Adabas System Coordinator

Adabas System Coordinator Introduction

Version 8.1.2

June 2014

Adabas System Coordinator

This document applies to Adabas System Coordinator Version 8.1.2.

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

Document ID: COR-INTRO-812-20140626

Table of Contents

Preface v
1 Adabas System Coordinator Benefits and Features 1
The Role of the Adabas System Coordinator 2
How the Adabas System Coordinator Evolved 2
Adabas Client Integration 5
Adabas Server Integration 5
Additional Daemon 5
Consistent Runtime Environment 4
Online Administration and Configuration 4
Client Runtime Controls 4
Context Management 5
Performance and Reliability 6
Local Mode and Daemon Mode 14
Versioning Feature 7
2 Dynamic Transaction Routing 9
Benefits and Use 10
Clustered Applications 10
3 Adabas System Coordinator Components 17
Client and Server Components 18
Daemon Component 18
Administration Component 19
Configuration File 19
Node Error Program 19
Plug-in Service Routine 20

Preface

This document provides an introduction to the Adabas System Coordinator.

The following topics are provided:

Adabas System Coordinator Benefits and Features
Dynamic Transaction Routing
Adabas System Coordinator Components

1 Adabas System Coordinator Benefits and Features

- The Role of the Adabas System Coordinator 2
- How the Adabas System Coordinator Evolved 2
- Adabas Client Integration 5
- Adabas Server Integration 5
- Additional Daemon 5
- Consistent Runtime Environment 4
- Online Administration and Configuration 4
- Client Runtime Controls 4
- Context Management 5
- Performance and Reliability 6
- Local Mode and Daemon Mode 14
- Versioning Feature 7

This section provides an overview of the benefits and features provided by Adabas System Coordinator.

The Role of the Adabas System Coordinator

Adabas System Coordinator provides infrastructure technology for the optional Adabas Fastpath, Adabas Vista, Adabas SAF Security and Adabas Transaction Manager features, thereby enabling them to function in the most efficient manner possible. These optional features have many things in common. Architecturally they all have footprints in the Adabas client and the Adabas server. The functionality provided by these options is very much focused on the client. In very basic terms the functionalities of each product are:

- *Adabas Transaction Manager (ATM): Distributed Transaction Integrity.* ATM monitors all client transaction activity and generates transparently the processing required to ensure distributed transaction integrity. The application remains unchanged.
- *Adabas Vista (AVI): Data Partitioning.* AVI monitors all client activity and re-directs transparently Adabas processing according to the defined partition distribution and translation rules. The application remains unchanged.
- *Adabas SAF Security (AAF): Data Protection.* AAF monitors all client activity and ensures transparently that processing rules defined in RACF (etc.) are used to protect data managed by Adabas. The application remains unchanged.
- *Adabas Fastpath (AFP): Data Access Optimization.* AFP monitors all client activity and optimizes transparently processing according to the defined rules. The application remains unchanged.

How the Adabas System Coordinator Evolved

The Adabas System Coordinator first appeared with version 7.3.1, but the technology actually existed for quite a while before that. This technology originally appeared as part of the Adabas Fastpath releases prior to version 7.1. At the same time, similar technology appeared in Adabas Vista, but it was not engineered in exactly the same way, as is usually the case with duplicated effort. This resulted in inconsistency and error in addition to extended engineering cycles.

With version 7.1, Software AG decided it was time to make a common technology framework so that the existing and emerging options could reuse as much technology as possible. This saw the introduction of largely invisible software that we called the common runtime environment. This was a major evolutionary step on the way to introducing the Adabas System Coordinator. Technology from Adabas Fastpath was taken and enhanced so that it could be shared by more than one option. Both Adabas Fastpath and Adabas Vista made extensive use of the common runtime environment at the version 7.1 level.

The Adabas System Coordinator technology officially appeared at version 7.3, when Adabas SAF Security made use of it for the first time in addition to continued and extended use by Adabas Fastpath and Adabas Vista. With version 7.4, the technology was developed even further and the interfaces were fortified to achieve even more efficiency. Adabas Transaction Manager version 7.4 now makes extensive use of the technology, as well.

Adabas Client Integration

All of the Adabas options interface to the Adabas link module (Adabas proxy) in the client process. The Adabas proxy represents Adabas in the client space to make it easier for applications to use Adabas by simply calling the local Adabas proxy program. The Adabas proxy handles communication with Adabas servers running in the Software AG network.

