Operational Changes and Enhancements

This section describes the operational changes and enhancements provided by Version 8.1.

- Improved Operational Efficiency
- Terminology Change: Job Parameters Replaced By Client Runtime Controls
- Avoid Error RSP240 Subcode 444
- Alternate Configuration File
- Revised ETID Support
- CL Command to a Database at ET Status
- Terminating a Transaction – Application Protocol
- Support for Triggers
- Removal of the Global User Queue
- Pending Response Codes
- Mandatory Use of Unmodified ADALNK
- Default Change: ATM (Activate ATM Processing) Parameter
- Change of Parameter Length: TMCIDPREF (Dynamic Client ID Prefix) Parameter
- New Client Control: Allow Serial Remote RM Coordination
- Deactivation of TMGTNA (Non-Activity Timeout) Parameter
- Deactivation of TMDYNTCIDS (Dynamic Client IDs) Parameter
- Deactivation of SVC Client Control

Improved Operational Efficiency

Adabas Transaction Manager has been enhanced internally to become more efficient, in addition to the changes to support the new Adabas Version 8.1 evolution. For example, the Transaction Manager now focuses only on your currently active transactions whereas previous releases took on more internal overheads. This will show some gains in efficiency over previous releases for sites that have a high number of concurrent Adabas sessions. Over the whole system, in a single-system laboratory test, cpu reductions were experienced across the sum of clients, Transaction Manager, and Adabas RMs. These reductions are likely to be seen in higher-end production systems rather than small-scale test systems.

Terminology Change: Job Parameters Replaced By Client Runtime Controls

ATM job parameters are now referred to as Client Runtime Controls.

For more information, see section ATM Parameters.

Avoid Error RSP240 Subcode 444

In earlier versions of Adabas Transaction Manager, the screen for defining client runtime controls (job parameters) for ATM offered the value YES for the parameter `UseHostSystemTransactionManager`. ATM Version 8.1 validates this runtime control more rigorously than earlier versions, to ensure that the host-system transaction manager is used when, and only when it is required. It is therefore possible, if you use a configuration file that has been converted from a previous version, that inappropriate settings for this control will exist, and will cause runtime errors.

At Version 8.1, if you try to use a client session for which the runtime control `UseHostSystemTransactionManager` is set to YES and ATM’s client proxy has *not* been configured to use the host system’s transaction manager, response code 240, subcode 444 will be given, and you will be unable to continue processing normally. You should therefore check that the runtime control `UseHostSystemTransactionManager` is set correctly in all client runtime controls; set to YES only for client jobs or environments for which ATM supports the host system’s transaction manager, and which have been correctly configured to use the host transaction manager.

Alternate Configuration File

ATM’s client runtime controls are stored in a system file (configuration file). Many sites consider the configuration file to be a critical point in operations. Consequently, availability is critical. ATM now works with the Adabas System Coordinator to provide an alternate file facility, so that if one becomes unavailable the other is used automatically and dynamically. The switching to and from alternate is done on a session-by-session basis, dynamically. This means that at any point in time some sessions may be using one file, and some the other. This is as intended. If you wish to force all sessions to use the same file you can make the other unavailable for a long period and all sessions will move to it over time.

For more details please refer to the *Adabas System Coordinator* documentation.

Note:
It is your responsibility to make sure the contents of both configuration files are identical. See the *Adabas System Coordinator* documentation for more details.

Revised ETID Support

Previous versions of Adabas Transaction Manager worked on the assumption that a client would make consistent use of single ETID across all database session, or else no ETID would be used. However, there are requirements for a client to use an ETID for certain database sessions, but no ETID for others.

From Version 8.1, ATM supports more flexible use of ETIDs. A client session can use ETIDs for some database sessions, while using other databases concurrently without ETID. Software AG strongly recommends that a client use no more than one non-blank ETID at a time.

For further details, see Introduction: Interfacing to Adabas Applications.

CL Command to a Database at ET Status

If a `CL` command is issued to a database that is at `ET` status, when a global transaction is in progress against other databases, the `CL` command is processed, but the global transaction is not affected.

Terminating a Transaction – Application Protocol

Applications are required to behave correctly when terminating (commiting or backing out) a global transaction. The correct protocol is that a sequence of consistent transaction directives must be issued – one directive to each of the databases that was affected by the global transaction. A transaction directive is an `ET` or `BT` command (or a `CL` or `OP` command that implies the required `ET` or `BT` operation). Applications that do not behave correctly in this way will receive response code 240 subcode 496 when trying to continue with transactional work without first having issued all required directives.

