

Adabas Transaction Manager

Adabas Transaction Manager Parameters

Version 8.1.2

June 2008

Adabas Transaction Manager

This document applies to Adabas Transaction Manager Version 8.1.2 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.

Copyright © Software AG 2008. All rights reserved.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.

Table of Contents

1 Adabas Transaction Manager Parameters	1
2 Runtime Parameters	3
ATM Runtime Parameters	4
Recommendations for Adabas Runtime Parameters when using ATM	11
Enforced Runtime Parameters	12
3 ATM Client Runtime Controls	13
Usage of Client Runtime Controls	14
Descriptions of Client Runtime Controls	14
4 Parameter Quick Reference	27
Index	29

1 Adabas Transaction Manager Parameters

This document describes the parameters used for Adabas Transaction Manager.

- [ATM Runtime Parameters](#)
- [ATM Client Runtime Controls](#)
- [Parameter Quick Reference](#)


2 Runtime Parameters


- ATM Runtime Parameters 4
- Recommendations for Adabas Runtime Parameters when using ATM 11
- Enforced Runtime Parameters 12

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 ATM transaction manager, 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 ATM transaction manager.

The following section describes each ATM runtime parameter:

- [TMDRQ : Size of ATM Request Queue](#)
- [TMETDATA : ET Data Control](#)
- [TMGTT : Global Transaction Time Limit](#)
- [TMLOG : ATM Manager Logging Control](#)
- [TMMSGSEV : Severity Threshold for Warning Messages](#)
- [TMRESTART : ATM Restart Control](#)
- [TMSYNMGR : Support for External Transaction Coordinators](#)
- [TMTCIDPREF : Dynamic Client ID Prefix](#)

TMDRQ : Size of ATM Request Queue

Parameter Type	Use	Possible Values	Default
Runtime	Sets the number of entries allowed in the transaction manager’s internal request queue.	10 - 32767	10

The ATM transaction manager uses an internal request queue for scheduling some of 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 ATM manager. 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

ATM will allow 250 entries in its internal request queue.

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

TMETDATA : ET Data Control

Parameter Type	Use	Possible Values	Default
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 Adabas Transaction Manager (ATM) instances on the network.

Possible values:

Value	Explanation
ATM	<p>ET data is always stored in and read from the ATM transaction manager's database, without regard to the database ID indicated by the Adabas command.</p> <ul style="list-style-type: none"> ■ treats ET data as belonging to a global transaction rather than to a database; ■ there is just one copy of a client'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 <code>ETDB</code> parameter; ■ should not be used if the <code>EmergencySerialETs</code> runtime control will be set to YES or FORCE for any client environment in which ET data is stored or read.
TARGETS	<p>ET data is stored during the commit process in all changed databases that are running with <code>DTP=RM</code>.</p> <p>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 <code>DTP=RM</code>. If the database runs with <code>DTP=NO</code>, the ET data is written to the database only after any open global transaction has been committed.</p> <p>If the client 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.</p>

Value	Explanation
	<ul style="list-style-type: none"> ■ treats ET data as belonging to a database or a local database transaction rather than to a global transaction; ■ a client's ET data might exist in several databases in a system; ■ a client can have several different ET data values in different databases at the same time; ■ 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.



Notes:

1. The recommended setting for this parameter is the default, TARGETS. This setting eliminates the dangers associated with running in serial mode when the transaction manager is unavailable.
2. See also the section entitled *ET Data Storage* in your Adabas Transaction Manager operations documentation.

Example

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

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

TMGTT : Global Transaction Time Limit

Parameter Type	Use	Possible Values	Default
Runtime	Sets time limit during which a global transaction can be open without being prepared.	1 - 16777215	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, ATM automatically backs out the global transaction from all affected databases. The next time the client issues a transactional Adabas command, a response code 9 is returned.

It is recommended that TMGTT be smaller than the lowest TT parameter of any database running with DTP=RM.

The TMGTT parameter does not supersede the TT parameter settings of other databases. If a database times out a client who has an unprepared global transaction open, the global transaction is subsequently backed out. A response code 9 is returned if the client then issues a transactional command.



