♫ software AG

Adabas Transaction Manager

Adabas Transaction Manager Configuration

Version 8.2.2

March 2013

Adabas Transaction Manager

Table of Contents

Preface

This document provides information on configuring Adabas Transaction Manager.

The following topics are provided:

- **ATM Transaction Managers and Adabas System Coordinator Groups**
- **Using Adabas Transaction Manager with Adabas Cluster Services**
- **Using Adabas Transaction Manager with Adabas Parallel Services**
- **Configuration Examples**

> **Note:** For information on the specific job and runtime controls which are used to configure and control Adabas Transaction Manager, see the section Parameters.

1 Adabas Transaction Managers and Adabas System Coordinator Groups

A job or TP system that requires ATM services must execute alongside an ATM transaction manager, executing in the same system and under the same Adabas SVC or ID table.

An ATM transaction manager runs as a service within an Adabas System Coordinator (SYSCO) daemon. It is therefore implicitly associated with the same Adabas System Coordinator (COR) group as the SYSCO daemon and makes use of services provided by the SYSCO daemon.

A System Coordinator group is a collection of System Coordinator daemons that communicate and inter-operate with each other across coupling facility or Entire Net-Work connections. Currently, on IBM systems, every daemon in a group must execute under the same Adabas SVC. This means that the daemons execute on different logical partitions (LPARs). An ATM transaction manager can execute within each of these daemons.

If there is only one SYSCO daemon in the COR group, there can only be one ATM transaction manager. This transaction manager will provide full support for global transactions that change `DTP=RM` databases executing under the same Adabas SVC or ID table. If a transaction changes any remote databases – that is, databases that execute in a different network node – these databases will be treated as if they were running with `DTP=NO`.

If you need to execute transactions that change `DTP=RM` databases across several network nodes, you will need to define a COR group that encompasses all of these network nodes, and run a SYSCO daemon and an ATM transaction manager in each node. The COR group effectively defines a network of ATM transaction managers that will cooperate in coordinating transactions that change `DTP=RM` databases across the corresponding network nodes. If a transaction changes any `DTP=RM` databases that execute in a network node that is not included in the COR group, these databases will be treated as if they were running with `DTP=NO`.

If appropriate, you could simply define one COR group that includes all network nodes. In this case, all the ATM managers in the network will interact, and will provide full transaction coordination across all `DTP=RM` databases in the network.

2 Using Adabas Transaction Manager with Adabas Cluster Services

A clustered operating system is a collection of independent operating system images working together as one. The objective is to spread work around the cluster to achieve better load balancing, throughput, and availability.

Clustered applications appeared before their operating system counterpart. Application clusters operated over multiple processes within a single operating system image. Now clustered applications are enhanced to operate throughout an operating system cluster.

Adabas Cluster Services allows multiple instances of the Adabas server to operate against the same database across multiple operating system images, which enhances the general clustering approach for high-end systems.

Clustered Applications

A clustered application is a collection of independent jobs operating together as a single service.

Each implementation of a clustered application allows client sessions to migrate from one job to another across one or more images in an operating system cluster. The memory associated with a particular client session must be migrated along with the client session itself.

The following clustered applications may be implemented within a single operating system image:

- CICS/MRO with dynamic transaction routing in z/OS and VSE/ESA

- IMS TM in z/OS

- UTM in BS2000

The following clustered applications are available for operating system clusters:

- CICSplex in z/OS

- IMSplex in z/OS

Adabas Cluster Services

Adabas Cluster Services allows you to use multiple images of the Adabas server across an operating system cluster in order to provide truly scalable throughput. The cluster of Adabas servers can take part in distributed transaction processing, under the control of Adabas Transaction Manager. This is controlled by the `ADARUN DTP` parameter, in the same way as for a standard Adabas nucleus. The same value must be specified for the `DTP` parameter for every nucleus in the cluster.

Adabas System Coordinator

Support for clustered applications within a single system image or across multiple system images in an operating system cluster is enabled by the Adabas System Coordinator when running one or more of the following Adabas add-on products:

- Adabas Fastpath
- Adabas Vista
- Adabas Transaction Manager
- Adabas SAF Security

Adabas Fastpath also uses the Adabas System Coordinator to host the Asynchronous Buffer Manager service.

