

Maintaining Destination Definitions


A destination definition defines the destination of the replicated data. Destination definitions can be created for Adabas, webMethods EntireX, WebSphere MQ, File, and Null destinations. At least one definition is required for every Event Replicator for Adabas destination you intend to use.

Destination Type	Description
Adabas	Data is replicated to one or more Adabas files.
webMethods EntireX	Replicated data is written to an output queue via webMethods EntireX.
WebSphere MQ	Replicated data is written to an output queue via IBM WebSphere MQ.
Null	Data replication is tested without actually sending the data to a destination.
File	Replicated data is written to the CLOG, using TLOG URBLTDOD records.

This chapter covers the following topics:

- Listing Destination Definitions
- Creating an Adabas Destination Definition
- Creating a webMethods EntireX Destination Definition
- Creating a WebSphere MQ Destination Definition
- Creating a File Destination Definition
- Creating a Null Destination Definition
- Modifying Destination Definitions
- Copying Destination Definitions
- Activating and Deactivating Destination Definitions
- Opening and Closing Destinations
- Deleting Destination Definitions

Listing Destination Definitions

 To use Event Replicator Administration to list the destination definitions stored in the Replicator system file:

1. Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*.
2. Click and expand **Replication Definitions** in tree-view under the selected database.
3. Click on **Destinations** in the tree-view under **Replication Definitions**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

Creating an Adabas Destination Definition

Using Adabas destination definitions, data can be replicated to one or more Adabas files. This section describes how to create an Adabas destination definition using Event Replicator Administration.

Note:

An Adabas destination can be referenced by no more than one subscription.

To create an Adabas destination definition in Event Replicator Administration, complete the following steps:

- Step 1. Access the Adabas Destination Definition Creation Area
- Step 2. Specify General and TLOG Adabas Destination Parameters
- Step 3. Specify Input and Target Adabas Destination Databases and Files
- Step 4. (Optional) Specify File-Related Parameters for the Adabas Destination
- Step 5. Save the Adabas Destination Definition

Step 1. Access the Adabas Destination Definition Creation Area

 **To access the Adabas destination definition creation area of Event Replicator Administration:**

1. List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*.

The destination definitions are listed in detail-view.

2. Right-click on **Destinations** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3. Click on **Create New Adabas Destination** in the drop-down menu.

A blank **Adabas Destination** panel appears in detail-view.

Step 2. Specify General and TLOG Adabas Destination Parameters

 **To use Event Replicator Administration to supply general and TLOG specifications for an Adabas destination definition, complete the following steps:**

1. Supply general parameter specifications for the Adabas destination. The fields for these general parameters can be found at the top of the Adabas Destination panel:

General Values	
Name	Value
Destination Name	<input type="text"/> *
Destination Active	Yes <input type="button" value="v"/>
Allow Logging	No <input type="button" value="v"/>
Replicate Utility Changes	No <input type="button" value="v"/>
Open Retry Count	0 *
Open Retry Interval	10 *
Error Action	A - ALTACTION <input type="button" value="v"/>
Opened at Start	Global <input type="button" value="v"/>

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Destination Name (DESTINATION NAME)	The unique name for the Adabas destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No

Parameter Name	Specify	Default
Replicate Utility Changes (DREPLICATEUTI)	<p>Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup or whether it should be activated for a specific target file.</p> <p>This parameter appears twice on this screen. In the general location, it specifies whether the utility change replication should be activated for a destination at Event Replicator Server startup; in the location (next to the Target File parameter), it specifies whether the utility change replication should be activated for the associated target file.</p> <p>Valid values are "Yes" and "No". If "Yes" is specified, utility replication is activated; if "No" is specified, utility replication is not activated.</p> <p>For more information about replicating utility functions, read <i>Replicating Utility Functions</i>, in <i>Event Replicator for Adabas Concepts</i>.</p>	No
Open Retry Count (DRETRYCOUNT)	<p>The number of times that an attempt to open the destination will be retried at the interval specified by the Retry Interval parameter. This is the equivalent of specifying the DRETRYCOUNT parameter directly in the Event Replicator Server startup job.</p> <p>Valid values range from 0 through 2,147,483,647 or the literal "GLOBAL".</p> <p>If the value "GLOBAL" is specified for this parameter, the specification for the Retry Count global variable will be used. Any retry attempts will occur at the interval specified by the Retry Interval parameter. A value of zero indicates that no retry attempt to open this destination should occur.</p>	The value of the Retry Count global variable.
Open Retry Interval (DRETRYINTERVAL)	<p>The default number of seconds between retry attempts to open the destination. This is the equivalent of specifying the DRETRYINTERVAL parameter directly in the Event Replicator Server startup job.</p> <p>Valid values are 0, 5 through 2,147,483,647, or the literal "GLOBAL".</p> <p>If the value "GLOBAL" is specified for this parameter, the specification for the Retry Interval global variable 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.</p>	The value of the Retry Interval global variable.

Parameter Name	Specify	Default
<p>Error Action (DAERROR)</p>	<p>The action to be taken when an error occurs during replication to an Adabas destination. This is the equivalent of specifying the DAERROR parameter directly in the Event Replicator Server startup job. Valid values are ALTACTION, BACKOUT, or CLOSE.</p> <p>In all cases (ALTACTION, BACKOUT, and CLOSE), if response 148 is returned and the SLOG system file is available, the destination is closed.</p> <p>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 is set to ISN, the text "ISN" will be appended to this message as well as the ISN value. If the DATMETHOD is set to KEY, the record key will be written in both hexadecimal and readable format in a separate message.</p> <p>Additional actions are taken, based on the value of this parameter. These actions are:</p> <ul style="list-style-type: none"> ● A - ALTACTION: Processing continues with the next update that is part of the same transaction. Some special processing occurs when DAERROR=ALTACTION: <ul style="list-style-type: none"> ○ 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. <p>For other errors, the record is skipped.</p> <ul style="list-style-type: none"> ● B - 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. ● C - 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. 	<p>A- ALTACTION</p>

Parameter Name	Specify	Default
Opened at Start (DOPEN parameter)	<p>Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default.</p> <p>When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is <i>not</i> opened at Event Replicator Server startup.</p> <p>When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the Global OPEN Value (GOPEN) global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened.</p> <p>This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.</p>	Global


