Adabas Transaction Manager

Adabas Transaction Manager Parameters

Version 7.5.1

September 2009

Adabas Transaction Manager

This document applies to Adabas Transaction Manager Version 7.5.1 and to all subsequent releases.

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

Table of Contents

1 Adabas Transaction Manager Parameters

This document describes the parameters used for Adabas Transaction Manager.

- ATM Runtime Parameters
- ATM Job Parameters
- ATM Parameter Summary Table

2 Runtime Parameters

This section describes the runtime parameters used for Adabas Transaction Manager.

ATM Runtime Parameters

ATM execution is controlled mainly through ADARUN (ADARUN DDCARD input) runtime parameters. When executing the Adabas Transaction Manager daemon, the parameter setting `ADARUN DTP=TM` must be provided. All other ATM runtime parameters, all of which begin with "TM", will be effective only when `ADARUN DTP=TM` is specified. They are ignored for all other values of DTP. For more information on the DTP parameter, see the *Adabas Operations* documentation.

Note: Adabas databases that participate in two-phase commit processing must execute as resource managers with `ADARUN DTP=RM`.

Note: `ADARUN NONDES=YES` (non-descriptor search allowed) is required for the Adabas Transaction Manager daemon.

The following section describes each ATM runtime parameter:

TMDRQ : Size of ATM Request Queue

Parameter	Parameter Type	Use	Possible Values	Default
`TMDRQ`	Runtime	Sets the number of entries allowed in the ATM daemon internal request queue.	10 - 32767	10

The ATM daemon uses an internal request queue for scheduling the tasks it needs to perform. The `TMDRQ` parameter determines the total number of entries in the queue that is used to service certain incoming, asynchronous requests. This queue is currently used only when the ATM RRMS interface is active.

RRMS Considerations

When the RRMS interface is in use, the ATM dynamic request queue (DRQ) is used for communication between ATM RRMS exits and the daemon. The value specified for the TMDRQ parameter should therefore be increased if RRMS is to be used. Consider the number of global transactions that may reach a syncpoint at the same time, and increase TMDRQ by this number. You can specify a higher value, for safety, without incurring any performance degradation.

You can use Online Services to monitor usage of the DRQ. If you specify too small a value and the DRQ becomes full, backouts resulting in response code 9 are likely to occur.

Example

The ATM daemon will allow 250 entries in its internal request queue.

```
ADARUN PROG=ADANUC,DTP=TM,TMDRQ=250
```

TMDYNTCIDS : Number of Dynamic Client Identifiers

Parameter	Parameter Type	Use	Possible Values	Default
TMDYNTCIDS	Runtime	Sets the maximum number of Client IDs that ATM can dynamically allocate.	32 - 65536	1024

To aid interpretation of online displays and diagnostic logs, ATM assigns an 8-byte client identifier (TCID) to each user that engages in transactional activity. If the user issues an Adabas OP command with an ETID, the TCID takes the same value as the ETID. If a user does not have an ETID, ATM dynamically allocates a TCID. A dynamically allocated TCID becomes available for reuse when the associated user closes all its sessions with Adabas databases or times out.

The TMDYNTCIDS parameter determines the maximum number of TCIDs available for use by the ATM daemon during its lifetime. The value should be at least as large as the maximum number of users who may be active at one time without ETIDs.

Note: A TCID is allocated temporarily to every user when the user first comes to the attention of the ATM daemon. If the user supplies an ETID, the temporarily allocated TCID will quickly be freed. Therefore, TCIDs are required, even in environments in which every user has an ETID.

Example

The ATM daemon is to handle up to 2048 concurrent users who do not have ETIDs.

```
ADARUN PROG=ADANUC,DTP=TM,TMDYNTCIDS=2048
```

TMETDATA : ET Data Control

Parameter	Parameter Type	Use	Possible Values	Default
`TMETDATA`	Runtime	Location of `ET` data	ATM | TARGETS	TARGETS

The `TMETDATA` parameter determines the database or databases that will store `ET` data. The same `TMETDATA` parameter value must be specified for all ATM instances on the network.