There are many types of Adabas clients, including Com-plete, CICS, batch, TSO, IMS/DC, CMS, UTM, and TIAM. There are also many types of operating system such as z/OS, VSE/ESA, BS2000 and z/VM. The original and basic reason for evolving the Adabas System Coordinator was to produce common technology that resolves the technical challenge of running in all of these environments. This avoids duplication of complex technology in each of the Adabas options described above.

Adabas Server Integration

All of the Adabas options also interface to the Adabas server. Again, the Adabas System Coordinator provides common technology to eliminate duplication in the options. There are many internal interfaces with Adabas that are implemented in the Adabas System Coordinator, including initialization, before command processing, after command processing, and PLOG write. The options are insulated from these interfaces, which improves reliability.

Additional Daemon

Some of the Adabas options require a daemon of their own (a server process that is not an Adabas database server). For example, Adabas Fastpath requires a daemon to act as the Adabas Fastpath Asynchronous Buffer Manager.

The Adabas System Coordinator provides a daemon environment in which the options can be executed. Again, this common technology insulates the products from the complex technology that is required to operate an effective daemon. In addition, duplication of technology is avoided. The Adabas Fastpath Asynchronous Buffer Manager runs as a service within the Adabas System Coordinator daemon. There is another common service provided to support dynamic transaction routing as described in the section **Dynamic Transaction Routing**.

Consistent Runtime Environment

The Adabas System Coordinator provides a consistent runtime environment for the Adabas options. No matter where an option operates it is always housed in an identical system coordinator environment. The products do not have to be concerned with the differences between running in batch, CICS, an Adabas server or a daemon service.

This simplifies the engineering of the options immensely. The options are developed to use system coordinator interfaces, not interfaces to batch, CICS, Adabas, etc. All these complexities are removed so the engineering of the options can focus solely on providing functionality rather than managing such factors as the environment or communication. Over time, this brings increasing benefits to the sites that use these options.

Online Administration and Configuration

A design goal for all Adabas optional features as well as the Adabas System Coordinator is to provide the administrator with total control of the software operation from an online command center (administration tool). There is a deliberate focus on trying to avoid the need for control card inputs, JCL options, or parameter modules. In that it is not always possible to achieve 100% online operation, there are a few bootstrap configuration settings required but in essence the administrator can use the Adabas System Coordinator Online Services application for online administration and monitoring.

Client Runtime Controls

The Adabas options and Adabas System Coordinator allow configuration by runtime controls through the administration tools. These can often be left to take their default values, and the defaults can be overridden for specific jobs when necessary. Consequently, the operation of specific jobs can be controlled remotely without having to gain access to the JCL. Furthermore, it is not necessary to install different various options in different libraries for use by jobs with different JCL. This provides for very flexible operational management.

Additionally there is increasing need for Adabas client sessions to operate differently within the same job. For example:

- Client ABC in CICSXYZ needs special tracing controls to be in use, all other clients do not
- Transaction D412 in CICSXYZ must be able to operate with a lower timeout limit than other transactions
- Stepname S0010 in job PRODA032 must be excluded from using the Adabas System Coordinator

This level of runtime control is becoming extremely important. For example, tracing options can be directed at a very few sessions rather than globally. This can mean overall memory consumption can be kept to a minimum while at the same time aggressively pursuing a problem investigating for only the sessions to be scrutinized.

Adabas System Coordinator Version 8.1 allows these configuration controls to be prescribed in advance by adding optional override controls to the original base job level controls. It is possible to preconfigure overrides as follows:

- Batch job:
  - 1. Stepname
  - 2. LOGIN (for example, RACF LOGIN userid)
  - 3. Special API
- TSO, CMS, TIAM, etc
  - 1. Special API
- COM-PLETE, CICS, IMS, UTM
  - 1. Special API
  - 2. LOGIN
  - 3. Transaction code

For example, as a terminal operator moves from one transaction to another the runtime behaviors will alter dynamically according to what is prescribed in the configuration file. In addition to being able to pre-set the different configurations to be adopted at runtime it is also now possible to dynamically change the runtime controls for your "current" session. So, you may decide to switch tracing on or off, for example, regardless of what is prescribed in the configuration file.

Context Management

The Adabas System Coordinator provides many services to the Adabas options. Overall, the most fundamental one is client context management. As stated previously, the functionality of all these options is very much oriented toward the Adabas client. At runtime, each option must maintain status information in memory about each Adabas client session in the client job (process).

