

Adabas Cluster Services

Concepts and Facilities

Version 8.1.3

June 2014

This document applies to Adabas Cluster Services Version 8.1.3.

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

Copyright © 2014 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors..

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://documentation.softwareag.com/legal/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Document ID: ALS-INTRO-813-20140626

Table of Contents

Introduction	v
Conventions	vi
1 Features of a Sysplex	1
Sysplex References	2
2 Components and Processes	3
Cluster of Adabas Nuclei	4
Adabas Sysplex Cluster	4
Single Physical Database Per Cluster	5
ADACOM	5
ADASVC Component SVCCLU	6
Cache Structure	7
Lock Structure	8
XCF Group	9
Messaging Services	9
3 Logs and Log Merges	11
Protection Logs	12
Command Logs	14
Coordinated Log Switching	15
4 Parallel Participant Table (PPT)	17
Acquiring the PPT	18
Registering Nuclei during Session Initialization	18
Registering PLOG, CLOG, and Work Data Sets for Each Nucleus	18
Nucleus Termination Processing	19
Using PPT Information	19
PPT Unavailable	19
SBLKNUM Parameter for ADARES PLCOPY NOPPT	20
5 Dynamic Allocation	21
JCL Considerations	22
6 Component Summary	23
7 Adabas Cluster Services and Entire Net-Work	25
XCF Line Driver	27
Summary of Entire Net-Work Features	28
8 Getting Started	31
First Cluster Nucleus Starts	32
Additional Cluster Nuclei on the Local Image	33
Additional Cluster Nuclei on a Remote Image	33
Images with Cluster Users But No Nuclei	33
Environment Maintenance	34
Assigning Users to Nuclei	34
9 Adabas Cluster Services and Other Software AG Products	35
Adabas Online System	36
Adabas Caching Facility	36
Adabas Delta Save Facility	36

Adabas Fastpath	37
Adabas Review	37
Adabas Transaction Manager	37
Adabas Vista	38
Adabas SAF Security	38
Index	39

Introduction

- Conventions vi

This documentation provides information on Adabas Cluster Services (formerly Adaplex+), a selectable unit of Adabas that helps you to make full use of IBM's parallel sysplex environment by allowing a single physical database to be accessed simultaneously by up to 32 nuclei installed across multiple operating system images.

Adabas Cluster Services extends the earlier Adaplex+ capability to allow all nuclei in a "cluster" to process updates concurrently.

Adabas Cluster Services implements multinucleus, multithread parallel processing and optimizes Adabas in z/OS parallel sysplex (SYStems comPLEX) environments.

In addition to the increased throughput that results from parallel processing, Adabas Cluster Services increases database availability during planned or unplanned outages: the database can remain available when a particular operating system image or cluster nucleus requires maintenance or goes down unexpectedly.

The Adabas Cluster Services Introduction is organized in the following topics:

[Features of a Sysplex](#)

[Components and Processes](#)

[Logs and Log Merges](#)

[Parallel Participant Table \(PPT\)](#)

[Dynamic Allocation](#)

[Component Summary](#)

[Adabas Cluster Services and Entire Net-Work](#)

[Getting Started](#)

[Adabas Cluster Services and Other Software AG Products](#)

Conventions

Syntax

The following syntax conventions are used in this documentation when describing ADARUN parameters and operator commands:

- The vertical bar (|) separates parameter options or values, one of which must be chosen.
- Curly brackets or braces ({ }) delimit the series of options or values.
- (Square) brackets ([]) indicate a term that may optionally be specified.
- The underline (_) identifies the default value.

- Three dots preceded by a comma or a slash (, ... or / ...) indicates that the previous term can be repeated but must be preceded by the comma or slash as indicated.
- A range of values is indicated by two instances of a variable (for example, *fnr*, *rabn*, *csn*) separated by a hyphen (for example, *fnr - fnr*). The first instance represents the low value and the second the high value of the range. The range specification comprises the two numbers separated by the hyphen with no intervening spaces.
- Parentheses (()) are part of the term and must be included as part of the parameter statement.

Data Set Names

Data set names starting with DD are referred to in the Adabas documentation with a slash separating the DD from the remainder of the data set name to accommodate VSE/ESA data set names that do not contain the DD prefix. The slash is not part of the data set name.

1 Features of a Sysplex

- Sysplex References 2

A "sysplex" comprises hardware components and software services that allow z/OS systems to communicate and cooperate with each other to process a user's work. For certain kinds of work, the sysplex provides parallel processing and improved data sharing. It also provides a way to remove a single point of failure in the data processing (DP) enterprise.

High-speed caching, list processing, and locking functions are provided by the sysplex coupling facility, which allows up to 32 system images to communicate shared data. The coupling facility resides in a special logical partition where storage is divided into objects called "structures". Structure types are cache, list, and lock. Adabas Cluster Services implements lock and cache services using the sysplex cross-system extended services (XES).

The coupling facility resource management (CFRM) "policy" is used to manage coupling facility storage. It contains information about each coupling facility structure to be used. Once created, the policy resides in the CFRM COUPLE_{xx} data sets.

Once defined in the CFRM policy, coupling facility structures are allocated by the first cluster nucleus that starts. Defined and allocated structures can be viewed using the system activity monitor. The structures are not persistent; that is, they disappear when the last active nucleus terminates, either normally or abnormally. No cleanup is required.

z/OS systems are directly connected to the coupling facility through high-bandwidth, high-speed channels that can be a kilometer in length in some cases. These channels provide a fast and efficient means for systems in the sysplex to access data in the coupling facility. Using the sysplex cross-system coupling facility (XCF) services, Adabas Cluster Services implements group and signaling services.

The chapter covers the following topics:

Sysplex References

For additional information about the sysplex environment, refer to the IBM documentation

- Parallel Sysplex Overview: Introducing Data Sharing and Parallelism in a Sysplex
- MVS Programming: Sysplex Services Guide
- MVS Programming: Sysplex Services Reference
- MVS Setting Up a Sysplex

2 Components and Processes

- Cluster of Adabas Nuclei 4
- Adabas Sysplex Cluster 4
- Single Physical Database Per Cluster 5
- ADACOM 5
- ADASVC Component SVCCLU 6
- Cache Structure 7
- Lock Structure 8
- XCF Group 9
- Messaging Services 9

The Adabas Cluster Services components and processes ensure intercommunicability and data integrity among the operating system images and the associated Adabas nuclei in each cluster in a sysplex environment.