Possible values:

Value	Explanation
ATM	`ET` data is always stored in and read from the ATM daemon's database, without regard to the Database ID indicated by the Adabas command. ■ treats `ET` data as belonging to a global transaction rather than to a database; ■ there is just one copy of a user's `ET` data in a system; ■ eliminates confusion resulting from the existence of different `ET` data in different databases under the same ETID; ■ overrides the Natural `ETDB` parameter; ■ should not be used if the `SERIAL=YES` or `SERIAL=FORCE` option will be specified for any client environment in which `ET` data is stored or read.
TARGETS	`ET` data is stored during the commit process in all changed databases that are running with `DTP=RM`. `ET` data that is stored by an `ET` or `CL` command is always stored in the database that is the target of the command, whether or not that database runs with `DTP=RM`. It the database runs with `DTP=NO`, the `ET` data is written to the database only after any open global transaction has been committed. If the user storing `ET` data is at global transaction status, ATM is not required to carry out any commit processing; the data is simply stored in the database indicated by the Adabas command. `ET` data is always read from the database specified in the Adabas command. ■ treats `ET` data as belonging to a database or a local database transaction rather than to a global transaction; ■ a user's `ET` data might exist in several databases in a system; ■ a user can have several different `ET` data values in different databases at the same time;

Value	Explanation
	■ means that 3GL application programs using ET data must have knowledge of the database that holds ET data for a given transaction; ■ honors the Natural ETDB parameter.

Note: See also section ET Data Storage.

Example

ET data is to be stored in the transaction manager daemon's database only.

```
ADARUN PROG=ADANUC,DTP=TM,TMETDATA=ATM
```

TMGTNA : Global Non-Activity Time Limit

Parameter	Parameter Type	Use	Possible Values	Default
TMGTNA	Runtime	Sets the user inactivity time limit	1 - 1677215	720

The TMGTNA parameter approximately specifies the maximum elapsed time, in units of 1.048576 seconds, a user known to the ATM daemon can remain active without issuing a transactional Adabas command that invokes ATM on the user’s behalf. If a user exceeds the specified time, the ATM daemon frees any of its resources associated with this user.

The TMGTNA value should be smaller than the lowest TNAx parameter of any database that runs with DTP=RM, and greater than the value specified for TMGTT.

The TMGTNA parameter does not supersede the TNAx parameter settings of other databases. When a user exceeds the period specified by TMGTNA, other databases do not free that user’s user queue elements (UQEs); this occurs only when the user exceeds the TNAx values of the local databases.

Example

The ATM daemon will time out an inactive user after a period of about 7200 seconds (2 hours).

```
ADARUN PROG=ADANUC,DTP=TM,TMGTNA=7200
```

TMGTT : Global Transaction Time Limit

Parameter	Parameter Type	Use	Possible Values	Default
TMGTT	Runtime	Sets time limit during which a global transaction can be open without being prepared.	1 - 1677215	720

The TMGTT parameter approximately specifies the maximum elapsed time, in units of 1.048576 seconds, that a global transaction can remain open without being prepared. If the limit is exceeded, the ATM daemon automatically backs out the global transaction from all affected databases. The next time the user issues a transactional Adabas command, a response code 9 is returned, provided that the global non-activity time has not been exceeded for the user.

TMGTT should be smaller than the lowest TT parameter of any database running with DTP=RM and smaller than the value specified for TMGTNA.

The TMGTT parameter does not supersede the TT parameter settings of other databases. If a database times out a user who has an unprepared global transaction open, the global transaction is subsequently backed out. A response code 9 is returned if the user then issues a transactional command before the corresponding non-activity time limit is exceeded.

Example

The ATM daemon will back out global transactions that are not prepared within about 600 seconds (10 minutes) of the first transactional command.

```
ADARUN PROG=ADANUC,DTP=TM,TMGTT=600
```

TMLOG : ATM Logging Control

Parameter	Parameter Type	Use	Possible Values	Default
TMLOG	Runtime	Controls the logging option for the current execution of the ATM daemon.	YES | NO | NEVER	NO

The ATM daemon can log internal events for diagnostic purposes using a dual logging system. It can

- maintain a log buffer area in memory that contains details of the latest logged events;
- maintain transaction-related event details in storage areas related to specific global transactions;
- write every log record to dual sequential ATMLOG datasets.

Possible values:

Value	Description
YES	The ATM daemon writes every record to an ATMLOG dataset. The ATM daemon keeps a record of the latest internal events in its global log buffer area. It also maintains the contents of log buffer areas related to open global transactions. Logging to the ATMLOG datasets can be deactivated at any time using the operator command `TM NOLOG`.
NO	The ATM daemon will not write log records to the ATMLOG datasets. Logging to the ATMLOG datasets can be activated at any time using the operator command `TM LOG`. The ATM daemon keeps a record of the latest internal events in its global log buffer area; it also maintains the contents of log buffer areas related to open global transactions.
NEVER	ATM logging is disabled for the duration of this execution of the ATM daemon.