Note: Transaction timeout can happen when a client session stops issuing commands for a period of time. In these circumstances the transaction manager has no opportunity to give

a response code to the client, to indicate that the timeout has taken place. In such cases, the pending response code will be preserved and returned to the client at the first possible opportunity. For further details, please refer to *Pending Response Codes* in the Adabas Transaction Manager introduction.

Example

ATM 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 Manager Logging Control

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

The ATM manager 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 manager writes every record to an ATMLOG dataset. The ATM manager 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 <code>TM NOLOG</code> .
NO	The ATM manager will not write log records to the ATMLOG datasets. Logging to the ATMLOG datasets can be activated at any time using the operator command <code>TM LOG</code> . The ATM manager 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 manager.

On z/OS and VSE systems, if the ATM manager writes log records to the ATMLOG (`TMLOG=YES` or `TM LOG` issued) and the dataset currently in use becomes full, the manager 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 manager is to maintain its internal log buffer areas and write all log records to the ATMLOG datasets.

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

TMSGSEV : Severity Threshold for Warning Messages

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

Every message that the ATM manager sends to the console and to the DDPRINT dataset has a severity level. Message severity levels are explained in the Adabas Transaction Manager *Messages and Codes*.

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

All messages created by the ATM manager are written to DDPRINT regardless of the `TMSGSEV` 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 manager should not send message of severity less than 4 to the console.

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

TMRESTART : ATM Restart Control

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

When an ATM manager restarts, it rebuilds its global transaction list as nearly as possible to its latest state when it was last executing. It then attempts to complete or back out any incomplete global transactions. The `TMRESTART` parameter determines the manager'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 manager. The details of these transactions are deleted from ATM's recovery information, the originating clients 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 clients 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 manager 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 Type	Use	Possible Values	Default
Runtime	Indicates whether or not the ATM manager is to interact with an external transaction coordinator.	NONE RRMS	NONE

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

Possible values:

Value	Description
NONE	The ATM manager 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 ATM's client proxy component and not with the ATM manager.
RRMS	The ATM manager 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 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 Com-plete or 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
```

TMTCIDPREFIX : Dynamic Client ID Prefix

Parameter Type	Use	Possible Values	Default
Runtime	Defines the first one or two characters of dynamically allocated Client IDs.	up to 2 alphanumeric characters	TM

To aid interpretation of Adabas Online Services displays and diagnostic logs, the ATM manager dynamically allocates an 8-byte client identifier (TCID) for each client session that engages in transactional activity. These client identifiers are simply labels, and have no operational significance.

The format of a dynamically allocated TCID is

```
ppnnxxxx
```

– where

pp is the value of the TMTCIDPREFIX parameter, left-justified and with trailing blanks if required.

nn is the 2-byte binary ATM database ID.

xxxx is a 4-byte binary suffix. Suffixes are allocated in sequence and are reused. The first TCID to be allocated has a suffix of zeros.

Example

ATM will assign dynamic TCIDs beginning with the string "DY".