- Supply TLOG parameter specifications for the Adabas destination. The fields for these TLOG parameters can be found at the bottom of the Adabas Destination panel:

TLOG Values	
Name	Value
Assign Level	0 - No Logging
Completion Level	0 - No Logging
SLOG Write Level	0 - No Logging
SLOG Read Level	0 - No Logging
Adabas Level	0 - No transaction logging should occur

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Assign Level (DTLASSIGN)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (DTLCOMP)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (DTLSLOGWRITE)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Read Level (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. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Adabas Level (DTLADABAS)	The level of transaction logging that should occur when a transaction for an Adabas destination incurred an error. This is the equivalent of specifying the DTLADABAS parameter in the Event Replicator Server startup job. Valid values range from 0 through 3, as described below: <ul style="list-style-type: none"> ● 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. 	0

Step 3. Specify Input and Target Adabas Destination Databases and Files

 To use Event Replicator Administration to supply input-target file pairings for an Adabas destination definition, complete the following steps:

- 1. The fields for input-target file pairing specifications in an Adabas destination definition can be found in the middle of the Adabas Destination panel:

All	Input DBID	Input File	Target DBID	Target File					
<input type="checkbox"/>	<input type="text" value="0"/>	*	<input type="text" value="0"/>	*	<input type="text" value="0"/>	*	<input type="text" value="0"/>	*	<input checked="" type="checkbox"/>

For the first input-target destination file pairing, fill in the fields in Value columns as described in the following table:

Parameter Name	Specify	Default
Input DBID (DAIDBID)	<p>The database ID associated with an input file (see the Input File field) for this Adabas destination. The database ID is numeric and can range from one to 65535.</p> <p>The input database ID and file specified must also be included in an SFILE definition in the subscription.</p> <p>Multiple input database files can be specified in a single destination definition, as needed:</p> <ul style="list-style-type: none"> ● To add an additional input database file to the definition, click New. A blank row in the table of input database files appears. Fill in the blank row. ● To delete an input database ID from the definition, click on the check box in the corresponding All column and then click Delete. To delete all database files from the list, click the All check box in the column heading and then click Delete. The input database file is removed from the list of input database files. <p>Only unique combinations of Input DBID and Input File parameters can be specified in 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.</p>	---

Parameter Name	Specify	Default
Input File (DAIFILE)	<p>The number of the input file for this Adabas destination. The input database ID and file specified must also be included in an SFILE definition in the subscription. At least one file must be listed for an Adabas destination definition.</p> <p>Multiple input database files can be specified in a single destination definition, as needed:</p> <ul style="list-style-type: none"> ● To add an additional input database file to the definition, click New. A blank row in the table of input database files appears. Fill in the blank row. ● To delete an input database ID from the definition, click on the check box in the corresponding All column and then click Delete. To delete all database files from the list, click the All check box in the column heading and then click Delete. The input database file is removed from the list of input database files. <p>Only unique combinations of Input DBID and Input File parameters can be specified in 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.</p>	---

Parameter Name	Specify	Default
Target DBID (DATDBID)	<p>The database ID associated with the target file for the replicated data. The database ID is numeric and can range from one to 65535.</p> <p>Multiple target database IDs and files can be specified in a single destination definition, as needed.</p> <ul style="list-style-type: none"> ● To add an additional target database file to the definition, click New. A blank row in the table of target database files appears. Fill in the blank row. ● To delete a target database ID from the definition, click on the check box in the corresponding All column and then click Delete. To delete all database files from the list, click the All check box in the column heading and then click Delete. The target database file is removed from the list of target database files. 	---
Target File (DATFILE)	<p>The number of the target (output) file for the replicated data associated with the input file in this destination definition. There is no default.</p> <p>Multiple target database IDs and files can be specified in a single destination definition, as needed.</p> <ul style="list-style-type: none"> ● To add an additional target database file to the definition, click New. A blank row in the table of target database files appears. Fill in the blank row. ● To delete a target database ID from the definition, click on the check box in the corresponding All column and then click Delete. To delete all database files from the list, click the All check box in the column heading and then click Delete. The target database file is removed from the list of target database files. 	---


Parameter Name	Specify	Default
<input checked="" type="checkbox"/>	This button allows you to supply additional parameter specifications for each input and target DBID/file pair listed in this Adabas destination definition. The description of these additional parameters is provided in the later steps of this procedure.	

2. If you want to add additional input-target destination file pairings, click the **New** button to add new lines to the list of input-target file pairings. Fill in the fields for each line as described in the previous step.
3. If you want to delete an input-target file pairing from the list, click in the check box in the **All** column corresponding to the line for the pairing you want to delete and click the **Delete** button.

The file pairing is deleted from the list.

Step 4. (Optional) Specify File-Related Parameters for the Adabas Destination

 **To use Event Replicator Administration to specify file-related parameters for the Adabas destination, complete the following steps:**

1. On the Adabas Destination panel, click on the  button to the right of the input-target file pairing for which you want to specify file-related parameter settings.

A new panel appears allowing you to specify additional parameters for the input/target DBID/file pair. The fields for file-related parameter settings appears at the bottom of this panel:

File-Related Parameters	
Name	Value
Replicate Utility	No
Replication Method	ISN
After Image Offset	
After Image Key Length	
Before Image Offset	
Before Image Key Length	
Key Offset	
Key Length	
Search Buffer	