Correct transaction behavior is quite normal for the vast majority of applications, since most pre-date the introduction of Adabas Transaction Manager. For example, Natural generates correct transaction behavior, and has done so for many releases.

Adabas Transaction Manager acts to provide full global transaction integrity as soon as the first directive in the series is received from the application. The application must complete the sequence of transaction directives for two main reasons. First, you may wish to deactivate Transaction Manager in an emergency, so the application must be programmed to behave correctly when terminating transactions. Second, it is imperative that the transaction state for all databases is mirrored within the application and in Adabas Transaction Manager; correct transactional behavior is the only way to ensure that an application's operation can be regulated to be fully consistent with the distributed transaction model.

Support for Triggers

Triggers can execute global transactions under the control of Adabas Transaction Manager. Refer to the documentation for the Adabas System Coordinator to find out how to configure the System Coordinator for a trigger environment.

If your application causes participating triggers to be fired, and you require a global transaction to include changes made by both the application program and the participating trigger, you must ensure that the very first change command (store, delete or update) of the global transaction is issued by the application program, not by the trigger.

Removal of the Global User Queue

Operation of an ATM transaction manager has been streamlined by eliminating the Global User Queue. The Transaction Manager focuses on active transactions, without the need to maintain a list of active clients.

Pending Response Codes

Sometimes a transaction is terminated in such a way that the owner should receive a non-zero response code, but in circumstances in which it is not possible to return a response code; for example, the transaction manager backs the transaction out because its global transaction time limit has been exceeded. When this happens, the manager stores details of the pending response code in a new list, and returns it to the client at the first possible opportunity. Pending response codes can be listed and displayed using ATM's Online Services application.

Mandatory Use of Unmodified ADALNK

As in previous releases, there is a requirement to use an unmodified ADALNK in certain areas. With Version 8.1 this restriction has been reduced to unmodified ADALNK being needed only when running standalone Adabas utility jobs.

Default Change: ATM (Activate ATM Processing) Parameter

The default value for the `ATM ON/OFF` (Activate ATM Processing) parameter has been changed from ON to OFF.

For more information, see ATM Parameters.

Change of Parameter Length: TMCIDPREF (Dynamic Client ID Prefix) Parameter

The maximum length of this parameter has been reduced from 3 to 2 characters, to allow for a greater number of client IDs. A three-character value will be accepted, but only the first two characters will be used.

For more information, see ATM Parameters.

New Client Control: Allow Serial Remote RM Coordination

With previous versions of Adabas Transaction Manager, any attempt to change a `DTP=RM` database which is not signed on for Distributed Transaction Processing would be rejected with response code 240, subcode 188. To allow greater flexibility in configuring your network of systems, Transaction Managers and RMs, this behavior is now governed by a new client control.

By default, ATM will assume that a remote RM which is not signed on to any Transaction Manager in the scope of the client's COR group should be treated as if it were running with `DTP=NO`. That is, if the client session changes such a database, those changes will be committed using serial `ET` commands, since the database cannot currently support the two-phase commit protocol. The new client control, `AllowSerialRemoteRMCoordination`, can be used to override this behavior, so that response 240, subcode 188 will be given, as before, when an attempt is made to change a remote RM which is not signed on for DTP.

For further details, see the description for `AllowSerialRemoteRMCoordination`.

Deactivation of TMGTNA (Non-Activity Timeout) Parameter

Previous versions of Adabas Transaction Manager provided a global non-activity timeout feature, to prevent dormant or lost client sessions from retaining resources. The Adabas System Coordinator now provides such a facility in an integrated manner, which is effective not only for Adabas Transaction Manager, but also for other products which use the System Coordinator. Therefore, from Version 8.1, ATM's non-activity timeout facility has been removed, and the ADARUN parameter `TMGTNA` is no longer effective.

Deactivation of TMDYNTCIDS (Dynamic Client IDs) Parameter

Previous versions of Adabas Transaction Manager used the runtime parameter `TMDYNTCIDS` to limit the number of temporary, dynamic Client Identifiers that a transaction manager could assign, and there was a theoretical maximum of 64k. From Version 8.1, a transaction manager will be able to assign up to 16,777,215 Client Identifiers. The ADARUN parameter `TMDYNTCIDS` is therefore no longer effective.

Deactivation of SVC Client Control

With previous versions of Adabas Transaction Manager, a client control (job parameter) was used to identify the Adabas SVC for which ATM was required to provide transaction coordination. ATM would not interfere in any way with commands that were issued under a different Adabas SVC.

From Version 8.1, the client session's Adabas SVC will be taken to be the SVC used by the Adabas link module.