```
ADARUN  PROG=ADANUC , DTP=TM , TMTCIDPREFIX=DY
```

Recommendations for Adabas Runtime Parameters when using ATM



Caution: Before you use 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 ATM transaction manager 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 client proxy forwards these commands to the transaction manager for scrutiny.

Furthermore, the transaction manager 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, ATM transaction manager 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 manager, the NU parameter needs to be high enough for the maximum concurrent number of open transactions in the system, plus the value of the NT parameter.
NC	The command queue size for the transaction manager 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 manager should be higher than might be expected.
TMLOG=NO	Numerous I/O operations to the TMLOG 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 manager.

Before using Adabas Transaction Manager in a production environment, remember also to review the client runtime controls for jobs and TP systems that will use ATM. Pay particular attention to the setting of [NumberLogRecordEntries](#).

Enforced Runtime Parameters

When the transaction manager 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 manager. 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 Client Runtime Controls

- Usage of Client Runtime Controls 14
- Descriptions of Client Runtime Controls 14

Usage of Client Runtime Controls

Correct operation of the ATM transaction manager and the Adabas Transaction Manager client proxy depends on the settings of client runtime controls. 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 client proxy checks the Adabas System Coordinator system file for client runtime controls that match the name of the currently executing job. If no exact match is found, a wild-card match is attempted. Failing this, the control values 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.

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



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

Descriptions of Client Runtime Controls

This section describes each ATM runtime control:

- `ATM ON/OFF`: Activate ATM Processing
- `System Coordinator Group Name`
- `Maximum Number of Databases`
- `Number of Log Record Entries`
- `Transaction Control`
- `Emergency Serial ET Commands`
- `Allow Serial Remote RM Coordination`
- `Generate OP Commands`
- `Syncpoint Processing Mode`
- `Transaction Model`
- `Generate External Syncpoint on BT Command`
- `Generate External Syncpoint on CL Command`
- `Generate External Syncpoint on ET Command`
- `Use Client-Side Transaction Manager`
- `Use Host System Transaction Manager`

- Extended Hold

ATM ON/OFF: Activate ATM Processing

Parameter Type	Use	Possible Values	Default
Client runtime	Indicates whether or not ATM is to provide transaction coordination for this client session.	ON OFF	OFF

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 client session.

System Coordinator Group Name

Parameter Type	Use	Possible Values	Default
Client runtime	Identifies the Adabas System Coordinator group in which the client session or TP system will execute.	Any valid Adabas System Coordinator group name	A valid group name must be provided

Maximum Number of Databases

Parameter Type	Use	Possible Values	Default
Client runtime	Sets the maximum number of databases with which a client session can have an active session at one time.	number	4

The setting should include all databases without regard to their DTP parameter settings.

Number of Log Record Entries

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

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

Possible values:

Value	Description
0	The client 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 `NumberLogRecordEntries` client control has no effect on the logging that takes place in the transaction manager's address space. Logging by the transaction manager is controlled by the ADARUN parameter `TMLOG`.

Transaction Control

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

The setting of this control 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 client proxy, or by a software component other than ATM or ATM client proxy.

Unsolicited syncpoint requests can occur if:

- The option `Client-sideTransactionManager` is in effect 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 transaction manager 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 client proxy.

Possible values 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 client proxy. If unsolicited syncpoints are not valid in your environment, specify <code>LOCAL</code> so that any unsolicited syncpoint request is reported as an error. Specifying <code>TRNCTL=LOCAL</code> 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.

Value	Description
GLOBAL	<p>Decision to commit or back out a global transaction can be made by a software component other than ATM or ATM client proxy. GLOBAL is required if unsolicited syncpoint requests are acceptable.</p> <p>Note: Running with GLOBAL in any environment other than CICS/RMI can cause the ATM client proxy to make additional calls to the ATM transaction manager to check transaction status. It is recommended to set <code>Transaction Control</code> to LOCAL wherever possible to eliminate these extra calls.</p>

CICS/RMI Considerations

When running with the `Client-side Transaction Manager` option set to 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 client proxy is aware of the syncpoint decision.