2. Fill in the fields in Value columns for the file-related parameters, as described in the following table:

Parameter Name	Description	Default
Replicate Utility (DAREPLICATEUTI)	<p>This parameter can be specified regardless of the Replication Method selected.</p> <p>Specify whether Adabas utility change replication should be activated for a specific target file at Event Replicator Server startup. Valid values are "Yes" and "No".</p> <p>If "Yes" is specified, utility replication is activated for the target file at Event Replicator Server startup; if "No" is specified, utility replication is not activated for the target file.</p> <p>For more information about replicating utility functions, read <i>Replicating Utility Functions</i>, in <i>Event Replicator for Adabas Concepts</i>.</p>	No
Replication Method (DATMETHOD)	<p>Specify the method to be used when searching for a record on the target database. Valid values are "ISN" and "KEY".</p> <p>When this parameter is set to "KEY", the parameters Search Buffer, After Image Offset, After Image Key Length, Before Image Offset, Before Image Key Length, Key Offset, and Key Length may also be supplied to indicate where to find the key in the replicated data. All of these parameters are optional except Search Buffer and After Image Length, which are required.</p>	ISN
After Image Offset (DATKEYAIO)	<p>This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified.</p> <p>Specify the offset of the key to be used in the after image of the data buffer. Valid values range from 0 through 2,147,483,646 bytes. The sum of this parameter value and the length of the key to be used in the after image (After Image Key Length parameter), must be less than or equal to 2,147,483,647 bytes.</p>	0
After Image Key Length (DATKEYAIL)	<p>When the Replication Method parameter is set to "KEY", this parameter is required. If the Replication Method parameter is set to ISN, this parameter may not be specified.</p> <p>Specify the length (in bytes) of the key to be used in the after image of the data buffer. Valid values range from 1 through 32,767 bytes.</p>	none

Parameter Name	Description	Default
<p>Before Image Offset (DATKEYBIO)</p>	<p>This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified.</p> <p>Specify the offset of the key to be used in the before image of the data buffer. Valid values range from 0 through 2,147,483,646 bytes. The sum of this parameter value and the length of the key to be used in the before image (Before Image Key Length parameter), must be less than or equal to 2,147,483,647 bytes.</p> <p>If this parameter is specified, the Before Image Key Length parameter must also be specified with a value greater than zero.</p>	<p>0</p>
<p>Before Image Key Length (DATKEYBIL)</p>	<p>This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified. If the Before Image Offset parameter is specified, this parameter must also be specified with a value greater than zero.</p> <p>Specify the length (in bytes) of the key to be used in the before image of the data buffer. Valid values range from 1 through 32767 bytes.</p>	<p>none</p>
<p>Key Offset (DATKEYKYO)</p>	<p>This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified.</p> <p>Specify the offset of the key to be used in the before image of the primary key. Valid values range from 0 through 32767 bytes. The sum of this parameter value and the length of the key to be used in the before image of the primary key (Key Length parameter), must be less than or equal to 32767 bytes.</p> <p>If this parameter is specified, the Key Length parameter must also be specified with a value greater than zero.</p>	<p>0</p>

Parameter Name	Description	Default
Key Length (DATKEYKYL)	<p>This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified. If the Key Offset parameter is specified, this parameter must also be specified with a value greater than zero.</p> <p>Specify the length (in bytes) of the key to be used in the before image of the primary key. Valid values range from 1 through 32767 bytes.</p>	none
Search Buffer (DATKEYSB)	<p>When the Replication Method parameter is set to "KEY", this parameter is required. If the Replication Method parameter is set to ISN, this parameter may not be specified.</p> <p>Specify a search buffer to be used for keyed replication. Up to 60 alphanumeric characters can be specified.</p>	none

- When you are satisfied with the file-related parameters pairings listed on the panel, click the **OK** button to return to the first panel of Adabas destination definitions.

Step 5. Save the Adabas Destination Definition

To use Event Replicator Administration to save an Adabas destination definition:

- When all specifications have been made to your satisfaction, click **OK** on the Adabas destination panel.

The Adabas destination definition is saved in the Replicator system file.

Creating a webMethods EntireX Destination Definition

Using a webMethods EntireX destination definition, replicated data is written to an output queue via webMethods EntireX. Prior to using webMethods EntireX as the messaging subsystem, be sure to read *Using webMethods EntireX as the Messaging System*, in *Event Replicator for Adabas Administration and Operations Guide* provided with your Event Replicator Administration documentation.

To create a webMethods EntireX destination definition in Event Replicator Administration, complete the following steps:

- Step 1. Access the webMethods EntireX Destination Definition Creation Area
- Step 2. Specify General and TLOG webMethods EntireX Destination Parameters
- Step 3. (Optional) Specify Destination Class Information, If Applicable

- Step 4. Save the webMethods EntireX Destination Definition

Step 1. Access the webMethods EntireX Destination Definition Creation Area

▶ To access the webMethods EntireX destination definition creation area of Event Replicator Administration:

1. List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*.

The destination definitions are listed in detail-view.

2. Right-click on **Destinations** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3. Click on **Create New EntireX Communicator Destination** in the drop-down menu.

A blank **EntireX Communicator Destination** panel appears in detail-view.

Step 2. Specify General and TLOG webMethods EntireX Destination Parameters

▶ To use Event Replicator Administration to supply general and TLOG specifications for a webMethods EntireX destination definition, complete the following steps:

1. Supply general parameter specifications for the webMethods EntireX destination. The fields for these general parameters can be found at the top of the EntireX Communicator Destination panel:

General Values	
Name	Value
Destination Name	<input type="text"/> *
Architecture	2 - High Order Byte First, EBCDIC Encoding Family ▾
Commit Threshold	5 *
EntireX Broker ID	<input type="text"/> *
Broker Service	<input type="text"/> *
Service Class	<input type="text"/> *
Service Name	<input type="text"/> *
Destination Active	Yes ▾
Allow Logging	No ▾
Event Logging	No ▾
Destination Class	<input type="text"/>
Destination Class Parameter	<input type="text"/>
Replicate Utility Changes	No ▾
Maximum Output Size	0
Open Retry Count	0 *
Open Retry Interval	10 *
Opened at Start	Global ▾

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Destination Name (DESTINATION NAME)	The unique name for the webMethods EntireX destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---
Architecture (DARC)	The data architecture for fields in the URB* control structures sent to the webMethods EntireX destination. For complete information on calculating a value for this parameter, read about the DARC parameter in <i>Event Replicator for Adabas Reference Guide</i> provided with your Event Replicator Administration documentation.	2 (High-order byte first, EBCDIC encoding)
Commit Threshold (DCOMMITTHRESHOLD)	The number of messages that will be sent to the webMethods EntireX destination before a commit is performed for those messages.	5

