a destination with DTYPE=ADABAS. Sub-parameters (DAIDBID, DATDBID, and DATFILE) will be specified for each occurrence of DAIFILE.

For each combination of parameters DAIDBID and DAIFILE, the related subscription must have the same combination of SFDBID and SFILE.

Combinations of parameters DAIDBID and DAIFILE must be unique within a single Adabas destination definition. This implies that the input from a database and file combination can only have a single Adabas target within a specific Adabas destination definition. If you want the input from a database and file combination to go to more than one Adabas target, define multiple Adabas destination definitions.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Input File` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DARC Parameter

Parameter	Specify . . .	Possible Values	Default
DARC	defines the data architecture for fields in the URB* control structures sent to a webMethods EntireX, WebSphere MQ, or NULL destination.	integer sum of byte order and encoding family (*see below*)	2 (byte order 0 + encoding family 2)

The `DARC` parameter value is the sum of the following numbers:

Control Structure Element	Values	Description
Byte order (b)	0	High order byte first
	1	Low order byte first
Encoding Family (e)	0	ASCII encoding family
	2	EBCDIC encoding family

This parameter is not valid for file or Adabas destinations (DTYPE=FILE or DTYPE=ADABAS).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Architecture` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DAREPLICATEUTI Parameter

Parameter	Specify . . .	Possible Values	Default
`DAREPLICATEUTI`	whether Adabas utility change replication should be activated for the target file in the Adabas destination at Event Replicator Server startup.	"YES" or "NO"	NO

If "YES" is specified, utility replication is activated for the file in and Adabas destination at Event Replicator Server startup; if "NO" is specified, utility replication is not activated for the file.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Replicate Utility` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

For more information about replicating utility functions, read *Replicating Adabas Utility Functions*, in *Event Replicator for Adabas Concepts*.

DASUBSCRIPTION Parameter

⚠️ **Important:** This parameter is no longer used; it remains only for compatibility with past versions of Event Replicator for Adabas.

Parameter	Specify . . .	Possible Values	Default
`DASUBSCRIPTION`	an input subscription name associated with the destination with DTYPE=ADABAS.	subscription name	none

DATDBID Parameter

Parameter	Specify . . .	Possible Values	Default
`DATDBID`	the target database ID for replication of the input data.	numeric; 1-65,535	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Target DBID` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DATFILE Parameter

Parameter	Specify . . .	Possible Values	Default
`DATFILE`	the target file for replication of the input data.	file name	none

When DTYPE=ADABAS is specified for the destination, replicated data for the input subscription, input database, and input file number will be applied to the related target Adabas database ID and file number.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Target File` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DATKEYAIL Parameter

Parameter	Specify . . .	Possible Values	Default
`DATKEYAIL`	the length (in bytes) of the key to be used in the after image of the data buffer.	1-32,767	none

When the `DATMETHOD` parameter is set to KEY, this parameter is required. If the `DATMETHOD` parameter is set to ISN, this parameter may not be specified.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `After Image Key Length` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DATKEYAIO Parameter

Parameter	Specify . . .	Possible Values	Default
`DATKEYAIO`	the offset (in bytes) of the key to be used in the after image of the data buffer.	0 - 2,147,483,646	0

This optional parameter can only be set if the `DATMETHOD` parameter is set to KEY; if the `DATMETHOD` parameter is set to ISN, this parameter may not be specified.

The sum of this parameter and the length of the key to be used in the after image (`DATKEYAIL` parameter), must be less than or equal to 2,147,483,647 bytes.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `After Image Offset` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DATKEYBIL Parameter

Parameter	Specify . . .	Possible Values	Default
`DATKEYBIL`	the length (in bytes) of the key to be used in the before image of the data buffer.	1-32,767	none

This optional parameter can only be set if the `DATMETHOD` parameter is set to KEY; if the `DATMETHOD` parameter is set to ISN, this parameter may not be specified.

If the `DATKEYBIO` parameter is specified, this parameter must also be specified with a value greater than zero.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Before Image Key Length` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DATKEYBIO Parameter

Parameter	Specify . . .	Possible Values	Default
`DATKEYBIO`	the offset (in bytes) of the key to be used in the before image of the data buffer.	0 - 2,147,483,646	0

This optional parameter can only be set if the `DATMETHOD` parameter is set to KEY; if the `DATMETHOD` parameter is set to ISN, this parameter may not be specified.

The sum of this parameter and the length of the key to be used in the before image (`DATKEYBIL` parameter), must be less than or equal to 2,147,483,647 bytes.

If this parameter is specified, the `DATKEYBIL` parameter must also be specified with a value greater than zero.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Before Image Offset` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DATKEYKYL Parameter

Parameter	Specify . . .	Possible Values	Default
`DATKEYKYL`	the length (in bytes) of the key to be used in the before image of the primary key.	1-32,767	none

This optional parameter can only be set if the `DATMETHOD` parameter is set to KEY; if the `DATMETHOD` parameter is set to ISN, this parameter may not be specified.

If the `DATKEYKYO` parameter is specified, this parameter must also be specified with a value greater than zero.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Key Length` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DATKEYKYO Parameter

Parameter	Specify . . .	Possible Values	Default
`DATKEYKYO`	the offset (in bytes) of the key to be used in the before image of the primary key.	0-32,767	0

This optional parameter can only be set if the `DATMETHOD` parameter is set to KEY; if the `DATMETHOD` parameter is set to ISN, this parameter may not be specified.

The sum of this parameter and the length of the key to be used in the before image (`DATKEYKYL` parameter), must be less than or equal to 32,767 bytes.

If this parameter is specified, the `DATKEYKYL` parameter must also be specified with a value greater than zero.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Key Offset` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DATKEYSB Parameter

Parameter	Specify . . .	Possible Values	Default
`DATKEYSB`	the name of a search buffer to be used for keyed replication.	up to 60 alphanumeric characters	none

When the `DATMETHOD` parameter is set to KEY, this parameter is required. If the `DATMETHOD` parameter is set to ISN, this parameter may not be specified.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Search Buffer` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DATMETHOD Parameter

Parameter	Specify . . .	Possible Values	Default
DATMETHOD	the method to be used when searching for a record on the target database.	"ISN" or "KEY"	ISN

This parameter is valid only when the **destination type** is Adabas (DTYPE=ADABAS).

When DATMETHOD=KEY is specified, the subparameters **DATKEYSB**, **DATKEYAIO**, **DATKEYAIL**, **DATKEYBIO**, **DATKEYBIL**, **DATKEYKYO**, and **DATKEYKYL** may also be supplied to indicate where to find the key in the replicated data. All of these subparameters are optional except **DATKEYSB** and **DATKEYAIL**.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Replication Method` parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DCLASS Parameter

Parameter	Specify . . .	Possible Values	Default
DCLASS	the class type for this destination definition.	SAGTARG or SAGTRGB	none

The DCLASS parameter specifies the destination class. There are two possible values for this parameter: SAGTARG or SAGTRGB. There is no default. You should chose one of these values if the output is to be sent to the Event Replicator Target Adapter for the destination.

When set to SAGTARG the output produced is in an XML format that can be processed by the Event Replicator Target Adapter. If it is set to SAGTRGB the output is in a binary raw data format that can be processed by the Event Replicator Target Adapter.

Otherwise, leave this parameter blank unless otherwise requested by a Software AG support representative.

> **Notes:**

1. DCLASS=SAGTARG cannot be specified for destinations with DTYPE=ADABAS or FILE. It is only valid for webMethods EntireX, WebSphere MQ, or NULL destinations. When DCLASS=SAGTARG is specified, the ADARUN RPLPARMS parameter must be set to "FILE" or "BOTH" to provide access to any field table (GFFT) definitions.

2. The Adabas file should be defined with the RPLDSBI option. When this option is on, the before image of data storage is included with the replication data for updates. SAGTARG compares the after image fields with the before image fields (when available), to determine if the field was changed, and only sends the field if it was changed. If the before image is not included, then there is no way for SAGTARG to know if a field was changed or not and has to send all

the fields in the after image. This can result in unneeded overhead. In addition, the before image is needed by Event Replicator Target Adapter for processing updated MU's. Failure to include it can result in duplicate key errors. See also the section *Restrictions and Limitations*, in the *Event Replicator Target Adapter Release Notes*.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Destination Class` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DCLASSPARM Parameter

Parameter	Specify . . .	Possible Values	Default
`DCLASSPARM`	up to 120 bytes of character data to be passed to the optional destination output user exit.	any character data	none

The DCLASSPARM parameter specifies up to 120 bytes of character data to be passed to the optional destination output user exit.

DCLASSPARM cannot be specified for destinations with DTYPE=ADABAS or DTYPE=FILE and can only be specified if a value is set for the `DCLASS` or `DEXIT` parameters.

If either `DCLASS=SAGTARG` or `DCLASS=SAGTRGB` is specified to invoke support for the Event Replicator Target Adapter, you may want to specify one or more of the following keyword parameters:

> **Note:** These parameter keywords must be specified in uppercase.

ADA

The ADA keyword is available for use with the `SAGTARG` DCLASS exit.

Specify the ADA keyword in the DCLASSPARM parameter if the target of the destination requires the Adabas 2-byte short name be included with the schema for all fields.

NOSPRE

The NOSPRE keyword is available for use with both `SAGTARG` and `SAGTRGB` DCLASS exits.

Specify the NOSPRE keyword in the DCLASSPARM parameter if you do not want the subscription name to prefix the names of the tables produced by the Event Replicator Target Adapter. When NOSPRE is specified, the schema file name (Predict view name) alone is used for the table names; when NOSPRE is *not* specified, the subscription name prefixes the schema file name in the table names.

> **Note:** Oracle identifiers are limited to 30 characters. If NOSPRE is *not* specified and an Oracle RDBMS is used by the Event Replicator Target Adapter, the identifier names may exceed 30 characters and errors may occur. We recommend using NOSPRE if an Oracle RDBMS is also used.

SPRE

Specify the SPRE keyword in the DCLASSPARM parameter if you do want the subscription name to prefix the names of the tables produced by the Event Replicator Target Adapter.

The notation is `SPRE=xxxxxxxx`. Where *xxxxxxxx* can be a subscription prefix to be used instead of the subscription name used in the job execution. The prefix can be 1 to 8 characters in size.

The SPRE keyword is available for use with both `SAGTARG` and `SAGTRGB` DCLASS exits.

OPTIONS

The OPTIONS keyword parameter can be used to specify options for the destination. Specify the OPTIONS keyword parameter, using the syntax `OPTIONS=nnnn`. Possible values of the OPTIONS (*nnnn*) are listed in the following table. However, if you want to combine options, add their values together and enter the total value.

For example, if you want to combine option 32 (to send the full image on an update) and option 64 (to set the XML transaction committed time value to local time instead of GMT/UTC), specify `OPTIONS=96` (with 96 being the sum of 32 and 64).

Option Value	Description
1	Specify `OPTIONS=1` to send the full image on an insert.
2	Specify `OPTIONS=2` to indicate that long names should be used. This option will cause long names to be sent in place of the default short names used for various elements and attributes. Short names are the default and save on the amount of data being transferred. Long names make for better readability. For example, the short name <F> would appear as <Field> using long names.
4	Specify `OPTIONS=4` to ensure that invalid XML characters found in alphanumeric fields are not translated to spaces.
8	Specify `OPTIONS=8` to ensure that trailing blanks in alphanumeric fields are not removed.
16	Specify `OPTIONS=16` to ensure that characters used by XML are not replaced automatically with predefined entity references. For example, if `OPTIONS=16` is set, the ampersand (&) character would not be replaced with the literal "&".
32	Specify `OPTIONS=32` to send the full image on an update. The full before image (if available) and after image of all fields are sent for an update, even if the field values were not changed or are null. **Note:** This option is mutually exclusive with option 128. If both options are specified, then `OPTIONS=32` will be set and `OPTIONS=128` ignored.
64	Specify `OPTIONS=64` to set the XML transaction committed time value to local time instead of GMT/UTC.

Option Value	Description
128	Specify `OPTIONS=128` if the before image is not to be included when a field is updated/changed if the before image data storage records are included. Primary field before images will still be included because the key is needed to find the record that is to be changed. **Note:** This option is mutually exclusive with option 32. If both options are specified, then `OPTIONS=32` will be set and `OPTIONS=128` ignored.
256	Specify `OPTIONS=256` to suppress ARMXL004I information messages.

TRACE

The TRACE keyword parameter can be used to specify the contents of the trace. Specify the TRACE keyword parameter using the syntax `TRACE=`*nnnn*. Possible values of the TRACE (*nnnn*) are listed in the following table. However, if you want to trace multiple control blocks, add their trace values together and enter the total value. For example, to trace the before and after images of the URBD control blocks, you would specify `TRACE=24` because the sum of 8 (URBD control block before image) and 16 (URBD control block after image) is 24.

> **Note:** If tracing is enabled using this keyword parameter, be sure to include the following JCL statement in the startup JCL of the Event Replicator Server:
> `//DDTRACE1 DD SYSOUT=X`

Settings for DCLASS=SAGTARG

Trace Value	Description
1	Trace the URBS control block.
2	Trace the URBT control block.
4	Trace the URBR control block.
8	Trace the URBD control block before image.
16	Trace the URBD control block after image.
32	Trace the URBF/URBG control block before image.
64	Trace the URBF/URBG control block after image.
128	Trace the send buffer.
256	Trace the URBY control block.
512	Trace the URBO control block.
1024	Trace the output parameters.
2048	Trace the subscription table.

Settings for DCLASS=SAGTRGB

Trace Value	Description
1	Trace the output buffer.
2	Trace the URBD control block after image.
4	Trace the URBR control block.
8	Trace the URBT control block.
16	Trace the URBH control block.
32	Trace the URBF/URBG control block before image.
64	Trace the URBS control block.
128	Trace the URBY control block.
256	Trace the URBO control block.
512	Trace the URBD AI control blocks
1024	Trace the URBF/G AI control blocks
2048	Trace the subscription table.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Destination Class Parameter Data` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DCOMMITTHRESHOLD Parameter

Parameter	Specify . . .	Possible Values	Default
DCOMMITTHRESHOLD	For destinations defined with DTYPE=ETBROKER, DTYPE=MQSERIES, or DTYPE=NULL, specify a threshold for the number of messages that will be sent to the destination before a commit is performed for those messages. For destinations defined with DTYPE=File, specify the number of URBLTDOD TLOG record bytes that will be written to the CLOG before the buffers are flushed. This parameter may not be specified for a destination defined with DTYPE=ADABAS.	numeric, within the range 0 through 2,147,483,647	5 (except for File destinations)

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Threshold` or `Commit Threshold` parameters in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

EntireX Broker, WebSphere MQ, and Null Destination Considerations

For webMethods EntireX, WebSphere MQ, and null destinations, the term *commit* in this context means that the Event Replicator Server informs the messaging system that all messages sent (since the last commit) should be made permanent. In the case of WebSphere MQ, commit means that the Event Replicator Server will issue an MQCMIT call for the queue.

The actual number of messages sent to the destination prior to the commit call will vary. The number of messages sent prior to the commit may be less than the DCOMMITTHRESHOLD value when all data queued to the destination has been sent. The number of messages sent prior to the commit may be greater than the DCOMMITTHRESHOLD value when a transaction's worth of replicated data must be sent in multiple messages. The commit will only be done after an entire transaction has been sent to the destination.

The replicated transactions occupy space in the Event Replicator Server's replication pool until the commit has been performed. If the commit threshold is set to a high value, the LRPL parameter also must be set to a sufficiently high value, so that the replication pool can accommodate the replicated transactions until they are committed.

File Destination Considerations

The check on the threshold set for File destinations will only occur when a complete transaction has been written. The default value for this threshold is calculated using the following formula:

`MAX(MIN(a,b),c)`

where

- *a*= `0.9 * CLOG-block-size * CLOG-blocks-per-track * CLOG-tracks-per-cylinder`
- *b*= `0.05 * LRPL-setting`
- *c*= The maximum URBL record size that can be written to the TLOG.

DETBBROKERID Parameter

Parameter	Specify . . .	Possible Values	Default
`DETBBROKERID`	defines the webMethods EntireX Broker ID.	up to 32 characters	none

If this parameter is *not* specified, the default EntireX Broker ID (specified in parameter `ETBBROKERID`) will be used.

Broker IDs come in two formats: one for TCP/IP communications and one for Adabas SVC communications. For TCP/IP communications, the format is:

`ip-address:port-number:TCP`

In this case, the *ip-address* setting is the TCP/IP IP address and the *port-number* setting should match the webMethods EntireX `PORT` parameter.

For Adabas SVC communications in z/OS and VSE environments, the format is:

```
'broker-id:SVCnnn:NET'
```

In z/OS and VSE environments, the *broker-id* setting should match the webMethods EntireX `BROKER-ID` parameter in the Broker ETBFILE DD. The *nnn* setting should match either the EntireX Broker `ADASVC` or `ADA5SVC` parameters in the Broker PARMS DD statement.

The use of the following parameter form is *not* recommended in BS2000 environments:

```
'broker-id: :NET'
```

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `EntireX Broker ID` parameter in *Creating an EntireX Broker Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DETBSERVICE Parameter

Parameter	Specify . . .	Possible Values	Default
`DETBSERVICE`	the webMethods EntireX service.	up to 32 characters	none

This should be the same as the value specified for the SERVICE parameter in webMethods EntireX.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Broker Service` parameter in *Creating an EntireX Broker Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DETBSERVICECLASS Parameter

Parameter	Specify . . .	Possible Values	Default
`DETBSERVICECLASS`	the webMethods EntireX service class.	up to 32 characters	none

This should be the same as the value specified for the CLASS parameter in webMethods EntireX.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Service Class` parameter in *Creating an EntireX Broker Destination Definition*, in the .

DETBSCONV Parameter

Parameter	Specify . . .	Possible Values	Default
`DETBSCONV`	whether or not the destination is to run in single conversation mode for Node to Node support.	YES | NO	NO

This parameter should be used for high volume destinations.

When this optional parameter is set to "YES" for a destination, all messages are sent using the same EntireX Broker conversation ID.

In the event of a failure, or if the destination is closed, the same conversation ID will be used the next time the destination connection to the EntireX Broker is opened. If there is no active conversation, then a new conversation ID will be established.

Additional considerations when using Single Conversation mode

- The EntireX Broker will only accept the Single Conversation mode if either
  - the Single Conversation mode is already active or
  - there are no active conversations for the specified class, server or service.
- A switch to Single Conversation mode requires that there are no "old" conversations active (see section about Migration to Single Conversation mode below). A mixture of conversation types is not possible.
- A mixture of Single Conversation and non-Single Conversation mode per specified class, server or service is not possible.

Migration to Single Conversation mode

To migrate an environment with a Sending Replicator Server, EntireX Broker and Receiving Replicator Server working with multiple Conversations.

1. The EntireX Broker class, server or service must be empty or drained.
2. The EntireX Broker must be started and run with EXB970L023 / EXX970L023 or higher.
3. The Sending Event Replicator Server "Destination" must be modified with the new Single Conversation Parameters, and the Sending Event Replicator Server must be running Event Replicator for Adabas Version 3.5. SP1 or higher.
4. The Receiving Event Replicator Server "Input Queue" must be modified with the new Single Conversation Parameters, and the Receiving Event Replicator Server must be running Event Replicator for Adabas Version 3.5. SP1 or higher.
5. Open the Single Conversation Destination and input queue.

If the open is successful, ADAFS0 messages will indicate the destination successfully opened with Single Conversation Mode as shown below.

```
ADAFSO 36889 ETBBROKER destination OUTPUT1 connected
ADAFSO 36889 with single conversation mode set
ADAFSO 36889 ETBBROKER Token = REPTOR1
ADAFSO 36889 ETBBROKER Userid = OUT1
```

If the class, server or service was not empty when the Single Conversation mode opens were attempted, error messages as shown below are received.

```
ADAF8A 36889 Destination: OUTPUT1 Open failed: Unexpected reason code
ADAF87 36889 AMI R=Unexpect Q=OUTPUT1 RC=012/036/0000/0000

ADAF9C 36890 Input Queue: INPUT1 Open failed: Unexpected reason co
ADAF87 36890 AMI R=RPIDBG * Q=INPUT1 RC=012/036/0000/0000
ADAF9B 36890 Initialization error in input subtask 001
ADAF8H 36890 Open for input queue INPUT1 failed
```

DETBTOKEN Parameter

Parameter	Specify . . .	Possible Values	Default
DETBTOKEN	the webMethods EntireX token.	up to 32 characters	REPTOR-DEST- <destination name>

This optional parameter can only be specified when DETBSCONV=YES.