Runtime Controls for Clustered Applications

When defining a job parameter for CICS/MRO with dynamic transaction routing or for CICSplex, select the job type CICS DTR.

CICS DTR jobs have user context information allocated from shared memory rather than from local (job) memory.

Refer to the *Adabas System Coordinator* documentation for more information.

3 Using Adabas Transaction Manager with Adabas Parallel Services

Adabas Parallel Services allows you to use multiple images of the Adabas server within a single operating system image, in order to provide improved scalability. The cluster of Adabas servers can take part in distributed transaction processing, under the control of Adabas Transaction Manager. This is controlled by the `ADARUN DTP` parameter, in the same way as for a standard Adabas nucleus. The same value must be specified for the `DTP` parameter for every nucleus in the Parallel Services cluster.

4 Configuration Examples

This section provides example configurations of Adabas Transaction Manager with the Adabas System Coordinator. It guides you through typical configurations using the SYSATM and SYSCOR administration tools. It is best to read through all the examples in turn because the configuration issues become increasingly complex with each successive example.

The Role of the Adabas System Coordinator

The Adabas System Coordinator is a prerequisite technology for Adabas Transaction Manager, Adabas Fastpath, Adabas Vista and Adabas SAF Security. Traditionally these products have focused their functionality and benefit on Adabas client sessions rather than explicitly on Adabas servers. The type of Adabas session that has been required in the past can be described as "static". Static Adabas client sessions are not able to move from one job to another within the system for load balancing purposes (this is called dynamic transaction routing). However, an increasing requirement is emerging for Adabas clients to be able to support dynamic transaction routing (DTR). One of the main purposes of the Adabas System Coordinator is to provide transparent DTR support for Adabas Transaction Manager, Adabas Fastpath, Adabas Vista and Adabas SAF Security.

These products always use Adabas System Coordinator services to support their clients. Traditional (static) clients only need to use the default (local) mode of Adabas System Coordinator services. In local mode, the Adabas System Coordinator software is embedded in the application job and provides local support for client sessions. However, DTR clients must use the Adabas System Coordinator daemon services to move client sessions around the system. In daemon mode, the local Adabas System Coordinator software works in conjunction with counterparts in the daemon to make sure the client sessions can be dynamically moved around the system. You must control these options by configuration.

> **Note:** You may also configure static clients to use daemon services if you wish. This is achieved by using SYSCOR to set the "Statistics externally viewed using group:" client control. This will make it possible to monitor users in these jobs and TP systems from SYSCOR or SYSATM running in any environment which has access to the System Coordinator daemon. Note that this option increases the use of shared memory in the system.

The Adabas System Coordinator daemon also provides services to ATM transaction managers. In particular, it provides transaction managers with details of resource managers and other ATM managers in the network. The following examples are provided for running Adabas Transaction Manager with the Adabas System Coordinator.

■ **Single System with Static Clients**

Static client support from the Adabas System Coordinator in local mode. The Adabas System Coordinator daemon is only needed to provide services to the ATM manager.

■ **Multi System with Static Clients**

Static client support from the Adabas System Coordinator in local mode. The Adabas System Coordinator daemon is only needed to provide services to the ATM manager in each system image.

■ **Single System with Dynamic Transaction Routing Clients**

DTR client support from the Adabas System Coordinator in daemon mode. The Adabas System Coordinator daemon is also needed to provide services to the ATM manager in each system image.

■ **SYSPLEX with Dynamic Transaction Routing Clients**

DTR client support from the Adabas System Coordinator in daemon mode. The Adabas System Coordinator daemon is also needed to provide transaction manager services in each system image.

Single System with Static Clients

A single system is one in which only one operating system image is used, perhaps in isolation within a larger complex site. In this type of environment, you will need to do the following:

■ Obtain a new Node ID from your administrator to use for the Adabas System Coordinator daemon. In this example, Node ID 9001 is used.

■ Define the Adabas System Coordinator group. This example shows a group called TESTSING.

■ Define the sole member of the Adabas System Coordinator group. In this example, the member is SYSCO1.