Parameter Name	Specify	Default
EntireX Broker ID (DETBBROKERID)	<p>The webMethods EntireX Broker ID for which this destination definition applies. The name can be up to 32 characters long.</p> <p>Broker IDs come in two formats: one for TCP/IP communications and one for Adabas SVC communications. For TCP/IP communications, the format is:</p> <div data-bbox="552 651 1043 743" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <code>addr:port-number:TCP</code> </div> <p>In this case, the <i>addr</i> setting is either the TCP/IP address or the host name. The <i>port-number</i> setting should match the webMethods EntireX PORT parameter.</p> <p>For Adabas SVC communications, the format is:</p> <div data-bbox="552 1050 1043 1142" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <code>'broker-id:SVCnnn:NET'</code> </div> <p>In this case, the <i>broker-id</i> setting should match the webMethods EntireX BROKER-ID parameter in the Broker ETBFILE DD. The <i>nnn</i> setting should match either the webMethods EntireX ADASVC or ADA5SVC parameters in the Broker PARMS DD statement.</p> <p>If no name is specified, the default webMethods EntireX Broker ID specified by the ETBBROKERID parameter is used.</p>	---
Broker Service (DETBSERVICE)	The webMethods EntireX Broker service for which this destination definition applies. The service identification can be up to 32 characters long.	---
Service Class (DETBSERVICECLASS)	The webMethods EntireX Broker service class name for which this destination definition applies. The name can be up to 32 characters long.	---
Service Name (DETBSERVICENAME)	The webMethods EntireX service name for which the webMethods EntireX destination definition applies. The name can be up to 32 characters long.	---

Parameter Name	Specify	Default
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No
Event Logging (DEVENTLOG)	<p>Whether or not events should be logged by the Event Replicator Server and sent to this destination. This is the equivalent of specifying the <i>DEVENTLOG</i> directly in the Event Replicator Server startup job. Valid values are "Yes" or "No". When this optional parameter is set to "Yes", Event Replicator Server events are logged to the destination. When this parameter is set to "No" (the default), they are not.</p> <p>Event Replicator Server events are logged in URBS elements. In releases prior to 3.2.1, these URBS elements were sent to destinations related to the event itself. Starting with release 3.2.1, the URBS elements are also sent to any other destinations you have defined "Event Logging =Yes". If a related destination also is defined with "Event Logging =Yes", it will only receive one instance of the URBS element.</p> <p>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.</p>	No
Destination Class (DCLASS)	For the moment, leave this field blank. It is described later in this section.	---
Destination Class Parameter (DCLASSPARM)	For the moment, leave this field blank. It is described later in this section.	---

Parameter Name	Specify	Default
Replicate Utility Changes (DREPLICATEUTI)	<p>Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup. Valid values are "Yes" and "No".</p> <p>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.</p> <p>For more information about replicating utility functions, read <i>Replicating Utility Functions</i>, in <i>Event Replicator for Adabas Concepts</i>.</p>	No
Maximum Output Size (DMAXOUTPUTSIZE parameter)	<p>The maximum output size (in bytes) for the destination. This is the equivalent of specify the DMAXOUTPUTSIZE parameter directly in the Event Replicator Server startup job. Valid values are 0 or any integer ranging from 4096 through 2,147,483,647. 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.</p> <p>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.</p> <p>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.</p>	0

Parameter Name	Specify	Default
Open Retry Count (DRETRYCOUNT parameter)	<p>The number of times that an attempt to open the destination will be retried at the interval specified by the Retry Interval parameter. This is the equivalent of specifying the DRETRYCOUNT parameter directly in the Event Replicator Server startup job.</p> <p>Valid values range from 0 through 2,147,483,647 or the literal "GLOBAL".</p> <p>If the value "GLOBAL" is specified for this parameter, the specification for the Retry Count global variable will be used. Any retry attempts will occur at the interval specified by the Retry Interval parameter. A value of zero indicates that no retry attempt to open this destination should occur.</p>	The value of the Retry Count global variable
Open Retry Interval (DRETRYINTERVAL parameter)	<p>The default number of seconds between retry attempts to open the destination. This is the equivalent of specifying the DRETRYINTERVAL parameter directly in the Event Replicator Server startup job.</p> <p>Valid values are 0, 5 through 2,147,483,647, or the literal "GLOBAL".</p> <p>If the value "GLOBAL" is specified for this parameter, the specification for the Retry Interval global variable 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.</p>	The value of the Retry Interval global variable.

Parameter Name	Specify	Default
Opened at Start (DOPEN parameter)	<p>Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default.</p> <p>When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is <i>not</i> opened at Event Replicator Server startup.</p> <p>When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the Global OPEN Value (GOPEN) global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened.</p> <p>This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.</p>	Global

- Supply TLOG parameter specifications for the webMethods EntireX destination. The fields for these TLOG parameters can be found at the bottom of the EntireX Communicator Destination panel:

TLOG Values	
Name	Value
Assign Level	0 - No Logging
Completion Level	0 - No Logging
SLOG Write Level	0 - No Logging
SLOG Read Level	0 - No Logging

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Assign Level (DTLASSIGN)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (DTLCOMP)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (DTLSLOGWRITE)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Read Level (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. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging

Step 3. (Optional) Specify Destination Class Information, If Applicable

The Destination Class and Destination Class Parameter fields can be used to invoke and pass parameters to the Event Replicator Target Adapter for this destination. The fields are located in the middle of the general fields for the webMethods EntireX destination. Do not use these fields unless you want to invoke and pass parameters to the Event Replicator Target Adapter for the destination or unless otherwise requested by a Software AG support representative.

If applicable, use the Destination Class (DCLASS) field to specify the destination class for this destination definition. Valid values are "NONE" (interpreted as a blank) or "SAGTARG" (if Event Replicator Target Adapter processing should be invoked). There is no default.

If you specify a value for the Destination Class field, you can optionally use the Destination Class Parameter (DCLASSPARM) field to specify up to 120 bytes of character data to be passed to the optional destination output user exit.

If DCLASS=SAGTARG is specified (if the Destination Class field is set to "SAGTARG") to invoke the Event Replicator Target Adapter for a destination, you may want to specify one or more of the following keyword parameters:

Note:

These parameter keywords must be specified in uppercase.

NOSPRE

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.