Refer to *TOKEN* in the *EntireX Broker ACI for Assembler* section of the *webMethods EntireX* documentation for more information on usage and syntax.

DETBUSERID Parameter

Parameter	Specify . . .	Possible Values	Default
DETBUSERID	the webMethods EntireX user ID.	up to 32 characters	REPTOR--<node>-----<jobname>

This optional parameter can only be specified when DETBSCONV=YES.

Refer to *USER-ID* in the *EntireX Broker ACI for Assembler* section of the *webMethods EntireX* documentation for more information on usage and syntax.

DETBSERVICENAME Parameter

Parameter	Specify . . .	Possible Values	Default
`DETBSERVICENAME`	the webMethods EntireX service name.	up to 32 characters	none

This should be the same as the value specified for the SERVER parameter in webMethods EntireX.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Service Name` parameter in *Creating an EntireX Broker Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DEVENTLOG Parameter

Parameter	Specify . . .	Possible Values	Default
`DEVENTLOG`	whether or not events should be logged by the Event Replicator and sent to this destination.	YES | NO	NO

Note: This parameter is not valid for Adabas destinations.

When this optional parameter is set to "YES" for a destination, Event Replicator Server events are logged to the destination. When DEVENTLOG is set to NO (the default), they are not.

Event Replicator Server events are logged in URBS elements. These URBS elements are sent to destinations related to the event itself. The URBS elements are also sent to any other destinations you have defined with DEVENTLOG=YES. If a related destination also is defined with DEVENTLOG=YES, it will only receive one instance of the URBS element.

To access this log of Event Replicator Server events in the destination queue, you must supply your own application that reads the event URBS elements in the destination queue. If such an application does not exist, the logged events simply sit in the queue.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Event Logging` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DEXIT Parameter

Parameter	Specify . . .	Possible Values	Default
DEXIT	the destination output user exit name.		none

⚠ **Caution:** This parameter is not available for general use.

DEXITWORKSIZE Parameter

Parameter	Specify . . .	Possible Values	Default
DEXITWORKSIZE	the size of the destination output user exit work area.		

⚠ **Caution:** This parameter is not available for general use.

DHBINTERVAL Parameter

Parameter	Specify . . .	Possible Values	Default
DHBINTERVAL	the interval (in seconds) at which the heartbeat should be published to a destination.	0, 60-86400	0

📄 **Note:** This parameter is to be used with *Replication Monitoring* only.

Use the DHBINTERVAL destination subparameter to set the interval at which heartbeat events are sent to a particular destination. Information for a heartbeat event is collected if at least the defined amount of seconds has passed since a heartbeat event was last sent to the destination. This heartbeat information is inserted into the replication data stream at the exit from the Event Replicator Server (destination processing).

A value of zero (0) indicates that a heartbeat event should not be published for a particular destination.

The DHBINTERVAL parameter is only allowed for destinations defined with DTYPE=ETBROKER, DTYPE=MQSERIES, or DTYPE=NULL.

The DHBINTERVAL parameter is only allowed for destinations defined with DCLASS=SAGTARG.

📄 **Note:** The interval between creating heartbeat events may be slightly longer (e.g. 10 seconds longer) than the interval specified for this parameter.

DLOG Parameter

Parameter	Specify . . .	Possible Values	Default
DLOG	whether subscription logging should be activated for the destination at Event Replicator Server startup.	"YES" or "NO"	NO

If "YES" is specified, subscription logging is activated for the destination at Event Replicator Server startup; if "NO" is specified, subscription logging is not activated. For more information about subscription logging, also known as the SLOG facility, read *Using the Subscription Logging Facility* in *Event Replicator for Adabas Administration and Operations Guide*.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Allow Logging` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DMAXOUTPUTSIZE Parameter

Parameter	Specify . . .	Possible Values	Default
DMAXOUTPUTSIZE	the maximum output size (in bytes) for the destination.	0, 4096 - 2,147,483,647.	0

> **Note:** You can specify the value for this parameter in a purely numeric form or use K at the end of the number to specify kilobytes. For example, DMAXOUTPUTSIZE=4K is the same as DMAXOUTPUTSIZE=4096.

The value for this parameter will be used if it is less than or equal to the maximum output size for the Event Replicator Server (specified using the `MAXOUTPUTSIZE` global parameter) and less than or equal to the maximum output allowed for the messaging system queue being defined. If this value is larger than the `MAXOUTPUTSIZE` specification or the maximum output size allowed by the messaging system, the smaller value will be used.

A value of 0 indicates that no specific limit is set for this destination. Instead, the smaller of the `MAXOUTPUTSIZE` specification or the maximum output size allowed by the messaging system will be used.

> **Note:** This parameter is not valid for FILE destinations (DTYPE=FILE). The size of the records written for a destination with DTYPE=FILE are affected solely by the record size supported by the CLOG data set.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Max Output Size` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DMQCCSID Parameter

Parameter	Specify . . .	Possible Values	Default
DMQCCSID	the destination-specific coded character set ID (CCSID) for the WebSphere MQ destination.	0 - 2,147,483,647	0

This optional parameter can only be specified when the DCLASS or DEXIT parameters are specified.

The Event Replicator Server does not attempt to verify the value of this parameter as the character codes may be changed or added to as time goes on. The value for this parameter is simply passed in the appropriate WebSphere MQ request as the CCSID.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the MQSeries Coded Character Set ID parameter in *Creating a WebSphere MQ Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DMQDYNQNAME Parameter

Parameter	Specify . . .	Possible Values	Default
DMQDYNQNAME	the WebSphere MQ dynamic queue name	up to 48 characters	(blanks)

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Dynamic Queue Name parameter in *Creating a WebSphere MQ Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DMQFORMAT Parameter

Parameter	Specify . . .	Possible Values	Default
DMQFORMAT	the optional MQ format name, only if a value for the DCLASS parameter has been specified.	up to eight characters	(blanks)

> **Note:** You cannot specify a value for the DMQFORMAT parameter if a value has not been specified for the DCLASS **parameter**.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the MQ Format parameter in *Creating a WebSphere MQ Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DMQQMGRNAME Parameter

Parameter	Specify . . .	Possible Values	Default
`DMQQMGRNAME`	defines the WebSphere MQ queue manager name.	up to 48 characters	none

If this parameter is *not* specified, the value specified for parameter MQQMGRNAME will be used. This parameter is required if DTYPE=MQSERIES and parameter MQQMGRNAME is *not* specified.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Queue Manager Name` parameter in *Creating a WebSphere MQ Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DMQQNAME Parameter

Parameter	Specify . . .	Possible Values	Default
`DMQQNAME`	the WebSphere MQ queue name.	up to 48 characters	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Queue Name` parameter in *Creating a WebSphere MQ Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DOPEN Parameter

Parameter	Specify . . .	Possible Values	Default
`DOPEN`	whether or not the destination should be opened at Event Replicator Server startup.	YES | NO | GLOBAL	GLOBAL

When DOPEN=YES is specified, the destination is opened at Event Replicator Server startup. When DOPEN=NO is specified, the destination is *not* opened at Event Replicator Server startup.

When DOPEN=GLOBAL is specified, the decision to open the destination at Event Replicator Server startup depends on the setting of the `GOPEN` global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Open at Startup` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DQFULLDELAY Parameter

Parameter	Specify . . .	Possible Values	Default
`DQFULLDELAY`	the number of seconds between retry attempts when resending output transactions to a specific and previously-full webMethods EntireX or MQSeries destination..	5 - 300 or "GLOBAL"	GLOBAL

This parameter specifies the number of seconds between retry attempts to resend output transactions to a specific, previously-full webMethods EntireX or MQSeries destination. If the value GLOBAL is specified, the number seconds between retry attempts is set to the value of the **GQFULLDELAY** initialization parameter.

> **Note:** This parameter is valid only for webMethods EntireX and MQSeries destinations (DTYPE=ETBROKER or DTYPE=MQSERIES).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `QUEUE FULL DELAY` parameter in *Creating an EntireX Broker Destination Definition* or *Creating a WebSphere MQ Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DREPLICATEUTI Parameter

Parameter	Specify . . .	Possible Values	Default
`DREPLICATEUTI`	whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup.	"YES" or "NO"	NO

If "YES" is specified, utility replication is activated for the destination at Event Replicator Server startup; if "NO" is specified, utility replication is not activated for the destination.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Replicate Util Chgs` or `Replicate Utility Changes` parameters in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

For more information about replicating utility functions, read *Replicating Adabas Utility Functions*, in *Event Replicator for Adabas Concepts*.

DRETRYCOUNT Parameter

Parameter	Specify . . .	Possible Values	Default
`DRETRYCOUNT`	the number of times that an attempt to open the destination will be retried at the interval specified by `DRETRYINTERVAL`.	0 - 2,147,483,647 or "GLOBAL"	`RETRYCOUNT` parameter setting

This parameter specifies the number of retry attempts that should occur to open this destination. If the value "GLOBAL" is specified for this parameter, the specification for the global `RETRYCOUNT` parameter will be used. Any retry attempts will occur at the interval specified by the `DRETRYINTERVAL` parameter. A value of zero indicates that no retry attempt to open this destination should occur.

> **Note:** This parameter is not valid for FILE or NULL destinations (DTYPE=FILE or DTYPE=NULL).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `RETRY COUNT` parameter in *Creating an Adabas Destination Definition*, *Creating an EntireX Broker Destination Definition*, or *Creating a WebSphere MQ Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DRETRYINTERVAL Parameter

Parameter	Specify . . .	Possible Values	Default
`DRETRYINTERVAL`	the default number of seconds between retry attempts to open the destination.	0, 5 - 2,147,483,647, or "GLOBAL"	`RETRYINTERVAL` parameter setting.

This parameter specifies the number of seconds between retry attempts to open the destination. If the value "GLOBAL" is specified for this parameter, the specification for the global `RETRYINTERVAL` parameter will be used. A value of zero indicates that no retry attempt to open this destination should occur. Except for a specification of zero, the minimum value that can be specified for this parameter is 5 seconds.

> **Note:** This parameter is not valid for FILE or NULL destinations (DTYPE=FILE or DTYPE=NULL).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `RETRY INTERVAL` parameter in *Creating an Adabas Destination Definition*, *Creating an EntireX Broker Destination Definition*, or *Creating a WebSphere MQ Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DSTATLOG Parameter

Parameter	Specify . . .	Possible Values	Default
DSTATLOG	whether or not the associated destination should receive interval statistics published by the Event Replicator Server.	YES | NO	NO

> **Note:** This parameter is valid for ETBROKER, MQSERIES, or NULL destinations. This parameter is to be used with *Replication Monitoring* only.

A value of YES indicates that interval statistics should be published to the destination being set up for use with *Replication Monitoring*; a value of NO indicates they should not.

The Replication Monitoring subsystem will only work if this Destination is defined and the Global Variable STATINTERVAL is greater than zero.

The DEVENTLOG parameter should be set to YES, if Replication events should be sent to the Adabas Analytics Collector MF component. If NO is specified, no event information will be sent.

A destination defined with DSTATLOG=YES cannot be specified in a subscription.

DTLADABAS Parameter

Parameter	Specify . . .	Possible Values	Default
DTLADABAS	the level of transaction logging that should occur when a transaction for an Adabas destination incurs an error.	0-3	0

Valid values range from 0 through 3, as described in the following table:

Value	Description
0	No transaction logging should occur.
1	Log event and Adabas error information.
2	Log event, Adabas error information, and file and record data.
3	Log event, Adabas error information, file and record data, and the actual payload.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Adabas Level parameter in *Creating an Adabas Destination Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

DTLASSIGN Parameter

Parameter	Specify . . .	Possible Values	Default
DTLASSIGN	the level of transaction logging that should occur when a transaction is assigned to a destination for output processing.	"0", "1", "2", or "3"	0

Valid values are "0" (no logging), "1" (log event and output transaction data), "2" (log event, output transaction, and file/record data), or "3" (log event and all available output information for the event).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Assign Level` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DTLCOMP Parameter

Parameter	Specify . . .	Possible Values	Default
DTLCOMP	the level of transaction logging that should occur when a transaction has been successfully output to the messaging system.	"0", "1", "2", or "3"	0

Valid values are "0" (no logging), "1" (log event and output transaction data), "2" (log event, output transaction, and file/record data), or "3" (log event and all available output information for the event).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Completion Level` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DTLSLOGREAD Parameter

Parameter	Specify . . .	Possible Values	Default
DTLSLOGREAD	the level of transaction logging that should occur when a transaction has been successfully read from the SLOG and is about to be queued for output to the destination.	"0", "1", "2", or "3"	0

Valid values are "0" (no logging), "1" (log event and output transaction data), "2" (log event, output transaction, and file/record data), or "3" (log event and all available output information for the event).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `SLOG Read Level` parameter in *Maintaining Destination Definitions*,

in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DTLSLOGWRITE Parameter

Parameter	Specify . . .	Possible Values	Default
DTLSLOGWRITE	the level of transaction logging that should occur when a transaction has been successfully written to the SLOG file.	"0", "1", "2", or "3"	0

Valid values are "0" (no logging), "1" (log event and output transaction data), "2" (log event, output transaction, and file/record data), or "3" (log event and all available output information for the event).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `SLOG Write Level` parameter in *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

DTYPE Parameter

Parameter	Specify . . .	Possible Values	Default
DTYPE	the type of output destination.	ADABAS | ETBROKER | MQSERIES | FILE | NULL	ETBROKER

Parameter DTYPE defines the type of output destination. Valid values are ADABAS, ETBROKER, MQSERIES, FILE, or NULL. The default value is ETBROKER.

- If DTYPE=ADABAS is specified, data is replicated to one or more Adabas files.

- If DTYPE=ETBROKER is specified, replicated data is written to the output queue via webMethods EntireX .

- If DTYPE=MQSERIES is specified, replicated data is written to the output queue via WebSphere MQ.

  > **Note:** If you are running on z/OS using IBM WebSphere MQ Series definitions for your Event Replicator DESTINATION or IQUEUE definitions, an 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.

- If DTYPE=FILE is specified, replicated data is written to the CLOG using TLOG URBLTDOD records. You can use these records in the CLOG file to create a sequential output file of the replicated data. For more information, read *Creating a Sequential Output File* in *Event Replicator for Adabas Administration and Operations Guide*.

> ⊖ **Caution:** Be sure that the CLOG is defined in the Event Replicator Server startup JCL (via one or more DDCLOGR*n* DD statements) if you will be using a File destination definition during Event Replicator for Adabas processing. If you do not, a warning message will be issued and the File destination will be set to "Unavailable". For more information about the CLOG, refer to your Adabas documentation.

■ If DTYPE=NULL is specified, replication is tested without actually sending the data to a destination.

For additional information about using DTYPE=ADABAS, see *Special Information about Destinations with DTYPE=ADABAS*.

This parameter can be selected online in Adabas Event Replicator Subsystem (SYSRPTR). SYSRPTR assigns the destination type based on the destination you select to create in SYSRPTR. For more information, read *Maintaining Destination Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*.

Example

The following is an example using the DESTINATION and **SUBSCRIPTION** parameters to specify that replicated data from DBID 6, file 2 is replicated to DBID 60, file 20:

```
ADARPD DESTINATION NAME=ADA1
ADARPD   DTYPE=ADABAS
ADARPD   DAIFILE=2,DAIDBID=6,DATDBID=60,DATFILE=20
*
ADARPD SUBSCRIPTION NAME=SUB300
ADARPD  SDESTINATION='ADA1'
ADARPD  SFILE=2
ADARPD  SFDBID=6
ADARPD  SFBAI='AA,BA,BB,BC,BD,CA,CB,CD,CE,CF.'
```

Special Information about Destinations with DTYPE=ADABAS

The destination type ADABAS may be used for replicating data from one Adabas database to another Adabas as long the following conditions are met:

■ The layout of the target file must be consistent with the format buffer used to decompress the input data from the source file

■ The ISNs used by the target file must be identical to the ISNs used by the source file (i.e. if record A is stored on ISN 100 in the source file, it must also be stored on ISN 100 in the target file). Insert commands (i.e., N1 or N2) on the source database are applied to the target database via N2 commands. Ensure that the ADALOD LOAD MAXISN setting on the target file is high enough to accommodate all modifications on the source file.

- The target Adabas database (the mainframe or open systems Adabas database) must be available to the Event Replicator Server either locally or remotely via Entire Net-Work.

Adabas destinations can be referenced by no more than one subscription.

The following will apply to replication using destination type ADABAS:

- The file/ISN will uniquely identify the replicated record.

- An insert on the source database/file will result in an insert (using an N2 command) to the same ISN on the target database/file.

- An update on the source database/file will result in an update to the same ISN on the target database/file.

- A delete on the source database/file will result in a delete of the same ISN on the target database/file.

- The format buffer used during an insert or an update will be the after image format buffer defined in the subscription for the related input database/file.

  > **Note:** The format buffer defined in the subscription must conform to the format buffer rules for both a read and an update.

- If the destination target for replicated data is an Adabas mainframe database running Adabas 8.1.3 or later, you can now specify the symbolic notation "C." alone as the global format buffer (`GFB` parameter) or the after-image format buffer (`SFBAI` parameter) or after-image global format buffer (`SGFORMATAI` parameter). This notation will cause the Event Replicator Server to replicate the entire source record to the target mainframe Adabas database.

  > **Caution:** This "C." option may be used *only* if the destination target file has been defined with the same fields in the same order as the fields in the source file; if there are differences in the definitions of the files, the replication of the data is likely to incur errors. The only exception to this rule is that the definitions of the descriptors and superdescriptors in the target and source files may be different.

- The record buffer used during an insert or an update will be the decompressed after image associated with the record.

The following will *not* be supported on the target database:

- A file defined with a cipher code
- A file protected with Adabas security
- A file defined as a multi-client file

You must set up the target databases and files correctly. Each target file must be sufficiently large (e.g. DSSIZE, UISIZE, NISIZE, MAXISN) to contain all of the data related to the source file. When the user has set up the target file(s) correctly, no errors will be expected when applying the replic-

ation data to the target file(s). If one or more errors are received while applying the replication data to the target file(s), information will be printed in the Event Replicator Server to help the user determine the cause of the problem(s).

≫ **To make use of this special destination, perform the following steps:**

1   Load identical files on the source and target.

> **Note:** The target file could have a subset of the fields in the source file. Although this is possible, it is up to the user to ensure the test is set up correctly.

2   Ensure replication is turned on in the source Adabas nucleus.

3   Turn replication on for the source file.

4   Define a subscription that includes the source file and sends output to the destination with type Adabas.

5   Define a destination of type ADABAS. Define the destination with the source database/file and the target database/file. This destination can be referenced by only one subscription.

6   Ensure the source nucleus, the target nucleus, and the Event Replicator Server are active.

If you would like to verify that replication is working correctly, you can use an Adabas destination to check it.

≫ **To do this, perform the following steps:**

1   Complete Steps 1 - 6 in the procedure above to set up the Adabas destination properly.

2   Modify the source file with one or more transactions from one or more users.

3   Wait a sufficient amount of time for replication to complete.

4   Unload the source and target files in ISN sequence.

5   Decompress the source file.

> **Note:** If the target file contains a subset of the fields in the source file, the FORMAT parameter should be used to ensure the decompressed source file contains only the fields defined in the target file.

6   Decompress the target file.

7   Compare the source and target files. Replication may be considered successful if the two unloaded/decompressed files are identical. Replication may be considered unsuccessful if differences are found *and* the user has set up the test correctly *and* no related errors were reported by the Event Replicator Server.