A search for client context is performed whenever an option detects an Adabas command. Without the Adabas System Coordinator, each option would have to perform context search and management. With the Adabas System Coordinator, there is one context search and management service that shields the options from all this complexity. This is one of the ways in which the Adabas System Coordinator provides great benefit. This context management has also enabled the support for **dynamic transaction routing**.

Performance and Reliability

The Adabas System Coordinator adds logic to the Adabas client, so it is understandable to assume that it brings with it some performance degradation. But this warrants further examination. For example, if the Adabas System Coordinator is used to house the Adabas Fastpath option, the chances are that performance is improved, not worsened. This obviously depends on the levels of optimization gained by the Adabas Fastpath functionality used and the relevance of the rules put in place.

The performance profile changes when Adabas Vista is used with the Adabas System Coordinator. The use of Adabas Vista functionality introduces an unavoidable overhead as the price of additional functionality. The Adabas System Coordinator helps to minimize this by providing highly tuned services, but there is nevertheless an overhead. If both Adabas Fastpath and Adabas Vista are used together with the Adabas System Coordinator, both benefit from a single context search so the combination of multiple options means that any overhead expected by an option on its own is lessened by the amount of shared facilities. This is also true if Adabas Transaction Manager is added.

Each Adabas System Coordinator service is shared by all of the Adabas options. Therefore, in the future it may be decided to enhance various services to be more efficient. This will benefit multiple options at the same time. And, the reliability of each shared service helps to improve the reliability of all the options simultaneously. These are just some of the benefits of using a common technology framework.

The Adabas System Coordinator Online Services tool can be used to set job parameters and to obtain feedback from a running operation. For example, the list of Adabas clients being processed at the moment can be viewed to review certain statistics, such as the amounts of memory being used. These online facilities can prove very useful for locating and resolving performance problems.

Local Mode and Daemon Mode

The Adabas System Coordinator can be used in local or daemon mode by a job. The default is local mode. In local mode memory is allocated by the coordinator from within the job (process, region, partition, address space, etc.) in which it is running.

Alternatively, the administrator can configure a job to run in daemon mode. Obviously, this requires an Adabas System Coordinator daemon to be running in the same operating system image as the client job. In daemon mode the coordinator logic in the client job arranges for all context related memory to be allocated through the daemon. The daemon allocates this from shared memory.

By using daemon mode for all jobs it is possible to use the administration tool to obtain feedback from all jobs in the system simultaneously. This is referred to as “single seat administration”. This

is one reason for using daemon mode. In local mode the feedback for a job can only be viewed from within that job since the allocated memory is only available within that job.

Daemon mode is also required for dynamic transaction routing as described in the next section.

Versioning Feature

Introducing a new release of base Adabas collection, or the client-based add-on collection of products can be a big challenge in complex IT sites. Many sites will opt to update the base Adabas products separate from the client-based products to simplify the scope of the project and manage risk. More and more we have seen sites with stringent change-management that require you to perform implementation of new client-based releases in a very gradual, controlled fashion. This allows the switchover from one release to another to be managed job by job, client by client, database by database.

The Versioning Feature of Adabas System Coordinator enables this fine-grained ability to perform upgrades in a very stealthy, managed way that provides great benefits to your goals for continuous operation. By using the versioning feature you can:

- Run two releases of client-based products within an Adabas database
- Run two System Coordinator daemon versions in the same system-image
- Run client jobs that use different releases

Consequently, you are able to convert one client job at a time, step by step, until all clients are running the new release. At that point you decommission the daemon running for the old release and you decommission the old release within your databases.

2 Dynamic Transaction Routing

- Benefits and Use 10
- Clustered Applications 10

This section describes the dynamic transaction routing feature provided by Adabas System Coordinator:

Benefits and Use

Many TP systems can operate with a collection of jobs working together to provide a single TP service. Once this ability is available it becomes logical to load-balance client sessions across these jobs. The simplest implementation of this is to allocate new sessions to jobs on a “round-robin” basis. However, this does not cater to clients that are doing more work than others at varying times of the day.

The most effective way to load balance is to employ dynamic transaction routing. Algorithms are implemented into the TP system to move sessions from job to job according to availability and demand, thereby gaining the best throughput performance and service. This means that client sessions may be moved from one job to another at any time.