The chapter covers the following topics:

Cluster of Adabas Nuclei

The ADARUN parameter `CLUSTER=SYSPLEX` identifies a nucleus as a member of an Adabas cluster in a sysplex environment. The `NUCID` parameter is used to number the Adabas cluster nuclei 1-65000. A `NUCID` of zero (0) identifies a noncluster nucleus. The `NUCID` is the only ID table entry for a nucleus.

Each nucleus has its own

- local buffer pool (LBP) and manager;
- Work data set, which must have the same device type and physical size for each nucleus in the cluster;
- (optional) direct-access protection log (PLOG) data sets; and
- (optional) command log (CLOG) data sets.

Protection logs are optional but if supplied, each nucleus must have its own. The same is true for command logs.

Adabas Sysplex Cluster

Each Adabas sysplex cluster is defined by its

- database ID;
- sysplex group name;
- sysplex cache and lock structure names.

In a sysplex environment, the Adabas nuclei in a cluster can be distributed over multiple operating system images that are synchronized by an IBM Sysplex Timer™.

Although more than one Adabas cluster nucleus may be run under an operating system image, several nuclei typically run under different operating system images to minimize the impact on the cluster when an operating system image must be stopped for maintenance or upgrades. The cluster database would continue to be serviced by the cluster nuclei that are active under different operating system images. For more information, read *Planning an Outage*, in *Adabas Cluster Services Operations*.

Single Physical Database Per Cluster

Using Adabas Cluster Services, up to 32 Adabas nuclei running under multiple operating system images access a single physical database simultaneously. A "single physical database" is one set of Associator and Data Storage data sets identified by a single database ID number (DBID).

An unlimited number of Adabas clusters can operate in the same sysplex system under the same or different SVCs; that is, an unlimited number of separate databases can be processed, each with its own Adabas cluster of up to 32 nuclei.

Each database has an ID table header (IDTH) prefix element.

Applications see only one database target; that is, no interface changes are required. Applications still communicate with their intended databases and communicate with the Adabas cluster of nuclei without modification.

ADACOM

The ADACOM initialization module or a cluster nucleus, whichever is the first to start, sets up the Adabas sysplex cluster environment on an operating system image. See the section [Getting Started](#). An ECSA area is allocated for maintaining information about active users and the nuclei to which they are assigned. Using the ADACLU module and Entire Net-Work, any changes are sent to all peer ADACOMs or cluster nuclei.

The ADACOM initialization task:

- must be run at least once per IPL on any operating system image that has cluster users but no cluster nuclei to set up the environment.
- can optionally be used to monitor and control the nuclei of one or more sysplex clusters.

Parameters specify the SVC/DBID combinations (sets) that the ADACOM is to manage. The DBID identifies the external physical database shared by a particular cluster of nuclei and known to the application. The same SVC may be used for different clusters.

The first nucleus or ADACOM started governs the value for NU: different values set for subsequent nuclei or ADACOMs are ignored. The NU parameter in each ADACOM and each cluster nucleus should be large enough for all possible parallel local users and locally assigned remote users, so that the user table in common storage and the user queue in each cluster nucleus is big enough to continue providing service in case only one nucleus remains active in the cluster. If NU is set to zero, ADACOM frees the environment and exits. If no value is set for NU, it defaults to 200.

After initialization, ADACOM can either be quiesced or maintained in operation as an internucleus command manager to monitor and control cluster nuclei running under specified SVC/DBID sets and to dynamically create or terminate SVC/DBID sets. ADACOM commands are available to:

- exercise some control over the assignment of users to nuclei;
- display the number of commands and user assignments for each active nucleus on the local image only or on every image known to the local image; and
- dynamically create or terminate SVC/DBID combinations (sets).

Software AG recommends that you execute and maintain the ADACOM module in each operating system image that participates in the sysplex so that the internucleus commands are available.

For operating system images that have users but no nuclei, Software AG recommends that you keep ADACOM in operation so that it is available not only as a command manager but also to reset the environment if Entire Net-Work goes down for any reason and comes back up.

Although a single ADACOM job can run all SVC/DBID sets in a sysplex environment, it is possible to run multiple ADACOM tasks simultaneously with the same, mixed, or completely different SVC/DBID sets. Two subtasks are attached for each SVC/DBID pair in each ADACOM task in which it occurs.

The COMPRINT data set prints global messages that apply to all SVC/DBID sets defined to an ADACOM task. In addition, two SYSOUT data sets are dynamically allocated for command output to each SVC/DBID set. The format of these file names are "P*ss*dddd" and "D*ss*dddd" where *ss* is the last two digits of the SVC and *dddd* is the DBID.

ADASVC Component SVCCLU

Prelinked to the version 8.1 Adabas SVC is the cluster SVC component SVCCLU that routes commands to local and, using Entire Net-Work, to remote nuclei. To make routing decisions, it uses local ECSA tables, which are updated based on nucleus or ADACOM information. When a user is moved to another system by CICSplex, for example, the ADASVC containing the SVCCLU on that system knows where to route the user's command.

An unlimited number of Adabas clusters each servicing a separate database can run under a single version 8.1 ADASVC.

Cache Structure

Each Adabas cluster in a sysplex environment uses a cache structure to hold ASSO and DATA blocks that have been updated during the session. Caching only the updated blocks improves the efficiency of information transfer and reduces the overall size of the required cache area. The cache structure synchronizes the nuclei, users, and operating system images and ensures data integrity.

The sysplex cache structure resides in the coupling facility in accordance with IBM's definition of coupling facility cache operations; it is thus accessible to all nuclei. The name and size of the cache structure is defined in the coupling facility resource management (CFRM) policy. The cache structure name for an Adabas cluster in a sysplex environment is specified for each cluster nucleus using the ADARUN parameter `CLUCACHENAME`. The same name must be provided to each nucleus in the cluster, and the same structure name may not be used outside the cluster.

Every nucleus in an Adabas cluster has a local buffer pool and manager. The buffer pool manager (BPM) oversees all nucleus requests for database blocks.

For each block in its local buffer pool, the BPM

- registers its interest in the block with the coupling facility;
- checks the coupling facility for the status of the registered block to ensure that its nucleus always has the most current copy of the block; and
- writes changed blocks to the coupling facility cache structure.

Whenever an updated block is written to the cache structure, the coupling facility notifies all nuclei that have "registered" an interest in that particular block that their local copy has become invalid.

Buffer Flush

From the cache structure, modified blocks are "cast out" to the flush I/O pool (FIOP) buffer before they are written to disk. The FIOP buffer is sized using the ADARUN `LFIOIP` parameter, and the frequency of buffer flushing depends on the limit set. Until such blocks are written to the database, the cache structure holds more current information than the database.