OPTIONS

The OPTIONS keyword parameter can be used to specify options for the destination. Specify one or more of the options described in the following table, using the syntax `OPTIONS=val1, val2, . . .`

Option Value	Description
1	This option is no longer supported. If specified, it will be ignored.
2	Specify <code>OPTIONS=2</code> 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 <code>OPTIONS=4</code> to ensure that invalid XML characters found in alphanumeric fields are not translated to spaces.
8	Specify <code>OPTIONS=8</code> to ensure that trailing blanks in alphanumeric fields are not removed.
16	Specify <code>OPTIONS=16</code> to ensure that characters used by XML are not replaced automatically with predefined entity references. For example, if <code>OPTIONS=16</code> is set, the ampersand (&) character would not be replaced with the literal "&";
32	Specify <code>OPTIONS=32</code> 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.

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
```

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.

Step 4. Save the webMethods EntireX Destination Definition

 **To use Event Replicator Administration to save a webMethods EntireX destination definition:**

1. When all specifications have been made to your satisfaction, click **OK** on the webMethods EntireX destination panel.

The webMethods EntireX destination definition is saved in the Replicator system file.

Creating a WebSphere MQ Destination Definition

Using a WebSphere MQ destination definition, replicated data is written to an output queue via IBM WebSphere MQ. Prior to using WebSphere MQ as the messaging subsystem, be sure to read *Using WebSphere MQ as the Messaging System*, in *Event Replicator for Adabas Administration and Operations Guide* provided with your Event Replicator Administration documentation.

To create a WebSphere MQ destination definition in Event Replicator Administration, complete the following steps:

- Step 1. Access the WebSphere MQ Destination Definition Creation Area
- Step 2. Specify General and TLOG WebSphere MQ Destination Parameters
- Step 3. (Optional) Specify Destination Class Information, If Applicable
- Step 4. Save the WebSphere MQ Destination Definition

Step 1. Access the WebSphere MQ Destination Definition Creation Area

 To access the WebSphere MQ destination definition creation area of Event Replicator Administration:

1. List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*.

The destination definitions are listed in detail-view.


2. Right-click on **Destinations** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3. Click on **Create New MQSeries Destination** in the drop-down menu.

A blank **MQSeries Destination** panel appears in detail-view.

Step 2. Specify General and TLOG WebSphere MQ Destination Parameters

 To use Event Replicator Administration to supply general and TLOG specifications for a WebSphere MQ destination definition, complete the following steps:

1. Supply general parameter specifications for the WebSphere MQ destination. The fields for these general parameters can be found at the top of the MQSeries Destination panel:

General Values	
Name	Value
Destination Name	<input type="text"/> *
Architecture	2 - High Order Byte First, EBCDIC Encoding Family ▾
Commit Threshold	5 *
Queue Manager Name	<input type="text"/> *
Queue Name	<input type="text"/> *
Dynamic Queue Name	<input type="text"/>
Destination Active	Yes ▾
Allow Logging	No ▾
Event Logging	No ▾
Destination Class	<input type="text"/>
Destination Class Parameter	<input type="text"/>
Coded Character Set ID	0
Optional MQ Format Name	<input type="text"/>
Replicate Utility Changes	No ▾
Maximum Output Size	0
Open Retry Count	0 *
Open Retry Interval	10 *
Opened at Start	Global ▾

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Destination Name (NAME)	The unique name for the WebSphere MQ destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---
Architecture (DARC)	The data architecture for fields in the URB* control structures sent to the WebSphere MQ destination. For complete information on calculating a value for this parameter, read <i>DARC</i> , in <i>Event Replicator for Adabas Reference Guide</i> provided with your Event Replicator Administration documentation.	2 (High-order byte first, EBCDIC encoding)

Parameter Name	Specify	Default
Commit Threshold (DCOMMITTHRESHOLD)	The number of messages that will be sent to the WebSphere MQ destination before a commit is performed for those messages. 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.	5
Queue Manager Name (DMQQMGRNAME)	The WebSphere MQ queue manager name. The name can be up to 48 characters long.	---
Queue Name (DMQQNAME)	The WebSphere MQ queue name. The name can be up to 48 characters long. There is no default.	---
Dynamic Queue Name (DMQDYNQNAME)	The WebSphere MQ dynamic queue name. The name can be up to 48 characters long.	blanks
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No

Parameter Name	Specify	Default
Event Logging (<i>DEVENTLOG</i>)	<p>Whether or not events should be logged by the Event Replicator Server and sent to this destination. This is the equivalent of specifying the <i>DEVENTLOG</i> directly in the Event Replicator Server startup job. Valid values are "Yes" or "No". When this optional parameter is set to "Yes", Event Replicator Server events are logged to the destination. When this parameter is set to "No" (the default), they are not.</p> <p>Event Replicator Server events are logged in URBS elements. In releases prior to 3.2.1, these URBS elements were sent to destinations related to the event itself. Starting with release 3.2.1, the URBS elements are also sent to any other destinations you have defined "Event Logging =Yes". If a related destination also is defined with "Event Logging =Yes", it will only receive one instance of the URBS element.</p> <p>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.</p>	No
Destination Class (<i>DCLASS</i>)	For the moment, leave this field blank. It is described later in this section.	---
Destination Class Parameter (<i>DCLASSP</i>)	For the moment, leave this field blank. It is described later in this section.	---

Parameter Name	Specify	Default
Coded Character Set ID (DMQCCSID)	<p>The destination-specific coded character set ID (CCSID) for the WebSphere MQ destination. This is the equivalent of specifying the DMQCCSID parameter in the Event Replicator Server startup job. Valid values range from 0 through 2,147,483,647.</p> <p>This optional parameter can only be specified when the DCLASS or DEXIT parameters are specified.</p> <p>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.</p>	0
Optional MQ Format Name (DMQFORMAT)	<p>The optional MQ format name. The format name can be up to eight characters long.</p> <p>Note: You cannot specify a value for this parameter if a value has not also been specified for the Destination Class parameter.</p>	---
Replicate Utility Changes (DREPLICATEUTI)	<p>Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup. Valid values are "Yes" and "No".</p> <p>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.</p> <p>For more information about replicating utility functions, read <i>Replicating Utility Functions</i>, in <i>Event Replicator for Adabas Concepts</i>.</p>	No

Parameter Name	Specify	Default
Maximum Output Size (DMAXOUTPUTSIZE)	<p>The maximum output size (in bytes) for the destination. This is the equivalent of specify the DMAXOUTPUTSIZE parameter directly in the Event Replicator Server startup job. Valid values are 0 or any integer ranging from 4096 through 2,147,483,647. 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.</p> <p>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.</p> <p>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.</p>	0

Parameter Name	Specify	Default
<p>Open Retry Count (DRETRYCOUNT)</p>	<p>The number of times that an attempt to open the destination will be retried at the interval specified by the Retry Interval parameter. This is the equivalent of specifying the DRETRYCOUNT parameter directly in the Event Replicator Server startup job.</p> <p>Valid values range from 0 through 2,147,483,647 or the literal "GLOBAL".</p> <p>If the value "GLOBAL" is specified for this parameter, the specification for the Retry Count global variable will be used. Any retry attempts will occur at the interval specified by the Retry Interval parameter. A value of zero indicates that no retry attempt to open this destination should occur.</p>	<p>The value of the Retry Count global variable</p>
<p>Open Retry Interval (DRETRYINTERVAL)</p>	<p>The default number of seconds between retry attempts to open the destination. This is the equivalent of specifying the DRETRYINTERVAL parameter directly in the Event Replicator Server startup job.</p> <p>Valid values are 0, 5 through 2,147,483,647, or the literal "GLOBAL".</p> <p>If the value "GLOBAL" is specified for this parameter, the specification for the Retry Interval global variable 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.</p>	<p>The value of the Retry Interval global variable</p>

Parameter Name	Specify	Default
Opened at Start (DOPEN parameter)	<p>Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default.</p> <p>When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is <i>not</i> opened at Event Replicator Server startup.</p> <p>When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the Global OPEN Value (GOPEN) global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened.</p> <p>This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.</p>	Global

- Supply TLOG parameter specifications for the WebSphere MQ destination. The fields for these TLOG parameters can be found at the bottom of the MQSeries Destination panel:

TLOG Values	
Name	Value
Assign Level	0 - No Logging
Completion Level	0 - No Logging
SLOG Write Level	0 - No Logging
SLOG Read Level	0 - No Logging

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Assign Level (DTLASSIGN)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (DTLCOMP)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (DTLSLOGWRITE parameter)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Read Level (DTLSLOGREAD parameter)	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. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging

Step 3. (Optional) Specify Destination Class Information, If Applicable

The Destination Class and Destination Class Parameter fields can be used to invoke and pass parameters to the Event Replicator Target Adapter for this destination. The fields are located in the middle of the general fields for the WebSphere MQ destination. Do not use these fields unless you want to invoke and pass parameters to the Event Replicator Target Adapter for the destination or unless otherwise requested by a Software AG support representative.

If applicable, use the Destination Class (DCLASS) field to specify the destination class for this destination definition. Valid values are "NONE" (interpreted as a blank) or "SAGTARG" (if Event Replicator Target Adapter processing should be invoked). There is no default.

If you specify a value for the Destination Class field, you can optionally use the Destination Class Parameter (DCLASSPARM) field to specify up to 120 bytes of character data to be passed to the optional destination output user exit.

If DCLASS=SAGTARG is specified (if the Destination Class field is set to "SAGTARG") to invoke the Event Replicator Target Adapter for a destination, you may want to specify one or more of the following keyword parameters:

Note:

These parameter keywords must be specified in uppercase.

NOSPRE

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.

OPTIONS

The OPTIONS keyword parameter can be used to specify options for the destination. Specify one or more of the options described in the following table, using the syntax `OPTIONS=val1, val2, . . .`

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.
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.

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
```