Using Adabas System Coordinator daemon mode makes it possible for Adabas client sessions to operate in a dynamic transaction routing environment. The client context for the options is made available automatically to all jobs in the service by the Adabas System Coordinator.

Clustered Applications

There is a standard service always running in the Adabas System Coordinator daemon called the Clustered Application Service (CAS). It is this service that provides support for dynamic transaction routing (daemon mode). In Adabas System Coordinator terms any TP service that is able to perform transaction routing is called a clustered application. The “cluster” refers to the fact that multiple jobs act together as members of one TP service.

Note: Clustered applications are not supported in z/VM systems.

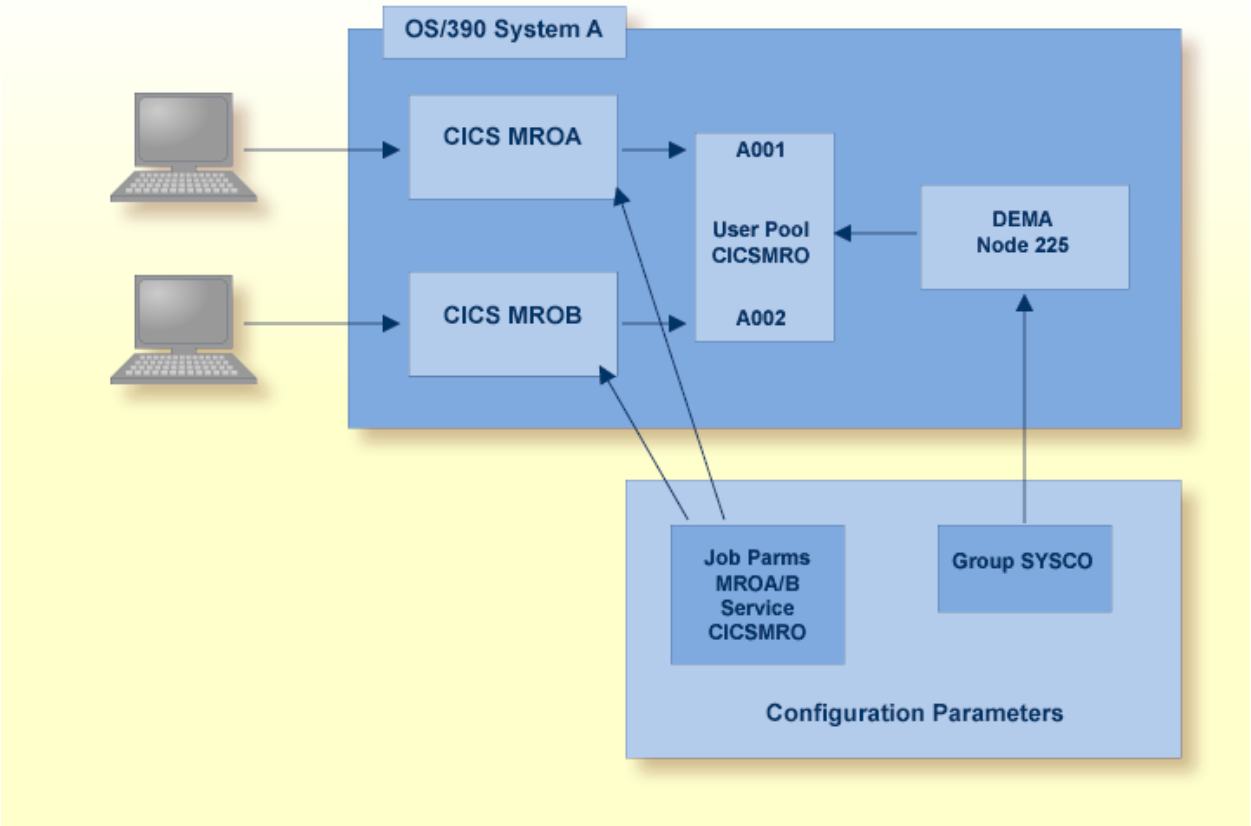
Clustered Applications in a Single Operating System Image

The following TP systems provide dynamic transaction routing across multiple jobs within a single operating system image:

- CICS/ESA or CICS Transaction Server for z/OS running with MRO (multi-region option)
- CICS for VSE/ESA or CICS Transaction Server for VSE/ESA running with MRO (multi-region option)
- IMS/ESA
- Universal Transaction Monitor (UTM) for BS2000

Configuration Example: CICS Cluster in a Single Operating System Image

The following graphic illustrates a CICS (MRO) cluster of two jobs running in a single system image. Terminal users (A001, A002) can be scheduled in any region and dynamic transaction routing can occur at any time.



