Adabas System Coordinator Benefits and Features

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.