In all other cases, from the client proxy's point of view, the syncpoint is unsolicited. If the client session 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 GLOBAL for the `TransactionControl` client control - otherwise, specify 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/Os occurs. If ATM CICS/RMI is installed and active, and client runtime control `TransactionModel` is set to MESSAGE, 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 [TransactionModel](#) control for details of a different way of responding to unsolicited syncpoints.

If the client control `TransactionControl` is set to LOCAL 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 client runtime control `TransactionModel` set to MESSAGE, such functions are likely to

- fail with a NAT3243 error when `TransactionControl` is set to LOCAL; or

- be unable to back out in case of errors when `TransactionControl` is set to `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 client session that operates with the client runtime control `TransactionModel` set to `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 ATM's client proxy is aware of the syncpoint decision. In other cases, from ATM's point of view, the syncpoint is unsolicited. If the client session 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, set the client control `TransactionControl` to `GLOBAL` - otherwise, set `TransactionControl` to `LOCAL` to treat unsolicited syncpoint requests as errors. If `TransactionControl` is set to `LOCAL` and an unsolicited syncpoint causes a client sessions's Adabas changes to be committed, the next Adabas call issued by the user returns response code 243.

If ATM runs with the `HostSystemTransactionManager` option is specified for IMS/TM systems whose transactions are coordinated by RRMS, `TransactionControl` must be set to `GLOBAL` for these IMS systems.

Emergency Serial ET Commands

Parameter Type	Use	Possible Values	Default
Client runtime	Indicates whether or not the ATM client proxy is to switch a client session from DTP to serial ET/BT mode if the local ATM transaction manager becomes unavailable.	YES NO FORCE	FORCE

Possible values:

Value	Description
YES	The ATM client proxy is to switch a client session from DTP to serial ET/BT mode if the local transaction manager becomes unavailable. A response code will be returned to indicate that the local TM is unavailable.
NO	The ATM client proxy is not to switch a client session from DTP to serial ET/BT mode if the local TM becomes unavailable.
FORCE	A client session 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 client session to serial ET/BT mode when the session has a transaction open, the client might receive unexpected response codes, even if the runtime control value for serial processing is set to FORCE. For example, the transaction might be backed out, and response 9 returned to the client. 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 transaction manager 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 clients to continue processing without noticing the switch.

Allow Serial Remote RM Coordination

Parameter Type	Use	Possible Values	Default
Client runtime	Indicates whether the client's local Transaction Manager should use serial ET commands to remote RMs which are not signed on for Distributed Transaction Processing in the client's System Coordinator group.	YES NO	YES

Possible values:

Value	Description
YES	If the client session changes one or more remote RMs which are currently not signed on for Distributed Transaction Processing, to any Transaction Manager within the scope of the client's System Coordinator Group, the client's local Transaction Manager will treat these RMs as if they were running with <code>DTP=NO</code> . That is, such a transaction will be committed in the normal way, using the two-phase commit protocol with respect to any signed-on RMs, and then ATM will issue <code>ET</code> commands to any changed remote RMs that are not signed on for DTP.
NO	If the client session attempts a transaction that changes an RM which is currently not signed on for DTP to any Transaction Manager within the scope of the client's COR Group, the transaction will usually be rejected with a suitable response code. In some circumstances the transaction may be allowed to complete normally, but only if it can be completed safely without the need of a two-phase commit operation.



Note: If YES is specified, and a transaction changes one or more remote RMs which are not signed on for DTP, and a system or component failure occurs during the commit process, ATM cannot guarantee integrity of the transaction.



Note: If the `EmergencySerialETCommands` control is set to YES or FORCE, and a client session is switched into serial ET/BT mode, serial remote RM coordination will take effect as long as the session is in serial ET/BT mode, regardless of the setting of the `AllowSerialRemoteRMCoordination` control.

Generate OP Commands

Parameter Type	Use	Possible Values	Default
Client runtime	Indicates whether or not the ATM client proxy is to generate an <code>OP</code> command when a session uses a database without having issued an <code>OP</code> .	NO YES	NO

Possible values:

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

Syncpoint Processing Mode




This parameter has been replaced by the `TransactionModel` runtime control.

Transaction Model

Parameter Type	Use	Possible Values	Default
Client runtime	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 <code>ET</code> , <code>BT</code> , <code>OP</code> and <code>CL</code> 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; <code>ET</code> and <code>CL</code> 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:** Setting `TransactionModel` to `MESSAGE` is equivalent to `SYNCMODE=FULL` with ATM Version 1.2.