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.

Step 4. Save the WebSphere MQ Destination Definition

 **To use Event Replicator Administration to save a WebSphere MQ destination definition:**

1. When all specifications have been made to your satisfaction, click **OK** on the WebSphere MQ destination panel.

The WebSphere MQ destination definition is saved in the Replicator system file.

Creating a File Destination Definition

Using a File destination definition, 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* provided with your Event Replicator Administration documentation.

Caution:

Be sure that the CLOG is defined in the Event Replicator Server startup JCL (via one or more `DDCLOGRn 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, read your Adabas documentation.

To create a File destination definition in Event Replicator Administration, complete the following steps:

- Step 1. Access the File Destination Definition Creation Area
- Step 2. Specify General and TLOG File Destination Parameters
- Step 3. Save the File Destination Definition

Step 1. Access the File Destination Definition Creation Area

▶ To access the File destination definition creation area of Event Replicator Administration:

1. List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*.

The destination definitions are listed in detail-view.

2. Right-click on **Destinations** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3. Click on **Create New File Destination** in the drop-down menu.

A blank **File Destination** panel appears in detail-view.

Step 2. Specify General and TLOG File Destination Parameters

▶ To use Event Replicator Administration to supply general and TLOG specifications for a File destination definition, complete the following steps:

1. Supply general parameter specifications for the File destination. The fields for these general parameters can be found at the top of the File Destination panel:

General Values	
Name	Value
Destination Name	<input type="text"/> *
Commit Threshold	0 <input type="text"/> *
Destination Active	Yes <input type="button" value="v"/>
Allow Logging	No <input type="button" value="v"/>
Replicate Utility Changes	No <input type="button" value="v"/>
Event Logging	No <input type="button" value="v"/>
Opened at Start	Global <input type="button" value="v"/>

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Destination Name (NAME)	The unique name for the File destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---
Commit Threshold (DCOMMITTHRESHOLD)	The number of URBLTDOD TLOG record bytes that will be written to the CLOG sequential file before the buffers are flushed. 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.	5
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No
Replicate Utility Changes(DREPLICATEUTI)	Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup or whether it should be activated for a specific target file. Valid values are "Yes" and "No". If "Yes" is specified, utility replication is activated; if "No" is specified, utility replication is not activated. For more information about replicating utility functions, read <i>Replicating Utility Functions</i> , in <i>Event Replicator for Adabas Concepts</i> .	No

