

# Transaction Processing

- Transaction Coordination Priority
  - Heuristic Completion of Prepared Transactions
- 

## Transaction Coordination Priority

The ATM instances involved in a transaction can be running with TMSYNCMGR=RRMS or TMSYNCMGR=NONE.

When a transaction is under the control of a client-side transaction coordinator such as the CICS Syncpoint Manager, each ATM transaction manager is aware of this and does not involve RRMS, even if its own parameters indicate that its RRMS interface is to be activated (TMSYNCMGR=RRMS).

The transaction executing under the CICS/RMI can change Adabas and DB2 databases, for example, and these changes are coordinated by the CICS Syncpoint Manager. For such a transaction, the ATM manager behaves as if it were running with TMSYNCMGR=NONE.

In order for the transactions of a job or IMS TM system to be coordinated by RRMS, you must set the client runtime control HostSystemTransactionManager to YES, and the ATM manager must run with parameter TMSYNCMGR=RRMS.

## Heuristic Completion of Prepared Transactions

- When is Heuristic Completion Required?
- Adabas Termination of a Global Transaction
- Transaction Timeout (TT) Setting
- Online Save
- ADAEND Command
- Internal Synchronized Checkpoint

### When is Heuristic Completion Required?

In certain situations, it may be necessary for a resource manager to make a heuristic decision about completing a transaction that it has already prepared.

For example, suppose a resource manager is asked by its local transaction manager to prepare its part of a global transaction initiated by a remote client, and then the network fails. The resource manager would normally lock any resources held by the transaction and wait to receive an instruction from the transaction manager to commit or roll back.

The transaction resources could remain locked for a long time. While the locks remain, the resource manager is unable to perform certain functions, such as a database save.

**Note:**

A prepared transaction is not subject to the normal timeout rules.

**Adabas Termination of a Global Transaction**

An Adabas nucleus will terminate its part of a global transaction which it has successfully prepared, if all of the following are true:

- an internal ET syncpoint occurs;
- the TT time limit currently in effect for the user has expired.

An internal ET syncpoint occurs when:

- an ADAEND command is issued;
- an online save or delta save syncpoint occurs; or
- a SYNCC command is issued.

An Adabas nucleus will also terminate its part of a global transaction if a HALT command is received, or if its WORK 4 dataset overflows.

In most of the above cases, the Adabas nucleus decides to commit the local transaction. However, if HALT is issued or Work 4 overflows, the nucleus backs out the local transaction. In all cases, Adabas writes information about the terminated transaction to its DDPRINT dataset for audit purposes.

**Caution:**

It is strongly recommended not to use the HALT command for an Adabas nucleus running with the parameter setting DTP=RM.

**Transaction Timeout (TT) Setting**

The risk of heuristic transaction terminations would increase if an operator command were allowed to reduce the TT value (in preparation for an ADAEND command or an online save, for example). For this reason, an Adabas nucleus running with DTP=RM normally ignores a command to reduce its TT value if it has any prepared transactions that have not yet completed. However, if the nucleus has a pending ET syncpoint (at the end of an online save, for example), the new TT value will take effect.

**Online Save**

If a prepared transaction remains at the end of an online save operation, and has not been completed within a period of 60 seconds plus the value of the TT SYN parameter, measured from the beginning of the transaction, the Adabas nucleus will heuristically back out the transaction, to allow the pending ET syncpoint to complete.

**ADAEND Command**

When an Adabas nucleus running with DTP=RM receives an ADAEND command, it prevents new transactions from being prepared or started and asks its local ATM manager to quiesce any global transactions in which it is involved. If this process is successful, any prepared transactions are completed (committed or backed out) and the ADAEND command is then processed. If prepared transactions remain incomplete, they are heuristically terminated when their TT time limits expire, and then the ADAEND

command is processed.

## Internal Synchronized Checkpoint

An internal synchronized checkpoint takes place at the end of an online save operation. This requires all clients who are using the nucleus to be at ET status simultaneously. No client can begin a new transaction until the checkpoint has completed. The TTSYN parameter of ADASAV can be used to make a temporary reduction in the nucleus's TT value; this causes incomplete transactions to be backed out after the specified time. This logic applies only to unprepared transactions. If ATM is in use, there is a possibility for prepared transactions to remain open after the period specified by TTSYN. If this is the case, an additional period of approximately 60 seconds is allowed for prepared transactions to complete. Once this extra period has elapsed, the nucleus will heuristically back out any transactions remaining in prepared status. An ADAN89 message will be written to the console and DDPRINT for each such heuristic termination.

### **Caution:**

Heuristic termination can result in the loss of transaction integrity. When part of a global transaction is terminated, the global transaction as a whole may have mixed completion, which means that parts of it are committed while other parts are backed out. Mixed completion means a loss of global transaction integrity. However, since unilateral local transaction completion applies only to prepared transactions, the likelihood of mixed completion is small.