On MVS and VSE systems, if the ATM daemon writes log records to the ATMLOG (`TMLOG=YES` or `TM LOG` issued) and the dataset currently in use becomes full, the daemon closes that dataset and switches to the other dataset. Automatic log switching currently does not take place on BS2000 systems. A system message might appear in the console log, indicating that the first log dataset has become full. You may also switch between the two ATMLOG datasets by using the operator command `TM FEOFLOG`. Whenever the ATMLOG datasets are switched, a message is displayed on the console indicating the dataset that is currently in use.

Example

The ATM daemon is to maintain its internal log buffer areas and write all log records to the ATMLOG datasets.

```
ADARUN PROG=ADANUC,DTP=TM,TMLOG=YES
```

TMMSGSEV : Severity Threshold for Warning Messages

Parameter	Parameter Type	Use	Possible Values	Default
`TMMSGSEV`	Runtime	Sets severity threshold for the suppression of warning messages.	0 | 4 | 8	0

Every message that the ATM daemon sends to the console and to the DDPRINT dataset has a severity level. Message severity levels are explained in the section Messages and Codes.

Using the `TMMSGSEV` parameter, you can prevent messages with low severity levels from being written to the console by the ATM daemon. This parameter has no effect on messages that are issued by the TM proxy, or during the early part of the daemon's initialization.

All messages created by the ATM daemon are written to DDPRINT regardless of the `TMMSGSEV` specification.

Possible values:

Value	Description
0	No messages are suppressed.
4	Messages of severity less than 4 are not sent to the console.
8	Messages of severity less than 8 are not sent to the console.

Example

The ATM daemon should not send message of severity less than 4 to the console.

```
ADARUN PROG=ADANUC,DTP=TM,TMMSGEV=4
```

TMRESTART : ATM Restart Control

Parameter	Parameter Type	Use	Possible Values	Default
TMRESTART	Runtime	Controls restart handling of problematic transactions.	NORMAL I FORCE I FORCEALL	NORMAL

When an ATM daemon restarts, it rebuilds its global user queue as nearly as possible to its latest state when ATM was last executing. It then attempts to complete or back out any incomplete global transactions. The TMRESTART parameter determines the daemon's restart action with regard to global transactions and transaction branches that remain incomplete at the end of this process.

Possible values:

Value	Description
NORMAL	Any incomplete transactions remain in an incomplete state until such time as they can be completed.
FORCE	ATM transfers to the suspect transaction journal (STJ) the details of every incomplete global transaction that has its root local to this ATM daemon. The details of these transactions are deleted from ATM's recovery information, the originating users are closed, and any related internal resources are freed. ATM can no longer guarantee integrity for such global transactions.
FORCEALL	Details of all unresolved global transactions and transaction branches are transferred to the STJ. The details are deleted from ATM's recovery information, the originating users are closed, and any related internal resources are freed. ATM can no longer guarantee integrity for the affected global transactions.

Example

After recording them in the STJ, the ATM daemon is to discard details of any global transactions that it owns, and that remain incomplete at the end of its restart processing.

```
ADARUN PROG=ADANUC,DTP=TM,TMRESTART=FORCE
```

TMSYNCMGR : Support for External Transaction Coordinators

Parameter	Parameter Type	Use	Possible Values	Default
TMSYNCMGR	Runtime	Indicates whether or not the ATM daemon is to interact with an external transaction coordinator.	NONE | RRMS	NONE

The TMSYNCMGR parameter determines whether or not the ATM daemon registers and interacts with an external transaction coordinator.

Possible values:

Value	Description
NONE	The ATM daemon will not interact proactively with an external transaction coordinator. This setting does not preclude the use of the CICS Syncpoint Manager, which interfaces directly with the TMP component and not with the ATM daemon.
RRMS	The ATM daemon will register with the IBM Recoverable Resource Management Services so that it can participate in transactions that involve other RRMS-enabled resource managers. This parameter value is valid only for z/OS or OS/390 systems in which RRS (a component of RRMS) is active. The current version of ATM offers participation in RRMS-coordinated two-phase commit for single-user, single-TCB batch applications and for applications running under IMS/TM.

Example

ATM registers with RRMS so that it can take part in transactions involving other RRMS-enabled resource managers.

```
ADARUN PROG=ADANUC,DTP=TM,TMSYNCMGR=RRMS
```

TMTCIDPREF : Dynamic Client ID Prefix

Parameter	Parameter Type	Use	Possible Values	Default
TMTCIDPREF	Runtime	Defines the first one, two, or three characters of dynamically allocated Client IDs.	up to 3 alphanumeric characters	TM

To aid interpretation of Online Services displays and diagnostic logs, the ATM daemon dynamically allocates an 8-byte client identifier (TCID) for each user who engages in transactional activity but who does not have an ETID.