Parameter Name	Specify	Default
Event Logging (<i>DEVENTLOG</i>)	<p>Whether or not events should be logged by the Event Replicator Server and sent to this destination. This is the equivalent of specifying the <i>DEVENTLOG</i> directly in the Event Replicator Server startup job. Valid values are "Yes" or "No". When this optional parameter is set to "Yes", Event Replicator Server events are logged to the destination. When this parameter is set to "No" (the default), they are not.</p> <p>Event Replicator Server events are logged in URBS elements. In releases prior to 3.2.1, these URBS elements were sent to destinations related to the event itself. Starting with release 3.2.1, the URBS elements are also sent to any other destinations you have defined "Event Logging =Yes". If a related destination also is defined with "Event Logging =Yes", it will only receive one instance of the URBS element.</p> <p>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.</p>	No

Parameter Name	Specify	Default
Opened at Start (DOPEN parameter)	<p>Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default.</p> <p>When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is <i>not</i> opened at Event Replicator Server startup.</p> <p>When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the Global OPEN Value (GOPEN) global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened.</p> <p>This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.</p>	Global

- Supply TLOG parameter specifications for the File destination. The fields for these TLOG parameters can be found at the bottom of the File Destination panel:

TLOG Values	
Name	Value
Assign Level	0 - No Logging
Completion Level	0 - No Logging
SLOG Write Level	0 - No Logging
SLOG Read Level	0 - No Logging

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Assign Level (DTLASSIGN)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (DTLCOMP)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (DTLSLOGWRITE)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Read Level (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. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging

Step 3. Save the File Destination Definition

► To use Event Replicator Administration to save a File destination definition:

1. When all specifications have been made to your satisfaction, click **OK** on the File destination panel.

The File destination definition is saved in the Replicator system file.

Creating a Null Destination Definition

Using null destinations, data replication is tested without actually sending the data to any destination.

To create a null destination definition in Event Replicator Administration, complete the following steps:

- Step 1. Access the Null Destination Definition Creation Area
- Step 2. Specify General and TLOG Null Destination Parameters
- Step 3. (Optional) Specify Destination Class Information, If Applicable

- Step 4. Save the Null Destination Definition

Step 1. Access the Null Destination Definition Creation Area

▶ To access the null destination definition creation area of Event Replicator Administration:

1. List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*.

The destination definitions are listed in detail-view.

2. Right-click on **Destinations** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3. Click on **Create New NULL Destination** in the drop-down menu.

A blank **NULL Destination** panel appears in detail-view.

Step 2. Specify General and TLOG Null Destination Parameters

▶ To use Event Replicator Administration to supply general and TLOG specifications for a null destination definition, complete the following steps:

1. Supply general parameter specifications for the null destination. The fields for these general parameters can be found at the top of the **NULL Destination** panel:

General Values	
Name	Value
Destination Name	<input type="text"/> *
Architecture	2 - High Order Byte First, EBCDIC Encoding Family ▾
Commit Threshold	5 *
Destination Active	Yes ▾
Allow Logging	No ▾
Event Logging	No ▾
Destination Class	<input type="text"/>
Destination Class Parameter	<input type="text"/>
Replicate Utility Changes	No ▾
Maximum Output Size	0
Opened at Start	Global ▾

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Destination Name (NAME)	The unique name for the null destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---
Architecture (DARC)	The data architecture for fields in the URB* control structures sent to the null destination. For complete information on calculating a value for this parameter, read <i>DARC</i> , in <i>Event Replicator for Adabas Reference Guide</i> provided with your Event Replicator Administration documentation.	2 (High-order byte first, EBCDIC encoding)
Commit Threshold (DCOMMITTHRESHOLD)	The number of messages that will be sent to the null destination before a commit is performed for those messages. 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.	5
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No

Parameter Name	Specify	Default
Event Logging (<i>DEVENTLOG</i>)	<p>Whether or not events should be logged by the Event Replicator Server and sent to this destination. This is the equivalent of specifying the <i>DEVENTLOG</i> directly in the Event Replicator Server startup job. Valid values are "Yes" or "No". When this optional parameter is set to "Yes", Event Replicator Server events are logged to the destination. When this parameter is set to "No" (the default), they are not.</p> <p>Event Replicator Server events are logged in URBS elements. In releases prior to 3.2.1, these URBS elements were sent to destinations related to the event itself. Starting with release 3.2.1, the URBS elements are also sent to any other destinations you have defined "Event Logging =Yes". If a related destination also is defined with "Event Logging =Yes", it will only receive one instance of the URBS element.</p> <p>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.</p>	No
Destination Class (<i>DCLASS</i>)	For the moment, leave this field blank. It is described later in this section.	---
Destination Class Parameter (<i>DCLASSPARAM</i>)	For the moment, leave this field blank. It is described later in this section.	---

Parameter Name	Specify	Default
Replicate Utility Changes (DREPLICATEUTI)	<p>Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup. Valid values are "Yes" and "No".</p> <p>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.</p> <p>For more information about replicating utility functions, read <i>Replicating Utility Functions</i>, in <i>Event Replicator for Adabas Concepts</i>.</p>	No
Maximum Output Size (DMAXOUTPUTSIZE)	<p>The maximum output size (in bytes) for the destination. This is the equivalent of specify the DMAXOUTPUTSIZE parameter directly in the Event Replicator Server startup job. Valid values are 0 or any integer ranging from 4096 through 2,147,483,647. 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.</p> <p>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.</p> <p>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.</p>	0

Parameter Name	Specify	Default
Opened at Start (DOPEN parameter)	<p>Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default.</p> <p>When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is <i>not</i> opened at Event Replicator Server startup.</p> <p>When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the Global OPEN Value (GOPEN) global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened.</p> <p>This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.</p>	Global

- Supply TLOG parameter specifications for the null destination. The fields for these TLOG parameters can be found at the bottom of the **NULL Destination** panel:

TLOG Values	
Name	Value
Assign Level	0 - No Logging
Completion Level	0 - No Logging
SLOG Write Level	0 - No Logging
SLOG Read Level	0 - No Logging

Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Assign Level (DTLASSIGN)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (DTLCOMP)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (DTLSLOGWRITE)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Read Level (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. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging

Step 3. (Optional) Specify Destination Class Information, If Applicable

The Destination Class and Destination Class Parameter fields can be used to invoke and pass parameters to the Event Replicator Target Adapter for this destination. The fields are located in the middle of the general fields for the null destination. Do not use these fields unless you want to invoke and pass parameters to the Event Replicator Target Adapter for the destination or unless otherwise requested by a Software AG support representative.