The coupling facility handles the deletion of blocks in the cache structure when it becomes necessary to overwrite them.

Lock Structure

Each Adabas cluster in a sysplex environment uses a lock structure to manage the setting, status, and release of various locks imposed during multiple update nucleus processing. The lock structure synchronizes the nuclei, users, operating system images, and transaction processing to ensure data integrity.

The sysplex lock structure resides in the sysplex coupling facility in accordance with IBM's definition of coupling facility lock operations; it is thus accessible to all nuclei. The name and size of the lock structure is defined in the sysplex coupling facility resource management (CFRM) policy. The lock structure name for an Adabas cluster in a sysplex environment is specified for each cluster nucleus using the ADARUN parameter `CLULOCKNAME`. The same name must be provided to each nucleus in the cluster, and the same structure name may not be used outside the cluster.

Adabas Cluster Services uses the lock structure name when initializing its lock manager. The lock manager

- connects to and disconnects from the lock structure;
- obtains (conditional or unconditional), releases, and alters ownership (shared or exclusive) of resource locks; and
- reads recovery information about a failed lock.

Lock manager calls may be asynchronous, meaning that the nucleus may continue processing in other threads before a call has completed.

If a lock request is conditional, it is rejected if the lock is not free; if the lock request is unconditional, the nucleus thread waits until the requested lock is free before continuing.

In general, locks are used to prevent two cluster nuclei from using the same resource at the same time. Such resources include

- data records, which are protected by hold queue element (HQE) locks;
- unique descriptor values, which are protected by unique descriptor element (UQDE) locks;
- end-transaction IDs, which are protected by the ETID lock; and
- various other single-instance resources.

XCF Group

Each Adabas cluster in a sysplex environment uses a cross-system coupling facility (XCF) group to control communication among the cluster nuclei. XCF automatically creates an XCF group for a cluster when the first-starting cluster nucleus attempts to join the group.

When the sysplex COUPLE_{xx} data sets are first formatted, the maximum number of XCF groups to be used for the entire sysplex environment is defined. However, XCF group names are *not* predefined in the CFRM policy.

A name for a cluster's XCF group must be provided to each cluster nucleus using the ADARUN parameter CLUGROUPNAME. The same name must be provided to each nucleus in the cluster and the name may not be used outside the cluster.

Messaging Services

The messaging services component is essential to communication among the active nuclei of an Adabas sysplex cluster. For example, messages need to be sent between nuclei when:

- the general control block (GCB) has been modified
- a global parameter has been modified
- a buffer flush occurred
- a PLOG is switching

Adabas cluster messaging service statistics are produced during normal nucleus termination and in response to the DXMSG operator command.

3

Logs and Log Merges

- Protection Logs 12
- Command Logs 14
- Coordinated Log Switching 15

In an Adabas cluster environment, the protection logs (and optionally, the command logs) of all individual nuclei in the cluster are merged into single log files in chronological order for the cluster database shared by all the nuclei as a whole. The chronological order is determined by timestamps on all individual nucleus log records, which are synchronized across systems in a parallel sysplex environment by an IBM Sysplex Timer™.

ADARES is the only utility that handles both the unmerged and merged PLOGs and CLOGs. The log merge process

- merges PLOG or CLOG records into correct chronological sequence.
- avoids a single point of failure by partially merging as much PLOG or CLOG data as possible if a nucleus is unable to respond to a switch.
- is automatic; merges directly from the multiple PLOGs or CLOGs, which eliminates the need to retain control information for tracking unmerged logs.
- quickly copies and merges so that the PLOG or CLOG can be reused.
- uses an identical PLOG or CLOG record layout for an unmerged or a merged record.
- produces a consistent amount of data from each PLCOPY or CLCOPY.

For more detailed information, read *Adabas Cluster Services Operations* and read about the ADARES utility in *Adabas Utilities* documentation.

The chapter covers the following topics:

Protection Logs

Either all nuclei in an Adabas cluster must run without protection logs (PLOGs) or all nuclei must run with PLOGs. The same ADARUN PLOGRQ setting is enforced for all nuclei in a cluster. This value cannot be modified during a session.

If the nuclei in the cluster are running with PLOGs, they must be dual or multiple PLOGs: sequential PLOGs are not allowed. For more information, read your *Adabas Operations*.

Each nucleus in the cluster writes its protection data to its own Work and PLOG data sets: no synchronization is required. However, every protection record is preceded by an 8-byte timestamp indicating the time of its creation. The timestamp is the only difference between protection records written by a cluster nucleus and those written by a noncluster nucleus. The timestamp is required for the merge process and is removed after the protection log is merged but before it is written to the sequential data set.

Merging Protection Logs

For recovery processing, all protection log data sets are automatically merged into a single log "stream" for each cluster database when ADARES PLCOPY is executed. The PLCOPY process accesses the parallel participant table (see section *PPT*) to determine which protection logs to copy and opens the appropriate data sets using dynamic allocation. PLCOPY copies/merges as much data as possible; if a nucleus is still writing to a protection log data set, PLCOPY 'partially' merges the data set.

ADARES expects that at least one of the protection logs being merged is at 'completed' status. If this is not the case, ADARES reports that there is no data to be copied.

The merge begins with the lowest timestamp from all protection logs being merged and ends with the lowest of the ending timestamps from all data sets. Records beyond this point are written to an 'intermediate' data set, which must be supplied as input to the next merge.

Intermediate Data Sets

It is necessary to supply two intermediate data sets to the PLCOPY merge process: one is used as input and the other as output. ADARES analyzes both data sets to determine which is which.

A cross-check ensures that the correct input intermediate data set has been supplied; the utility will not continue if the correct data set has not been supplied. The input intermediate data set contains "leftover unmerged" data from the last run of the utility, without which the merged protection data is incomplete.

User Exits for Merging Protection Logs

A user exit is necessary for merging protection logs automatically: user exit 2 (UEX2) for dual logs or user exit 12 (UEX12) for multiple logs. However, you may choose to run without the user exit. This may be advisable for testing purposes, for example. If one nucleus in a cluster has the user exit specified, then all other nuclei in the cluster must also supply the user exit. The user exit supplied must be the same across all nuclei.

Sample user exits (UEX2 and UEX12) are provided that illustrates the necessary changes for the intermediate data set. Once DD statements for the PLOG data sets have been supplied on the session startup JCL, you do not need to supply them again for ADARES as these are opened using dynamic allocation. If the DD statements are supplied, they are ignored.