Note:

After establishing the correct Network links, it is important to issue a Network "D T" operator command on the Net-work node where the Replication Server resides to confirm:

1. The target database is seen as a "Translation/UES-enabled database" in the NET0124I message (T Translation / UES-enabled database)

2. The Replication Server is also seen as a "Translation/UES-enabled database" in the NET0124I message(T Translation / UES-enabled database)

ETBBROKERID (Default webMethods EntireX ID) Parameter

Parameter	Specify . . .	Possible Values	Default
`ETBBROKERID`	a default webMethods EntireX Broker ID destination.	up to 32 characters	none

This value is used when an webMethods EntireX Broker ID is not specified for a specific webMethods EntireX destination.

Broker IDs come in two formats: one for TCP/IP communications and one for Adabas SVC communications. For TCP/IP communications, the format is:

```
ip-address:port-number:TCP
```

In this case, the *ip-address* setting is the TCP/IP IP address and the *port-number* setting should match the webMethods EntireX `PORT` parameter.

For Adabas SVC communications in z/OS and VSE environments, the format is:

```
'broker-id:SVCnnn:NET'
```

In z/OS and VSE environments, the *broker-id* setting should match the webMethods EntireX `BROKER-ID` parameter in the Broker ETBFILE DD. The *nnn* setting should match either the EntireX Broker `ADASVC` or `ADA5SVC` parameters in the Broker PARMS DD statement.

For Adabas XCOM communications in BS2000 environments, the format is:

```
'broker-id: :NET'
```

In BS2000 environments, the *broker-id* setting should match the webMethods EntireX `BROKER-ID` parameter in the Broker ETBFILE DD. This file should also contain an IDTNAME parameter in the NET definition section that matches the one in the file in the DDLNKPAR link name. Note also that the Broker ETBFILE link file should contain the parameter TRANSPORT=NET or TRANSPORT=TCP-NET.

Syntax:

```
[ETBBROKERID=entire-broker-id]
```

ETBBROKERNAME (EntireX Broker Stub Program) Parameter

Parameter	Specify . . .	Possible Values	Default
ETBBROKERNAME	a default EntireX Broker stub program.	up to 8 characters	BROKER

This value specifies the name of the EntireX Broker stub program to be used by the Event Replicator Server.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `EntireX Broker Stub Name` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

> **Notes:**

1. EntireX Broker is a component of webMethods EntireX.

2. 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.

Syntax:

```
[ETBBROKERNAME=entirex-broker-stub-program]
```

FBVALIDATION Parameter

Parameter	Specify . . .	Possible Values	Default
FBVALIDATION	the level of format buffer validation that should occur for subscription definitions.	ABORT DEAC NONE WARN	blank (no validation is performed)

This value specifies the level of format buffer validation that should occur for subscription definitions.

During the subscription phase of transaction processing, all before and after images are decompressed using the format buffers defined for the subscription and its associated SFILE definitions.

If an invalid format buffer is specified, this decompression processing fails and a nonzero response code is set in the `URBRRSP` field of the URBR record. This occurs regardless of the setting of `FBVALIDATION` initialization parameter. The advantage of the `FBVALIDATION` initialization parameter is that, when set, format buffer validation occurs *before* transaction processing uses a subscription.

Format buffer validation set by the `FBVALIDATION` parameter occurs at Event Replicator initialization for active databases and after the initial handshake for databases that are activated at a later time. In addition, it is performed anytime an updated FDT is received.

If you do not specify the `FBVALIDATION` parameter (if you leave it blank) or if you specify a value of "NONE" for it, format buffer validation is *not* performed at Event Replicator Server startup, at the initial handshake for databases, or when an updated FDT is received. If you specify any of the other valid values for `FBVALIDATION`, format buffer validation is performed, but the action that occurs when the validation fails depends on which value you have selected, as described in the following table:

FBVALIDATION Value	Action Occurring When Validation Fails
ABORT	If validation errors occur during Event Replicator initialization, the Event Replicator Server is terminated. Validation errors found after successful Event Replicator initialization errors result in warning messages for each format buffer found to be in error. If validation errors occur during refresh processing (`RPLREFRESH` command), the refresh request fails.
DEAC	Warning messages are issued for each format buffer found to be in error. In addition, the subscription for which format buffer validation failed is deactivated . If validation errors occur during refresh processing (`RPLREFRESH` command), the refresh request fails.
WARN	Warning messages are displayed for each format buffer found to be in error.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Format Buffer Validation` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*. SYSRPTR allows you to specify this parameter for all destination definitions, regardless of destination type.

FILTER Parameter

Transaction filter definitions can be used to filter the records used for replication based on the values of fields in those records.

Syntax:

```
FILTER NAME=filter-name
     [FRECORDS=INCLUDE|EXCLUDE]
     FFIELD='fieldcode' [ [OR] FFIELD='fieldcode']...
```

One or more of the following group of subparameters are also specified for the FFIELD parameter:

```
          [FSBEGIN={startbyte | 1}]
          [FSLENGTH={length]
          [FSIMAGE=AI|BI]
          [FSPE=source-pe-number|0]
          [FSMU=source-mu-number|0]
          [FCOND=EQ|NE|LT|LE|GT|GE]
          FLIST=[']value-1[,value-n]...[']
          FTARGET='target-fieldcode'
          [FTBEGIN={startbyte | 1}]
          [FTLENGTH={length ]
          [FTIMAGE=AI|BI]
          [FTPE=target-pe-number|0]
          [FTMU=target-mu-number|0]
```

This section covers the following FILTER parameters and subparameters:

- NAME Parameter
- FCOND Subparameter
- FFIELD Parameter
- FLIST Subparameter
- FRECORDS Parameter
- FSBEGIN Subparameter
- FSIMAGE Subparameter
- FSLENGTH Subparameter
- FSMU Subparameter
- FSPE Subparameter
- FTARGET Subparameter
- FTBEGIN Subparameter
- FTIMAGE Subparameter
- FTLENGTH Subparameter
- FTMU Subparameter
- FTPE Subparameter
- OR Keyword

- FILTER Examples

NAME Parameter

Parameter	Specify . . .	Possible Values	Default
NAME	the name of the transaction filter definition.	alphanumeric; 1-8 characters	none

Parameter NAME is required and is used to uniquely identify a transaction filter definition. This value is only referenced within the Event Replicator Server. The value may be used when the Event Replicator Server writes a related message.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Transaction Filter Name` parameter in *Step 2. Specify a Transaction Filter Definition Name and Type*, in the *Adabas Event Replicator Subsystem User's Guide*.

FCOND Subparameter

Parameter	Specify . . .	Possible Values	Default
FCOND	a condition operator code for the filter.	EQ (equal to) NE (not equal to) LT (less than) LE (less than or equal to GT (greater than) GE (greater than or equal to)	EQ

When EQ or NE are specified, multiple target values and target values using wildcards can be tested. For all other condition codes, only single target values without wildcards can be tested (an FLIST subparameter is not valid).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Condition` parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

If NE is used to test multiple target values, the record is selected if the field is not equal to any of the listed target values. If EQ is used to test multiple target values, the record is selected if the field is equal to any of the listed target values.

For more information about rules of filter conditions, read *Rules for Writing Filter Conditions* in the *Adabas Event Replicator Subsystem User's Guide*.

FFIELD Parameter

Parameter	Specify . . .	Possible Values	Default
FFIELD	a specific field in the record. Use single quotation marks around the field. This field must be in the format buffer specified for the file in the SFILE definition of the subscription or the subscription will never process the filter.	A valid field code in single quotes.	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Source Field` parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

Details of the condition for the named field are specified in the `FSBEGIN`, `FSLENGTH`, `FSIMAGE`, `FSPE`, `FSMU`, `FCOND`, `FLIST`, `FTARGET`, `FTIMAGE`, `FTPE`, and `FTMU` subparameters.

If the specified condition is true, the record is selected to be included or excluded from replication, depending on the setting of the `FRECORDS` parameter.

Multiple FFIELD Parameters

More than one `FFIELD` parameter can be specified in a transaction filter definition. The maximum number of `FFIELD` parameters that can be specified in the DDKARTE statements of the Event Replicator Server startup job is controlled by the `MAXFFIELDS` parameter setting. If you use the Adabas Event Replicator Subsystem to define transaction filter definitions and their corresponding `FFIELD` parameters in the Replicator system file, the `MAXFFIELDS` parameter is not needed as the maximum number of `FFIELD`s that can be specified is 2500.

When more than one `FFIELD` parameter is specified in a transaction filter definition, the conditions of all of the `FFIELD`s must be true for the record to be selected. In other words, the conditions are logically ANDed. However, you can use the **OR keyword** to group the `FFIELD` specifications and enforce logical OR processing between the groups. In the following example, condition1, condition2, condition3, and condition4 must all be true for the record to be selected as they are logically ANDed:

```
FILTER NAME=MYINCLF
FRECORDS=INCLUDE
FFIELD='field',FCOND=cond1,FTARGET or FLIST values1
FFIELD='field',FCOND=cond2,FTARGET or FLIST values2
FFIELD='field',FCOND=cond3,FTARGET or FLIST values3
FFIELD='field',FCOND=cond4,FTARGET or FLIST values4
```

But in the following example, either both conditions 1 and 2 OR both conditions 3 and 4 must be true for the record to be selected:

```
FILTER NAME=MYINCLF
FRECORDS=INCLUDE
FFIELD='field',FCOND=cond1,FTARGET or FLIST values1
FFIELD='field',FCOND=cond2,FTARGET or FLIST values2
OR
FFIELD='field',FCOND=cond3,FTARGET or FLIST values3
FFIELD='field',FCOND=cond4,FTARGET or FLIST values4
```

For more information about rules of filter conditions, read *Rules for Writing Filter Conditions* in the *Adabas Event Replicator Subsystem User's Guide*.

Partial FFIELD Parameters

If you want to specify only part of a source field for the comparison, specify:

- a starting byte number in the `FSBEGIN` **parameter**.

- optionally, the length of the part to be compared in the `FSLENGTH` **parameter**.

> **Note:** The format of the complete field is used for partial field comparisons. If the formats of a source field or partial source field and a target field or partial target field do not match, the comparisons may always result in an unequal condition. For example, comparing an alphanumeric field to a packed field will always result in an unequal condition.

FLIST Subparameter

Parameter	Specify . . .	Possible Values	Default
FLIST	a list of values against which the source field will be compared.	Varies, based on the field identified in the `FFIELD` parameter	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Target Value n` parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

Separate values in the list with commas. Strings that include blanks should be enclosed in single quotes. Apostrophes in strings must be doubled (for example: 'six o''clock'). Each value may consist of either free-format characters or a mix of elements specified using the `A()` or `X()` notation.

- If free-format data consists entirely of numeric data (including an optional leading "+" or "-" character) it is treated as a numeric value.

- If a value (or part of a value) is specified using `A()` notation, it will be treated as alphabetic data.

- Hexadecimal values may be specified using `X()` notation.

A value must be specified entirely as free-format data, or composed of one or more `A()` or `X()` subelements. If a value begins with an `A()` or `X()` subelement all remaining subelements of the value must be so specified. Please refer to *Rules for Writing Filter Conditions* in *Adabas Event Replicator Subsystem User's Guide* for a full description of the permissible use of FLIST.

FLIST may specify its list of values spread over many lines, as depicted in the following example:

```
ADARPD FLIST='value1,value 2, value  3,value    4,value      5,val'
ADARPD FLIST='ue      6,value         7,value          8,value      '
ADARPD FLIST='   9'
```

This example defines the following list of nine string values:

```
'value1'
'value 2'
'value  3'
'value   4'
'value    5'
'value     6'
'value      7'
'value       8'
'value        9'
```

Consider the following example, defining eight values:

```
ADARPD FLIST='J         K'
ADARPD FLIST='L,M,N,OPQRS'
ADARPD FLIST='T,U     V,W'
ADARPD FLIST='X,Y      ,Z'
ADARPD FLIST='six o''clock'
```

The values defined by this example are:

```
'J         KL'
'M'
'N'
'OPQRST'
'U     V'
'WX'
'Y       '
'Zsix o'clock'
```

This parameter is mutually exclusive with the `FTARGET` subparameter and its related `FTIMAGE`, `FTPE`, and `FTMU` subparameters. You cannot specify values for these subparameters if an `FLIST` subparameter is specified. Only a value list or a specific field can be used in transaction filter conditions -- not both.

The `FLIST` parameter may only specify more than one value if the `FCOND` parameter is set to "EQ" or "NE". In addition, wildcards can only be used in `FLIST` values if the `FCOND` parameter is "EQ" or "NE".

Only one value can be specified in the `FLIST` parameter if the `FCOND` subparameter is set to "LT", "LE", "GT", or "GE". In addition, wildcard values cannot be specified for these conditions.

For more information about rules of filter conditions, read *Rules for Writing Filter Conditions* in the *Adabas Event Replicator Subsystem User's Guide*.

The size of the `FLIST` value list that can be specified in the DDKARTE statements of the Event Replicator Server startup job is controlled by the `MAXFLISTSIZE` parameter setting. If you use the Adabas Event Replicator Subsystem to define transaction filter definitions and their corresponding `FFIELD` and `FLIST` parameters in the Replicator system file, the `MAXFLISTSIZE` parameter is not needed as the maximum size of the `FLIST` value list is restricted to the number of bytes on one Replicator system file record (250*8=2000).

FRECORDS Parameter

Parameter	Specify . . .	Possible Values	Default
FRECORDS	"INCLUDE" or "EXCLUDE" to indicate whether the records selected by the transaction filter should be included or excluded from replication.	"INCLUDE" or "EXCLUDE"	INCLUDE

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Exclude or Include Records` parameter in *Step 2. Specify a Transaction Filter Definition Name and Type*, in the *Adabas Event Replicator Subsystem User's Guide*.

For more information about rules of filter conditions, read *Rules for Writing Filter Conditions* in the *Adabas Event Replicator Subsystem User's Guide*.

FSBEGIN Subparameter

> **Note:** This parameter can only be specified for fields with alphanumeric or binary formats.

Parameter	Specify . . .	Possible Values	Default
FSBEGIN	The starting byte number of the partial Adabas source field to be compared. This field is only used if you want to specify a partial field for comparison.	1 - 32,767	1

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Source Begin` parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

For fixed length fields, valid values range from "1" (the start of the field) through the maximum length of the field (the last byte of the field). For variable length fields, valid values range from "1" (the start of the field) to the maximum length allowed for that field type. Counting occurs from left to right beginning with 1 for fields defined with alphanumeric format, and from right to left beginning with 1 for fields defined with binary format.

> **Note:** The format of the complete field is used for partial field comparisons. Valid comparisons of different field types are listed in *Field Type Considerations*, in the *Adabas Event Replicator Subsystem User's Guide*.

Examples:

In the following example, four bytes of alphanumeric field AA, starting at byte 2, are compared against the values "A", "BB", "CCC" and "DDDD".

```
ADARPD   GFORMAT NAME=GF1
ADARPD   GFB='AA,8,A.'
*
ADARPD    FILTER NAME=MYFILT
ADARPD      FRECORDS=INCLUDE
ADARPD        FFIELD='AA',FSIMAGE=AI,FSBEGIN=2,FSLENGTH=4
ADARPD          FCOND=EQ
ADARPD           FLIST='A,BB,CCC,DDDD'         ↵
```

In the following example, four bytes of binary field BB, starting at byte 2, are compared against the values X'C1', X'C2C2', X'C3C3C3', and X'C4C4C4C4'.

```
ADARPD   GFORMAT NAME=GF2
ADARPD   GFB='BB,8,B.'
*
ADARPD    FILTER NAME=MYFILT
ADARPD      FRECORDS=INCLUDE
ADARPD        FFIELD='BB',FSIMAGE=AI,FSBEGIN=2,FSLENGTH=4
ADARPD          FCOND=EQ
ADARPD           FLIST='X(C1),X(C2C2),X(C3C3C3),X(C4C4C4C4)'          ↵
```

In the following example, four bytes of binary field BB, starting at byte 5, are compared against the after-image of packed field PP using a numeric comparison. If the numbers are equal, the record will be included.

> **Note:** If field BB had been an alphanumeric field, the comparison would always result in an unequal condition and the record would not be included because you cannot compare alphanumeric fields and packed fields.

```
ADARPD   GFORMAT NAME=GF3
ADARPD   GFB='BB,8,B,PP,4,P.'
*
ADARPD    FILTER NAME=MYFILT
ADARPD      FRECORDS=INCLUDE
ADARPD        FFIELD='BB',FSIMAGE=AI,FSBEGIN=5,FSLENGTH=4
ADARPD          FCOND=EQ
ADARPD          FTARGET='PP',FTIMAGE=AI       ↵
```

FSIMAGE Subparameter

Parameter	Specify . . .	Possible Values	Default
FSIMAGE	Whether the source field (FFIELD parameter) is in the after image (AI), before image (BI), or the default image of the record.	"AI" "BI"	AI for adds and updates; BI for deletes

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Source Image parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

FSLENGTH Subparameter

Note: This parameter can only be specified for fields with alphanumeric or binary formats.

Parameter	Specify . . .	Possible Values	Default
FSLENGTH	The numeric length of the partial Adabas source field that should be compared. This field is only used if you want to specify a partial field for comparison.	1 - 32,767	see below

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Source Length parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

For fixed length fields, errors will occur if the sum of the values of the FSBEGIN and FSLENGTH parameters exceeds the fixed length of the field. For variable length fields, the sum of the values of the FSBEGIN and FSLENGTH parameters must not exceed the maximum length of the field plus 1. For example, if a variable length field has format "A" with a maximum length of 253 bytes, settings of FSBEGIN=1 and FSLENGTH=253 are valid, but settings of FSBEGIN=2 and FSLENGTH=254 are not.

If FSBEGIN is not specified, the default value is the entire field. If FSBEGIN is specified, the default value is the maximum length of the field minus the value of the FSBEGIN parameter plus 1.

Note: The format of the complete field is used for partial field comparisons. Valid comparisons of different field types are listed in *Field Type Considerations*, in the *Adabas Event Replicator Subsystem User's Guide*.

Examples:

Good examples of the FSLENGTH parameter are provided in the description of the FSBEGIN **parameter**.

FSMU Subparameter

Parameter	Specify . . .	Possible Values	Default
FSMU	the index number of the multiple-value field (MU) to which the condition relates if the source field (FFIELD parameter) in this field filter is an MU field.	0-191	0

A value of "0" indicates that the source field is not an MU field.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Source MU` parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

FSPE Subparameter

Parameter	Specify . . .	Possible Values	Default
FSPE	the index number (occurrence) of the periodic group (PE) to which the condition relates if the source field (FFIELD parameter) in this field filter is a PE field.	0-191	0

A value of "0" indicates that the source field is not a PE field.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Source PE` parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

FTARGET Subparameter

> **Note:** This parameter can only be specified for fields with alphanumeric or binary formats.

Parameter	Specify . . .	Possible Values	Default
FTARGET	the two-byte Adabas field code for the target field with which the source field will be compared. Use single quotation marks around the field code. This field must be in the same record as the source field (FFIELD parameter).	A valid field code in single quotes	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Target Field` parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

This parameter is mutually exclusive with the FLIST subparameter. You cannot specify values for the FLIST subparameter if this subparameter is specified. Only a value list or a specific field can be used in transaction filter conditions -- not both.

For more information about rules of filter conditions, read *Rules for Writing Filter Conditions* in the *Adabas Event Replicator Subsystem User's Guide*.

Details regarding the target field are specified in the `FTBEGIN`, `FTLENGTH`, `FTIMAGE`, `FTPE`, and `FTMU` subparameters.

Partial FFIELD Parameters

If you want to specify only part of a target field for the comparison, specify:

- a starting byte number in the `FTBEGIN` **parameter**.
- optionally, the length of the part to be compared in the `FTLENGTH` **parameter**.

> **Note:** The format of the complete field is used for partial field comparisons. Valid comparisons of different field types are listed in *Field Type Considerations*, in the *Adabas Event Replicator Subsystem User's Guide*.