■ Specify the Adabas System Coordinator group with which the ATM transaction manager will be associated. This example uses the group SYSCO1.

■ Define the job(s) for which Adabas transaction manager is to provide transaction coordination. This example uses job CICTSING.

You can perform the required configuration by taking the following steps:

1. In this SYSCOR example the Adabas System Coordinator group name is TESTSING, the SVC is 253, and the system type is Standard (Single-system image). It is not necessary to specify a cluster facility name in this case because this is only required for running in SYSPLEX mode.

```
16:22:19                    Modify                    2006-05-29
            System Coordinator Group Member      C11230M1


Group Name: TESTSING       SVC ID: 253

System Type: X  Standard (Single-system image)
(Mark one)   _  Standard (Multi-system images)
             _  Sysplex (IBM Parallel Sysplex)
Cluster Facility Name  : _____
Automatic Pool Recovery: Y




 Command ==>
          PF1 Help          PF3 Exit          PF5 Upd
```

2. Now you must define the daemon member of the Adabas System Coordinator group in SYSCOR.
   There is usually one daemon running in each system image. In a single system there is only
   one member required. The name of the member must be the same as the name of the job to be
   run, otherwise the controls will not be located at runtime. In addition to the name, you must
   also specify the (database or) node number in the Software AG network to be used by the daemon
   member. This node number must not be currently used for any other purpose. In our example
   the member name will be SYSCO1. The Node ID allocated in this example is 9001. It is entered
   in the member definition, as shown below in the expanded group:

   > **Note:** The daemon job (SYSCO1) must specify DDCARD input for PRODUCT=CAS. This
   > defines the services that will operate in the daemon job.

```
16:29:29 ***** A D A B A S  SYSTEM COORDINATOR 8.2.1 (I004) ***** 2006-05-29
             -  System Coordinator Group Members  -           C11260M1


Runmode: Local                                          Session: Local

Group Name: TESTSING                 Cluster Facility Name:
   SVC ID: 253                       Operating System    : Single

              Member                                                      ↵

    Purge(P)      Job Name          Node ID
       _          SYSCO10            9010     <== End of List




 Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12
     Help         Exit  Refr  Upd                            Add         Menu
```

3. Next, use SYSCOR to set the System Coordinator runtime controls for the application jobs and TP systems that are to use Adabas Transaction Manager in your system. Refer to the *Adabas System Coordinator* documentation for details. In this case you can set the control `Managedbydaemon` to N.

4. Now you must use SYSATM to define runtime controls for all the client applications that are to use Adabas Transaction Manager in your system - in this example, a standard CICS job with started task name CICTSING. At runtime, this CICS system produces static Adabas clients managed by the local Adabas System Coordinator (without help from the daemon). Default parameter settings are often sufficient (and can be modified later if necessary), but there are certain runtime controls that need to be entered to ensure that Adabas Transaction Manager will be active for the named job. Where ATM's transaction management is to be activated for a job for the first time, you are asked to add a new set of runtime controls in SYSATM. Here you must first identify the job name and the type. The job type allows Adabas Transaction Manager to assume suitable defaults for most runtime controls and be ready to use the correct operating system or TP system interfaces at runtime. In our example, the job name is CICTSING and the type is CICS (not CICS Cluster), as follows:

```
16:32:41           Add                      2006-05-29
             Client Runtime Controls          F11310MC


          Name: CICTSING
          (D = Default for Type)


              _      Batch
              _      COM-PLETE
              _      CICS Cluster
              X      CICS
              _      IMS/DC
              _      UTM
              _      TSO
              _      CMS
              _      TIAM
              _      None above
          Mark to Select a Type


          Command ==>
            PF1 Help        PF3 Exit
```

5. Refer to the section Client Runtime Controls for details of runtime controls and how to set them. Default values will suffice for some of the controls, but you must specify appropriate values for the following:

   ■ `ATM ON/OFF`: set Activate ATM Processing: to ON

   ■ `System Coordinator Group Name`: provide the name of the System Coordinator group of the local System Coordinator daemon – TESTSING in this example.

