

Restart and Recovery

- ATM Recovery Records
 - Suspect Transaction Records
 - Adabas Resource Locks
 - ATM Transaction Completion
 - Recovery with the CICS Resource Manager Interface
 - Recovery with RRMS
-

ATM Recovery Records

The ATM transaction manager records details of incomplete, prepared transactions in the recovery file. Whenever a transaction is completed, its recovery record is deleted. If the recovery records in this file are lost at a time when incomplete, prepared transactions exist in the system, ATM is not able to guarantee the integrity of those transactions.

You can use the Online Services application to check for incomplete transactions in the system.

Caution:

When incomplete transactions exist, you must *not*

- change the Node ID of the Adabas System Coordinator daemon within which the ATM transaction manager is running as a service;
- change any of the following transaction manager parameter values: TMETDATA , TMSYNCMGR, or TMTCIDPREF.

Suspect Transaction Records

ATM uses suspect transaction records (STJ) in the recovery file to record all known details of incomplete transactions that have been purged from the system as a result of intervention by the operator or database administrator.

Incomplete transactions can be purged as follows:

- using the Stop Transaction function provided by Online Services, or
- as a result of a forced restart using the runtime parameter TMRESTART.

Online Services can be used to browse through the suspect transaction records..

Alternatively, you may use the sample program ATMSPRNT in the supplied JOBS library to produce a readable printout of the suspect transaction records. See Print Suspect Transaction Records for more information. Use the comments in the job when modifying it to conform to site requirements.

There is no automatic housekeeping of suspect transaction records. It is intended for emergency use only. The database administrator should purge these records from time to time, after making sure that the information contained in it is no longer required.

Adabas Resource Locks

During the life of a transaction, an application gains ownership of certain database resources associated with the transaction. For example, records that are changed, or new and old unique descriptor values. Adabas locks these resources against use by other users or applications until the transaction is completed; that is, committed or backed out.

In the case of a global transaction, Adabas secures this information, together with the necessary recovery information, in its Work dataset. The information is not discarded until the end of the two-phase commit process for the owning transaction. It also survives database restart if the transaction has successfully completed the prepare phase.

ATM Transaction Completion

- How ATM Handles Incomplete Transactions
- Transaction Manager failover
- Undetected Database Restarts

How ATM Handles Incomplete Transactions

When a database nucleus with the runtime parameter `DTP=RM` is started, it signs on to the local ATM transaction manager and provides the details of any incomplete global transactions.

The TM then attempts to complete each of the transactions by instructing each relevant database to commit its transaction or roll back its changes, as appropriate. If any of the incomplete transactions have branches in other systems, the partner ATM managers in those systems are also instructed to commit or roll back, as appropriate.

Meanwhile, the resources that were changed by the incomplete transaction remain unavailable to other users.

When started, an ATM manager obtains, from its recovery file and from any partner ATM managers in other systems, details of any global transactions that were incomplete when it last terminated. It then attempts to complete each of these transactions by instructing each relevant database to commit its transaction or roll back its changes, and each partner ATM manager to commit or roll back its transaction branches, as appropriate.

The integrity of global transaction is thus secured across restarts of critical components.

ATM does not decide whether to commit or back out a prepared transaction that is controlled by an external transaction coordinator. Any such transaction remains in doubt until the external coordinator resolves it.

Transaction Manager failover

Quite obviously, if the transaction management service fails (this usually implies the System Coordinator daemon has also failed) it should be restarted immediately. In most systems automatic restart management makes sure this happens immediately.

Unplanned outage in a single system

When the transaction manager is unavailable, existing clients will continue to run; they might only be impacted when they require services from the transaction manager, for example to seek transaction completion. Transactions in mid-completion at the time when the transaction manager outage occurs will be resolved when the transaction manager returns. Transactions that were known to the failed transaction manager but which had not yet been committed by the application will be terminated as in previous releases. New transactions will proceed according to the configuration you use, refer to Serial Mode Transaction Control for more information.

Unplanned outage in a multi-system

When multiple Transaction management services collaborate as peers in a group across multi-systems they automatically help each other by taking on responsibility for any transactions that were in-flight when the failure occurred.

In general, the same things happen as far as possible as though it is a single-system running. But in a multi-system there is also the possibility that dynamic transaction routing clients are running. If so, these are allowed to continue according to the usual rules of dynamic transaction routing systems; they will migrate (if allowed to do so) to other systems where the local transaction manager will take over responsibility for them, on demand, in exactly the same way as they are allowed to migrate across systems in normal operation.

Similar to a single system, an unplanned outage of a transaction manager usually causes an automatic restart so everything resolves immediately with no need for assistance by peer transaction managers. Every chance is given for this automatic activity to take place. Therefore, no peer transaction managers will become involved until the unplanned outage lasts for more than 60 seconds without the failing manager resuming.

After 60 seconds one of the other peer managers automatically takes over responsibility for the duties of the failed manager for the period of the outage. This role is referred to as being the agent for the failed manager. The agent will try to complete the transactions that were in-flight and will terminate those that cannot be completed, etc. And of course, any dynamic routing migration will be allowed to take place on demand as normal.

At some point the failed manager is likely to return, at that point it will negotiate return to normal duties with the agent, and the agent ceases any further involvement.

Undetected Database Restarts

In an Adabas environment without Transaction Manager, it can happen that a client session changes a database, and that the database is then recycled without the client session being made aware. This can be prevented (see below). However, if it is allowed to happen, the client might then make more changes and then commit, receiving a positive response from Adabas - not knowing that the changes made in the previous Adabas session were backed out as a result of the recycle. The same situation applies when Transaction Manager is present. To avoid this problem, one or both of the following approaches must be

used:

- Set the ADARUN parameter OPENRQ=YES for all databases, so that a restarted database will give response code 9, subcode 66, indicating that a new OP is required.
- Use ETIDs for all client sessions, so that “backout during system open” sets a pending response 9 for the affected session.

Recovery with the CICS Resource Manager Interface

For a system which has been configured to use the CICS Resource Manager Interface, the following recovery process occurs at CICS startup (or soon after):

1. The Adabas Transaction Manager CICS re-synchronization driver program (ATMRMIRS) obtains from the local transaction manager a list of all prepared (but incomplete) transactions that were controlled by this CICS system.
2. CICS is then instructed to re-synchronize each of these transactions.
3. During this process, CICS indicates whether each of these transactions should be backed out or unconditionally committed.
4. When the last incomplete transaction has been processed, the transaction manager writes a console message indicating that the re-synchronization process is complete.

In order to re-synchronize incomplete transactions in this way, CICS logging must be active and CICS must be warm started. If CICS logging is not in use or if CICS is cold started when there are incomplete transactions in the system, transaction integrity cannot be guaranteed.

Recovery with RRMS

If RRMS is already active when the ATM transaction manager starts up, which is normally the case, ATM re-synchronizes in cooperation with RRMS to resolve any incomplete transactions that were under RRMS control.

If RRMS is unavailable when the ATM manager starts, the ATM manager issues a warning message to the console and waits until RRMS becomes available. Then it re-synchronizes.

If a critical component of RRMS becomes unavailable while ATM is operating, a warning message is issued to the console. In some cases, ATM is able to continue processing and initiates re-synchronization processing as soon as the missing component is reactivated.