The format of a dynamically allocated TCID is

```
pppnnxxx
```

—where

ppp is the value of TMTCIDPREF, left-justified and with trailing blanks if required.

nn is the 2-byte binary ATM Database ID.

xxx is a 3-byte binary suffix. Suffixes are allocated in sequence and are reused. The first TCID to be allocated has a suffix of zeros.

The TMTCIDPREF value must be chosen to ensure that dynamic client identifiers are always different from any ETIDs used in your network.

Example

ATM will assign dynamic TCIDs beginning with the string “DYN”.

```
ADARUN PROG=ADANUC,DTP=TM,TMTCIDPREF=DYN
```

Recommendations for Adabas Runtime Parameters when using ATM

Caution: Prior to using Adabas Transaction Manager in a production environment, the settings for Adabas runtime parameters should be reviewed and modified as necessary according to the recommendations provided in this section.

The Adabas Transaction Manager daemon tracks and coordinates all Adabas transactional activity. This includes:

- OP,CL,ET,BT and RE commands
- The first change-type command (A1,E1,N1, etc.) to each DTP=RM database modified in a transaction

The originating client's Transaction Manager proxy forwards these commands to the Adabas Transaction Manager daemon for scrutiny.

Furthermore, the Adabas Transaction Manager daemon acts as a focal point for transactional activity for all Adabas activity within the system image. Consequently, there may be times when significant load is experienced. Therefore, for production running, an Adabas Transaction Manager daemon needs to be configured to handle high levels of concurrent activity.

Some Adabas parameters need to be reviewed. In general it is usually appropriate to take the values set for the busiest Adabas database in the system and use at least those values. In any case, setting values high, and reviewing them later, after a successful period of operation, is a reasonable safety first approach for most parameters. The following table describes the most important parameters which should be reviewed, but all other parameters should be considered also.

Adabas Parameter	Description
NU	For the ATM daemon, the NU parameter needs to be high enough for the maximum concurrent number of Adabas clients in the system, which may include two for each Natural session (depending on the Natural ADAMODE setting). NU should be set to at least twice the value of TMGUQ.
NC	The command queue size for the Transaction Manager daemon needs to be set for high levels of throughput.
NAB	For each concurrent activity a portion of the attached buffers is used.
NT	Some transactional activity can take more time to complete than a normal Adabas command (for example, coordinating multiple databases to complete a transaction safely), so the number of threads in the ATM daemon should be higher than might be expected.
TMLOG=NO	Numerous I/O operations to the TM LOG dataset are performed when the TMLOG parameter is set to TMLOG=YES. In a busy system, the I/O rate for TMLOG=YES will have a significant impact on performance; in such cases, set TMLOG=NO for the ATM daemon.

Before using Adabas Transaction Manager in a production environment, remember also to review the job parameter settings for jobs and TP systems that will use ATM. Pay particular attention to the setting of the LGRECNO parameter for TP systems.

Enforced Runtime Parameters

When the Adabas Transaction Manager daemon is started, certain ADARUN parameter values are enforced. If other values are supplied through the DDCARD DD statement, they are ignored and overwritten by the values preset for the daemon. The enforced parameter values include the following, although other parameters might also be affected:

```
FASTPATH=NO
PLXID=0
READONLY=NO
SMPID=0
SPT=NO
UTIONLY=NO
VISTA=NO
```

3 ATM Job Parameters

Job Parameter Usage

Correct operation of the Adabas Transaction Manager and the Adabas Transaction Manager proxy depends on the settings of job parameters. These are defined and maintained using Online Services. Their settings determine the logic to be used by the proxy during transaction processing. These settings should be determined according to production system requirements, and should be tested before being used in production.

At execution time, the Adabas Transaction Manager proxy checks the Adabas System Coordinator system file for job parameters that match the name of the currently executing job. If no exact match is found, a wild-card match is attempted. Failing this, the parameters for job name DEFAULT will be returned. If there is no definition under the name DEFAULT, hard-coded default values in the Adabas Transaction Manager kernel will be used, and console messages will be issued to indicate their values.

It is therefore important to set appropriate values for each job that could load and execute the Adabas Transaction Manager proxy.

Note: The `LGRECNO` parameter setting requires particular attention. This parameter setting determines the amount of memory, per user, that is allocated for logging diagnostic information. In a TP system that services many users, an `LGRECNO` parameter setting of 100, for example, could quickly consume a large amount of memory. In such systems, choose a small value for the `LGRECNO` setting. See the section User Sessions for more information.