If applicable, use the Destination Class (DCLASS) field to specify the destination class for this destination definition. Valid values are "NONE" (interpreted as a blank) or "SAGTARG" (if Event Replicator Target Adapter processing should be invoked). There is no default.

If you specify a value for the Destination Class field, you can optionally use the Destination Class Parameter (DCLASSPARM) field to specify up to 120 bytes of character data to be passed to the optional destination output user exit.

If DCLASS=SAGTARG is specified (if the Destination Class field is set to "SAGTARG") to invoke the Event Replicator Target Adapter for a destination, you may want to specify one or more of the following keyword parameters:

Note:

These parameter keywords must be specified in uppercase.

NOSPRE

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.

OPTIONS

The OPTIONS keyword parameter can be used to specify options for the destination. Specify one or more of the options described in the following table, using the syntax `OPTIONS=val1, val2, . . .`

Option Value	Description
1	This option is no longer supported. If specified, it will be ignored.
2	Specify <code>OPTIONS=2</code> 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 <code>OPTIONS=4</code> to ensure that invalid XML characters found in alphanumeric fields are not translated to spaces.
8	Specify <code>OPTIONS=8</code> to ensure that trailing blanks in alphanumeric fields are not removed.
16	Specify <code>OPTIONS=16</code> to ensure that characters used by XML are not replaced automatically with predefined entity references. For example, if <code>OPTIONS=16</code> is set, the ampersand (&) character would not be replaced with the literal "&";
32	Specify <code>OPTIONS=32</code> 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.

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
```

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.

Step 4. Save the Null Destination Definition

▶ To use Event Replicator Administration to save a null destination definition:

1. When all specifications have been made to your satisfaction, click **OK** on the NULL Destination panel.

The null destination definition is saved in the Replicator system file.

Modifying Destination Definitions

▶ To use Event Replicator Administration to modify a destination definition in the Replicator system file:

1. List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*.

The destination definitions are listed in detail-view.

2. Locate the definition you want to modify in the table in detail-view and click on it.


The Destination Definition panel appears in detail-view listing the current settings for the destination definition you selected.

3. Click the **Modify** button.

The destination parameters you can modify become editable in detail-view. For information on modifying these parameters, read the description of adding that type of destination definition, elsewhere in this section.

4. When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the changes.

Copying Destination Definitions

 **To use Event Replicator Administration to copy a destination definition in the Replicator system file:**

1. List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*.

The destination definitions are listed in detail-view.

2. Locate the definition you want to copy in the table in detail-view and click on it.

The Destination Definition panel appears in detail-view listing the current settings for the destination definition you selected.

3. Click the **Copy** button.

A copy of the destination definition is created and its parameter values appear in detail-view.

4. Specify a new, unique name for the copy of the destination definition in the Value column for the **Destination Name** parameter.

5. If you wish, modify any other parameters for the new copy in detail-view. For information on modifying the parameters, read the description of adding that type of destination definition, elsewhere in this section.

6. When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the copy.

Activating and Deactivating Destination Definitions

You can use Event Replicator Administration to activate and deactivate destination definitions.

**Warning:**

Be careful when you activate and deactivate replication definitions and databases, especially if replication is ongoing at the time. Whenever you activate or deactivate definitions or databases, you run the risk of altering what data is replicated and how that replication occurs. If the Event Replicator Server receives data from an Adabas database for which it has no active definitions, replication simply does not occur.

This section covers the following topics:

- Activating Destination Definitions
- Deactivating Destination Definitions

Activating Destination Definitions

 **To use Event Replicator Administration to activate a destination definition:**

1. Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*.
2. Click and expand **Active Parameters** in tree-view under the selected database.
3. Click and expand **Replication** in tree-view under **Active Parameters**.
4. Click on **Destinations** in the tree-view under **Replication**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

5. In detail-view, click on the name of the definition you want to activate.

Details about the destination appear in detail-view.

6. Click the **Activate** button.

Note:

This button will not be available if the definition is already activated.

The destination definition is activated.

Deactivating Destination Definitions

 **To use Event Replicator Administration to deactivate a destination definition:**

1. Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*.
2. Click and expand **Active Parameters** in tree-view under the selected database.
3. Click and expand **Replication** in tree-view under **Active Parameters**.
4. Click on **Destinations** in the tree-view under **Replication**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

5. In detail-view, click on the name of the definition you want to deactivate.

Details about the destination appear in detail-view.

6. Click the **Deactivate** button.

Note:

This button will not be available if the definition is already deactivated.

The destination definition is deactivated.

Opening and Closing Destinations

You can use Event Replicator Administration to open and close destinations. This section covers the following topics:

- Opening Destinations
- Closing Destinations

Opening Destinations

 **To use Event Replicator Administration to open a destination:**

1. Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*.
2. Click and expand **Active Parameters** in tree-view under the selected database.
3. Click and expand **Replication** in tree-view under **Active Parameters**.
4. Click on **Destinations** in the tree-view under **Replication**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

5. In detail-view, click on the name of the definition you want to open.

Details about the destination appear in detail-view.

6. Click the **Open** button.

Note:

This button will not be available if the destination is already opened.

The destination is opened.

Closing Destinations

 **To use Event Replicator Administration to close a destination:**

1. Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*.
2. Click and expand **Active Parameters** in tree-view under the selected database.
3. Click and expand **Replication** in tree-view under **Active Parameters**.
4. Click on **Destinations** in the tree-view under **Replication**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

5. In detail-view, click on the name of the definition you want to close.

Details about the destination appear in detail-view.

6. Click the **Close** button.

Note:

This button will not be available if the destination is already closed.

The destination is closed.

Deleting Destination Definitions

 **To use Event Replicator Administration to delete a destination definition in the Replicator system file:**

1. List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*.

The destination definitions are listed in detail-view.

2. Locate the definition you want to delete in the table in detail-view and click on it.

The Destination Definition panel appears in detail-view listing the current settings for the destination definition you selected.

3. Click the **Delete** button.

A confirmation panel appears verifying that you want to delete the definition. If you click **Yes** (indicating that you do want to delete the definition), the definition is deleted. If you click **No** (indicating that you do not want to delete the definition), the definition is not deleted.