FTBEGIN Subparameter

> **Note:** This parameter can only be specified for fields with alphanumeric or binary formats and only if the `FTARGET` parameter is also specified.

Parameter	Specify . . .	Possible Values	Default
`FTBEGIN`	The starting byte number of the partial Adabas target field at which the comparison should begin. This field is only used if you want to specify a partial field for the comparison.	1 - 32,767	1

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Target Begin` parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

For fixed length fields, valid values range from "1" (the start of the field) through the maximum length of the field (the last byte of the field). For variable length fields, valid values range from "1" (the start of the field) to the maximum length allowed for the field type. Counting occurs from left to right beginning with 1 for fields defined with alphanumeric format, and from right to left beginning with 1 for fields defined with binary format.

> **Note:** The format of the complete field is used for partial field comparisons. Valid comparisons of different field types are listed in *Field Type Considerations*, in the *Adabas Event Replicator Subsystem User's Guide*.

FTIMAGE Subparameter

Parameter	Specify . . .	Possible Values	Default
FTIMAGE	Whether the target field (FTARGET subparameter) is in the after image (AI), before image (BI), or the default image of the record.	"AI" "BI"	AI for adds and updates; BI for deletes

This subparameter can only be specified if the FTARGET subparameter is also specified.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Target Image parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

FTLENGTH Subparameter

> **Note:** This parameter can only be specified for fields with alphanumeric or binary formats and only if the FTARGET parameter is also specified.

Parameter	Specify . . .	Possible Values	Default
FTLENGTH	The numeric length of the partial Adabas target field that should be used for the comparison. This field is only used if you want to specify a partial field for the comparison.	1 - 32,767	see below

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Target Length parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

For fixed length fields, errors will occur if the sum of the values of the FTBEGIN and FTLENGTH parameters exceeds the fixed length of the field. For variable length fields, the sum of the values of the FTBEGIN and FTLENGTH parameters must not exceed the maximum length of the field plus 1. For example, if a variable length field has format "A" with a maximum length of 253 bytes, settings of FTBEGIN=1 and FTLENGTH=253 are valid, but settings of FTBEGIN=2 and FTLENGTH=254 are not.

If FTBEGIN is not specified, the default value is the entire field. If FTBEGIN is specified, the default value is the maximum length of the field minus the value of the FTBEGIN parameter plus 1.

> **Note:** The format of the complete field is used for partial field comparisons. Valid comparisons of different field types are listed in *Field Type Considerations*, in the *Adabas Event Replicator Subsystem User's Guide*.

FTMU Subparameter

Parameter	Specify . . .	Possible Values	Default
FTMU	the index number of the multiple-value field (MU) to which the condition relates if the target field (FTARGET subparameter) in this field filter is an MU field.	0-191	0

This subparameter can only be specified if the FTARGET subparameter is also specified. A value of "0" indicates that the target field is not an MU field.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Target MU parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

FTPE Subparameter

Parameter	Specify . . .	Possible Values	Default
FTPE	the index number (occurrence) of the periodic group (PE) to which the condition relates if the target field (FTARGET subparameter) in this field filter is a PE field.	0-191	0

This subparameter can only be specified if the FTARGET subparameter is also specified. A value of "0" indicates that the target field is not a PE field.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Target PE parameter in *Step 3. Add Filter Conditions to the Transaction Filter Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

OR Keyword

When more than one FFIELD parameter is specified in a transaction filter definition, the conditions of all of the FFIELDs must be true for the record to be selected. In other words, the conditions are logically ANDed. However, you can use the **OR keyword** to group the FFIELD specifications and enforce logical OR processing between the groups.

In the following example, condition1, condition2, condition3, and condition4 must all be true for the record to be selected (they are logically ANDed):

```
FILTER NAME=MYINCLF
FRECORDS=INCLUDE
FFIELD condition1
FFIELD condition2
FFIELD condition3
FFIELD condition4
```

But in the following example, either both conditions 1 and 2 OR both conditions 3 and 4 must be true for the record to be selected:

```
FILTER NAME=MYINCLF
FRECORDS=INCLUDE
FFIELD condition1
FFIELD condition2
OR
FFIELD condition3
FFIELD condition4
```

For more information about rules of filter conditions, read *Rules for Writing Filter Conditions* in the *Adabas Event Replicator Subsystem User's Guide*.

FILTER Examples

The following filter not replicate any updates in which fields CA, CB, and CC do not change. In other words, the update will only be replicated when fields CA, CB or CC are changed.

```
FILTER NAME=SAG1
  FRECORDS=EXCLUDE
    FFIELD='CA',FSIMAGE=BI
      FCOND=EQ
      FTARGET='CA',FTIMAGE=AI
    FFIELD='CB',FSIMAGE=BI
      FCOND=EQ
      FTARGET='CB',FTIMAGE=AI
    FFIELD='CC',FSIMAGE=BI
      FCOND=EQ
      FTARGET='CC',FTIMAGE=AI
```

The following filter will replicate all updates in which the values of fields CA, CB, or CC are increased. If any of these field values are increased, the update will be replicated.

```
FILTER NAME=SAG2
  FRECORDS=INCLUDE
    FFIELD='CA',FSIMAGE=BI
      FCOND=LT
      FTARGET='CA',FTIMAGE=AI
    OR
    FFIELD='CB',FSIMAGE=BI
      FCOND=LT
      FTARGET='CB',FTIMAGE=AI
```

```
      OR
   FFIELD='CC',FSIMAGE=BI
      FCOND=LT
      FTARGET='CC',FTIMAGE=AI  ↵
```

The following filter will replicate records in which field BA contains the characters "J", "K", "L", "M" or "NO", *or* in which field BB is set to a value ending in 30 consecutive "P" characters *and* field BC is set to contain the strings "ALPHABETIC", "STRINGS" or any string containing 15 consecutive "Q" characters.

```
FILTER NAME=SAG3
  FRECORDS=INCLUDE
    FFIELD='BA',FSIMAGE=AI
      FCOND=EQ
      FLIST='J,K,L,M,NO'
    OR
    FFIELD='BB',FSIMAGE=AI
      FCOND=EQ
      FLIST='*PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP'
    FFIELD='BC',FSIMAGE=AI
      FCOND=EQ
      FLIST='ALPHABETIC,*QQQQQQQQQQQQQQQ*,STRINGS'
```

The following example replicates records for which the second MU (FSMU=2) within PEs "1", "2", or "3" of field BA are set equal to the contents of field AA.

```
FILTER NAME=SAG4
  FRECORDS=INCLUDE
    FFIELD='BA',FSMU=2,FSPE=1
      FCOND=EQ
      FTARGET='AA'
    FFIELD='BA',FSMU=2,FSPE=2
      FCOND=EQ
      FTARGET='AA'
    FFIELD='BA',FSMU=2,FSPE=3
      FCOND=EQ
      FTARGET='AA'                    ↵
```

GFORMAT (Global Format) Parameter

A global format buffer definition may be used for decompression of replication data from a specific DBID/file. Once a global format buffer definition is defined, it can be used by one or more subscriptions.

```
GFORMAT NAME=format-name
     GFB='format-buffer'  ↵
```

- NAME Subparameter

- GFB Subparameter
- GFORMAT Example

NAME Subparameter

The NAME subparameter defines the name of the global format buffer.

Parameter	Specify . . .	Possible Values	Default
NAME	the global format buffer name.	alphanumeric; 1 - 7 characters	none

The format name must be unique across all defined global formats. Subparameter GFB is also required when GFORMAT is specified.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the GFB Name parameter in *Adding GFB Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*.

GFB Subparameter

A global format may be used for decompression of replication data from a specific DBID/file for one or more subscriptions. The following may be defined in the definition of a global format: 1) Global format name 2) Format buffer.

Parameter	Specify . . .	Possible Values	Default
GFB	the global format buffer	varies	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the blank lines below the GFB Name parameter in *Adding GFB Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*.

The format buffer specified in the GFB parameter must conform to the format buffer requirements (for read commands) documented in the Adabas command reference documentation (*Adabas Command Reference*). Subparameter NAME is also required when GFORMAT is specified.

If the destination target for replicated data is an Adabas mainframe database running Adabas 8.1.3 or later, you can specify the symbolic notation "C." alone as the GFB parameter value (for example, GFB='C.'). This notation will cause the Event Replicator Server to replicate the entire source record to the target mainframe Adabas database.

PE and MU fields cannot use the range notation 1-N in format buffers for a subscription that is sent to a destination that has specified the replication initialization parameter DCLASS=SAGTARG. The SAGTARG application invoked requires that the range of occurrences specified are contained in the record buffer even if they are empty occurrences. 1-N results in a range of 1-191, but unless there are 191 occurrences containing data, space in the record buffer is not allocated for any empty occurrences, resulting in incorrect field positioning when processing the record.

⚠ **Caution:** This "C." option may be used *only* if the destination target file has been defined with the same fields in the same order as the fields in the source file; if there are differences in the definitions of the files, the replication of the data is likely to incur errors. The only exception to this rule is that the definitions of the descriptors and superdescriptors in the target and source files may be different.

GFORMAT Example

In the following example, the first and second `GFB` subparameters define the global format buffer of the global format definition named "G1". The third through fifth `GFB` subparameters define the global format buffer of the global format definition named "G2".

```
ADARPD GFORMAT NAME=G1
ADARPD  GFB='AA,8,A,'
ADARPD  GFB='AB,8,A.'
*
ADARPD GFORMAT NAME=G2
ADARPD  GFB='CC,8,A,'
ADARPD  GFB='CD,8,A,'
ADARPD  GFB='CE,8,A.'
*                                              ↵
```

GLOBALS Parameter

The GLOBALS parameter allows you to specify global settings for the Event Replicator Server. All of these global settings have defaults, so none of them are required.

```
GLOBALS
   [NUMSFILE=number-of-SFILE-groups]
   [MAXFFIELDS=n]
   [MAXFLISTSIZE=n]
```

This section covers the following GLOBALS parameters and subparameters:

- NUMSFILE (Maximum Number of SFILEs) Subparameter
- MAXFFIELDS (Maximum Number of FFIELDs) Setting

- MAXFLISTSIZE (Maximum FLIST Size) Setting

NUMSFILE (Maximum Number of SFILEs) Subparameter

The `NUMSFILE` subparameter specifies the number of SFILE groups to be allowed for a SUBSCRIPTION. Once set, all subsequent subscriptions will allow up to `NUMSFILE` number of SFILE groups.

Parameter	Specify . . .	Possible Values	Default
`GLOBALS NUMSFILE`	the maximum number of SFILE groups allowed.	Minimum value of 1 Maximum value is dependent on available memory	10

Example:

In the following example, subsequent subscriptions will allow up to 30 SFILE parameter groups.

```
GLOBALS NUMSFILE=30
```

MAXFFIELDS (Maximum Number of FFIELDs) Setting

Parameter	Specify . . .	Possible Values	Default
`MAXFFIELDS`	the number of `FFIELD` definitions that can be specified in a single `FILTER` definition.	Minimum value of 1 Maximum value is dependent on available memory	32

The Event Replicator Server will getmain storage to accommodate the number of `FFIELD` parameters allowed by this parameter and based on the maximum size of the FLIST specifications allowed by the `MAXFLISTSIZE` parameter.

MAXFLISTSIZE (Maximum FLIST Size) Setting

Parameter	Specify . . .	Possible Values	Default
`MAXFLISTSIZE`	the maximum size allowed for the `FLIST` subparameter. Values are expressed in bytes or kilobytes (if you add a K after the value).	1-32768	128 bytes

The Event Replicator Server will getmain storage to accommodate the size specified by this parameter and based on the number of `FFIELD` specifications allowed by the `MAXXFIELDS` parameter.

GOPEN Parameter

Parameter	Specify . . .	Possible Values	Default
GOPEN	whether or not destinations with DOPEN=GLOBAL and input queues with IQOPEN=GLOBAL specified in their definitions should be opened at Event Replicator Server startup. This parameter specifies the global policy determining whether destinations with DOPEN=GLOBAL settings are opened at Event Replicator Server startup.	YES | NO	YES

When GOPEN=YES is specified, any destinations with DOPEN=GLOBAL or input queues with IQOPEN=GLOBAL specified in their definitions are opened at Event Replicator Server startup. When GOPEN=NO is specified, any destinations with DOPEN=GLOBAL or input queues with IQOPEN=GLOBAL specified in their definitions are *not* opened at Event Replicator Server startup.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Open at Start` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax

```
GOPEN={ YES | NO }
```

GQFULLDELAY Parameter

Parameter	Specify . . .	Possible Values	Default
GQFULLDELAY	the number of seconds between retry attempts when resending output transactions to a previously-full webMethods EntireX or MQSeries destination.	5-300	60

This is a global parameter.

> **Note:** This parameter only affects webMethods EntireX and MQSeries destinations (DTYPE=ETBROKER or DTYPE=MQSERIES).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR) in both the webMethods EntireX and MQSeries destinations as well as as a global parameter. For more information, read about the `Queue Full Delay` parameter in *Step 2. Specify General and TLOG EntireX Broker Destination Parameters*, *Step 2. Specify General and TLOG WebSphere MQ Destination Parameters*, and *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax

```
GQFULLDELAY=n
```

INITIALSTATE Parameter

The INITIALSTATE parameter and its subparameters define an initial-state process that may be requested from the Event Replicator Server. Each initial-state process is identified by its name. Different initial-state processes must have different names.

> **Note:** At least one subscription (ISUBSCRIPTION), destination (IDESTINATION), or file (IFILE) must be specified in an initial-state definition.

Syntax:

```
INITIALSTATE NAME=init-state-name
     [ISUBSCRIPTION='subscription-name-list']
     [IDESTINATION='destination-name-list']
     [IMAXREQ=n]
```

Zero, one or more of the following group of sub-parameters will be specified:

```
[IFILE=file-number]
     [IDBID=database-id]
     [ISNLIST | SELCRIT='search-buffer'] ↵
```

This section covers the following INITIALSTATE parameters and sub-parameters:

- NAME
- ISUBSCRIPTION
- IDESTINATION
- IMAXREQ
- IFILE
- IDBID
- ISNLIST or SELCRIT

NAME

Parameter	Specify . . .	Possible Values	Default
NAME	an initial-state process that may be requested from the Event Replicator Server.	alphanumeric; 1-8 characters	none

Each initial-state process is identified by its name. Different initial-state processes must have different names.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Initial-State Name` parameter in *Step 2. Supply a Name for the Initial-State Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

ISUBSCRIPTION

Parameter	Specify . . .	Possible Values	Default
ISUBSCRIPTION	the subscriptions for which initial-state data is to be delivered.	alphanumeric list of up to 32 subscription names	none

Zero, one, or more subscription names (up to 32 names) can be specified. If one or more IFILE parameters are specified, the ISUBSCRIPTION parameter restricts the delivery of initial-state data to the specified subscriptions. If no IFILE parameter is specified, ISUBSCRIPTION also defines the set of files for which initial-state data is to be delivered (see below).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about selecting subscriptions in *Select Subscriptions for the Initial-State Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IDESTINATION

Parameter	Specify . . .	Possible Values	Default
IDESTINATION	the destinations to which initial-state data is to be delivered.	alphanumeric list of destination names	none

Zero, one, or more destination names can be specified. If one or more IFILE parameters are specified, the IDESTINATION parameter restricts the delivery of initial-state data to the specified destinations. If neither IFILE nor ISUBSCRIPTION is specified, IDESTINATION also defines the set of files for which initial-state data is to be delivered (see below).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about selecting destinations in *Select Destinations for the Initial-State Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IMAXREQ

Parameter	Specify . . .	Possible Values	Default
IMAXREQ	number of concurrent initial-state requests that can occur in a given instance of this initial-state definition.	numeric; 1-2,147,483,647	1

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Num of Concurrent Initial State Requests` field in *Step 3. Specify the Number of Concurrent Initial-State Requests Possible for the Initial-State Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IFILE

Parameter	Specify . . .	Possible Values	Default
IFILE	the files from which initial-state data is to be delivered.	numeric file number	none

Zero, one or more file numbers can be specified.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about selecting files in *Select Files for the Initial-State Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IDBID

Parameter	Specify . . .	Possible Values	Default
IDBID	the database containing the specified file (for each IFILE parameter).	numeric; 1-65,535	none

If IDBID is not specified, the default DBID is assumed. This parameter may only be specified when the IFILE parameter is specified.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about selecting files in *Select Files for the Initial-State Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

ISNLIST or SELCRIT

Parameter	Specify . . .	Possible Values	Default
ISNLIST or SELCRIT	data selection (filtering) method.	MAX 60 byte search buffer for SELCRIT	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the selection criteria in *Select Files for the Initial-State Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

The optional keyword parameters ISNLIST and SELCRIT are mutually exclusive and may only be specified when the IFILE parameter is specified.

The ISNLIST parameter specifies that only the records with the ISNs provided in the actual initial-state request will be delivered. The alternative optional SELCRIT parameter, which takes an Adabas Search Buffer as its argument, specifies that only those records that satisfy a search criterion will be delivered. The search criterion consists of the specified Search Buffer and the Value Buffer provided in the actual initial-state request.

If you elect to use an ISN list, be sure to supply an ISN list when you submit the initial-state request. The ISN list can be specified using the Adabas Event Replicator Subsystem (read *Populating a Database With Initial-State Data*, in the *Adabas Event Replicator Subsystem User's Guide*) or in the client

request (read *ISN List Format* in the section entitled *Initial-State Requests* in *Event Replicator for Adabas Application Programmer's Reference Guide*).

If one or more file numbers are specified, initial-state data will be delivered for the specified files, subject to selection (filtering) via the ISNLIST or SELCRIT parameters. The data will be delivered only for the specified subscriptions and destinations. If no subscriptions or destinations are specified, the data will be delivered for each subscription that includes any of the specified files. It will be delivered to all destinations defined in these subscriptions.

If no file numbers are specified and one or more subscriptions names are specified, initial-state data will be delivered for all files defined in any of the specified subscriptions. The data will be delivered to the specified destinations. If no destination is specified, the data will be delivered to all destinations defined in any of the specified subscriptions.

If no file numbers and no subscription names are specified, initial-state data will be delivered for all files defined in any subscription that is delivered to any of the specified destinations.

An initial-state definition with the SELCRIT option may not be used for a file with spanned Data Storage record support.

> **Note:** Initial-state processes that replicate all records may be used for a file with spanned Data Storage record support. In addition, initial-state processes that replicate records matching an ISN list may also be used for a file with spanned Data Storage record support.

IQUEUE Parameter

One or more input queues may be defined. The Event Replicator Server will listen on each of its defined queues. Defined input queues should match queues used by target applications when sending requests to an Event Replicator Server.

Syntax:

An input queue is defined with the following parameters:

```
IQUEUE NAME=input-queue-name
[IQBUFLEN={nnnn|2048}]
[IQOPEN={GLOBAL|NO|YES}
[IQRETRYCOUNT={n|GLOBAL}
[IQRETRYINTERVAL={n|GLOBAL}
[IQTYPE={ETBROKER|MQSERIES}]
```

The following parameters are used when IQTYPE=MQSERIES is used:

```
[IQMQQMGRNAME=mqseries-queue-manager-name]
[IQMQQNAME=mqseries-queue-name]
```

> **Note:** If you are running on z/OS 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.

The following parameters are used when IQTYPE=ETBROKER is used:

```
[IQETBBROKERID=entire-broker-id]
[IQETBSERVICE=entire-broker-service]
[IQETBSERVICENAME=entire-broker-service-name]
[IQETBSERVICECLASS=entire-broker-service-class]
[IQETBSCONV={YES|NO}]
[IQETBTOKEN=entire-broker-token]
[IQETBUSERID=entire-broker-userid]
```

This section covers the following IQUEUE parameters and subparameters:

- NAME Parameter
- IQBUFLEN Parameter
- IQOPEN Parameter
- IQRETRYCOUNT Parameter
- IQRETRYINTERVAL Parameter
- IQTYPE Parameter
- IQMQQMGRNAME Parameter
- IQMQQNAME Parameter
- IQETBBROKERID Parameter
- IQETBSERVICE Parameter
- IQETBSERVICENAME Parameter
- IQETBSERVICECLASS Parameter
- IQETBSCONV Parameter
- IQETBTOKEN Parameter
- IQETBUSERID Parameter

NAME Parameter

The IQUEUE NAME defines an input queue name. It is used to uniquely identify an input definition. This value is only referenced within the Event Replicator Server. The value may be used when the Event Replicator Server writes a message related to an input queue.