Job Parameter Descriptions

This section describes each ATM job parameter:

- ATM : Activate ATM Processing
- CORGROUP: Adabas System Coordinator Group Name
- GENOP : Generate OP Commands
- HSTSYNC : Use Host System Transaction Manager
- LCLSYNC : Use Client-Side Transaction Manager
- LGRECNO : Number of Log Record Entries
- MAXDB : Maximum Number of Current Databases
- SERIAL : Emergency Serial ET Commands
- SVC : SVC Number
- SYNCMODE : Syncpoint Processing Mode
- SYNCONBT : Generate External Syncpoint on BT Command
- SYNCONCL : Generate External Syncpoint on CL Command
- SYNCONET : Generate External Syncpoint on ET Command
- TRANMODE : Transaction Model

- TRNCTL : Transaction Control
- XH : Extended Hold Feature

ATM : Activate ATM Processing

Parameter	Parameter Type	Use	Possible Values	Default
ATM	Job	Indicates whether or not ATM is to provide transaction coordination for this job.	ON | OFF	ON

Possible values:

Value	Description
ON	Users will perform distributed transaction processing, and ATM is to provide transaction coordination.
OFF	ATM is not to be used for coordinating transactions for users in this job.

CORGROUP: Adabas System Coordinator Group Name

Parameter	Parameter Type	Use	Possible Values	Default
CORGROUP	Job	Identifies the System Coordinator group in which the job or TP system will execute.	Any valid System Coordinator group name	A valid group name must be provided

GENOP : Generate OP Commands

Parameter	Parameter Type	Use	Possible Values	Default
GENOP	Job	Indicates whether or not the ATM proxy is to generate an OP command when a session uses a database without having issued an OP.	NO | YES	NO

Possible values:

Value	Description
NO	The ATM proxy will not generate OP commands on behalf of users.
YES	Whenever a user session uses a new database for which no OP command has been issued, the ATM proxy will issue an OP command on behalf of the user.

HSTSYNC : Use Host System Transaction Manager

Parameter	Parameter Type	Use	Possible Values	Default
`HSTSYNC`	Job	Indicates whether or not transactions in this client environment are to be controlled by the host system native transaction manager.	YES | NO	NO

Possible values:

Value	Description
YES	Transactions in this client environment are to be controlled by the host system native transaction manager.
NO	Transactions in this client environment are not to be controlled by the host system native transaction manager.

Note: This parameter is currently supported only for single-user, single-TCB batch jobs or TSO sessions, and for IMS/TM systems, running under z/OS or OS/390.

Note: The parameters `HSTSYNC=YES` and `LCLSYNC=YES` are mutually exclusive.

LCLSYNC : Use Client-Side Transaction Manager

Parameter	Parameter Type	Use	Possible Values	Default
`LCLSYNC`	Job	Indicates whether or not transactions in this client environment are to be controlled by the environment native syncpoint manager.	YES | NO	NO

Possible values:

Value	Description
YES	Transactions in this client environment are to be controlled by the environment native syncpoint manager.
NO	Transactions in this client environment are not to be controlled by the environment native syncpoint manager.

Note: CICS under z/OS, OS/390 or VSE/ESA is the only client environment for which this parameter is currently supported.

Note: The parameters `LCLSYNC=YES` and `HSTSYNC=YES` are mutually exclusive. However, `LCLSYNC=NO` does not preclude the use of RRMS as an external transaction coordinator for other jobs, since RRMS interfaces directly with the ATM daemon and not with the ATM proxy.

CICS/RMI Considerations

When defining a parameter module for a CICS/RMI environment:

- `LCLSYNC=NO` means that ATM coordinates transactions in the normal way, but not under the control of the CICS Syncpoint Manager. That is, even when the RMI is enabled, no transaction coordination occurs between Adabas and other RMI-enabled resource managers.
- `LCLSYNC=YES` activates the RMI so that the CICS Syncpoint Manager coordinates transactions across all resource managers.

LGRECNO : Number of Log Record Entries

Parameter	Parameter Type	Use	Possible Values	Default
`LGRECNO`	Job	Indicates whether or not internal logging is to be performed, and if so, how much space, per user, will be used for this purpose.	0-4096	0

The ATM proxy is capable of logging internal events for diagnostic purposes. The log buffer area is reused cyclically.

If necessary, Software AG support can provide assistance regarding analysis of the contents of a user's proxy log buffer.

Possible values:

Value	Description
0	The proxy will not acquire log buffer areas and will not log internal events.
1-4096	The number of entries to be reserved for internal event logging. Each entry requires 96 bytes.