If there is a problem with the user exit, the Extended Error Recovery logic allows you to disable the exit and load a new exit without bringing the nucleus down.



Note: While a non-cluster nucleus runs with the PSW in problem state, an Adabas Cluster Services nucleus runs with the PSW in supervisor state. User exits must return in the same state (i.e., problem state or supervisor state) in which they receive control. In fact, user exits

must return with all PSW fields the same except for the next instruction address and the condition code.

Command Logs

Each nucleus has its own command log (CLOG) data sets where it logs commands during a session. Command logging is activated by setting the ADARUN parameter `LOGGING=YES`.

Multiple command logging requires that you set ADARUN parameters correctly. See the *Adabas Operations* documentation for more information.

Merging Command Logs Manually

You can copy the dual or multiple command logs into a single sequential data set for a nucleus using the default ADARES CLCOPY function. Then, using the output from the CLCOPY execution as input, you can use the ADARES MERGE CLOG function to create a single cluster-wide command log containing, in chronological order, all Adabas commands executed by any of the cluster nuclei in the time period covered by the log:

```
ADARES MERGE CLOG NUMLOG=n
```

where *n* is the number of command logs to be merged, but not more than 32.

The ADARES MERGE CLOG function takes as input the sequential command log data sets from each of the cluster nuclei that were produced by running CLCOPY, merges them into chronological order and copies the resulting command log data set from disk to a sequential data set.

Merging Command Logs Automatically

For accounting or other tracking purposes, you may want to automate the CLOG merge process. When ADARUN `CLOGMRG=YES` is set, CLOGs are merged automatically as part of the ADARES CLCOPY process in much the same way that PLOGs are merged as part of the ADARES PLCOPY process.

Existing CLCOPY jobs must be modified to include intermediate data sets. The sample CLCOPY job ADARESMC illustrates the necessary addition of the intermediate data sets. When intermediate data sets are used for both PLCOPY and CLCOPY merge processes, the data set names must be unique so that they are not overwritten.

You can use the same exit for the CLOG merge process that is supplied for the PLOG merge process (see the sample exit USEREX2P):

- When `CLOGMRG=YES`, use a single user exit 2 or user exit 12 for both PLOG and CLOG merging; both PLOG and CLOG data sets are dynamically allocated. Data sets specified in the user exit 2 or user exit 12 JCL are ignored.

- When `CLOGMRG=NO` (the default), CLOG data sets are *not* dynamically allocated and must be specified in the ADARES JCL; therefore, user exit 2 or user exit 12 must submit different CLCOPY jobs that are appropriate for the nucleus that calls the exit.

Likewise, CLOG data sets with slightly modified names can be included in the user exit 2 or user exit 12 JCL used for other NUCIDs.



Note: While a non-cluster nucleus runs with the PSW in problem state, an Adabas Cluster Services nucleus runs with the PSW in supervisor state. User exits must return in the same state (i.e., problem state or supervisor state) in which they receive control. In fact, user exits must return with all PSW fields the same except for the next instruction address and the condition code.

Dynamically Modifying the CLOGMRG Parameter Setting

The `CLOGMRG` parameter setting can be changed dynamically with the new operator command `CLOGMRG`, the new ADADBS OPERCOM `CLOGMRG` function, or with the Modify Parameters function of Adabas Online System.

`CLOGMRG` is a global (that is, cluster-wide) parameter. A change made to one nucleus is automatically propagated to all nuclei in the cluster.

Coordinated Log Switching

The automated log merge process makes it necessary to coordinate the log switch process so that the merge process can copy as much data as possible without causing too many switches.

The first cluster nucleus that needs to switch logs (a log is full or the `FEOFPL` or `FEOFCL` command has been issued) sends a broadcast message to the other nuclei in the cluster indicating that it is switching. Then it goes on as normal, switching and calling the appropriate user exit.



Note: User exit 2 for dual logs or user exit 12 for multiple logs is recommended and is required if `PLOGRQ=FORCE` is set. If it is supplied to one nucleus in a cluster, it must be supplied to all the other nuclei. `PLOGRQ` must have the same setting across all nuclei in the cluster. If a cluster nucleus is to run with PLOGs, they must be dual or multiple PLOGs; sequential PLOGs are not allowed.

When a cluster nucleus receives the message that another nucleus is switching logs, it must decide if it will switch based on the status of its current log and whether it has an available log:

- If it switches, it does so without calling the user exit (user exit 2 or user exit 12) .
- If it does not switch, ADARES copies off all protection or command data up to the point in time where the first nucleus switched. The next PLCOPY or CLCOPY begins where this one stops.



Note: While a non-cluster nucleus runs with the PSW in problem state, an Adabas Cluster Services nucleus runs with the PSW in supervisor state. User exits must return in the same state (i.e., problem state or supervisor state) in which they receive control. In fact, user exits must return with all PSW fields the same except for the next instruction address and the condition code.

Global FEOFPL and FEOFCL Commands

The GLOBAL option has been added to both the ADADBS OPERCOM FEOFPL and FEOFCL commands and the corresponding console operator commands:

- When the GLOBAL option is not specified, consideration is given to how full the PLOG or CLOG is before an attempt is made to switch them.
- When the GLOBAL option is specified, a log switch is forced for each nucleus in the cluster if there is an available log data set to write to. This may be useful in situations where all nuclei need to switch at the same time to capture all merged records in the output data set.

If there is no log data set available to write to, the switch does not occur even if GLOBAL has been specified. This functions the same way when the FEOFPL or FEOFCL command is issued for a single nucleus.

4 Parallel Participant Table (PPT)

- Acquiring the PPT 18
- Registering Nuclei during Session Initialization 18
- Registering PLOG, CLOG, and Work Data Sets for Each Nucleus 18
- Nucleus Termination Processing 19
- Using PPT Information 19
- PPT Unavailable 19
- SBLKNUM Parameter for ADARES PLCOPY NOPPT 20

The parallel participant table (PPT) located in a database's Associator tracks all active Adabas nuclei in the cluster.

The chapter covers the following topics:

Acquiring the PPT

Before you can run an existing Adabas database with Adabas Cluster Services, you must run ADACNV to convert it to the version of Adabas that underlies Adabas Cluster Services. ADACNV accesses the free space table (FST) to acquire 32 contiguous blocks from the Associator and reserve them for the PPT.

ADACNV acquires 32 contiguous blocks in the Associator for each existing database and ADADEF acquires 32 contiguous blocks in the Associator for each new database. The location of the 32 contiguous blocks of Associator space is written to the general control block (GCB).