The Adabas option job parameters for the CICS regions CICSMROA and CICSMROB exist and reference the System Coordinator group SYSCO as responsible for enabling dynamic transaction routing. Both job parameters reference the service name CICSMRO, which enables dynamic transaction routing across the two regions.

The System Coordinator group has one member called DEMA defined to run using the node 225 in the Software AG network. The daemon DEMA must be started before any Adabas activity is generated in the CICS service.

Clustered Operating Systems

A clustered operating system connects multiple operating system images together to work as a single system image. The IBM parallel sysplex is an example of a clustered operating system. A clustered operating system improves throughput by providing more parallel capacity and more resilience for component failure.

A clustered operating system requires software that takes advantage of its cluster features. Adabas Cluster Services is used to access a single database resource from multiple Adabas nuclei operating on multiple operating system images coupled together in a sysplex cluster. Adabas Cluster Services balances loads across the nuclei to increase throughput and availability.

Clustered Applications in a Clustered Operating System

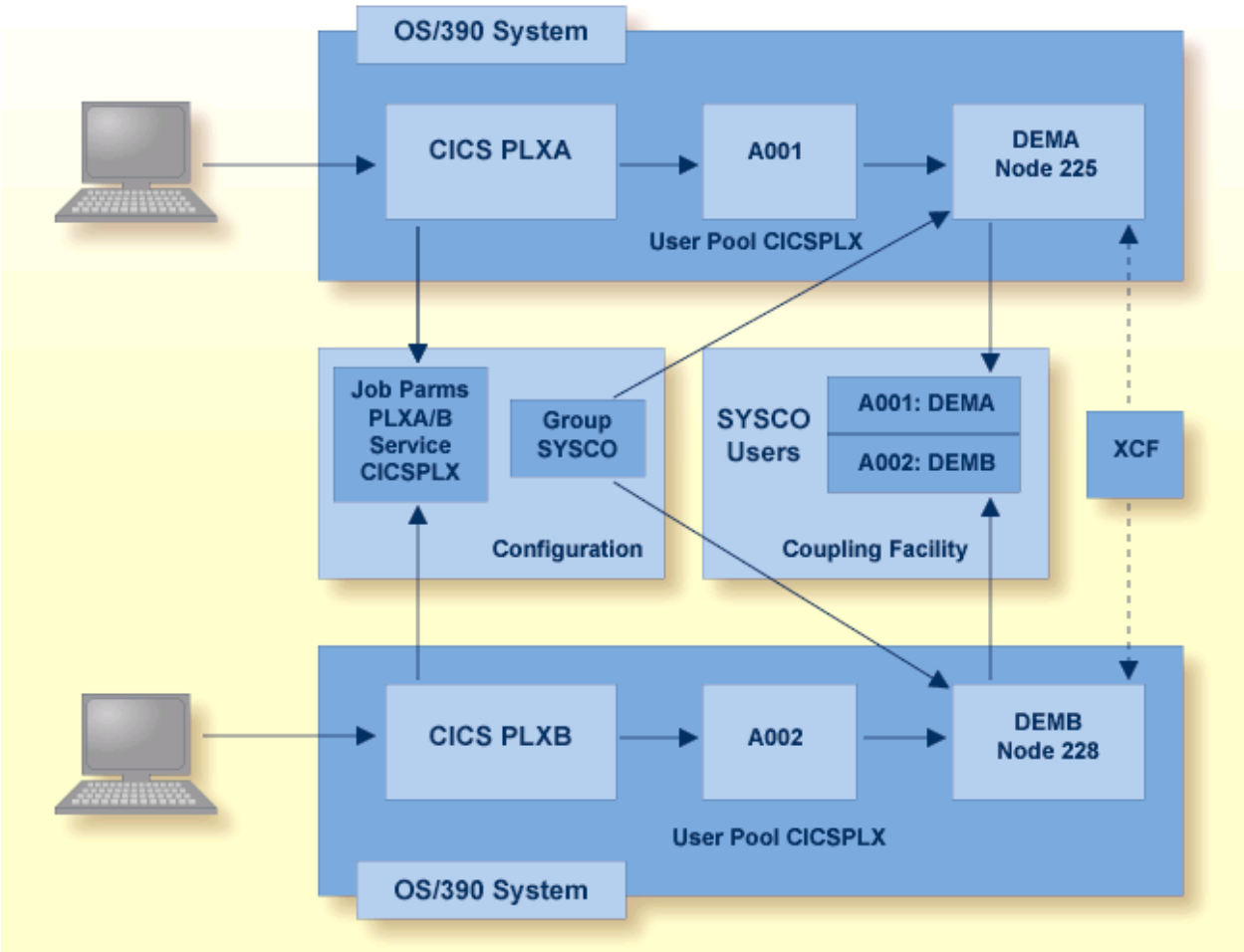
There are TP services that support dynamic transaction routing across multiple operating system images such as CICS in a Parallel Sysplex. In these systems one Adabas System Coordinator daemon must run in each of the operating system images. The daemons work together as members of a peer group. If dynamic transaction routing is required throughout an IBM Parallel Sysplex, the Adabas System Coordinator daemons in the group must all share resources in the IBM Coupling Facility as described later in this section.