Note: Do not specify a high value for a TP system that services a large number of users since this may result in memory becoming exhausted.

The `LGRECNO` parameter has no effect on the logging that takes place in the ATM daemon address space. Logging by the ATM daemon is controlled by the ADARUN parameter `TMLOG`.

MAXDB : Maximum Number of Current Databases

Parameter	Parameter Type	Use	Possible Values	Default
`MAXDB`	Job	Sets the maximum number of databases with which a user can have an active session at one time.	number	4

The `MAXDB` setting includes all databases without regard to the DTP parameter setting.

SERIAL : Emergency Serial ET Commands

Parameter	Parameter Type	Use	Possible Values	Default
`SERIAL`	Job	Indicates whether or not the ATM proxy is to switch a user from DTP to serial `ET/BT` mode if the local ATM daemon becomes unavailable.	YES | NO | FORCE	FORCE

Possible values:

Value	Description
YES	The ATM proxy is to switch a user from DTP to serial `ET/BT` mode if the local ATM daemon becomes unavailable. A response code will be returned to indicate that the local ATM daemon is unavailable.
NO	The ATM proxy is not to switch a user from DTP to serial `ET/BT` mode if the local ATM daemon becomes unavailable.
FORCE	A user is to be switched to serial mode if necessary. No notification is provided.

For more information regarding DTP and Serial mode processing, refer to the section Processing Modes.

Note: If ATM automatically switches a user to serial `ET/BT` mode when the user has a transaction open, the user might receive unexpected response codes, even if the job parameter value for serial processing is set to FORCE. For example, the transaction might be backed out, and response 9 returned to the user. Other response codes, such as response 240 subcode 88, might be given, depending on how far the transaction had progressed, and there is a possibility that the transaction will remain unresolved until the ATM daemon is once again able to carry out its responsibilities. Even so, the setting of value FORCE will ensure minimum disruption when a switch to serial `ET/BT` mode is necessary, and will probably allow most users to continue processing without noticing the switch.

SVC : SVC Number

Parameter	Parameter Type	Use	Possible Values	Default
SVC	Job	Sets the number of the Adabas SVC used by the ATM daemon that is local to users executing under these job parameters.	number	249

SYNCMODE : Syncpoint Processing Mode

The SYNCMODE parameter has been replaced by the TRANMODE parameter.

SYNCONBT : Generate External Syncpoint on BT Command

Parameter	Parameter Type	Use	Possible Values	Default
SYNCONBT	Job	Indicates whether or not a BT command should cause ATM to request a rollback syncpoint from the external transaction coordinator.	YES | NO	YES

Possible values:

Value	Description
YES	Any BT command will cause ATM to request a rollback syncpoint from the external transaction coordinator that controls the current transaction. If there is no external transaction coordinator in use, this parameter setting is ignored.
NO	ATM will not request a rollback syncpoint from any external coordinator when it processes a BT command. Furthermore, if ATM itself decides that a backout must be performed, it will not request a rollback from any external coordinator.

Regardless of the SYNCONBT parameter setting, a BT command causes all Adabas changes to be backed out.

SYNCONCL : Generate External Syncpoint on CL Command

Parameter	Parameter Type	Use	Possible Values	Default
SYNCONCL	Job	Indicates whether or not a CL command should cause ATM to request a commit syncpoint from the external transaction coordinator.	YES | NO	YES

Possible values:

Value	Description
YES	Any CL command will cause ATM to request a commit syncpoint from the external transaction coordinator that controls the current transaction. If there is no external transaction coordinator in use, this parameter setting is ignored.
NO	ATM will not request a commit syncpoint from any external coordinator when it processes a CL command.

For IMS/TM systems for which the parameter setting HSTSYNC=YES is specified, the parameter setting SYNCONCL=YES will be ignored, and the parameter setting SYNCONCL=NO will take effect.

SYNCONET : Generate External Syncpoint on ET Command

Parameter	Parameter Type	Use	Possible Values	Default
SYNCONET	Job	Indicates whether or not an ET command should cause ATM to request a commit syncpoint from the external transaction coordinator.	YES | NO	YES

Possible values:

Value	Description
YES	Any ET command will cause ATM to request a commit syncpoint from the external transaction coordinator that controls the current transaction. If there is no external transaction coordinator in use, this parameter setting is ignored.
NO	ATM will not request a commit syncpoint from any external coordinator when it processes an ET command.

Regardless of the SYNCONET parameter setting, an ET command causes any pending Adabas changes to be committed.