PPT blocks exist for both cluster nuclei and a single noncluster nucleus, eliminating the need to run the conversion process if switching from an Adabas cluster session to an Adabas noncluster session.

Registering Nuclei during Session Initialization

During session initialization as each Adabas cluster nucleus becomes active, it registers in the PPT based on the setting of its ADARUN `NUCID` parameter. The nucleus chooses the first of the 32 PPT blocks that is not yet assigned to another `NUCID`. If this is not the first run of this nucleus, the nucleus reuses its previously used PPT block. The relative block number of the chosen PPT block within the PPT (a number between 1 and 32) is called the 'internal nucleus ID'.



Note: A noncluster nucleus always occupies the first block of the PPT and its `NUCID` is always zero.

Registering PLOG, CLOG, and Work Data Sets for Each Nucleus

Each active nucleus writes information to the Work, protection log (PLOG), and command log (CLOG) data sets that it is using for that session. The internal nucleus ID is also written to the PLOG records.

As each Work, PLOG, and CLOG data set becomes active, it is registered in the PPT. Information about these data sets is logged in the PPT entry of its nucleus.

When the PLOG or CLOG is merged, its PPT entry is updated with the information. ADARES and the cluster nuclei synchronize their PPT updates against one another to ensure that only one modification occurs at a time.

Work data set, protection log, and command log information remains in the PPT, even when CLOGs or PLOGs are merged or the session terminates.

Nucleus Termination Processing

If the nucleus terminates normally, the PPT entry is retained and marked as inactive.

If a nucleus terminates abnormally, its entry in the PPT remains unchanged until the recovery process is complete.

Using PPT Information

The information about each Adabas nucleus maintained in the PPT is used when ADARES PLCOPY is copying and merging all active protection logs in the cluster.

It is also used during nucleus initialization to ensure that the data set information required for autorestart and PLOG merge is available:

- If the previously used Work data set and PLOG data sets are already in use by another nucleus, the nucleus will not start.
- If a different Work data set is provided and the previously used Work data set contains autorestart information, the nucleus will start but a warning will be printed.
- If different PLOGs or no PLOGs are provided and the previous session had PLOGs still to be copied, the nucleus will not start until those PLOGs have been copied/merged if PLOGRQ=FORCE is specified. If PLOGRQ=FORCE is not specified, the nucleus will start but a warning will be printed.

PPT Unavailable

In the case where the PPT becomes unavailable for some reason, the new parameter NOPPT is added to the ADARES PLCOPY function for emergency use. It specifies that the PPT is to be ignored and that the PLOG data sets of all cluster nuclei are to be taken from the JCL.

If you need to merge CLOGs while the PPT is unavailable, run a normal CLCOPY and then use the CLOG MERGE function to merge the sequential data sets.

SBLKNUM Parameter for ADARES PLCOPY NOPPT

The parameter `SBLKNUM` for the ADARES PLCOPY NOPPT function can be used in an emergency case when a last set of PLOGs must be copied and merged after the parallel participant table (PPT) in the Associator has been destroyed. In this case the ADARES utility does not know the last serial block number of the sequential PLOGs produced so far. By default, it starts numbering the blocks at 1 on the next sequential PLOG data set. If this sequential PLOG is concatenated to its predecessor in a REGENERATE function, ADARES would report an error in the PLOG block number sequence. The `SBLKNUM` parameter can be used to specify the correct starting block number for the sequential PLOG produced by the PLCOPY NOPPT function.

There are two ways to copy and merge the PLOGs when the PPT has been destroyed:

1. Without using the `SBLKNUM` parameter:

- Run ADARES PLCOPY NOPPT, specifying all direct-access PLOG data sets of all cluster nuclei as DD/PLOGnn data sets for the ADARES utility. ADARES copies and merges the relevant PLOGs and produces a sequential PLOG data set starting with PLOG block number 1.
- In case the database or individual files need to be restored and regenerated, run the regenerate in two (or more) job steps. The first REGENERATE should cover all sequential PLOGs produced by PLCOPY functions without the NOPPT parameter; these PLOGs may be concatenated into one sequential input. If the REGENERATE is performed with ET logic (REGENERATE database or with CONTINUE parameter), specify the NOAUTOBACKOUT parameter. The second REGENERATE should cover the sequential PLOG produced by the PLCOPY with NOPPT.

2. With using the `SBLKNUM` parameter:

- Determine the highest PLOG block number written so far by the regular PLOG merge processes. This number can be determined from an up-to-date checkpoint listing, from an ADARAI LIST protocol, or from the job protocol of the most recent PLCOPY execution.
- Run ADARES PLCOPY NOPPT, specifying all direct-access PLOG data sets of all cluster nuclei as DD/PLOGnn data sets for the ADARES utility. Specify the `SBLKNUM` parameter with value $n+1$, where n is the highest PLOG block number written so far, as determined in the previous step. ADARES copies and merges the relevant PLOGs and produces a sequential PLOG data set starting with the specified PLOG block number.
- In case the database or individual files need to be restored and regenerated, no special arrangements need to be made to run the REGENERATE in separate job steps.

5 Dynamic Allocation

- JCL Considerations 22

If used, protection log and command log data sets for use in Adabas cluster environments are specified in the session startup JCL. Session initialization writes the log data set names to the parallel participant table (PPT) and ADARES uses dynamic allocation to access them.

The same is true of the Work data set: it is specified in the session startup JCL. Session initialization writes the Work data set name to the PPT and, in the event of an autorestart, a nucleus is able to access the Work data set of a peer nucleus using dynamic allocation.

Dynamic allocation is currently available only in cluster environments (when the NUCID is greater than zero). However, dynamic allocation is used by a noncluster nucleus to determine that the previous nucleus session was a cluster session and PLOGs or CLOGs remain to be copied.

JCL Considerations

Files that are eligible for dynamic allocation by another database must be cataloged, rather than being accessed using explicit UNIT and VOLSER keywords in the JCL. If JOBCAT or STEPCAT DD statements are used to define a catalog search structure, these statements must be used in a consistent way on all database startup jobs in the sysplex.