Parameter	Specify . . .	Possible Values	Default
NAME	the input queue to which messages are to be written.	alphanumeric, 1-8 characters	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Input Queue Name` parameter in *Creating an EntireX Broker IQUEUE Definition* and *Creating a WebSphere MQ IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQBUFLEN Parameter

Parameter	Specify . . .	Possible Values	Default
`IQBUFLEN`	the length, in bytes, of the input buffer associated with this input queue.	2048 - 2,147,483,647	2048

The maximum value of the input queue buffer length is restricted by the amount of virtual storage available in the Event Replicator address space. In addition, if you enter a value less than "2048", "2048" will be used.

The IQBUFLEN value should be set to a value greater than or equal to the largest message that will be received by the input queue. When the input queue will receive data as a part of node-to-node replication, the largest message will be limited by the minimum of the following Event Replicator Server settings: `MAXOUTPUTSIZE` **parameter**, `DMAXOUTPUTSIZE` **parameter** (if specified for the destination), or the message limit imposed by the messaging system.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Input Queue Buffer Length` parameter in *Creating an EntireX Broker IQUEUE Definition* and *Creating a WebSphere MQ IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQOPEN Parameter

Parameter	Specify . . .	Possible Values	Default
`IQOPEN`	whether the input queue should be opened automatically when the Event Replicator Server starts up or when an `RPLREFRESH` command is run.	GLOBAL	NO	YES	GLOBAL

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Open Queue at Start` parameter in *Creating an EntireX Broker IQUEUE Definition* and *Creating a WebSphere MQ IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

Valid values for IQOPEN are:

- GLOBAL (the default): This setting indicates that the decision about automatically opening the input queue should be based on the setting of the **GOPEN** global parameter. If the **GOPEN** global parameter is set to "YES", the input queue is opened; if **GOPEN** is set to NO, the input queue is not opened.

- NO: This setting indicates that the input queue should never be opened automatically when the Event Replicator Server starts up or when an `RPLREFRESH` command is run.

- YES: This setting indicates that the input queue should always be opened automatically when the Event Replicator Server starts up or when an `RPLREFRESH` command is run.

IQRETRYCOUNT Parameter

Parameter	Specify . . .	Possible Values	Default
`IQRETRYCOUNT`	the number of times that an attempt to open the input queue will be retried at the interval specified by `IQRETRYINTERVAL`.	0 - 2,147,483,647 or "GLOBAL"	`RETRYCOUNT` parameter setting

This parameter specifies the number of retry attempts that should occur to open the input queue identified by this definition. If the value "GLOBAL" is specified for this parameter, the specification for the global `RETRYCOUNT` parameter will be used. Any retry attempts will occur at the interval specified by the `IQRETRYINTERVAL` parameter. A value of zero indicates that no retry attempt to open this input queue should occur.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Retry Count` parameter in *Creating an EntireX Broker IQUEUE Definition* and *Creating a WebSphere MQ IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQRETRYINTERVAL Parameter

Parameter	Specify . . .	Possible Values	Default
`IQRETRYINTERVAL`	the default number of seconds between retry attempts to open the input queue.	0, 5 - 2,147,483,647, or "GLOBAL"	`RETRYINTERVAL` parameter setting.

This parameter specifies the number of seconds between retry attempts to open the input queue identified by this definition. If the value "GLOBAL" is specified for this parameter, the specification for the global `RETRYINTERVAL` parameter will be used. A value of zero indicates that no retry attempt to open this input queue should occur. Except for a specification of zero, the minimum value that can be specified for this parameter is 5 seconds.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Retry Interval` parameter in *Creating an EntireX Broker IQUEUE Definition* and *Creating a WebSphere MQ IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQTYPE Parameter

Parameter	Specify . . .	Possible Values	Default
IQTYPE	the type of input queue.	ETBROKER | MQSERIES	ETBROKER

If IQTYPE=ETBROKER, the input queue will be accessed via EntireX Broker. If IQTYPE=MQSERIES, the input queue will be accessed via WebSphere MQ.

> **Note:** If you are running on z/OS 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.

This parameter can be selected online in Adabas Event Replicator Subsystem (SYSRPTR). SYSRPTR assigns the input queue type based on the input queue you select to create in SYSRPTR. For more information, read *Maintaining Input Queue (IQUEUE) Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQMQQMGRNAME Parameter

Parameter	Specify . . .	Possible Values	Default
IQMQQMGRNAME	the WebSphere MQ queue manager name. A maximum of 48 characters can be specified.	Check with your WebSphere MQ development staff and the WebSphere MQ documentation.	none

If this parameter is NOT specified, the value specified for parameter MQQMGRNAME will be used. This parameter is required if IQTYPE=MQSERIES and parameter MQQMGRNAME is NOT specified. Up to 48 characters may be specified for this parameter.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Queue Manager Name` parameter in *Creating a WebSphere MQ IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQMQQNAME Parameter

Parameter	Specify . . .	Possible Values	Default
IQMQQNAME	the WebSphere MQ queue name. A maximum of 48 characters can be specified.	Check with your WebSphere MQ development staff and the WebSphere MQ documentation.	IQUEUE NAME

If this parameter is NOT specified, the value specified for parameter IQUEUE NAME will be used.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `MQ Queue Name` parameter in *Creating a WebSphere MQ IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQETBBROKERID Parameter

Parameter	Specify . . .	Possible Values	Default
`IQETBBROKERID`	the webMethods EntireX Broker ID. A maximum of 32 characters can be specified.	Check with your webMethods EntireX Broker development staff and the webMethods EntireX Broker documentation.	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Broker ID` parameter in *Creating an EntireX Broker IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

If this parameter is *not* specified, the default EntireX Broker ID (specified in parameter ETB-BROKERID) will be used.

Broker IDs come in two formats: one for TCP/IP communications and one for Adabas SVC communications. For TCP/IP communications, the format is:

```
ip-address:port-number:TCP
```

In this case, the `ip-address` setting is the TCP/IP IP address and the `port-number` setting should match the webMethods EntireX `PORT` parameter.

For Adabas SVC communications in z/OS and VSE environments, the format is:

```
'broker-id:SVCnnn:NET'
```

In z/OS and VSE environments, the `broker-id` setting should match the webMethods EntireX `BROKER-ID` parameter in the Broker ETBFILE DD. The `nnn` setting should match either the EntireX Broker `ADASVC` or `ADA5SVC` parameters in the Broker PARMS DD statement.

The use of the following parameter form is *not* recommended in BS2000 environments:

```
'broker-id: :NET'
```

IQETBSERVICE Parameter

Parameter	Specify . . .	Possible Values	Default
`IQETBSERVICE`	the webMethods EntireX service. A maximum of 32 characters can be specified.	Check with your webMethods EntireX Broker development staff and the webMethods EntireX Broker documentation.	none

This should be the same as the value specified for the SERVICE parameter in webMethods EntireX.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Broker Service` parameter in *Creating an EntireX Broker IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQETBSERVICENAME Parameter

Parameter	Specify . . .	Possible Values	Default
`IQETBSERVICENAME`	the webMethods EntireX service name. A maximum of 32 characters can be specified.	Check with your webMethods EntireX Broker development staff and the webMethods EntireX Broker documentation.	none

This should be the same as the value specified for the SERVER parameter in webMethods EntireX.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Service Name` parameter in *Creating an EntireX Broker IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQETBSERVICECLASS Parameter

Parameter	Specify . . .	Possible Values	Default
`IQETBSERVICECLASS`	the webMethods EntireX service class. A maximum of 32 characters can be specified.	Check with your webMethods EntireX Broker development staff and the webMethods EntireX Broker documentation.	none

This should be the same as the value specified for the CLASS parameter in webMethods EntireX.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Service Class` parameter in *Creating an EntireX Broker IQUEUE Definition*, in the *Adabas Event Replicator Subsystem User's Guide*.

IQETBSCONV Parameter

Parameter	Specify . . .	Possible Values	Default
IQETBSCONV	whether or not the input queue is to run in single conversation mode for Node to Node support.	YES | NO	NO

This parameter should be used for high volume input queues. At this time only one destination may connect to an input queue that has this parameter set to yes. If more than one were to connect and send messages, a message out of sequence error can occur and the input queue will be closed.

When this optional parameter is set to "YES", it will only accept messages from a destination that also has single conversation mode set to "YES".

In the event of a failure or if the input queue is closed, no DEREGISTER or LOGOFF is issued, thus preserving the conversation ID that was in effect.

Additional considerations when using Single Conversation mode

- The EntireX Broker will only accept the Single Conversation mode if either
    - the Single Conversation mode is already active or
    - there are no active conversations for the specified class, server or service.
- A switch to Single Conversation mode requires that there are no "old" conversations active (see section about Migration to Single Conversation mode below). A mixture of conversation types is not possible.
- A mixture of Single Conversation and non-Single Conversation mode per specified class, server or service is not possible.

Migration to Single Conversation mode

To migrate an environment with a Sending Replicator Server, EntireX Broker and Receiving Replicator Server working with multiple Conversations.

1. The EntireX Broker class, server or service must be empty or drained.

2. The EntireX Broker must be started and run with EXB970L023 / EXX970L023 or higher.

3. The Sending Event Replicator Server "Destination" must be modified with the new Single Conversation Parameters, and the Sending Event Replicator Server must be running Event Replicator for Adabas Version 3.5. SP1 or higher.

4. The Receiving Event Replicator Server "Input Queue" must be modified with the new Single Conversation Parameters, and the Receiving Event Replicator Server must be running Event Replicator for Adabas Version 3.5. SP1 or higher.

5. Open the Single Conversation Destination and input queue.

If the open is successful, ADAFS0 messages will indicate the destination successfully opened with Single Conversation Mode as shown below.

```
ADAFS0 36889 ETBBROKER destination OUTPUT1 connected
ADAFS0 36889 with single conversation mode set
ADAFS0 36889 ETBBROKER Token = REPTOR1
ADAFS0 36889 ETBBROKER Userid = OUT1
```

If the class, server or service was not empty when the Single Conversation mode opens were attempted, error messages as shown below are received.

```
ADAF8A 36889 Destination: OUTPUT1 Open failed: Unexpected reason code
ADAF87 36889 AMI R=Unexpect Q=OUTPUT1 RC=012/036/0000/0000

ADAF9C 36890 Input Queue: INPUT1 Open failed: Unexpected reason co
ADAF87 36890 AMI R=RPIDBG * Q=INPUT1 RC=012/036/0000/0000
ADAF9B 36890 Initialization error in input subtask 001
ADAF8H 36890 Open for input queue INPUT1 failed
```

IQETBTOKEN Parameter

Parameter	Specify . . .	Possible Values	Default
IQETBTOKEN	the webMethods EntireX token.	up to 32 characters	IQIN<input queue name>

This optional parameter can only be specified when IQETBSCONV=YES.

Refer to *TOKEN* in the *EntireX Broker ACI for Assembler* section of the *webMethods EntireX* documentation for more information on usage and syntax.

IQETBUSERID Parameter

Parameter	Specify . . .	Possible Values	Default
IQETBUSERID	the webMethods EntireX user ID.	up to 32 characters	REPTOR--<node>-----<jobname>

This optional parameter can only be specified when IQETBSCONV=YES.

Refer to *USER-ID* in the *EntireX Broker ACI for Assembler* section of the *webMethods EntireX* documentation for more information on usage and syntax.

IRMSGINTERVAL Parameter

Parameter	Specify . . .	Possible Values	Default
IRMSGINTERVAL	the interval during which the message limit specified by the IRMSGLIMIT parameter applies.	1 - 2,147,483,647	60

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Input Request Msg Interval` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
IRMSGINTERVAL=n
```

IRMSGLIMIT Parameter

Parameter	Specify . . .	Possible Values	Default
IRMSGLIMIT	the maximum number of input request error messages issued by the Event Replicator Server during the interval specified by the IRMSGINTERVAL parameter	3 - 2,147,483,647	10

This optional parameter can be used to limit the number of input request error messages issued by the Event Replicator Server in the interval specified by the IRMSGINTERVAL parameter. These error messages relate to invalid information sent to a WebSphere MQ or EntireX Broker input queue. Use of this parameter allows you to prevent the console from being overrun with error messages when invalid data is sent to these input queues.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Input Request Msg Limit` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
IRMSGLIMIT=n
```

LOGINPUTTRANSACTION (Log Input Transaction) Parameter

Parameter	Specify . . .	Possible Values	Default
LOGINPUTTRANSACTION	whether and when database-related input transactions are written to the SLOG system file.	ALL	NO	1-99	NO

You can reduce the risk of Event Replicator Server replication pool overflows by requesting that the Event Replicator Server temporarily write replication transactions received from the Adabas nucleus to the SLOG system file. These transactions are commonly referred to as *database-related input transactions*. Once database-related input transactions have been written to the SLOG system file, the Event Replicator Server then processes them using a throttling mechanism so that only a limited amount of Event Replicator Server replication pool space is used at a time.

Valid values for the LOGINPUTTRANSACTION parameter are described below:

Valid Value	Description
ALL	Specify ALL if you want SLOG system file usage to always be active for database-related input transactions. When ALL is specified, database-related input transactions from the Adabas nucleus are written to the SLOG system file.
NO	Specify NO if you do not want to activate SLOG system file usage for database-related input transactions from the Adabas nucleus. In this case, the transactions will be processed, as usual, using the Event Replicator Server replication pool. This is the default setting.
1-99	Specify an integer from 1 to 99 to set a percentage threshold for Event Replicator Server replication pool usage. If this threshold is exceeded, database-related input transactions will be stored in and processed from the SLOG system file. When the threshold is no longer exceeded and all database-related input transactions have been processed, database-related input transactions from the Adabas nucleus are stored in and processed from the replication pool. The value is specified as a percentage of the setting of the LRPL parameter (which specifies the size of the replication pool).

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Log Input Transaction` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
LOGINPUTTRANSACTION=[ALL | NO | nn]
```

LOGOUTPUTTRANSACTION (Log Output Transaction) Parameter

Parameter	Specify . . .	Possible Values	Default
LOGOUTPUTTRANSACTION	whether and when output items are written to the SLOG system file.	ALL | NO | 1-99	NO

Valid values for the LOGOUTPUTTRANSACTION parameter are described below:

Valid Value	Description
ALL	Specify ALL, if you want SLOG processing to always be turned on for destinations defined with DLOG=YES.
NO	Specify NO, if you want SLOG processing to be turned off for a destination related to space used in the Event Replicator Server replication pool. This is the default setting.
1-99	Specify an integer from 1 to 99 to set a percentage threshold for Event Replicator Server replication pool usage. SLOG processing will be turned on for a destination when: 1. SLOG processing is not already on for the destination, and 2. Space used in the Event Replicator Server replication pool is at or above the threshold, and 3. DLOG=YES is specified for the destination, and 4. The SLOG system file is defined, and 5. A new output item is ready to be assigned to the destination.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Log Output Transaction` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
LOGOUTPUTTRANSACTION=[ALL | NO | nn]
```

MAXOUTPUTSIZE (Maximum Output Message Size) Parameter

Parameter	Specify . . .	Possible Values	Default
MAXOUTPUTSIZE	the maximum output message size, in bytes or KB.	32,768 - 2,147,483,647 bytes	100,000

MAXOUTPUTSIZE may be specified in bytes or it may be specified with the suffix K to indicate kilobytes. The maximum value for MAXOUTPUTSIZE is 2,147,483,647 bytes. The practical maximum is limited by the region size of the Event Replicator Server. One output buffer is acquired for each output task.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Max Output Size` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
MAXOUTPUTSIZE=n[K]
```

MAXRECORDSIZE (Maximum Decompressed Record Length) Parameter

Parameter	Specify . . .	Possible Values	Default
MAXRECORDSIZE	the maximum length (in bytes) of any decompressed record that can be processed by the Event Replicator Server.	1 - 32,767 bytes (Adabas 7 installations) 1-32,767 or 50% of the LRPL setting, whichever is larger (Adabas 8 installations)	32,767 bytes

For an Event Replicator Server running with Adabas 7, the maximum value that can be specified for MAXRECORDSIZE is 32,767 bytes. However, in Adabas 8 systems, this limit has been lifted; the size of a decompressed record may be much larger than 32,767 bytes. Therefore, for an Event Replicator Server running with Adabas 8, the maximum value that can be specified for MAXRECORDSIZE is the larger of either 32,767 bytes or 50% of the setting of the **ADARUN** `LRPL` parameter.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Max Record Size` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
MAXRECORDSIZE=n
```

MAXVARRECORDSIZE (Maximum Variable Decompressed Record Length) Parameter

Parameter	Specify . . .	Possible Values	Default
`MAXVARRECORDSIZE`	the maximum length (in bytes) of variable decompressed records that can be processed by the Event Replicator Server.	1 - MAXRECORDSIZE setting	32,767 bytes

The value specified for MAXVARRECORDSIZE must be less than or equal to the setting of the **MAXRECORDSIZE** setting.