Multi System with Static Clients

A multi-system is one in which multiple operating system images are used in conjunction with each other. These system images must be connected by Software AG's Entire Net-Work product and each connected system image must run its own instance of an ATM transaction manager executing as a service within an Adabas System Coordinator daemon. In these cases, it is likely that an application running on one system will change Adabas databases executing on more than one of the system images. The ATM transaction managers communicate with each other in managing the Adabas changes that are made around the network.

The example will use IMAGE1 and IMAGE2. Only static clients are used, so the local Adabas System Coordinator does not need the help of the daemon to manage its client sessions.

Here are the steps that you take:

- Define an Adabas System Coordinator group, for example, TESTMULT.

- Acquire/allocate a Node ID for each member (one per image), for example nodes 9010 and 9011 for systems IMAGE1 and IMAGE2.

- Define a member of the group for each system image, for example, SYSCO10 and SYSCO11.

- Specify the Adabas System Coordinator group with which the ATM transaction managers will be associated. This example uses the group TESTMULT.

- Define the static client jobs for which Adabas Transaction Manager is to provide transaction coordination, for example, job CICTMULT.

1. Below is the SYSCOR definition of Adabas System Coordinator Group TESTMULT using SVC number 253; the system type is Standard (Multi-system images). Cluster Facility name is not required since this is only required for running in SYSPLEX mode.

   Note: All members of an Adabas System Coordinator group must use the same SVC number at runtime.

```
16:38:44                  Modify                      2006-05-29
              System Coordinator Group Member      C11230M1


Group Name: TESTMULT        SVC ID: 253

System Type: -  Standard (Single-system image)
(Mark one)   X  Standard (Multi-system images)
             _  Sysplex (IBM Parallel Sysplex)
Cluster Facility Name  : _____
Automatic Pool Recovery: Y



Command ==>
          PF1 Help         PF3 Exit         PF5 Upd
```

2. The following shows the member definitions of the group when it has been expanded in SYSCOR.

> **Note:** The member names must exactly match the job names of the Adabas System Co-
> ordinator daemons.

```
16:39:23     ***** A D A B A S SYSTEM COORDINATOR 8.2.1 (0004) ***** 2006-05-29
                   -  System Coordinator Group Members   -         C11260M1

 Runmode: Local                                              Session: Local

 Group Name: TESTMULT                      Cluster Facility Name:
    SVC ID: 253                               Operating System: Multi


                    Member
    Purge(P)       Job Name          Node ID
       _           SYSCO10             9010          <== Top of List
       _           SYSCO11             9011          <== End of List
                  _____           _____
                  _____           _____



  Command ==>
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12
      Help         Exit  Refr  Upd                          Add         Menu
```

3. As in the single system example, use SYSCOR to set the System Coordinator runtime controls
   for the application jobs and TP systems that are to use Adabas Transaction Manager in your
   system. In this case you can set the control Managedbydaemon to N.

4. As in the single system example, you must now use SYSATM to define runtime controls for all
   the client applications that are to use Adabas Transaction Manager in your system. Be sure that

the runtime controls for each job are stored in a system file which is accessible from the system image in which the job executes.

Single System with Dynamic Transaction Routing Clients

The following are examples of technologies that offer dynamic transaction routing (DTR) in a single system image:

- CICS/MRO

- IMS TM

- UTM

> **Note:** The activation and use of DTR in the technologies listed above is under the control of the system administrator. You may be able to use these technologies without necessarily using DTR. Please be sure to check.

DTR is the most flexible implementation of load balancing and fault tolerance for these technologies. This is where multiple jobs run together to provide a single service. We refer to DTR-enabled technologies as clustered applications. Clustered applications allow client sessions to move from running in one job to another (within the same service) at any time a message pair completes. Consequently, Adabas Transaction Manager, Adabas Vista, Adabas Fastpath and Adabas SAF Security must all be ready to react to this event, on demand. The Adabas System Coordinator provides an internal service to enable DTR support for these products.

Here are the steps that you take if you wish to use Adabas Transaction Manager's transaction co-ordination in DTR jobs within a single system:

- Define an Adabas System Coordinator group, for example, TESTDTR.

- Acquire/allocate a Node ID for the daemon, for example, 9020.

- Define the member of the group, for example, member SYSCO20.