When a client session is routed from the domain (one system image) of one Adabas System Coordinator daemon to another, the Clustered Application Service (CAS) in each daemon negotiates the transfer of client context from one operating system image to the other. This occurs completely transparently to the running application.

Currently, transaction routing across operating system images is only supported in an IBM sysplex environment using IBM coupling facility features in a cache structure. An Adabas System Coordinator group must be defined and implemented in order to support this level of dynamic transaction routing.

Configuration Example: CICS Cluster in an IBM Parallel Sysplex

The following graphic illustrates a CICSplex cluster of two jobs running in an IBM parallel sysplex. Terminal users can be scheduled in any region and can migrate at any time.



The Adabas option's job parameters for the CICS regions CICSPLXA and CICSPLXB exist and reference the System Coordinator group SYSCO as responsible for managing dynamic transaction routing. Both job parameters reference the service name CICSPLX, which becomes the name of the user memory pools on both systems and enables routing across the CICSplex.

The System Coordinator group has two members called DEMA and DEMB defined to run using the nodes 225 and 228, respectively. The daemons DEMA and DEMB must be started before any Adabas activity is generated in the CICSplex.

The System Coordinator group definition identifies the coupling facility cache structure to be used.

Adabas System Coordinator Group, Member and Service

- Group
- Member
- Service

Group

An Adabas System Coordinator daemon must be defined as a member of a System Coordinator Group with an eight-character group name (see Online Services, Maintain Daemon Groups). In a single operating system image there will normally be just one daemon member in the group. In a clustered operating system there will be one daemon member for each system image in the group. In an IBM Parallel Sysplex implementation the group name is also used as the XCF group name.

Jobs and TP services that are to be run in daemon mode identify the daemon group name to be used, rather than a specific daemon member name.

Member

The daemon member name is the job or task name by which the daemon service is known to the operating system.

Service

A single daemon member can provide DTR facilities to multiple TP services. Each discrete TP service (Production, Test, etc) must be identified with a unique service name within the daemon member. This provides isolation of one service from all others.

When the TP service starts up it registers its presence with the daemon. A shared memory pool is created for the service, for holding user context data. The service name is used for the pool name.

Identifying Clustered Applications to the System Coordinator

Support for dynamic transaction routing in a clustered application must be indicated in the job parameter for the Adabas option used. For example, to implement Adabas Fastpath, Adabas Vista, and Adabas Transaction Manager and support dynamic transaction routing in a CICS/MRO system, it is necessary to:

- create a job parameter using online administration services for one of the Adabas options,
- identify the job parameter as type “CICS cluster” rather than the standard CICS, and
- identify, as part of the job parameter, the Adabas System Coordinator group and service name to be used.

Adabas Fastpath, Adabas Vista, and Adabas Transaction Manager share the same job parameter in the same system configuration file. When the job type and system coordinator group are set in one Adabas option, they are used automatically by the others. Specifying a group and service name automatically identifies the job as a clustered application. When the job starts, it attempts to contact its local Adabas System Coordinator daemon to register its presence. All users that log on and use the application are also registered.

The Adabas System Coordinator daemon maintains user information in system shared memory such as ECSA in z/OS, not in local memory. The specific memory requirement differs depending on the activity level of the application and the Adabas options that are installed. To optimize the use of shared memory, which is a limited resource, Software AG recommends that jobs be defined as clustered applications only if dynamic transaction routing is to be supported in the job. It is still possible to administer groups from a job such as a TSO user that is not actually defined to an Adabas System Coordinator group.