6 Component Summary

Component	Number
Entire Net-Work	One per operating system image
ADASVC/SVCCLU	One per operating system image
ADACOM	Zero, one, or more per sysplex; required for each operating system image that has cluster users but no cluster nuclei; optional but recommended for operating system images that have cluster nuclei.
PPT	One per cluster; located in the database Associator
Adabas DBIDs	One per cluster; a practically unlimited number per sysplex
sysplex cache structures	One per cluster; specified in each nucleus
sysplex lock structures	One per cluster; specified in each nucleus
XCF group names	One per cluster; specified in each nucleus
Adabas nuclei	Up to 32 per cluster

7 Adabas Cluster Services and Entire Net-Work

- XCF Line Driver 27
- Summary of Entire Net-Work Features 28

Entire Net-Work is used to send Adabas and Adabas Cluster Services commands back and forth between operating system images. No changes have been made to Entire Net-Work to accommodate Adabas Cluster Services.

Although ADACOM and several Adabas cluster nuclei could run within a single operating system image without Entire Net-Work, there could then be no application programs for that database in any other operating system image.

To support a cluster environment that includes more than one operating system image, a limited Entire Net-Work library is included as part of Adabas Cluster Services.

One copy of Entire Net-Work is installed on each operating system image participating in the sysplex. The configuration for a given system comprises

- an Entire Net-Work control module;
- control module service routines; and
- the XCF line driver.



Note: For existing users of mainframe Entire Net-Work, the XCF line driver may be optional. See the section *What to Install* in the *Installation* documentation for more information.

Each system with Entire Net-Work installed becomes a "node" in the network. Each node's links to other adjacent nodes are defined by name and driver type.

Application requests are automatically routed; the application is totally insensitive to the actual location of the operating system images.

Each Entire Net-Work node maintains a "request queue" for incoming requests that is similar to the command queue used by Adabas. It allows the node to receive Adabas calls from locally executing applications, which Entire Net-Work then dequeues and transports to the nodes where the requested services reside.

Each local Entire Net-Work node also keeps track of all active network services, and therefore can determine whether the user's request can be satisfied or must be rejected. If the request can be serviced, the message is transmitted; otherwise, Entire Net-Work advises the calling user immediately with response code 148, just as the Adabas router would do for a local database request.

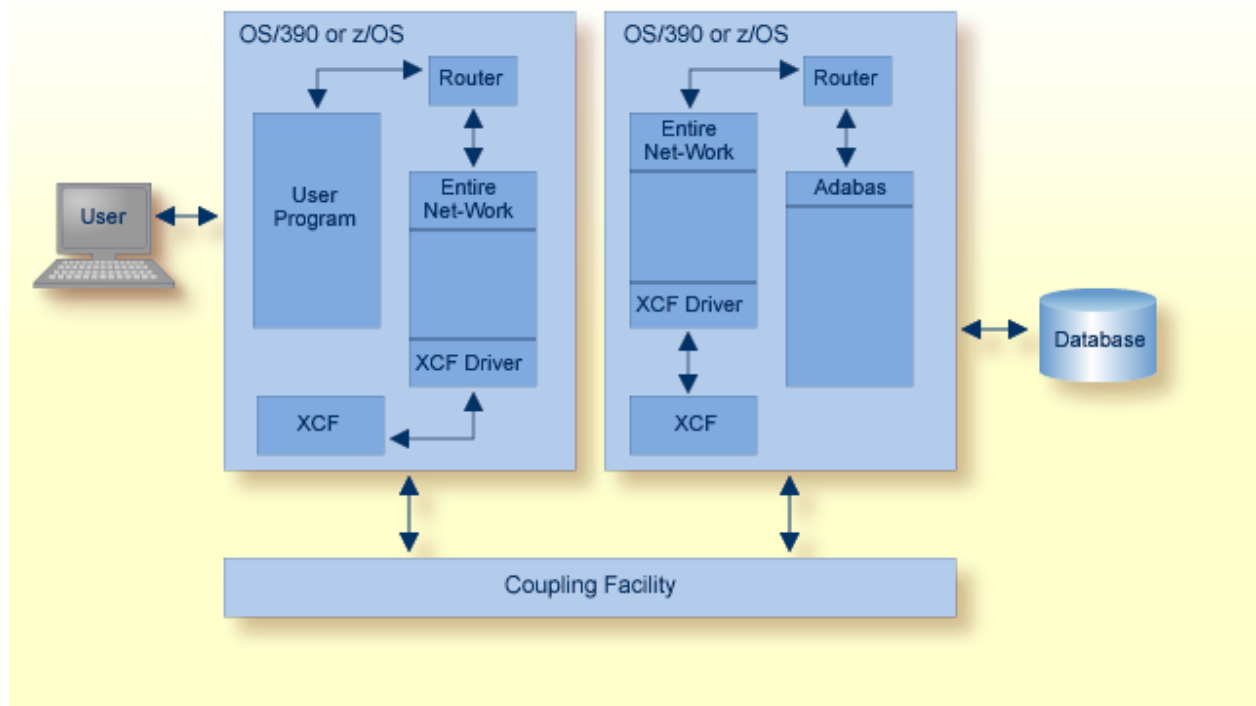
The chapter covers the following topics:

XCF Line Driver

Actual network data traffic is controlled by the Entire Net-Work XCF line driver, an interface to IBM's cross-system coupling facility (XCF) that allows authorized applications on one system to communicate with applications on the same system or on other systems. XCF transfers data and status information between members of a group that reside on one or more operating system images in a sysplex.

The XCF line driver, which is installed on each Entire Net-Work node, provides high performance, transparent communications between operating system images that reside on different central processors in the sysplex. Multiple connections to other nodes are supported, and the line driver's modular design permits easy addition of new access method support to the system.

A "member" is a specific function (one or more modules/routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one operating system image in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group. Each Entire Net-Work node running the XCF line driver is identified as a different member in a group specifically set up for Entire Net-Work connectivity.



Adabas Cluster Services with Entire Net-Work

The simple configuration shown works as follows:

- The user issues a call to Adabas.
- The call is presented to the local router by the Adabas link routine, which finds that the requested service is not available locally.
- Instead of returning the response code 148 to indicate "service not available", the router passes the call to the locally executing Entire Net-Work communicator.
- After accepting the call, Entire Net-Work determines whether the requested service is available on the network. If it is not available, the local Entire Net-Work communicator returns response code 148 to the user to indicate "service not available".
- If the target is active somewhere on the network, the call (including all required buffers) is packaged as a message and sent across the network to the target node by the XCF line driver. If the network topology allows, the message may be routed through many intermediate systems before it reaches its final destination.
- On the target node, the receiving Entire Net-Work communicator accepts the message and presents the contained call to the locally running service through the router.
- The target operating system image regards the Entire Net-Work call as equal to any other call issued from within its own local environment and returns any required reply in the normal manner through the local router.
- The router passes the reply information to the requestor, which is the local Entire Net-Work communicator, in the same way it passes reply information to a local request.
- The reply information is then packaged for the return trip to the originating node's Entire Net-Work communicator, which uses its local router to pass the reply to the user's request back to the application.