- Specify the Adabas System Coordinator group with which the ATM transaction manager will be associated - TESTDTR in this example.

- Define the client jobs that are to use Adabas Transaction Manager, for example, job name CICSDTR*; that includes jobs CICSDTR1 and CICSDTR2, that run together as a single DTR service.

1. Below is the definition of Adabas System Coordinator Group TESTDTR using SVC number 253. The System Type is Standard (Single-system image). Cluster Facility name is not required since this is only required for running in SYSPLEX mode:

```
17:39:46                  Modify                      2006-05-29
               System Coordinator Group Member         C11230M1


Group Name: TESTDTR       SVC ID: 253


System Type: X  Standard (Single-system image)
(Mark one)   -  Standard (Multi-system images)
             _  Sysplex (IBM Parallel Sysplex)
Cluster Facility Name  : _____
Automatic Pool Recovery: Y




Command ==>
          PF1 Help         PF3 Exit         PF5 Upd
```

2. The following shows the example member definition for the group expanded in SYSCOR:

```
17:42:08  ***** A D A B A S   SYSTEM COORDINATOR 8.2.1 (I004)  *****   2006-05-29
                 -  System Coordinator Group Members  -            C11260M1


Runmode: Local                                          Session: Local

Group Name: TESTDTR                      Cluster Facility Name:
   SVC ID: 253                              Operating System: Single

                    Member
     Purge(P)       Job Name          Node ID
        _           SYSCO20             9020            <== End of List
                   _____           _____
                   _____           _____                            ↵




  Command ==>
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12
      Help         Exit  Refr  Upd                         Add         Menu
```

3. Next, use SYSCOR to set the System Coordinator runtime controls for the application jobs and TP systems that are to use Adabas Transaction Manager in your system. Refer to the *Adabas System Coordinator* documentation for details. The control Managedbydaemon must be set to Y for any TP system that will use dynamic transaction routing, such as CICS/MRO.

4. As in the above examples, you must now use SYSATM to define runtime controls for all the client applications that are to use Adabas Transaction Manager in your system. Select the correct

job type for your clustered application environment, in this example, CICS Cluster. The wildcard (*) is used to reduce the number of job definitions required. You must also supply the name of the Adabas System Coordinator group which controls this cluster, and a service name for the cluster. The group must already have been defined using SYSCOR. In this example, the name CICSDTR has been chosen for the cluster service. Set ATM runtime controls, as described for the case of a single system with static clients.

> **Note:** It is important that all jobs of the same service have an identical setting for Clustered Application Service Name, especially when the wildcard option is not used. This is the only thing that relates jobs together as a single service. In this example, the cluster that we identify by service name CICSDTR consists of a number of CICS jobs, each of which executes with a job name CICSDTR with a single-character suffix.

```
07:57:12              Add                      2006-05-29
             Client Runtime Controls        T11100M3


         Type: CICS Cluster

         Name: CICSDTR*
         (D = Default for Type)
              ATM ON/OFF for Job: ON



         ----------- Daemon Mode ------------
         (Usually for Clustered Applications)
                  Service Name: CICSDTR
         Coordinator Group Name: TESTDTR
         -----------------------------------



         **********************
         *  PF5 to Confirm Add *
         **********************



         Command ==>

         PF1 Help    PF3 Exit
```

SYSPLEX with Dynamic Transaction Routing Clients

CICS in a Parallel Sysplex is an example of a technology that offers dynamic transaction routing (DTR) in a Clustered Operating System running in multiple images.

> **Note:** The activation and use of DTR in these technologies is under the control of the system administrator. You may be able to use these technologies without necessarily using DTR. Please be sure to check.

DTR is the most flexible implementation of load balancing and fault tolerance for these technologies. This is where multiple jobs run together to provide a single service. We refer to DTR-enabled technologies as clustered applications. Clustered applications allow client sessions to move from running in one job to another (within the same application service) at any time a message pair completes. Consequently, Adabas Transaction Manager, Adabas Vista, Adabas Fastpath and Adabas SAF Security must all be ready to react to this event, on demand. The Adabas System Coordinator provides an internal service to enable DTR support for these products.