For IMS/TM systems for which the parameter setting HSTSYNC=YES is specified, the parameter setting SYNCONET=YES will be ignored, and the parameter setting SYNCONET=NO will take effect.

TRANMODE : Transaction Model

Parameter	Parameter Type	Use	Possible Values	Default
TRANMODE	Job	Transaction model	MESSAGE | DYNAMIC	MESSAGE

Possible values:

Value	Description
MESSAGE	The message-based transaction model will be used. According to this model, a syncpoint always takes place when processing of a message is complete (normally, this means that screen I/O causes a syncpoint). Syncpoints caused by ET, BT, OP and CL commands are handled in the normal way. Unsolicited syncpoints cause ATM to commit (or back out) changes made to Adabas databases, and release held ISNs.
DYNAMIC	The dynamic transaction model will be used. According to this model, transactions are delimited by Adabas commands. Unsolicited commit syncpoints are ignored by ATM; ET and CL commands cause ATM to commit pending Adabas changes. Unsolicited rollback syncpoints cause ATM to back out changes from Adabas databases and release held ISNs.

Note: The setting TRANMODE=MESSAGE is equivalent to the setting SYNCMODE=FULL with ATM version 1.2.

Note: The setting TRANMODE=DYNAMIC is equivalent to the setting SYNCMODE=ADABAS with ATM version 1.2.

Note: There is no TRANMODE setting equivalent to the setting SYNCMODE=ALL with ATM version 1.2. The reason is that there is no distinction between the settings SYNCMODE=ALL and SYNCMODE=FULL with ATM version 7.4, since ATM will always cause held ISNs to be released when a syncpoint takes place, unless directed to do otherwise by command options and the XH parameter.

For more information, see also the section Syncpoint Processing Options.

TRNCTL : Transaction Control

Parameter	Parameter Type	Use	Possible Values	Default
TRNCTL	Job	Determines the action to be taken whenever an unsolicited syncpoint request is received.	GLOBAL | LOCAL	LOCAL

The TRNCTL setting determines what action is to be taken whenever an unsolicited syncpoint request is received, that is, whether a decision to commit or back out a global transaction is to be made by the application program itself using its local ATM proxy, or by a software component other than ATM or ATM proxy.

Unsolicited syncpoint requests can occur if:

- The parameter setting `LCLSYNC=YES` is specified and ATM could receive a request to commit or back out a global transaction other than through an `ET` or `BT` command; or
- the ATM daemon executes with the parameter setting `TMSYNCMGR=RRMS`; or
- an application could cause a participating trigger to execute, causing an `ET` command to be issued, all without the knowledge of the user's local ATM proxy.

Possible values for the `TRNCTL` setting are:

Value	Description
LOCAL:	Decision to commit or back out a global transaction is to be made by the application program itself using its local ATM proxy. If unsolicited syncpoints are not valid in your environment, specify `TRNCTL=LOCAL` so that any unsolicited syncpoint request is reported as an error. Specifying `TRNCTL=LOCAL` does not prevent unsolicited syncpoints from being processed; rather, it results in the user's next Adabas command being rejected with a response code 243 indicating that the syncpoint has occurred.
GLOBAL	Decision to commit or back out a global transaction can be made by a software component other than ATM or ATM proxy. `TRNCTL=GLOBAL` is required if unsolicited syncpoint requests are acceptable. **Note:** Running with `TRNCTL=GLOBAL` in any environment other than CICS/RMI can cause the ATM proxy to make additional calls to the ATM daemon to check transaction status. It is recommended to use `TRNCTL=LOCAL` wherever possible to eliminate these extra calls.

CICS/RMI Considerations

When running with the parameter setting `LCLSYNC=YES` in a CICS/RMI environment, a decision to commit or back out a global transaction can occur in several ways:

- the application program issues `ET` or `BT`;
- the application program issues `EXEC CICS SYNCPOINT` (or `SYNCPOINT ROLLBACK`);
- the CICS task terminates, normally or abnormally, when updates are pending.

In the first case, the ATM proxy is aware of the syncpoint decision.

In all other cases, from the proxy's point of view, the syncpoint is unsolicited. If the user is at global transaction status, an unsolicited syncpoint has no effect on Adabas databases, other than to cause held ISNs to be released. On the other hand, if an unsolicited syncpoint occurs when Adabas changes are pending, ATM is instructed to commit the changes; in this case also, held ISNs are released.

If such unsolicited syncpoints are acceptable in your CICS system, specify `TRNCTL=GLOBAL`. Otherwise, specify `TRNCTL=LOCAL` to treat unsolicited syncpoint requests as errors.