For an Event Replicator Server running with Adabas 7, the maximum value that can be specified for MAXRECORDSIZE or MAXVARRECORDSIZE is 32,767 bytes. However, in Adabas 8 systems, this limit has been lifted; the size of a decompressed record may be much larger than 32,767 bytes. Therefore, for an Event Replicator Server running with Adabas 8, the maximum value that can be specified for MAXRECORDSIZE or MAXVARRECORDSIZE is the larger of either 32,767 bytes or 50% of the setting of the **ADARUN** `LRPL` parameter.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Max Variable Record Size` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
MAXVARRECORDSIZE=n
```

MQQMGRNAME (Default WebSphere MQ Queue Manager) Parameter

Parameter	Specify . . .	Possible Values	Default
`MQQMGRNAME`	the default WebSphere MQ queue manager name.	MAX 48 characters	none

This parameter is used when a WebSphere MQ queue manager name is not specified for a specific WebSphere MQ destination.

Syntax

```
MQQMGRNAME=mqseries-queue-manager-name
```

NPADACALLS (Number of Parallel Adabas Calls) Parameter

Parameter	Specify . . .	Possible Values	Default
NPADACALLS	the maximum number of parallel Adabas calls that can be made.	1 - 50	3

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Num Parallel Adabas Calls` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Use this parameter to improve throughput of data replication to your Adabas destinations.

For example, assume a subscription sends its output to 5 Adabas destinations, each applying data to a file on a separate target database. Suppose a user issues the following commands for the one DBID/file combination in the subscription: `A1` (Update), `A1`, `ET` (End Transaction). If NPADACALLS is set to "1", the Event Replicator Server will send the following commands using a single Adabas user:

- `OP` (open)
- `A1,A1,ET` for the first Adabas destination
- `A1,A1,ET`for the second Adabas destination
- `A1,A1,ET` for the third Adabas destination
- `A1,A1,ET` for the fourth Adabas destination
- `A1,A1,ET` for the fifth Adabas destination
- `CL` (close)

However, if NPADACALLS is set to "5", the Event Replicator Server will send the following commands using five concurrent Adabas users to the five destinations: `OP,A1,A1,ET,CL`.

> **Note:** If NPADACALLS is set to a value greater than "1", you should make sure that the following ADARUN parameters are set appropriately for the target Adabas nucleus, ensuring the target nucleus can handle updates by multiple concurrent users: LBP, LDEUQP, LFIOP, LFP, LP, LWP, NAB, NC, NH, NT, and NU.

Syntax:

```
NPADACALLS=n
```

RECORDPLOGINFO Parameter

Parameter	Specify . . .	Possible Values	Default
RECORDPLOGINFO	whether or not PLOG information should be saved in the Replicator system file. Valid values are "YES" (store the information) or "NO" (do not store the information).	YES | NO	NO

If this parameter is set to "NO", PLOG information is not saved by the Reptor when the nucleus processes the SYNP checkpoint created by ADARES PLCOPY. The PLOG information screens in the Adabas Event Replicator Subsystem will either not display any PLOG data set information or will display outdated information on the screens. Replicator system file.

If you plan on using the automated replay facility provided with Event Replicator, you must set this parameter to "YES". The related information is collected when the source Adabas nucleus processes the SYNP checkpoint created by ADARES PLCOPY.

For more information about the PLOG information screens, read *Reviewing and Managing the PLOG Data Set List*, in the *Adabas Event Replicator Subsystem User's Guide*.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Record PLOG Information` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax

```
RECORDPLOGINFO={ YES | NO }
```

RESENDBUFFER Parameter

The Event Replicator Server can provide a circular buffer to expedite retransmission of a transaction. A single resend buffer can be shared among multiple subscriptions. Resend buffers are defined using the RESENDBUFFER initialization parameter.

RESENDBUFFER initialization parameters use the following syntax:

```
RESENDBUFFER NAME=name
        RSIZE={buffersize|32K}
```

This section covers the following topics:

- NAME

- RSIZE Parameter

NAME

Parameter	Specify . . .	Possible Values	Default
NAME	the unique name for the resend buffer definition.	alphanumeric; 1-8 characters (see constraints below)	none

There are some constraints on the name. It must:

- Be comprised of one to eight uppercase, alphanumeric characters and can include the special characters "@", "$", or "#". If the name is less than eight characters, it is automatically padded on the right with blanks.

- Not begin with a numeric character or a blank.

- Have no embedded blanks.

- Not begin with the letters "SYS".

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Resend Buffer Name` parameter in *Adding Resend Buffer Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*.

RSIZE Parameter

Parameter	Specify . . .	Possible Values	Default
RSIZE	the amount of storage allocated to the resend buffer. Values may be specified in bytes or kilobytes (specify K after the value).	32K - 2,097,151K	32K

If a resend buffer is defined for a subscription that delivers data to multiple destinations, multiple copies of the sent data may be saved, one copy for each destination. The specification of RSIZE must be large enough to accommodate these multiple copies of the data.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Resend Buffer Size` parameter in *Adding Resend Buffer Definitions*, in the *Adabas Event Replicator Subsystem User's Guide*.

RETRYCOUNT (Number of Retry Attempts) Parameter

Parameter	Specify . . .	Possible Values	Default
RETRYCOUNT	the default number of times that an attempt to retry opening a destination or input queue will be attempted at the interval specified by the **RETRYINTERVAL** parameter.	0 - 2,147,483,647	10

This global parameter specifies the number of attempts to retry opening any destination or input queue for which no specific retry count has been specified. A value of zero indicates that no retry processing should occur for any affected destinations or input queues.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Retry Count` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
RETRYCOUNT=n
```

RETRYINTERVAL (Retry Attempt Interval) Parameter

Parameter	Specify . . .	Possible Values	Default
RETRYINTERVAL	the default number of seconds between attempts to retry opening a destination or input queue.	0 - 2,147,483,647	0

This global parameter specifies the number of seconds between attempts to retry opening any destination or input queue for which no specific retry interval has been specified. A value of zero indicates that no retry processing should occur for any affected destinations or input queues.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Retry Interval` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
RETRYINTERVAL={n|GLOBAL}
```

SLOGCHECKINTERVAL Parameter

Parameter	Specify . . .	Possible Values	Default
SLOGCHECKINTERVAL	the interval (in minutes) between checks of the Associator and Data file space usage.	0, 5 - 2880	0

The SLOGCHECKINTERVAL is an optional parameter. If it is not specified, a value of zero (0) is assumed, indicating that no threshold monitoring should occur. If it is specified, valid values are zero (0) or any numbers in the range from 5 to 2880 minutes (48 hours).

Once a value is specified, the settings of the thresholds specified by the ASSOTHRESHOLD, DATATHRESHOLD, SLOGACTHRESHOLD, SLOGNITHRESHOLD, SLOGUITHRESHOLD, and SLOGDSTHRESHOLD initialization parameters are used to evaluate your system's Adabas and SLOG Associator and Data file space usage. If the thresholds are met or reached, warning messages are issued to the console and DDPRINT.

This section covers the following topics:

- ASSOTHRESHOLD
- DATATHRESHOLD
- SLOGACTHRESHOLD
- SLOGDSTHRESHOLD
- SLOGNITHRESHOLD
- SLOGUITHRESHOLD

ASSOTHRESHOLD

Parameter	Specify . . .	Possible Values	Default
ASSOTHRESHOLD	the percentage of the Event Replicator database Associator space that can be used before threshold warning messages are issued. Associator space usage must meet or exceed this threshold value to trigger the warning messages.	0 - 100	80

The ASSOTHRESHOLD is an optional parameter. If it is not specified, a default of 80% is assumed.

If the percentage of space used meets or exceeds the setting of this parameter, warning messages are issued to the console.

If zero (0) is specified for this parameter, no threshold monitoring will occur for the Event Replicator database Associator space.

> 📄 **Note:** If the `SLOGCHECKINTERVAL` initialization parameter is not specified, no Event Replicator database Associator space threshold monitoring will occur, regardless of the setting of this parameter.

DATATHRESHOLD

Parameter	Specify . . .	Possible Values	Default
DATATHRESHOLD	the percentage of the Event Replicator database Data space that can be used before threshold warning messages are issued. Data space usage must meet or exceed this threshold value to trigger the warning messages.	0 - 100	80

The `DATATHRESHOLD` is an optional parameter. If it is not specified, a default of 80% is assumed.

If the percentage of space used meets or exceeds the setting of this parameter, warning messages are issued to the console.

If zero (0) is specified for this parameter, no threshold monitoring will occur for the Event Replicator database Data space.

> 📄 **Note:** If the `SLOGCHECKINTERVAL` initialization parameter is not specified, no Event Replicator database Data space threshold monitoring will occur, regardless of the setting of this parameter.

SLOGACTHRESHOLD

Parameter	Specify . . .	Possible Values	Default
SLOGACTHRESHOLD	the percentage of the SLOG address converter space that can be used before threshold warning messages are issued. Address converter space usage must meet or exceed this threshold value to trigger the warning messages.	0 - 100	70

The `SLOGACTHRESHOLD` is an optional parameter. If it is not specified, a default of 70% is assumed.

If the percentage of space used meets or exceeds the setting of this parameter, warning messages are issued to the console. To determine if the address converter threshold has been reached, the percentage used is calculated using the following formula:

```
(TOPISN-MINISN+1)/(MAXISN-MINISN+1)
```

If zero (0) is specified for this parameter, no threshold monitoring will occur for the SLOG address converter space.

> **Note:** If the `SLOGCHECKINTERVAL` initialization parameter is not specified, no SLOG address converter space threshold monitoring will occur, regardless of the setting of this parameter.

SLOGDSTHRESHOLD

Parameter	Specify . . .	Possible Values	Default
SLOGDSTHRESHOLD	the percentage of the SLOG Data Storage space that can be used before threshold warning messages are issued. Data Storage space usage must meet or exceed this threshold value to trigger the warning messages.	0 - 100	70

The `SLOGDSTHRESHOLD` is an optional parameter. If it is not specified, a default of 70% is assumed.

If the percentage of space used meets or exceeds the setting of this parameter, warning messages are issued to the console. To determine if the Data Storage threshold has been reached, the percentage used is calculated using the following formula, where `acyl` represents the number of allocated cylinders, `uncyl` represents the number of unused cylinders, and `unbytes` represents the number of unused bytes (per the DSST):

```
(acyl - uncyl - unbytes) / acyl
```

If zero (0) is specified for this parameter, no threshold monitoring will occur for the SLOG Data space.

> **Note:** If the `SLOGCHECKINTERVAL` initialization parameter is not specified, no SLOG Data Storage space threshold monitoring will occur, regardless of the setting of this parameter.

SLOGNITHRESHOLD

Parameter	Specify . . .	Possible Values	Default
SLOGNITHRESHOLD	the percentage of the SLOG normal index space that can be used before threshold warning messages are issued. Normal index space usage must meet or exceed this threshold value to trigger the warning messages.	0 - 100	70

The `SLOGNITHRESHOLD` is an optional parameter. If it is not specified, a default of 70% is assumed.

If the percentage of space used meets or exceeds the setting of this parameter, warning messages are issued to the console. To determine if the normal index threshold has been reached, the per-

centage used is calculated using the following formula, where `acyl` represents the number of allocated cylinders and `ucyl` represents the number of used cylinders:

```
(ucyl / acyl)
```

If zero (0) is specified for this parameter, no threshold monitoring will occur for the SLOG normal index space.

> **Note:** If the `SLOGCHECKINTERVAL` initialization parameter is not specified, no SLOG normal index space threshold monitoring will occur, regardless of the setting of this parameter.

SLOGUITHRESHOLD

Parameter	Specify . . .	Possible Values	Default
SLOGUITHRESHOLD	the percentage of the SLOG upper index space that can be used before threshold warning messages are issued. Upper index space usage must meet or exceed this threshold value to trigger the warning messages.	0 - 100	70

The `SLOGUITHRESHOLD` is an optional parameter. If it is not specified, a default of 70% is assumed.

If the percentage of space used meets or exceeds the setting of this parameter, warning messages are issued to the console. To determine if the upper index threshold has been reached, the percentage used is calculated using the following formula, where `acyl` represents the number of allocated cylinders and `ucyl` represents the number of used cylinders:

```
(ucyl / acyl)
```

If zero (0) is specified for this parameter, no threshold monitoring will occur for the SLOG upper index space.

> **Note:** If the `SLOGCHECKINTERVAL` initialization parameter is not specified, no SLOG upper index space threshold monitoring will occur, regardless of the setting of this parameter.

STATINTERVAL Parameter

Parameter	Specify . . .	Possible Values	Default
STATINTERVAL	the interval (in seconds) at which Event Replicator statistics should be published by the Event Replicator Server.	0, 60-86400	0

Note: This parameter is to be used with *Replication Monitoring* only.

The STATINTERVAL initialization parameter must be set to a value greater than zero to activate *Replication Monitoring*. This will result interval statistics being generated by the Event Replicator Server.

A value of zero (0) indicates that interval statistics should not be published and Replication Monitoring will be deactivated.

When the value of the STATINTERVAL is greater than zero, a Destination with a DSTATLOG setting of YES must be setup. See the DSTATLOG parameter for more details.

SUBSCRIPTION Parameter

One or more subscriptions will be defined in the replication definitions. A subscription defines a set of rules to be applied to the replicated data. Each subscription defines the following:

1. The output alpha code used during decompression.

2. The output architecture key used during decompression.

3. One or more output destinations.

4. The version ID of the subscription.

5. The output wide-code used during decompression.

6. One or more combinations of database ID and file denoting the input replication data to be processed for the subscription.

7. In addition, the following is defined for each combination of database ID and file:

    a. The format buffer used when decompressing the data storage after image.

    b. The format buffer used when decompressing the data storage before image.

    c. The format buffer used when decompressing the key related before image.

    d. An optional subscription exit name.

The following parameters are used to define a SUBSCRIPTION:

SUBSCRIPTION initialization parameters use the following syntax:

```
SUBSCRIPTION NAME='subscription-name'
           [SACODE={user-data-alpha-key|0}]
           [SACTIVE=YES|NO]
           [SARC=architecture-key|2]
           [SDEACTIVATE=YES|NO]
           [SDESTINATION='destination-name-list']
           [SINCREMENTIS=YES|NO]
           [SIDESTINATION='NONE' | 'destination-name-list']
           [SRESENDBUFFER=resendbuffer-name]
           [STLFILTER={0 | 1 | 2 | 3 }]
           [STLIGNORE={0 | 1 | 2 }]
           [STLINPUT={0 | 1 | 2 | 3 }]
           [STLMATCH={0 | 1 | 2 }]
           [STLNOMATCH={0 | 1 | 2 }]
           [STLOUTPUT={0 | 1 | 2 | 3 }]
           [SVERSION=subscription-version-indicator]
           [SWCODE=user-data-wide-key]
```

One or more of the following group of subparameters are also specified:

```
SFILE='file-num'
           ↵
{SFBAI='format-buffer-used-with-data-storage-after-image'|SGFORMATAI=global-format-name}
           ↵
[SFBBI='format-buffer-used-with-data-storage-before-image'|SGFORMATBI=global-format-name]
           ↵
[SFBKEY='format-buffer-used-key-before-image'|SGFORMATKEY=global-format-name]
           [SFDBID=database-id|DBID-parm-value]
           [SFDEFAULTACODE=file-alpha-EBCDIC-key]
           [SFFILTER=transaction-filter-name]
           [SFFILTERGFBAI=global-format-name]
           [SFFILTERGFBBI=global-format-name]
           [SFREPLICATEDELETE=YES|NO|UPDATE]
           [SFREPLICATEINSERT=YES|NO]
           [SFREPLICATENOTCHANGED=YES|NO]
           [SFREPLICATEUPDATE=YES|NO]
           [SFSECURITYFILE=YES|NO]
           [SFSEXIT=subscription-exit-name]
```

This section covers the following topics:

- NAME Parameter
- SACODE Parameter
- SACTIVE Parameter
- SARC Parameter
- SDEACTIVATE Parameter
- SDESTINATION Parameter
- SFBAI and SGFORMATAI Parameters
- SFBBI and SGFORMATBI Parameters

NAME Parameter

Parameter	Specify . . .	Possible Values	Default
NAME	the unique name for the subscription definition.	alphanumeric; 1-8 characters	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Subscription Name` parameter in *Step 2. Supply General Subscription Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SACODE Parameter

Parameter	Specify . . .	Possible Values	Default
SACODE	the desired output character encoding. This can only be set when the Event Replicator Server has been started with Universal Encoding Support (UES) enabled.	numeric	0 (zero)

The A (alpha) field key code is documented in the record buffer section of the OP command in the Adabas command reference documentation (*Adabas Command Reference*). A value of zero means that no UES translation will be done.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `User Data Alpha Key` parameter in *Step 2. Supply General Subscription Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

Event Replicator Target Adapter Considerations

The subscription data sent to the Event Replicator Target Adapter is wrapped in XML messages. By, default the data is assumed to be encoded in US-EBCDIC (SACODE=37) and is converted to Unicode by the Event Replicator Target Adapter for interpretation. EBCDIC values for alphanumeric characters X'40' thru X'FF' are sent unchanged; EBCDIC values for alphanumeric fields under X'40' are translated to spaces.

If the source data is in a different code page, the parameter SACODE must be specified. The SACODE is sent with the subscription data so that the Event Replicator Target Adapter can convert it correctly. Not all source encodings can be specified because of conflicts with the Event Replicator Target Adapter protocol. Only SACODE settings of 37, 424, 813, 912, 915, 920, 922, 923, 1006, 1112, 1140, 1256, and 4091 are valid. The following table lists the single byte EBCDIC source encodings and their possible subscription encodings. If a specified encoding is not supported, a warning message will be printed and encoding 37 (EBCDIC) will be used.

> **Note:** For double-byte host encodings, SACODE should be set to 4091 (UTF-8). If you want to use UTF-8 character encoding, you must verify that your field lengths are increased as required to accommodate UTF-8 character encoding.

Source Encoding	Description	SACODE Setting
37	IBM037: USA, Canada, etc. (EBCDIC)	37
273	IBM273: Austria, Germany (EBCDIC)	
277	IBM277: Denmark, Norway (EBCDIC)	
278	IBM278: Finland, Sweden (EBCDIC)	
280	IBM280: Italy (EBCDIC)	
284	IBM284: Spanish (EBCDIC)	
285	IBM285: United Kingdom (EBCDIC)	

Source Encoding	Description	SACODE Setting
290	EBCDIC Japanese Katakana	4091
297	IBM297: France (EBCDIC)	37
420	IBM420 Arabic (presentation shapes) (EBCDIC)	1256
424	IBM424 Hebrew (EBCDIC)	424
500	IBM500: Belgium, Canada, Swiss International	37
803	Old Hebrew EBCDIC ibm-803_P100-1999	424
836	Simplified Chinese Host Extended SBCS (EBCDIC)	
870	IBM870 Multilingual: Latin-2 (EBCDIC)	912
871	IBM871 CECP: Iceland (EBCDIC)	37
875	Greek (EBCDIC)	813
918	IBM918 Urdu (EBCDIC)	1006
924	IBM00924: Latin-15, Belgium, Canada, Swiss, International	923
1025	Cyrillic Multilingual (EBCDIC)	915
1026	IBM1026 Turkey Latin-5 (EBCDIC) to ISO-8859-9	920
1027	Japanese Latin Host Extended SBCS (EBCDIC)	4091
1047	EBCDIC Latin 1 Open Systems	37
1112	Baltic, Multilingual (EBCDIC)	1112
1122	Estonia (EBCDIC)	922
1140	IBM01140: USA, Canada, (Euro)	1140
1141	IBM01141: Austria, Germany (EBCDIC Euro)	
1142	IBM01142: Denmark, Norway (EBCDIC Euro)	
1143	IBM01143: Finland, Sweden (EBCDIC Euro)	
1144	IBM01144: Italy (EBCDIC Euro)	
1145	IBM01145: Spanish (EBCDIC Euro)	
1146	IBM01146: United Kingdom (EBCDIC Euro)	
1147	IBM01147: France (EBCDIC Euro)	
1148	IBM01148: Belgium Canada Swiss (EBCDIC Euro)	
1149	IBM01149: Iceland (EBCDIC Euro)	
3585	DF.03 International CCSN:EDF03IRV	37
3586	DF.03 CCSN:EDF03 Austria, Germany	
3587	DF.03 CCSN:EDF03 Austria, Germany	
3588	DF.04-DRV LATIN 1, Austria, Germany	
3589	DF.04-1 LATIN 1, International	
3709	IBM28709: Trad. Chinese Host Ext. SBCS (EBCDIC)	
4094	Old TS default EBCDIC	

In the following example, IBM424 Hebrew (EBCDIC) encoding is requested for replicated data via the SACODE=424 parameter. Because the source database is not UES-enabled in this case, the SFDEFAULTACODE=803 parameter specifies the encoding of the source data (in this case, Old Hebrew EBCDIC).