Note: Adabas Fastpath always uses an Adabas System Coordinator to manage its asynchronous buffer manager (ABM). When defining job parameters for Adabas Fastpath, however, it is only necessary to specify a System Coordinator group for clustered applications.

3 Adabas System Coordinator Components

- Client and Server Components 18
- Daemon Component 18
- Administration Component 19
- Configuration File 19
- Node Error Program 19
- Plug-in Service Routine 20

This section describes the main Adabas System Coordinator components:

Client and Server Components

The Adabas System Coordinator client component is bound with the Adabas client link module during installation for use in batch, CICS, z/VM, UTM, etc. It functions as the interface between the link module and the installed Adabas options. It provides a common execution environment under which all options run, and a common set of system-dependent services that are used by all options.

The server component, ADAPOP, is loaded by the Adabas server when required. As the Adabas database interface component, it monitors various database functions and makes them known to the Adabas options that may have an interest in them. Like the client component, the server component provides a common execution environment under which all options run.

The common execution environment and common services are packaged in a single kernel module CORKRN that is used by both the client and server components.

The client and server components must be installed if any of the following Adabas options are to be used:

- Adabas Fastpath Version 8.1
- Adabas Vista Version 8.1
- Adabas SAF Security Version 8.1 (only requires the server component)
- Adabas Transaction Manager Version 8.1

Daemon Component

The Adabas System Coordinator daemon SYSCO is used by most installations, especially those using clustered applications, databases, or operating systems such as the IBM sysplex environment. The daemon component executes various services for the Adabas options such as the Adabas Fastpath Asynchronous Buffer Manager and the Clustered Application Service (CAS). The daemon component also uses the common services packaged in the kernel module CORKRN.

Administration Component

The Adabas System Coordinator Online Services tool (SYSCOR) is a Natural application which is used to administer the Adabas System Coordinator and the associated Clustered Application Service (CAS) by

- entering runtime controls for Adabas System Coordinator jobs and groups; and
- viewing active runtime information.

Configuration File

Runtime controls are maintained in a configuration file that is held in an Adabas database. The configuration file is shared with Adabas Fastpath, Adabas Vista and Adabas Transaction Manager.

The configuration file contains:

- Runtime controls for client jobs and TP systems using Adabas System Coordinator and/or Adabas Fastpath, Vista, and Transaction Manager executing in client jobs and TP systems
- Runtime controls for the Adabas System Coordinator daemon

The configuration file is now a vital part of the runtime operation. As such it can become a single point of failure. Version 8.1 now allows you to nominate a primary and an alternate configuration file. Each session will attempt to use the primary and if it is unavailable the alternate will be used if it is nominated. Once a configuration file has been identified for a session that file will continue to be the primary file for that session until it becomes unavailable, and then the other file will be used. Consequently, over time different sessions may be using different files at the same time until you forcibly cause all sessions to switchover by making one or the other unavailable for a long period. If an alternate Configuration file approach is used then both files must be available at both Coordinator daemon startup and shutdown. This is necessary because the same recovery/re-start information must be placed in both files so they do not get out of step.

Node Error Program

The node error program CORNEP is used by sites running CICS command–level applications in CICS/ESA, CICS Transaction Server for z/OS, CICS for VSE/ESA, or CICS Transaction Server for VSE/ESA.

CORNEP is not an essential component, but it does provide efficient memory reclamation for user sessions that terminate without releasing precious memory resources.

Important: Use of CORNEP requires modification of your installation CICS DFHZNEP program. CORNEP must only be called from DFHZNEP.

Plug-in Service Routine

The Adabas error handling and message buffering facility helps implement 24*7 operations by analyzing and recovering from certain types of errors automatically with little or no manual intervention. It also generates additional information so that the error can be diagnosed. See the *Adabas DBA* documentation for more information.

To work within this feature, the Adabas System Coordinator delivers a plug–in service routine PINCOR, which is established automatically when the Adabas System Coordinator server component (ADAPOP) initializes at nucleus startup.

If a program interrupt occurs in the Adabas System Coordinator server component, control is passed to PINCOR, which formats and prints the main memory areas used by the component.

These diagnostics are written to the DDPRINT dataset with the following title:

COMMON RUNTIME - memory–area–name : SNAP BY SMGT

PINCOR then returns control to the error handling and message buffering facility so that Adabas can terminate abnormally.