

Adabas System Coordinator

Adabas System Coordinator Introduction

Version 8.2.2

March 2012

Adabas System Coordinator

This document applies to Adabas System Coordinator Version 8.2.2.

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

Copyright © 2012 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

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: COR-INTRO-822-20120329

Table of Contents

1 Adabas System Coordinator Introduction	1
2 Features	3
Advanced control of the Adabas client environment	4
Dynamic control of the Adabas client environment	4
Display current activities	4
Client session latency management	5
Crash-recoverable client session latency management	5
Client session latency timeout	5
Adabas network display, discovery and tasks	5
Console message management	6
Automatic retry processing	6
Debug event diagnostic reports	6
Unified tracing for sibling products	6
Refresh executables	6
Adabas System Coordinator components	7
3 Client Component	9
4 Database Component	11
5 Group Component	13
6 Administration	15
Administration	16
Configuration	16
Operation and Performance	18
Versioning	19
Node Error Program	20
Plug-in Service Routine	20

1 Adabas System Coordinator Introduction

This document provides an introduction to the Adabas System Coordinator.

Adabas System Coordinator provides infrastructure technology for the optional Adabas Data Archiving, Adabas Fastpath, Adabas Vista, Adabas SAF Security and Adabas Transaction Manager features, thereby enabling them to function in the most efficient manner possible.

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.

The Adabas System Coordinator contains three major components (client component, database component, group component). Depending on how you use Adabas System Coordinator, you may need only one, two or all three of these components.

- **Features**
- **Client Component**
- **Database Component**
- **Group Component**
- **Administration**

2 Features

- Advanced control of the Adabas client environment 4
- Dynamic control of the Adabas client environment 4
- Display current activities 4
- Client session latency management 5
- Crash-recoverable client session latency management 5
- Client session latency timeout 5
- Adabas network display, discovery and tasks 5
- Console message management 6
- Automatic retry processing 6
- Debug event diagnostic reports 6
- Unified tracing for sibling products 6
- Refresh executables 6
- Adabas System Coordinator components 7

Advanced control of the Adabas client environment

Normally, the only way to control the runtime operation of Adabas clients is to modify JCL. This of course assumes that you know where the JCL is for the many hundreds of thousands of JCL pieces that exist in your site; and that you have permissions to access and/or modify them. The disparate nature of JCL often results in the loss of control, as opposed to the gain of it.

Adabas System Coordinator provides advanced control over the runtime processing of Adabas clients; ranging from things like control over activity displays to control of debug processing, for example. These controls are applied independently of JCL at runtime so the administrator can manage the entire Adabas client environment from an online administration tool without needing to access the many JCL objects that exist.

Dynamic control of the Adabas client environment

The advanced administrative control of the client runtime environment described above is applied from static configuration that you define online. This is excellent because these controls are applied out of hours as well as when the administrative systems are in use. However, further advanced control is needed for clients that are currently running; they run from the static controls initially but there may be reason to adjust controls dynamically, in an emergency for example. System Coordinator allows you to single out clients in any job (TP, batch, etc) and to alter the controls of individual sessions in your live system.

Display current activities

System Coordinator provides a live on-screen information feed about the current activities of Adabas client jobs and sessions. You are able to see every Adabas client running in your systems through these displays. You can do this from a single display session; your display session does not have to be with the same job as the clients that you are watching! You can see the activities of any Adabas clients you wish from one login.

Client session latency management

System Coordinator is a plug-in framework for sibling products such as Fastpath, Vista, SAF Security and Transaction Manager. Part of the framework performs latency management for Adabas client sessions. Latency is where a client session is at rest between terminal interactions, for example. At these times it is important that precious TP system resources reclaimed, such as memory, so they are not tied up while the terminal is temporarily dormant. Terminals are latent during user think-time for example. This latency can last for a few seconds up to many minutes or even hours; this is why careful latency management is so important. System Coordinator has options you can use to allow latency management to be carried out locally within the TP system or out to disk, etc.

Crash-recoverable client session latency management

As stated previously, latency management can be configured to use disk. This means latent TP sessions become crash-recoverable so that if you use static or dynamic transaction routing to accomplish failover within or across systems the Adabas client sessions managed by System Coordinator become crash-recoverable too!

Client session latency timeout

Part of latency management allows you to set timeout for excessively dormant client sessions. This allows resources to be recovered in a timely fashion.

Adabas network display, discovery and tasks

System Coordinator is aware of the surrounding Adabas network and provides display of the active nodes within it, across the System Coordinator group. In addition, various tasks can be performed on some nodes in the network, such as Adabas databases or Coordinator daemons. Some of these tasks may require the presence of a sibling product but others are provided directly by Coordinator. This allows you to discover the network and it allows you to see the network through the eyes (perspective) of other Coordinators in the network too.

Console message management

Console messages produced by Coordinator and by sibling products can be shown on the console, as expected. Alternatively, console messages can be redirected to a dataset.

Automatic retry processing

You can set automatic retry with optional pause between retries for various response codes. For example, retry for database down can avoid jobs failing so that they tolerate this and automatically retry repeatedly over a period until the database becomes available again. Or, contention for holding records can be retried rather than causing jobs to fail.

Debug event diagnostic reports

You can set automatic diagnostic reporting for various error situations. This is especially useful where a problem arises intermittently or out of hours. The diagnostic reporting happens automatically at the time the error occurs, without human intervention.

Unified tracing for sibling products

Another feature for helping to debug system or application problems allows a unified trace to be recorded for individual sessions. This allows all the sibling products to report their trace information into the same place which give a much more coordinated picture of what is happening at the time the trace is taken.

Refresh executables

You can instruct System Coordinator, and siblings, to re-load executable code dynamically. This allows you to introduce fix maintenance without having to stop systems for example.

Adabas System Coordinator components

The Adabas System Coordinator contains three major components client, database and group (represented by a daemon). Depending on the sibling products you use and how your site is configured you may need one, two or all three of these components. There is a fourth online administration tool too.

3

Client Component

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.

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 Complete, 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.

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.

4 Database Component

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.

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.

5

Group 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.

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.

6 Administration

- Administration 16
- Configuration 16
- Operation and Performance 18
- Versioning 19
- Node Error Program 20
- Plug-in Service Routine 20

Administration

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.

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

- [Configuration File](#)
- [Runtime Controls](#)

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 and above 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/restart information must be placed in both files so they do not get out of step.

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 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-LETE, 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.

Operation and Performance

- [Operation](#)
- [Performance](#)

Operation

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 managed by the daemon, enabling single- and multi-system dynamic transaction routing.

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.

Performance

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.

Versioning

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.

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.