Summary of Entire Net-Work Features

In an Adabas Cluster Services environment, Entire Net-Work allows users on various network nodes to query a logical database across multiple operating system images. Users access an Adabas cluster database as they would a conventional, single-node database.

A request to Adabas can be made from within an existing application, without change. The request is processed automatically by the system; the logistics of the process are transparent to the application.

Entire Net-Work uses Adabas-dependent service routines for the operating system interface as well as for interregion communication, thus avoiding incompatibility. The similarity between Entire Net-Work and Adabas means that the job control statements for running Entire Net-Work are much like those needed to run Adabas. For example, the EXEC statement invokes the ADARUN program for Entire Net-Work just as it does for Adabas, and the ADARUN parameters for Entire Net-Work are a subset of Adabas parameters.

All targets and services establishing or terminating communication with the network cause status information to be broadcast to all nodes, eliminating the need to maintain or refer to database or target parameter files at a central location.

Allowing only one Entire Net-Work task on each node enforces control over the network topology by maintaining all required information in one place. This avoids confusion in network operation and maintenance. If, however, more than one Entire Net-Work task is required, this can be accomplished by installing additional routers.

Each Entire Net-Work node maintains only one request queue and one attached buffer pool for economical use of buffer storage. All buffers that are not required for a particular command are eliminated from transmission. In addition, only those portions of the record buffer and ISN buffer that have actually been filled are returned to the user on a database reply. Software AG still recommends that you set the correct buffer sizes in the control block when coding direct calls; otherwise, unnecessary data may be transmitted.

Buffer size support in Entire Net-Work is comparable to that in Adabas, ensuring that all buffer sizes that are valid for Adabas can also be transmitted to remote nodes. Software AG recommends that you always run Entire Net-Work with the LU parameter value of 65535; this has no impact on actual storage requirements.

8 Getting Started

- First Cluster Nucleus Starts 32
- Additional Cluster Nuclei on the Local Image 33
- Additional Cluster Nuclei on a Remote Image 33
- Images with Cluster Users But No Nuclei 33
- Environment Maintenance 34
- Assigning Users to Nuclei 34

Nuclei start "open" to new users on the local image; on a remote image, they start "closed" to new users on the local image. To modify the status of nuclei throughout the sysplex cluster, read the description of the ADACOM SN command in *SN - Set Nucleus Status*, in *Adabas Cluster Services Reference*.



Note: ADACOM must be active on the image to use this command.

NUCIDs and DBIDs must have unique values in the range 1 through 65000:

- the range 65001 through 65481 is used by the system to identify operating system images to the network and is therefore restricted;
- Software AG strongly recommends that you used the range 65482 through 65535 to uniquely identify Entire Net-Work nodes. See section *TARGETID - Entire Net-Work Target Identification* in the *Initialization Parameters* documentation.

For one operating system image that has cluster nuclei, an additional network target ID is generated that is equal to the cluster database ID. This operating system image "owns" the cluster database and is marked with an asterisk in cluster statistics displays. If all cluster nuclei on that image terminate, Adabas Cluster Services broadcasts the information and another image with active cluster nuclei acquires ownership of the cluster DBID.



Note: Users and commands coming from systems that are not members of the parallel sysplex (for example, through TCP/IP, a UNIX or PC system) are routed to the image that "owns" the DBID. In fact, this is also true for users coming from members of the sysplex where no ADACOM or cluster nucleus has initialized the sysplex environment.

The chapter covers the following topics:

First Cluster Nucleus Starts

When a nucleus with a nonzero NUCID starts on a z/OS operating system image, it determines whether the image has a sysplex environment active. If not, it sets up the environment:

- it sets an ID table entry (IDTE) to its NUCID in the ID table in CSA space. The NUCID must be unique across all images in the range 1-65000.
- it sets an entry to its NUCID in the nucleus table in ECSA.
- it obtains from Entire Net-Work a global target ID that is equivalent to the sysplex cluster's common DBID (unless that target ID is already active on another image). The DBID must also be unique across all images in the range 1-65000.
- it obtains from Entire Net-Work a target ID for the local image in the range 65001 and 65481.

Additional Cluster Nuclei on the Local Image

If another nucleus for the same sysplex cluster comes up on the local image, the global DBID and local image target IDs are already set. The starting nucleus only acquires an IDTE and a nucleus table entry and sets both to its NUCID.

Additional Cluster Nuclei on a Remote Image

If a nucleus for the same sysplex cluster comes up on another image, it also determines whether the image has a sysplex environment active. If not, it sets up the environment:

- It sets an IDTE to its NUCID. The NUCID must be unique across all images in the range 1-65000.
- It determines that a global target ID equivalent to the sysplex cluster's common DBID has already been assigned.
- It obtains from Entire Net-Work an unused target ID for the local image in the range 65001 and 65481.

This process is the same for all images starting nuclei for the sysplex cluster.

Images with Cluster Users But No Nuclei

On an image that has users accessing a sysplex cluster but no local nuclei, ADACOM must be executed at least once per IPL to set up the proper environment:

- it sets the address of the PLXCB (control block) in the IDTH (header).
- it obtains from Entire Net-Work an unused target ID for the local image in the range 65001 and 65481.

Software AG strongly recommends that you maintain the ADACOM in operation on such an operating system image so that it is available to reset the environment should Entire Net-Work terminate for any reason and restart.

Environment Maintenance

The following statements apply when Entire Net-Work remains active:

- As long as a cluster nucleus is active, its IDTE entry remains active.
- As long as there are nuclei active on any image for a sysplex cluster, the global target ID (equivalent to the cluster database ID) remains active on one of those images.
- As long as an operating system image is active, the local image target ID set for the cluster remains active.

If Entire Net-Work terminates for any reason and restarts, and ADACOM is not active on an operating system image that has users but no cluster nuclei, you must rerun ADACOM on the image to reset the environment. See the section [ADACOM](#) for more information.

Assigning Users to Nuclei

The SVCCLU assigns new users to local image Adabas cluster nuclei, attempting to balance the number of users assigned to each nucleus based on the current number of users.

If no local nucleus is available, the SVCCLU assigns new users to remote Adabas cluster nuclei that are open for users from this image, attempting to balance first the number of users assigned to each image and then the number of users assigned to each nucleus within the image.