Most CICS applications run in pseudo-conversational mode; that is, the current CICS task terminates when a screen I/O occurs. If ATM CICS/RMI is installed and active, and the job parameter TRANMODE=MESSAGE is in effect, an application that runs in pseudo-conversational mode cannot keep an Adabas transaction open across screen I/Os because an implied (unsolicited) commit syncpoint occurs at every screen I/O. In this mode, it is not possible for a transaction to remain open across screen I/Os. See the TRANMODE parameter for details of a different way of responding to unsolicited syncpoints.

If TRNCTL=LOCAL is specified and the syncpoint resulted in Adabas changes being committed, the next Adabas call following a screen I/O returns response code 243.

Various Natural features including some SYSSEC functions maintain open Adabas transactions across screen I/Os. In a CICS/RMI environment operating with the job parameter TRANMODE=MESSAGE, such functions are likely to

- fail with a NAT3243 error when TRNCTL=LOCAL; or
- be unable to back out in case of errors when TRNCTL=GLOBAL; or
- fail with response code 144.

You should therefore execute Natural system functions in one of the following ways:

- execute Natural system functions in a system other than CICS, or in a CICS system using a link module in which the RMI is not active; or
- execute Natural system functions in a CICS system that operates with the job parameter TRANMODE=DYNAMIC; or
- execute Natural system functions in conversational mode when using the CICS Resource Manager Interface.

RRMS Considerations

When ATM's RRMS interface is active, a decision to commit or back out a global transaction can occur in several ways:

- the application program issues ET or BT;
- the application program issues SRRCMIT or SRRBACK;
- the application or some other agent issues ATRCMIT or ATRBACK.

In the first case, the TM proxy is aware of the syncpoint decision. In other cases, from ATM's point of view, the syncpoint is unsolicited. If the user is at global transaction status, an unsolicited syncpoint has no effect on Adabas databases, other than to cause held ISNs to be released. On the other hand, if an unsolicited syncpoint occurs when Adabas changes are pending, ATM is instructed to commit the changes; in this case, also, held ISNs will be released.

If unsolicited syncpoints are acceptable in your system, specify TRNCTL=GLOBAL. Otherwise, specify TRNCTL=LOCAL to treat unsolicited syncpoint requests as errors. If TRNCTL=LOCAL and an unsolicited syncpoint causes a user's Adabas changes to be committed, the next Adabas call issued by the user returns response code 243.

If ATM runs with `TMSYNCMGR=RRMS`, and `HSTSYNC=YES` is specified for IMS/TM systems whose transactions are coordinated by RRMS, `TRNCTL=GLOBAL` must also be specified for these IMS systems.

XH : Extended Hold Feature

Parameter	Parameter Type	Use	Possible Values	Default
`XH`	Job	Indicates whether P and M options on `ET` and `BT` commands will be honored.	YES | NO	NO

Possible values:

Value	Description
YES	P and M options on `ET` and `BT` commands will be honored. For any application environment in which prefetch or multifetch command options can be used, the setting YES should be used.
NO	P and M options on `ET` and `BT` commands will not be honored.

If Adabas Vista is present in an application job or TP environment, the value YES will take effect, regardless of the `XH` parameter setting.

For more information on extended hold processing, see Extended Hold Processing in section Termination Commands: ET and BT.

4 Parameter Summary Table

Job Parameters	Runtime Parameters
ATM : Activate ATM Processing	TMDRQ : Request Queue Size
CORGROUP : System Coordinator Group Name	TMDYNTCIDS : Dynamic Client Identifiers
GENOP : Generate OP Commands	TMETDATA : ET Data Control
HSTSYNC: Use Host System Transaction Manager	TMGTNA : Global Non-Activity Time Limit
LCLSYNC : Use Client-Side Transaction Manager	TMGTT : Global Transaction Time Limit
LGRECNO : Number of Log Record Entries	TMLOG : Logging Control
MAXDB : Maximum Number of Current Databases	TMMSGSEV : Severity Threshold for Warning Messages
SERIAL : Emergency Serial ET Commands	TMRESTART : Restart Control
SVC : SVC Number	TMSYNCMGR : Support for External Transaction Coordinators
SYNCMODE : Syncpoint Processing Mode	TMTCIDPREF : Dynamic Client Identifier Prefix
SYNCONBT : Generate External Syncpoint on BT Command	
SYNCONCL : Generate External Syncpoint on CL Command	
SYNCONET : Generate External Syncpoint on ET Command	
TRANMODE : Transaction Model	
TRNCTL : Transaction Control	
XH : Extended Hold Feature	

Index