```
ADARPD SUBSCRIPTION NAME=EMPLOYEE
ADARPD  SDESTINATION='OUT4'
ADARPD  SACODE=424         <- target encoding *)
ADARPD  SFILE=11,SFDBID=10006,SFDEFAULTACODE=803  <- source encoding **)
```

SACTIVE Parameter

Parameter	Specify . . .	Possible Values	Default
SACTIVE	whether the subscription definition should be activated once it is loaded.	"YES" or "NO"	YES

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Subscription Active parameter in *Step 2. Supply General Subscription Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

If "YES" is specified, the subscription definition is loaded and activated at Event Replicator Server startup; if "NO" is specified, the definition is loaded but not activated at Event Replicator Server startup (it is inactive).

Only activated subscription definitions can be used during replication. If you need to activate an inactive subscription definition after Event Replicator Server startup, use the Replicator Activate/Deactivate screen of the Adabas Online System. For more information, read *Adabas Online System Features Supporting Event Replicator for Adabas* in *Event Replicator for Adabas Administration and Operations GuideAdabas Online System Features Supporting Event Replicator for Adabas* in *Event Replicator for Adabas Administration and Operations Guide*.

SARC Parameter

Parameter	Specify . . .	Possible Values	Default
SARC	the desired output architecture key. This can only be set when the Event Replicator Server has been started with Universal Encoding Support (UES) enabled.	0-11	2

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Architecture Key parameter in *Step 2. Supply General Subscription Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

The architecture key is an integer that is calculated as the sum of the following numbers (this is the same as is documented in the record buffer section of the OP command in the Adabas Command Reference Guide):

byte order	b=0	high-order byte first
	b=1	low-order byte first
encoding family	e=0	ASCII encoding family
	e=2	EBCDIC encoding family (default for local calls)
floating-point format	f=0	IBM370 floating-point format
	f=4	VAX floating-point format
	f=8	IEEE floating-point format

If UES has not been enabled, leave this parameter set to "2".

> **Note:** If you want to transfer replicated and initial-state data to a relational database using the Event Replicator Target Adapter (in otherwords the `DCLASS` **parameter** is set to "SAGTARG"), set the SARC parameter to "2" -- regardless of the location of your relational database.

SDEACTIVATE Parameter

Parameter	Specify . . .	Possible Values	Default
`SDEACTIVATE`	whether the subscription should be deactivated when one or more of the files assigned to it are deactivated.	"YES" or "NO"	YES

If "YES" is specified, the subscription is deactivated when one or more of the files assigned to it are deactivated; if "NO" is specified, the subscription is *not* deactivated when one of its files is deactivated.

While this is an optional feature, we recommend that you use the default or specify SDEACTIV-ATE=YES to ensure that the integrity of your replicated data is maintained when a file becomes deactivated. If the subscription is deactivated because one of its files is deactivated, any other files referenced by this subscription are also deactivated unless they are also referenced by another active subscription.

If replication for this file in the subscription is unrelated to the replication for other files in the same subscription, the integrity of the replicated data may not be at risk. In this case, SDEACTIV-ATE=NO might be specified.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Deactivate if file deactivated` parameter in *Step 2. Supply General Subscription Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SDESTINATION Parameter

Parameter	Specify . . .	Possible Values	Default
SDESTINATION	the name or names of the destination definitions used by the subscription.	alphanumeric; 1-8 characters	none

This parameter is required. A matching destination name must be defined through the DESTINA-TION parameter (and its sub-parameters). Data processed for the subscription will be sent to this destination. This parameter can be specified more than once for one subscription.

At least one destination must be specified (and a maximum of 32 destinations may be specified) in the SDESTINATION or SIDESTINATION parameters.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about specifying destinations for the subscription in *Step 3. Specify One or More Destinations for the Subscription*, in the *Adabas Event Replicator Subsystem User's Guide*.

SFBAI and SGFORMATAI Parameters

Parameter	Specify . . .	Possible Values	Default
SFBAI	the format buffer used when decompressing the data storage after image.	Valid Adabas format buffer up to 40 characters	none
SGFORMATAI	the name of the global format buffer definition to be used when decompressing the data storage after image. The global format definition must be defined through the parameters GFORMAT and GFB.	Valid global format buffer definition name (maximum of seven characters)	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `After Image FB` and `After Image GFB Name` parameters in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

Format buffers for data storage after images must conform to the format buffer rules (for a read command) documented in the Adabas command reference documentation (*Adabas Command Reference*).

Parameters SFBAI and SGFORMATAI are mutually exclusive. One of these parameters must be specified for a specific input DBID/file.

> **Note:** The maximum length of a format buffer specified in parameter SFBAI is 40. This is a temporary restriction. If a longer format buffer is needed, take the following steps:

1. Specify the longer format buffer in the GFORMAT and GFB parameters.

2. Specify SGFORMATAI to use the related global format.

PE and MU fields cannot use the range notation 1-N in format buffers for a subscription that is sent to a destination that has specified the replication intialization parameter DCLASS=SAGTARG. The SAGTARG application invoked requires that the range of occurrences specified are contained in the record buffer even if they are empty occurrences. 1-N results in a range of 1-191, but unless there are 191 occurrences containing data, space in the record buffer is not allocated for any empty occurrences, resulting in incorrect field positioning when processing the record.

If the destination target for replicated data is an Adabas mainframe database running Adabas 8.1.3 or later, you can specify the symbolic notation "C." alone as the SFBAI parameter value (for example, `SFBAI='C.'`). This notation will cause the Event Replicator Server to replicate the entire source record to the target mainframe Adabas database. Likewise, the SGFORMATAI parameter can reference a GFB that uses this symbolic notation.

Caution: This "C." option may be used *only* if the destination target file has been defined with the same fields in the same order as the fields in the source file; if there are differences in the definitions of the files, the replication of the data is likely to incur errors. The only exception to this rule is that the definitions of the descriptors and superdescriptors in the target and source files may be different.

If you are using the Event Replicator Target Adapter, the value used for the **SGFORMATAI** parameter and the **SGFORMATBI** parameter must be the same. In addition, the key image global format buffer specified in the **SGFORMATKEY** parameter must have been built from a Predict user view with the same name as the user view used to build the before and after images, but with the suffix "-KEY" on the end. (Likewise, if you use the Data Mapping Tool, the **SGFORMATKEY** GFB should be built from a DDM with the same name as the DDM used to build the before and after image GFBs.) Event Replicator processing strips off the "-KEY" to ensure that any delete is associated with the before and after image file name that was used to build the table(s) in the RDBMS.

SFBBI and SGFORMATBI Parameters

Parameter	Specify . . .	Possible Values	Default
SFBBI	the format buffer used when decompressing the data storage before image.	Valid Adabas format buffer up to 40 characters	none
SGFORMATBI	the name of the global format buffer definition to be used when decompressing the data storage before image. The global format definition must be defined through the parameters GFORMAT and GFB.	Valid global format buffer definition name (maximum of seven characters)	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Before Image FB` and `Before Image GFB Name` parameters in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

Format buffers for data storage before images must conform to the format buffer rules (for a read command) documented in the Adabas command reference documentation (*Adabas Command Reference*).

The SFBBI and SGFORMATBI parameters are mutually exclusive. If neither of these parameters is specified, the parameter specified for the after image (for example, SFBAI or SGFORMATAI) will be used.

> **Note:** The maximum length of a format buffer specified in parameter SFBBI is 40. This is a temporary restriction. If a longer format buffer is needed, take the following steps:

1. Specify the longer format buffer in the GFORMAT and GFB parameters.
2. Specify SGFORMATBI to use the related global format.

PE and MU fields cannot use the range notation 1-N in format buffers for a subscription that is sent to a destination that has specified the replication intialization parameter DCLASS=SAGTARG. The SAGTARG application invoked requires that the range of occurrences specified are contained in the record buffer even if they are empty occurrences. 1-N results in a range of 1-191, but unless there are 191 occurrences containing data, space in the record buffer is not allocated for any empty occurrences, resulting in incorrect field positioning when processing the record.

If you are using the Event Replicator Target Adapter, the value used for the **SGFORMATAI** parameter and the **SGFORMATBI** parameter must be the same. In addition, the key image global format buffer specified in the **SGFORMATKEY** parameter must have been built from a Predict user view with the same name as the user view used to build the before and after images, but with the suffix "-KEY" on the end. (Likewise, if you use the Data Mapping Tool, the **SGFORMATKEY** GFB should be built from a DDM with the same name as the DDM used to build the before and after image GFBs.) Event Replicator processing strips off the "-KEY" to ensure that any delete is associated with the before and after image file name that was used to build the table(s) in the RDBMS.

SFBKEY and SGFORMATKEY Parameters

Parameter	Specify . . .	Possible Values	Default
SFBKEY	the format buffer used when decompressing the key-related before image.	Valid Adabas format buffer up to 40 characters	none
SGFORMATKEY	the name of the global format buffer definition to be used when decompressing the key-related before image. The global format definition must be defined through the parameters GFORMAT and GFB.	Valid global format buffer definition name (maximum of seven characters)	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Key-Related Before Image FB` and `Key-Related GFB Name`

parameters in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

Format buffers for key-related before images must conform to the format buffer rules (for an L9 command) documented in the Adabas command reference documentation (*Adabas Command Reference*).

The SFBKEY and SGFORMATKEY parameters are mutually exclusive. If a key-related before image is received and SFBKEY/SGFORMATKEY have *not* been specified, the field name followed by a period will be used for the format buffer.

If you are using the Event Replicator Target Adapter, the value used for the **SGFORMATAI** parameter and the **SGFORMATBI** parameter must be the same. In addition, the key image global format buffer specified in the **SGFORMATKEY** parameter must have been built from a Predict user view with the same name as the user view used to build the before and after images, but with the suffix "-KEY" on the end. (Likewise, if you use the Data Mapping Tool, the **SGFORMATKEY** GFB should be built from a DDM with the same name as the DDM used to build the before and after image GFBs.) Event Replicator processing strips off the "-KEY" to ensure that any delete is associated with the before and after image file name that was used to build the table(s) in the RDBMS.

SFDBID Parameter

Parameter	Specify . . .	Possible Values	Default
SFDBID	the DBID associated with the SFILE parameter.	Valid database ID	The value of the **DBID parameter**.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `DBID` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SFDEFAULTACODE Parameter

Parameter	Specify . . .	Possible Values	Default
SFDEFAULTACODE	a default, at the subscription file level, for the file's alpha character encoding. The encoding must belong to the EBCDIC encoding family; that is, the space character must be X'40'.'. This can only be set when the Event Replicator Server has been started with Universal Encoding Support (UES) enabled.	alpha character code page number	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Default Code` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

This parameter is meant to be used when *all* of the following conditions are met:

- The Event Replicator Server is UES-enabled,

- The source database is not UES-enabled

- The subscription definition requests UES translation (i.e. parameter `SACODE` is specified)

- The data in the source database file is stored in a code page other than the default alpha encoding.

At the time a *before* or *after* image is decompressed and translated in the Event Replicator Server, the file encoding is taken as follows:

1. from the source FCB;

2. if the encoding is not set above, from the GCB of the source database;

3. if the encoding is not set above, from the value set for this parameter (SFDEFAULTACODE);

4. if the encoding is not set above, from the GCB of the Event Replicator Server.

Note that if the source database is UES-enabled, the file encoding will be taken from either 1 or 2 above.

SFFILTER Parameter

Parameter	Specify . . .	Possible Values	Default
`SFFILTER`	a transaction filter definition that should be used by the `SFILE` definition.	Valid transaction filter name	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Transaction Filter` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

Transaction filter definitions specify filter conditions for specific fields within the `SFILE` record. When the conditions are met, the record is selected either to be included in replication processing or to be excluded from replication processing, depending on the `FRECORDS` setting in the transaction filter definition. Transaction filter definitions are defined by the `FILTER` initialization parameter and its related parameters and subparameters.

> **Note:** Any filter field specified in a transaction filter definition used by a subscription file (SFILE) definition must also be included in either the format buffers for the SFILE definition (**SFBAI**, **SGFORMATAI**, **SFBBI**, and **SGFORMATBI** parameters) or in filter format buffers for the SFILE definition (**SFFILTERGFBAI** and **SFFILTERGFBBI** parameters).

SFFILTERGFBAI Parameter

Parameter	Specify . . .	Possible Values	Default
SFFILTERGFBAI	the name of the global format buffer definition (GFB) to be used when filtering the data storage after image as described by the SFFILTER parameter. The global format definition must be defined through the parameters GFORMAT and GFB.	Valid global format buffer definition name (maximum of seven characters)	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Filter After Image FB` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

The SFFILTERGFBAI parameter specifies the after image filter format buffer for a subscription file (SFILE) definition. Ordinarily it is optional. It is only required if *both* of the following conditions are met:

1. A transaction filter definition is specified in the SFFILTER parameter of the SFILE definition.

2. The format buffer or global format buffer specified for the data storage after image in the SFILE definition does not contain one or more of the fields defined in the filter.

For example, suppose the SFFILTER parameter identifies a transaction filter that filters the data storage after image for the field AA. But suppose the normal after image format buffer for the SFILE definition does not include field AA (it might be `C.` or some other format buffer). Normally, this combination of specifications will not work. However, if you include an SFFILTERGFBAI parameter that specifies a global format buffer that includes field AA, after image records are processed as described by the filter and the associated after image filter format buffer.

This parameter can be used to improve the performance of Event Replicator processing. For example, if most records for a large format buffer are being rejected due to a filter based on the contents of a small number of fields, it may help to specify the key fields in a filter format buffer so that, for most records, only the fields required to make acceptance/rejection decisions are decompressed instead of the entire buffer.

SFFILTERGFBBI Parameter

Parameter	Specify . . .	Possible Values	Default
SFFILTERGFBBI	the name of the global format buffer (GFB) definition to be used when filtering the data storage before image as described by the SFFILTER parameter. The global format definition must be defined through the parameters GFORMAT and GFB.	Valid global format buffer definition name (maximum of seven characters)	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Filter Before Image FB` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

The SFFILTERGFBBI parameter specifies the before image filter format buffer for a subscription file (SFILE) definition. Ordinarily it is optional. It is only required if *both* of the following conditions are met:

1. A transaction filter definition is specified in the SFFILTER parameter of the SFILE definition.

2. The format buffer or global format buffer specified for the data storage before image in the SFILE definition does not contain one or more of the fields defined in the filter.

For example, suppose the SFFILTER parameter identifies a transaction filter that filters the data storage before image for the field AA. But suppose the normal before image format buffer for the SFILE definition does not include field AA (it might be `'C.'` or some other format buffer). Normally, this combination of specifications will not work. However, if you include an SFFILTERGFBBI parameter that specifies a global format buffer that includes field AA, before image records are processed as described by the filter and the associated before image filter format buffer.

This parameter can be used to improve the performance of Event Replicator processing. For example, if most records for a large format buffer are being rejected due to a filter based on the contents of a small number of fields, it may help to specify the key fields in a filter format buffer so that, for most records, only the fields required to make acceptance/rejection decisions are decompressed instead of the entire buffer.

SFILE Parameter

Parameter	Specify . . .	Possible Values	Default
SFILE	the file number of input data processed by the subscription.	Valid file number	none

The combination of `SFDBID` and `SFILE` is used to determine whether or not input data is processed for the subscription. One or more combinations of `SFDBID` and `SFILE` (along with the related subparameters) may be specified for a subscription. The number of `SFILE`s allowed for a given subscription depends on the `GLOBALS NUMSFILE` setting.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `File Number` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SFREPLICATEDELETE Parameter

Parameter	Specify . . .	Possible Values	Default
SFREPLICATEDELETE	whether replication should occur if an input record for the database file is for a delete.	"YES", "NO", or "UPDATE"	YES

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Replicate for Delete` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

> **Note:** IF SFREPLICATEDELETE is set to UPDATE, a value must be specified for the SFSEXIT parameter. In addition, the format buffer before and after images must be identical.

If SFREPLICATEDELETE=NO is specified and an input record for the DBID/file is for a delete, the input record will NOT be processed for this subscription. If SFREPLICATEDELETE=YES and an input record for the DBID/file is for a delete, the input record will be processed for this subscription.

If SFREPLICATEDELETE=UPDATE is specified and an input record for the DBID/file is for a delete, the before and after images of the input record will be passed to your subscription user exit. Therefore, if SFREPLICATEDELETE=UPDATE is specified, a subscription user exit name must be specified in the SFSEXIT parameter. In addition, the subscription before and after image format buffers must be identical and no primary key should be defined to the file (to ensure that the before image is a copy of data storage). The purpose of the UPDATE value of this parameter is to allow your subscription user exit to process replicated physical delete transactions on your target database as you choose. Your subscription user exit can decide if the physical delete transaction should be:

- physically deleted from your target database
- converted to an update
- ignored and not sent at all.

For more information about using the subscription user exit with SFREPLICATEDELETE=UPDATE, read *Controlling Delete Transaction Processing (SFREPLICATEDELETE=UPDATE Processing)*, in the *Event Replicator for Adabas Administration and Operations Guide*.

SFREPLICATEINSERT Parameter

Parameter	Specify . . .	Possible Values	Default
`SFREPLICATEINSERT`	whether replication should occur if an input record for the database file is for an insert.	"YES" or "NO"	YES

If SFREPLICATEINSERT=NO is specified and an input record for the DBID/file is for an insert, the input record will NOT be processed for this subscription. If SFREPLICATEINSERT=YES and an input record for the DBID/file is for an insert, the input record will be processed for this subscription.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Replicate for Insert` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SFREPLICATENOTCHANGED Parameter

Parameter	Specify . . .	Possible Values	Default
`SFREPLICATENOTCHANGED`	whether an update request should be replicated if the after image (AI) is the same as the before image (BI).	"YES" or "NO"	YES

If SFREPLICATENOTCHANGED=YES is specified and the AI and BI record images are the same, the record will be replicated, subject to any transaction filters that may be specified for the SFILE file. If SFREPLICATENOTCHANGED=NO is specified and the AI and BI record images are the same, no further processing will occur for the record.

> **Note:** The format buffer used for the AI and BI record images must be the same as the comparison to determine if they are identical is simply done against the decompressed records.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Include Identical Records` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SFREPLICATEUPDATE Parameter

Parameter	Specify . . .	Possible Values	Default
`SFREPLICATEUPDATE`	whether replication should occur if an input record for the database file is for an update.	"YES" or "NO"	YES

If SFREPLICATEUPDATE=NO is specified and an input record for the DBID/file is for an update, the input record will NOT be processed for this subscription. If SFREPLICATEUPDATE=YES and an input record for the DBID/file is for an update, the input record will be processed for this subscription.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Replicate for Update` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SFSECURITYFILE Parameter

Parameter	Specify . . .	Possible Values	Default
SFSECURITYFILE	whether replication of the security definitions should occur. The setting of this parameter allows you to define a subscription file (SFILE) definition specifically for security definitions in the Adabas security file on the database.	"YES" or "NO"	NO

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Replicate Security File` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

> **Notes:**

1. Adabas 8.2 or later must be installed in order to replicate updates to the Adabas security definitions. If SFSECURITYFILE=YES is specified, but Adabas 8.2 or later is *not* installed, no errors will occur, but updates to security definitions will *not* be replicated.

2. Adabas Security Facilities, including the Adabas security utility (ADASCR) can be obtained only by special request. If you are interested in Adabas Security Facilities, please contact your Software AG sales representative.

When SFSECURITYFILE=NO is specified, replication of the security definitions does not occur and replication processing proceeds normally.

When SFSECURITYFILE=YES is specified for a subscription file definition, replication of the security definitions in the Adabas security file will occur. In addition, the following parameter settings are required:

- A format buffer may *not* be specified. This means that no values may be specified for the SFBAI, SFBBI, SFBKEY, SGFORMATTAI, SGFORMATBI, or SGFORMATKEY parameters.

- No value may be specified for the SDEFAULTACODE parameter.

- A transaction filter definition may *not* be specified. This means that no value may be specified for the SFFILTER parameter.

- The default value of *YES* must be specified for the SFREPLICATEDELETE, SFREPLICATEINSERT, SFREPLICATENOTCHANGED, and SFREPLICATEUPDATE parameters.

- A subscription exit may *not* be specified. This means that no value may be specified for the SFSEXIT parameter.

- When you specify SFSECURITYFILE=YES, you indicate that the file specified in the SFILE parameter is the file number of the security file for the source database identified by the SFDBID

parameter. Therefore, if SFSECURITYFILE=YES is specified for other subscription SFILE definitions using the same source database (with the same SFDBID setting), the same value must be set for each of the SFILE parameters in the SFILE definition. In other words, it is invalid for a source database (SFDBID setting) to have different file numbers specified for the security file in different subscriptions. For example, source database 10 cannot have the security file specified as both file 15 in one SFILE definition and 20 in another SFILE definition.

- If SFSECURITYFILE=YES is specified for a source database (SFDBID parameter), then any Adabas destination definitions with same database specified as the destination input database (SFDBID setting equals the DAIDBID setting), must also specify identical file numbers for both the DAIFILE (input file) and DATFILE (target file) parameters.

> **Note:** Any security file you identify in an SFILE definition cannot also be used in an initial-state definition. In other words, Event Replicator for Adabas does not support the use of security files in initial-state definitions. If you want to populate a target database with an initial snapshot of the security definitions, read *Initial-State Processing of Security Definitions*, in the *Event Replicator for Adabas Administration and Operations Guide*.

For complete information about replicating security definitions, read *Replicating Security Definitions*, in the *Event Replicator for Adabas Administration and Operations Guide*.

SFSEXIT Parameter

Parameter	Specify . . .	Possible Values	Default
SFSEXIT	the name of a subscription exit	Valid subscription exit name	none

If a subscription exit is specified, it will be called when replication data is processed for one file/ISN for this subscription, database ID, and file number.

For more information about using the subscription user exit with SFREPLICATEDELETE=UPDATE, read *Controlling Delete Transaction Processing (SFREPLICATEDELETE=UPDATE Processing)*, in the *Event Replicator for Adabas Administration and Operations Guide*.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Subscription User Exit` parameter in *Step 2. Specify SFILE Definition Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SIDESTINATION Parameter

Parameter	Specify . . .	Possible Values	Default
SIDESTINATION	a destination name list for the subscription. It specifies where initial-state data is to be sent for the subscription.	Valid destination definition names in single quotes or NONE in single quotes.	The same list as specified for the SDESTINATION parameter

A matching destination name must be defined via parameter DESTINATION (and its subparameters) for each name in the list. If SIDESTINATION is specified, all initial-state data will be sent to each destination. If SIDESTINATION is NOT specified, the default is the same list of destinations specified for parameter SDESTINATION, that is, all data processed for the subscription, including initial-state data, will be sent to each destination. If SIDESTINATION='NONE' is specified, no initial-state process can be performed for the subscription.

At least one destination must be specified (and a maximum of 32 destinations may be specified) in SDESTINATION or SIDESTINATION.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about specifying destinations for the subscription in *Step 3. Specify One or More Destinations for the Subscription*, in the *Adabas Event Replicator Subsystem User's Guide*.

SINCREMENTIS Parameter

Parameter	Specify . . .	Possible Values	Default
SINCREMENTIS	whether the transaction sequence number should be incremented if an initial-state request is occurring.	"YES" or "NO"	NO

If "YES" is specified, the transaction sequence number is incremented if an initial-state request is occurring; if "NO" is specified, the transaction sequence number is not incremented.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the Increment Initial State Count parameter in *Step 2. Supply General Subscription Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SRESENDBUFFER Parameter

Parameter	Specify . . .	Possible Values	Default
SRESENDBUFFER	the name of the resend buffer to be associated with the subscription	Valid resend buffer definition name	none

A RESENDBUFFER **initialization parameter** with a corresponding name must be defined.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Resend Buffer Name` parameter in *Step 2. Supply General Subscription Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

STLFILTER Parameter

Parameter	Specify . . .	Possible Values	Default
`STLFILTER`	the transaction logging level when a record in a transaction is being excluded from replication due to extended subscription processing or the subscription user exit.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event, filter reason, and transaction information), "2" (log event, filter reason, transaction, and file/record information), or "3" (log event, filter reason, transaction information, file/record information, and payload data of available images).

This is the equivalent of specifying the **Filter Level** field in the Adabas Event Replicator Subsystem subscription definition screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Filter Level` parameter in *Step 5. (Optional) Modify the Transaction Logging Values for the Subscription, as Necessary*, in the *Adabas Event Replicator Subsystem User's Guide*.

STLIGNORE Parameter

Parameter	Specify . . .	Possible Values	Default
`STLIGNORE`	the transaction logging level when a filter condition cannot be evaluated.	0, 1, or 2	0

Valid values are "0" (no logging), "1" (log event and filter information), or "2" (log event, filter information, and payload data or available field values).

This is the equivalent of specifying the **Filter Ignored** field in the Adabas Event Replicator Subsystem subscription definition screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Filter Ignored` parameter in *Step 5. (Optional) Modify the Transaction Logging Values for the Subscription, as Necessary*, in the *Adabas Event Replicator Subsystem User's Guide*.

STLINPUT Parameter

Parameter	Specify . . .	Possible Values	Default
STLINPUT	the transaction logging level when a transaction is selected for subscription processing.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and input transaction data), "2" (log event, input transaction, and file/record data), or "3" (log event and all available input transaction data for the event).

This is the equivalent of specifying the **Input Level** field in the Adabas Event Replicator Subsystem subscription definition screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Input Level` parameter in *Step 5. (Optional) Modify the Transaction Logging Values for the Subscription, as Necessary*, in the *Adabas Event Replicator Subsystem User's Guide*.

STLMATCH Parameter

Parameter	Specify . . .	Possible Values	Default
STLMATCH	the transaction logging level when a filter condition is true.	0, 1, or 2	0

Valid values are "0" (no logging), "1" (log event and filter information), or "2" (log event, filter information, and payload data or available field values).

This is the equivalent of specifying the **Filter Matched** field in the Adabas Event Replicator Subsystem subscription definition screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Filter Matched` parameter in *Step 5. (Optional) Modify the Transaction Logging Values for the Subscription, as Necessary*, in the *Adabas Event Replicator Subsystem User's Guide*.

STLNOMATCH Parameter

Parameter	Specify . . .	Possible Values	Default
STLNOMATCH	the transaction logging level when a filter condition is false.	0, 1, or 2	0

Valid values are "0" (no logging), "1" (log event and filter information), or "2" (log event, filter information, and payload data or available field values).

This is the equivalent of specifying the **Filter Not Matched** field in the Adabas Event Replicator Subsystem subscription definition screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Filter Not Matched` parameter in *Step 5. (Optional) Modify the Transaction Logging Values for the Subscription, as Necessary*, in the *Adabas Event Replicator Subsystem User's Guide*.

STLOUTPUT Parameter

Parameter	Specify . . .	Possible Values	Default
STLOUTPUT	the transaction logging level when a transaction is to be output on behalf of a subscription.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and output transaction data), "2" (log event, output transaction, and file/record data), or "3" (log event and all available output information for the event).

This is the equivalent of specifying the **Output Level** field in the Adabas Event Replicator Subsystem subscription definition screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Output Level` parameter in *Step 5. (Optional) Modify the Transaction Logging Values for the Subscription, as Necessary*, in the *Adabas Event Replicator Subsystem User's Guide*.

SVERSION Parameter

Parameter	Specify . . .	Possible Values	Default
SVERSION	a user-specified subscription version indicator.	Up to 2 characters	blanks

The SVERSION parameter has no meaning to the Event Replicator Server. The subscription version indicator will be passed as a part of the header information sent to the output destination. Its use allows the target application to handle changes in the subscription definition.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Subscription Version` parameter in *Step 2. Supply General Subscription Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

SWCODE Parameter

Parameter	Specify . . .	Possible Values	Default
SWCODE	the desired output-wide code. This can only be set when the Event Replicator Server has been started with Universal Encoding Support (UES) enabled.	numeric	none

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `User Data Wide Key` parameter in *Step 2. Supply General Subscription Information*, in the *Adabas Event Replicator Subsystem User's Guide*.

The W (w-char) field key code is documented in the record buffer section of the OP command in the Adabas Command Reference documentation.

Wide-character fields sent to Event Replicator Target Adapter are translated to hexadecimal. However, support for UTF-8 encoding is available for internationalization purposes. To send wide-character fields to Event Replicator Target Adapter using UTF-8 format, make sure the `SWCODE` parameter is set to "4091" in the Event Replicator subscription definition used for Event Replicator Target Adapter processing.

Example

Below is an example using the DESTINATION and SUBSCRIPTION parameters to specify that replicated data from DBID 6, file 2 is replicated to DBID 60, file 20:

```
ADARPD DESTINATION NAME=ADA1
ADARPD   DTYPE=ADABAS
ADARPD   DAIFILE=2,DAIDBID=6,DATDBID=60,DATFILE=20
*
ADARPD SUBSCRIPTION NAME=SUB300
ADARPD  SDESTINATION='ADA1'
ADARPD  SFILE=2
ADARPD  SFDBID=6
ADARPD  SFBAI='AA,BA,BB,BC,BD,CA,CB,CD,CE,CF.'
```

SUBTASKS (Number of Subtasks) Parameter

Parameter	Specify . . .	Possible Values	Default
SUBTASKS	the number of output subtasks in the Event Replicator Server.	integer, 0-15	3

A specific destination will be handled by only one output subtask at a time.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Subtasks` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
SUBTASKS=n
```

SUBTASKWAIT (Subtask Wait Time) Parameter

Parameter	Specify . . .	Possible Values	Default
SUBTASKWAIT	the number of seconds the Event Replicator Server will wait for the successful initialization of an input or output subtask.	1 - 3600	10

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Subtask Activation Wait` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

Syntax:

```
SUBTASKWAIT=n
```

Transaction Log (TLOG) Parameters

Global transaction log settings can be specified using the following parameters:

```
[TLCOMP={0|1 }]
[TLINPUT={0|1|2|3 }]
[TLISTATE={0|1|2|3 }]
[TLISTATECOMP={0|1|2|3 }]
[TLMAX=[n|50 ]
[TLNOSUB={0|1|2|3 }]
[TLQCOMP={0|1|2|3 }]
[TLREQERR={0|1|2|3 }]
```

```
[TLREQRECV={0|1|2 }]
[TLREQREJECT={0|1|2|3 }]
[TLRESTART=[{n|40 ]
[TLRETRANS={0|1|2|3 }]
[TLSTATUS={0|1|2|3 }]
```

Transaction log settings can also be specified for specific destination and subscription definitions. For information on those TLOG settings, read *DESTINATION Settings* and *SUBSCRIPTION Settings*, elsewhere in this documentation.

This section covers the TLOG settings:

- TLCOMP Parameter
- TLINPUT Parameter
- TLISTATE Parameter
- TLISTATECOMP Parameter
- TLMAX Parameter
- TLNOSUB Parameter
- TLQCOMP Parameter
- TLREQERR Parameter
- TLREQRECV Parameter
- TLREQREJECT Parameter
- TLRESTART Parameter
- TLRETRANS Parameter
- TLSTATUS Parameter

TLCOMP Parameter

Parameter	Specify . . .	Possible Values	Default
TLCOMP	the level of transaction logging that should occur when a transaction has been fully processed and Adabas has been informed that the transaction was successfully replicated.	0 or 1	0

Valid values are "0" (no logging) or "1" (log event and input transaction data).

This is the equivalent of specifying the **Completion Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Completion Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLINPUT Parameter

Parameter	Specify . . .	Possible Values	Default
TLINPUT	the level of transaction logging that should occur when a transaction is taken off the input queue and put on the transaction assignment queue.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and input transaction data), "2" (log event, input transaction, and file/ record information), and "3" (log event and all available input information for the event.

This is the equivalent of specifying the **Input Queue Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Input Queue Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLISTATE Parameter

Parameter	Specify . . .	Possible Values	Default
TLISTATE	the level of transaction logging that should occur when a user request for initial-state information for a file has started.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and URBS), "2" (log event and URBI), or "3" (log event, URBS, and URBI).

This is the equivalent of specifying the **I-State Start Request Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `I-State Start Request Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLISTATECOMP Parameter

Parameter	Specify . . .	Possible Values	Default
TLISTATECOMP	the level of transaction logging that should occur when an initial-state information request has completed.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and URBS), "2" (log event and URBI, if available), or "3" (log event, URBS, and URBI, if available).

This is the equivalent of specifying the **I-State Completion Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `I-State Completion Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLMAX Parameter

Parameter	Specify . . .	Possible Values	Default
TLMAX	the maximum percentage of the Event Replicator replication pool that can be used for transaction log (TLOG) processing.	1 - 100	50

The setting for this parameter must always be greater than the setting for the TLRESTART **parameter**.

This is the equivalent of specifying the **Maximum RPL Usage** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Maximum RPL Usage` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLNOSUB Parameter

Parameter	Specify . . .	Possible Values	Default
TLNOSUB	the level of transaction logging that should occur when a transaction is not queued to any subscription in the Event Replicator Server.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and input transaction data), "2" (log event, input transaction, and file/record information), or "3" (log event and all available input information for the event).

This is the equivalent of specifying the **No Match Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `No Match Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLQCOMP Parameter

Parameter	Specify . . .	Possible Values	Default
TLQCOMP	the level of transaction logging that should occur prior to a transaction being assigned to the completion queue.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and input transaction data, "2" (log event, input transaction, and file/record information), or "3" (log event and all input information available for the event).

This is the same as specifying the **Queue Completion Level** global value directly in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Queue Completion Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLREQERR Parameter

Parameter	Specify . . .	Possible Values	Default
TLREQERR	the level of transaction logging that should occur when a user request is rejected due to an error in carrying out the request.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and URBS), "2" (log event and URBI), or "3" (log event, URBS, and URBI).

This is the equivalent of specifying the **Request Error Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Request Error Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLREQRECV Parameter

Parameter	Specify . . .	Possible Values	Default
TLREQRECV	the level of transaction logging that should occur when a user request has been received.	0, 1, or 2	0

Valid values are "0" (no logging), "1" (log event but no data), or "2" (log event and the entire input buffer before and after translation if appropriate).

This is the equivalent of specifying the **Request Received Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Request Received Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLREQREJECT Parameter

Parameter	Specify . . .	Possible Values	Default
`TLREQREJECT`	the level of transaction logging that should occur when a user request is rejected due to an error in interpreting the request.	0, 1, or 2	0

Valid values are "0" (no logging), "1" (log event but no data), or "2" (log event, error code, and entire input buffer).

This is the equivalent of specifying the **Request Rejected Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Request Rejected Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLRESTART Parameter

Parameter	Specify . . .	Possible Values	Default
`TLRESTART`	the amount of available Event Replicator replication pool storage that must be available before transaction logging (TLOG logging) can restart.	0 - 99	40

The setting for this parameter must always be less than the setting for the `TLMAX` **parameter**.

This is the equivalent of specifying the **Restart RPL Usage Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Restart RPL Usage` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLRETRANS Parameter

Parameter	Specify . . .	Possible Values	Default
`TLRETRANS`	the level of transaction logging that should occur when a user request to retransmit a specific transaction has been processed.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and URBS), "2" (log event and URBI), or "3" (log event, URBS, and URBI).

This is the equivalent of specifying the **Retransmit Request Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Retransmit Request Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

TLSTATUS Parameter

Parameter	Specify . . .	Possible Values	Default
TLSTATUS	the level of transaction logging that should occur when a user request for status on an Event Replicator Server resource has been processed.	0, 1, 2, or 3	0

Valid values are "0" (no logging), "1" (log event and URBS), "2" (log event and URBI), or "3" (log event, URBS, and URBI).

This is the equivalent of specifying the **Status Request Level** global value in the Adabas Event Replicator Subsystem screens.

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `Status Request Level` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

VERIFYMODE Parameter

Parameter	Specify . . .	Possible Values	Default
VERIFYMODE	whether the Event Replicator for Adabas should run in verify (test) mode or not.	YES | NO	NO

This parameter can be maintained online in Adabas Event Replicator Subsystem (SYSRPTR). For more information, read about the `VERIFYMODE` parameter in *Setting Global Values*, in the *Adabas Event Replicator Subsystem User's Guide*.

When `VERIFYMODE=YES` is specified during Event Replicator Server startup, Event Replicator for Adabas assumes that it is in test mode and performs the following processing:

- 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 messages 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`.

Syntax:

```
VERIFYMODE=[YES | NO]
```

18 C5 Command: Write User Data to Protection Log

The C5 command writes user data on SIBA/PLOG. If you have Event Replicator for Adabas installed, however, you can also use the C5 command to send messages from the originating application to one or more Event Replicator Server destinations.

Function and Use

The C5 command is used to write user data to the Adabas data protection log. This data may be read and displayed using the ADASEL utility. The data that is written has no effect on Adabas recovery processing. The ADASAV and ADARES utilities ignore all data written to the data protection log as a result of a C5 command.

If you have Event Replicator for Adabas installed, you can also use the C5 command to send a message from your application to one or more Event Replicator Server destinations. This C5 command *must* specify a file number (to identify the file to which the C5 command applies) and its Command Option 1 control block field *must* be set to "R". When a C5 command with the R-option is specified, Adabas sends the data to the Event Replicator Server as a user-defined event, in addition to writing the user data to the PLOG (where it can be read and displayed using the ADASEL utility). Adabas and the Event Replicator Server do not alter or convert the user data sent with a C5 command with the R option; therefore, you must ensure that the user data is sent in a format that the target application receiving the data understands.

If you specify the C5 command with the R option, you may optionally specify a subscription name in the Addition 1 control block field, however the subscription must be previously defined to the Event Replicator Server. The Event Replicator Server destinations to which the message is passed are determined from the subscription, if one is specified, or, if no subscription is specified, from all destinations of subscriptions for the specified file. If no destination can be determined (either because the specified subscription does not exist or because no subscriptions for the specified file exist), the message is not processed and will be discarded. The only auditing of discarded messages could be in the Replication transaction logging (TLOG) file.

Command: C5

User Control Block

Field	Position	Format	Before Adabas Call	After Adabas Call
	1-2	--	--	--
COMMAND CODE	3-4	alphanumeric	F	U
	5-8	--	--	--
FILE NUMBER**	9-10	binary	F	U
RESPONSE CODE	11-12	binary	--	A

Field	Position	Format	Before Adabas Call	After Adabas Call
	13-26	--	--	--
RECORD BUFFER LENGTH	27-28	binary	F	U
	29-34	--	--	--
COMMAND OPTION 1***	35	alphanumeric, only value of "R" allowed	F	U
	36	--	--	--
ADDITIONS 1***	37-44	alphanumeric	F	U
	45-72	--	--	--
COMMAND TIME	73-76	binary	--	A
USER AREA	77-80	--	--	U

User Buffer Areas

Buffer	Before Adabas Call	After Adabas Call
FORMAT BUFFER	*	
RECORD BUFFER	F	U

where:

F   Filled in by user before Adabas Call

A   Filled in by Adabas

U   Unchanged after Adabas call

*   Not used but must be included in parameter list of call statement

**   A database ID is only necessary if you are accessing multiple databases and need to limit the command operation to a specific database.

However, if you are using Event Replicator for Adabas and the Command Option 1 field is set to "R", a file number must be specified to identify the file to which the C5 command applies.

*** Only used if you are using Event Replicator for Adabas. For more information, read your Event Replicator for Adabas documentation.

--   Not used

Control Block

Command Code

C5

File Number

When "R" is specified in the Command Options 1 field, this field is required to identify the file to which the C5 command applies. The file number may be one or two bytes. If the specified file is not a replicated file, appropriate response codes are returned.

Adabas sends the user data and optional subscription name to the Event Replicator Server associated with the file for further processing.

If "R" is not specified in the Command Options 1 field, this field should not be specified.

Response Code

Adabas returns the response code for the command in this field. Response code 0 indicates that the command was executed successfully. If the C5 command returns a non-zero response code, the rightmost two bytes of the Adabas control block, bytes 45-48 (additions 2 field) may contain a subcode defining the exact response code meaning. Response codes and their subcodes are defined in the Adabas messages and codes documentation.

Record Buffer Length

The number of bytes specified in this field will be written to the Adabas data protection log.

The maximum length which may be specified is 2048 bytes.

Command Options 1

When the Event Replicator for Adabas is installed, a value of "R" is allowed in this field to identify that the command is intended for the Event Replicator Server.

If the Event Replicator for Adabas is not installed, this field must be blank.

Additions 1

When "R" is specified in the Command Options 1 field, an optional subscription name can be specified in this field. The subscription must have been previously defined to the Event Replicator Server.

If "R" is not specified in the Command Options 1 field, this field should not be specified.

Record Buffer

The information to be written to the data protection log is provided in this buffer. If option "R" has been specified in the Command Options 1 field, Adabas also sends the information to the Event Replicator Server associated with the specified file.

The written information may be selected subsequently using the ADASEL utility by specifying the beginning characters (1 to 30 characters) as originally contained in the record buffer. It is, therefore, recommended that the user provide a unique identification of his data in the beginning positions of the record buffer. This will ensure that the user data can be properly identified and selected.

Examples

This section provides the following examples:

- Non-Replication Example
- Replication Example

Non-Replication Example

In this example, the information "ULRR0422 UPDATES FOR JANUARY" is written to the Adabas data protection log.

Control Block

Command Code	C5
Record Buffer Length	28

Buffer Areas

Record Buffer	ULRR0422 UPDATES FOR JANUARY

Replication Example

In this example, the information "REPLICATION DATA RECEIVED" is sent to the Event Replicator Server destinations identified by subscription SUBSCR01 for Adabas file 12.

Control Block

Command Code	C5
File Number	12
Record Buffer Length	25
Command Options 1	R
Additions 1	SUBSCR01

Buffer Areas

Record Buffer	REPLICATION DATA RECEIVED

Index

Symbols