A user routed to a remote nucleus is *not* rerouted when a local nucleus becomes available until the user closes the Adabas session and opens a new one.

9 Adabas Cluster Services and Other Software AG Products

■ Adabas Online System	36
■ Adabas Caching Facility	36
■ Adabas Delta Save Facility	36
■ Adabas Fastpath	37
■ Adabas Review	37
■ Adabas Transaction Manager	37
■ Adabas Vista	38
■ Adabas SAF Security	38

This section describes the interaction of Adabas Cluster Services with other Software AG products.

The chapter covers the following topics:

Adabas Online System

Adabas Online System communicates with all nuclei within the Adabas cluster and with all clusters in the sysplex environment. It includes functions related to Adabas Cluster Services.

Adabas Caching Facility

Adabas Caching Facility can provide a performance boost to the Adabas cluster. It augments the Adabas buffer manager by reducing the number of read "execute channel programs" (EXCPs) to the database so that the available operating system facilities can be used without monopolizing valuable virtual memory resources.

Adabas Caching Facility is required if you want to take advantage of 64-bit virtual storage in an Adabas environment.



Note: Write EXCPs are always issued to maintain the integrity of the database.

Adabas Delta Save Facility

Software AG's Adabas Delta Save Facility (DSF) offers significant enhancements to ADASAV utility processing by backing up and restoring only the changed (delta) portions of Adabas databases. It supports the Adabas Cluster Services environment.

DSF is intended for Adabas sites with one or more large, heavily updated databases that need to be available most of the time. Especially for sites where the volume of data changed on a day-to-day basis is considerably smaller than the total database volume, DSF provides for

- more frequent saves without interrupting database availability;
- enhanced "24 x 7 x 52" operation;
- full offline saving in parallel with the active database; and
- shorter REGENERATE duration during recovery.

Adabas Fastpath

Whenever possible, Adabas Fastpath reduces CPU consumption and increases throughput for Adabas systems by processing Adabas calls to completion in the client process. Database processing is thus avoided and capacity is increased for the whole operation.

Fastpath optimizes two types of database calls: direct access and sequential access. The results of direct access calls to the database are saved in a cache and repeat calls are satisfied from the cache rather than directly from the database. Read-ahead optimization routines are applied to sequences of commands to reduce redundant activity and thus accelerate the sequence.

Adabas Fastpath seamlessly optimizes database processing across multiple nuclei in one or more operating system images.

Adabas Review

Adabas Review monitors the performance of Adabas environments and the applications executing within them. You can use information retrieved about Adabas usage when tuning application programs to achieve maximum performance with minimal resources.

Adabas Review supports Adabas Cluster Services version 8.1.

Adabas Transaction Manager

The Adabas Transaction Manager (ATM), a selectable unit of Adabas, is a server for coordinating "distributed transaction processing" (DTP) in distributed Adabas environments. It manages "global" transactions that are distributed across multiple Adabas databases by coordinating changes to the databases in a seamless, integrated way, using a "two-phase commit" protocol when necessary.

At any time, ATM can account for in-flight transactions, suspect transactions, participating databases, and more.

ATM addresses two basic needs of the enterprise object revolution:

- the need to deliver industrial strength enterprise objects for widespread commercial use in mainstream, critical business systems.
- the need to spread the masses of data that Adabas customers manage more evenly across the computer(s) and organization.

ATM includes an online administration system based on Natural and available through Adabas Online System.

ATM provides limited support for Adabas Cluster Services. For more information, see the *Adabas Transaction Manager Version 8.1 Release Notes*.

Adabas Vista

Adabas Vista is used to partition an Adabas file into multiple, separate files, each containing a part of the original larger whole. Partitioning makes it possible to distribute a file across multiple volumes or computers based on a criterion such as region or date. Adabas Vista is also used to translate Adabas database and file numbers, which allows an application to remain independent of the underlying physical layer.

Adabas Vista supports Adabas Cluster Services in sysplex and CICSplex environments.

Adabas SAF Security

Adabas SAF Security enhances the scope of standard security packages based on the System Authorization Facility (SAF) such as RACF, CA-ACF2, and CA-Top Secret to encompass Adabas resources. It integrates Adabas into a central security repository and enables you to derive maximum benefit from your investment in that repository.

Adabas SAF Security provides protection for Adabas resources in sysplex environments.

Index

A

- Adabas Caching Facility, 36
- Adabas Fastpath, 37
- Adabas Online System, 36
- Adabas Review, 37
- Adabas SAF Security, 38
- Adabas Transaction Manager, 37
- Adabas Vista, 38
- ADACNV
 - converting an existing database, 18
- ADACOM
 - component, 5
 - definition, 5
- ADACSH, 36
- ADADEF
 - defining a new database, 18
- ADARES
 - merging logs, 13
- Autorestart, 22

B

- Buffer flush, 7
- Buffer pool and manager
 - local, 7
- Buffers
 - setting sizes, 29
 - use of attached, 29

C

- Cache structure
 - component, 7
 - size, 7
- CLOG
 - activating, 14
 - merging, 14
 - synchronized switching, 15
- CLUSTER
 - ADARUN parameter, 4
- Conversion
 - to Adabas Cluster Services 7.2, 18
- Coupling facility
 - channels, 2
 - definition, 2

D

- Database
 - single physical
 - defined, 5
- Direct calls, setting buffer sizes for, 29
- Dynamic allocation, 22

E

- Entire Net-Work
 - component, 26
 - operation with Adabas Cluster Services, 25
 - request queue, 26
 - summary of features, 28
 - XCF line driver, 27

J

- JCL
 - like Adabas, 28

L

- Lock
 - accessibility, 8
 - ownership, 8
 - sequencing, 8
- Lock manager, 8
- Lock structure
 - component, 8
 - size, 8
- LU parameter
 - recommended setting, 29

M

- Messaging services, 9

N

- NUCID
 - ADARUN parameter, 4
- Nucleus
 - component, 4
 - components, 4
 - number per z/OS image, 4
- Nucleus cluster

- components, 4
- defined, 4
- maximum number of, 5

O

- Operating systems
 - supported, vi

P

- PLOG
 - merging, 13
 - requirements, 12
 - synchronized switching, 15
- PPT
 - definition, 18

R

- Remote processing, 28
- Request queue, 29

S

- Service not available (148) response, 28
- SVCCLU
 - component, 6
- Sysplex
 - definition, 2

U

- Users
 - assigning to a nucleus, 34

X

- XCF group
 - component, 9
- XCF line driver, 27
- XCF member
 - definition, 27