Open distributed transaction processing

- Using CICS RMI
- Using RRMS

Adabas Transaction Manager allows open transactions so that changes to Adabas databases can also be coordinated alongside transactions with other vendors (example DB2, etc). You must set *Open distributed transaction support* and related settings accordingly (such as *Adabas transaction dynamics*) to use either CICS/RMI or RRMS.

Open transactions also means you must consider the effect of control over transactions being exerted by more than the usual `ET` and `BT` commands. The default for Adabas is that transactions are able to span TP system message-pairs. However, this is restricted when running with other vendor transactions. Transactions are not able to span message-pairs. Basically, open transaction programming demands that screen interactions force a transaction boundary where any in-flight transaction is automatically committed. This can be a severe limitation on Adabas applications that have not been designed to work with such restrictions.

Many Adabas applications are not able to accommodate these restrictions. A number of settings exist to help try to alleviate the difficulties in the *Open distributed transaction support* area. Essentially, the design of the application is governed by the open transaction model that is used.

Using CICS RMI

- APPC Applications with CICS RMI
- Continuous operation mode with CICS RMI

Adabas Transaction Manager enables inter-operation with IBM’s CICS Resource Manager Interface (CICS RMI). This enables CICS applications to execute transactions that distribute across Adabas and any other vendor offerings that also work with CICS RMI (such as DB2, VSAM, etc).

CICS RMI must be configured correctly according to IBM documentation along with appropriate settings in the area of *Open distributed transaction support*. Correct configuration will enable automatic resynchronization as well as open transaction coordination interactions.

From a programming viewpoint transactions can be applied (committed) by the usual Adabas commands (`ET/CL`) or EXEC CICS SYNCPOINT in the application. You can set client runtime controls to specify whether you want some or all of these to be allowed. However, if there has been neither of these when a (pseudo-conversational) screen interaction takes place (or end of CICS task) then CICS RMI will itself enforce a unilateral commit to take place.

APPC Applications with CICS RMI

Some specialized applications are designed to use Advanced Peer-to-Peer Communication (APPC) in CICS RMI. APPC is a protocol for two programs to work together as peers, usually across different CICS (or compatible) TP systems in the same or different operating systems.

The APPC protocol demands the programmer is fully trained to implement peer programs that dynamically negotiate the explicit and implicit changing peer roles for control of transactions (this is governed by the rules of CICS RMI APPC not by Adabas Transaction Manager). For example, when one peer issues `SYNCPOINT ROLLBACK` either explicitly or by issuing a `BT` command the rules of APPC demand that it cannot then issue an `ET` (or `SYNCPOINT`)…this is all related to implied/explicit role play within the APPC protocol. When these protocol rules are broken CICS RMI will issue ASP2 abends (etc). These abends are not an indication of problems in Adabas or Adabas Transaction Manager; they are indication of the APPC programming protocol not being adhered to – in other words application program bugs. However, as stated the APPC protocol is very complex so it can be very difficult to solve such application problems.

Continuous operation mode with CICS RMI

Continuous operation mode can be used with CICS RMI, and when it is in play during outage Adabas transactions are managed with series of `ET/BT` commands as usual. And also as usual, when the outage is ended full DTP mode resumes automatically. If continuous operation mode takes effect in the middle of transaction completion the effects can vary according to the point at which it happens. This may mean the transaction is undone or is deferred or even a CICS transaction abend may occur (something CICS RMI often does unilaterally when symptoms occur during transaction completion). New transaction completions will follow the settings you choose for continuous operation until the outage ends and normal operation resumes.

Using RRMS

- Using RRMS with IMS TM

Adabas Transaction Manager enables inter-operation with IBM’s Recoverable Resource Management Services (RRMS). This enables any combination of resource managers (typically DBMSs) to participate in distributed transactions that are coordinated by Resource Recovery Services (RRS), a component of RRMS that drives the two-phase commit protocol.

RRMS must be configured correctly according to IBM documentation along with appropriate settings in the area of *Open distributed transaction support*. Correct configuration will enable automatic resynchronization as well as open transaction coordination interactions.

From a programming viewpoint transactions can be applied (committed) by the usual Adabas commands (`ET/CL`) or by a system call to RRMS. You can set client runtime controls to specify whether you want some or all of these to be allowed.

RRMS support is provided for single-user, single-TCB batch applications, and for applications running under IMS TM.

Using RRMS with IMS TM

If your applications run under IMS TM, and IMS allows its transactions to be coordinated by RRMS, Adabas Transaction Manager can ensure that their Adabas changes are committed (or backed out) in a synchronized manner, under the control of RRMS. An IMS commit syncpoint causes RRMS to carry out a commit operation for all changed resources that are managed by RRMS-enabled resource managers. An IMS rollback syncpoint causes all changes to be backed out.

The completion of an IMS message (normally this means screen I/O) causes a syncpoint to take place. In the case of successful completion, this is a commit syncpoint. Moreover, a commit syncpoint implies the completion of processing for the current message. For this reason, an `ET` or `CL` command that is issued during the processing of a message will not cause an RRMS commit syncpoint; any pending Adabas changes will be committed, but non-Adabas resources will be unaffected. Further, any held ISNs will be released, or will remain in held status, depending on the presence of `P` or `M` command options, and the setting of the Extended Hold client runtime control.