-  **Note:** Setting `TransactionModel` to `DYNAMIC` is equivalent to `SYNCMODE=ADABAS` with ATM Version 1.2.
-  **Note:** There is no `TransactionModel` setting equivalent to `SYNCMODE=ALL` with ATM Version 1.2. The reason is that there is no distinction between the settings `SYNCMODE=ALL` and `SYNCMODE=FULL` with more recent versions of ATM, 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 `ExtendedHold` client control.

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

Generate External Syncpoint on BT Command

Parameter Type	Use	Possible Values	Default
Client runtime	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 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 setting for this control, a BT command causes all Adabas changes to be backed out.

Generate External Syncpoint on CL Command

Parameter Type	Use	Possible Values	Default
Client runtime	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 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 `HostSystemTransactionManager` option is specified, a setting of YES for the client control `GenerateExternalSyncpointOnCL` will be ignored, and the value NO will take effect.

Generate External Syncpoint on ET Command

Parameter Type	Use	Possible Values	Default
Client runtime	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 setting is ignored.
NO	ATM will not request a commit syncpoint from any external coordinator when it processes an ET command.

Regardless of the setting for this control, ET command causes any pending Adabas changes to be committed.

For IMS/TM systems for which the `HostSystemTransactionManager` option is specified, a setting of YES for the client control `GenerateExternalSyncpointOnET` will be ignored, and the value NO will take effect.

Use Client-Side Transaction Manager

Parameter Type	Use	Possible Values	Default
Client runtime	Indicates whether or not transactions in this client environment are to be controlled by the environment's native syncpoint manager.	YES NO	NO

Possible values:

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



Note: CICS under z/OS or VSE/ESA is the only client environment for which a setting of YES is currently supported for this control.



Note: The `Client-sideTransactionManager` and `HostSystemTransactionManager` options are mutually exclusive. However, setting `Client-sideTransactionManager` to YES does not preclude the use of RRMS as an external transaction coordinator for client sessions in

other jobs or TP systems, since RRMS interfaces directly with the ATM client sessions in other jobs or TP systems and not with the client proxy

CICS/RMI Considerations

When defining client runtime controls for a CICS/RMI environment:


- Setting `Client-sideTransactionManager` to 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.
- Setting `Client-sideTransactionManager` to YES activates the RMI so that the CICS Syncpoint Manager coordinates transactions across all resource managers.

Use Host System Transaction Manager


Parameter Type	Use	Possible Values	Default
Client runtime	Indicates whether or not transactions in this client environment are to be controlled by the host system's native transaction manager.	YES NO	NO

Possible values:

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

 **Caution:** If this control is set to YES for an inappropriate type of job or client environment, or for a job or environment which has not been correctly configured to use the host system's transaction manager, sessions executing in that job will experience response code 240, subcode 444, and will not function correctly.

 **Note:** The setting YES is currently supported only for single-user, single-TCB batch jobs or TSO sessions, and for Com-plete and IMS/TM systems, running under z/OS.

 **Note:** The `HostSystemTransactionManager` and `Client-sideTransactionManager` options are mutually exclusive.

Extended Hold

Parameter Type	Use	Possible Values	Default
Client runtime	Indicates whether P and M options on ET and BT commands will be honored when a distributed transaction is terminated by a series of ET or BT commands.	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 setting specified for the `ExtendedHold` client control.

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

4 Parameter Quick Reference

Client Runtime Controls	Runtime Parameters
Activate ATM Processing	TMDRQ : Request Queue Size
System Coordinator Group Name	TMETDATA : ET Data Control
Maximum Number of Databases	TMGTT : Global Transaction Time Limit
Number of Log Record Entries	TMLOG : Logging Control
Transaction Control	TMSGSEV : Severity Threshold for Warning Messages
Emergency Serial ET Commands	TMRESTART : Restart Control
Allow Serial Remote RM Coordination	TMSYNCMGR : Support for External Transaction Coordinators
Generate OP Commands	TMTCIDPREF : Dynamic Client Identifier Prefix
Syncpoint Processing Mode	
Transaction Model	
Generate External Syncpoint on BT Command	
Generate External Syncpoint on CL Command	
Generate External Syncpoint on ET Command	
Use Client-Side Transaction Manager	
Use Host System Transaction Manager	
Extended Hold	

Index

A

Adabas Transaction Manager

- ATM manager logging option, 7
- external transaction coordinator support, 9
- global transaction time limit, 6
- identifying databases for ET data, 5
- restart action control, 9
- severity level for suppression of warning messages, 8
- size of the ATM request queue, 4
- TCID client ID prefix, 10

T

TMDRQ

- ADARUN parameter, 4

TMETDATA

- ADARUN parameter, 5

TMGTT

- ADARUN parameter, 6

TMLOG

- ADARUN parameter, 7

TMMMSGSEV

- ADARUN parameter, 8

TMRESTART

- ADARUN parameter, 9

TMSYNCMGR

- ADARUN parameter, 9

TMTCIDPREF

- ADARUN parameter, 10