In a SYSPLEX configuration it is possible for a client session to be routed from one system image to another within the SYSPLEX. In order to facilitate this DTR, the Clustered Application Service running in the Adabas System Coordinator daemons must communicate in order to negotiate the transfer of the client session context from one system image to the other. This system level communication is assisted by use of the IBM Coupling Facility.

Here are the steps that you take if you wish to use Adabas Transaction Manager in DTR jobs within a SYSPLEX:

- Define a Cache Structure in the Coupling Facility, for example, TESTMDTR-CFN.

- Define an Adabas System Coordinator group, for example, TESTMDTR.

- Acquire/allocate a Node ID for each member, for example, nodes 9030 and 9031.

- Specify the Adabas System Coordinator group with which the ATM transaction managers will be associated. This example uses the group TESTMDTR, defined as above.

- Define the members of the group, for example, members SYSCO30 and SYSCO31.

- Define the client jobs that are to use Adabas Transaction Manager. In this example we make a SYSPLEX DTR definition, with jobs CICMDTR1 and CICMDTR2 working together as a single service.

1. Below is the SYSCOR definition of Adabas System Coordinator Group TESTMDTR using SVC number 253; the System Type is SYSPLEX (IBM Parallel Sysplex). The IBM Coupling Facility is used in this configuration. In the example, this is a cache structure called TESTMDTR-CFN which must have been defined by your system administrator.

```
07:58:54                 Modify                      2006-05-29
            System Coordinator Group Member        C11230M1


Group Name: TESTMDTR        SVC ID: 253

System Type: -  Standard (Single-system image)
(Mark one)   -  Standard (Multi-system images)
             X  Sysplex (IBM Parallel Sysplex)
Cluster Facility Name  : _____
Automatic Pool Recovery: Y



Command ==>
         PF1 Help           PF3 Exit           PF5 Upd
```

2. The following shows the member definitions of the expanded group in SYSCOR.

> **Note:** The member names must exactly match the job names of the Adabas System Co-ordinator daemons.

```
07:59:58  ***** A D A B A S   SYSTEM COORDINATOR 8.2.1 (I004)  *****  2006-05-29
                -  System Coordinator Group Members  -            C11260M1

 Runmode: Daemon / 9020                             Session: Daemon / 9020

 Group Name: TESTMDTR              Cluster Facility Name: TESTMDTR-CFN
    SVC ID: 253                         Operating System: Multi

                  Member
    Purge(P)     Job Name          Node ID
        _        SYSCO30             9030         <== Top of List
        _        SYSCO31             9031         <== End of List
                 _____            _____

                 _____            _____


  Command ==>
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12
     Help        Exit  Refr  Upd                          Add        Menu
```

3. Next, use SYSCOR to set the System Coordinator job runtime controls for the application jobs and TP systems that are to use Adabas Transaction Manager in your system. Refer to the *Adabas System Coordinator* documentation for details. The control `Managedbydaemon` must be set to Y for any CICS/PLEX system that will use dynamic transaction routing.

4. Next, use SYSATM to define runtime controls for all the CICS/PLEX jobs that are to run together as a clustered service. Select the correct job type for your clustered application environment, in

this example, CICS Cluster. The wildcard (*) is used to reduce the number of job definitions required. You must also supply the name of the Adabas System Coordinator group which controls this cluster, and a service name for the cluster. The group, TESTMDTR in this example, must already have been defined using SYSCOR. In this example, the name CICSMDTR has been chosen for the cluster service. Set ATM runtime controls, as described for the case of a single system with static clients.

> **Note:** It is important that all jobs of the same service have an identical setting for Clustered Application Service Name, especially when the wildcard option is not used. This is the only setting that relates jobs together as a single service.

```
08:05:17                  Add                    2006-05-29
                 Client Runtime Controls        T11100M3


         Type: CICS Cluster
         Name: CICSDTR*
         (D = Default for Type)
             ATM ON/OFF for Job: ON



         ----------- Daemon Mode ------------
         (Usually for Clustered Applications)
                  Service Name: CICSDTR
          Coordinator Group Name: TESTDTR
         ----------------------------------



            ***********************
            *  PF5 to Confirm Add *
            ***********************



         Command ==>
              PF1 Help    PF3 Exit
```