ET data and ET identity processing

- Using ET identities (ETID)

- ET Data

- Continuous operation mode and ET Data

- External transaction coordinators and ET Data

Using ET identities (ETID)

It is possible for an application to use an ETID for some database sessions, and concurrently to use other database sessions with no ETID. It is also possible for a client to use different ETIDs concurrently in different database sessions; however, Software AG strongly recommends that the same ETID is used for all databases that are used.

ET Data

`ET` data is a special type of application data. Changes to `ET` data take effect if and only if a transaction completes successfully (is applied) with an `ET` command (or `CL`).

Adabas Transaction Manager supports the use of `ET` data with distributed transactions. If no new `ET` data is used when the transaction is applied the existing `ET` data remains unchanged, similar to how standard Adabas handles `ET` data.

When an application issues an `ET` command with `ET` data when no valid ETID has been provided, Adabas Transaction Manager, like Adabas will not store the `ET` data persistently. Instead, it will proceed as follows:

- If the target database of the `ET` command is being managed using the two-phase protocol, the `ET` data will be ignored, but Adabas Transaction Manager will not draw attention to this fact by means of a non-zero response code;

- Otherwise, the `ET` data will be passed to the database which the application specified on the `ET` command, and to no other database.

When `ET` data is stored by an `ET` or `CL` command for a session using a valid ETID, its location is determined by the setting of the client runtime control ET data storage location.

Note:
It is important that the same mechanism and store location is used from one execution to the next otherwise `ET` data will be lost. For example, if the application is configured to store `ET` data in a database one day, and then changed to use the `TM` recovery file the next, then it is likely that `ET` data will be lost during this changeover – unless special action has taken place to make sure the `ET` data has been moved from the old store to the new store.

Software AG strongly recommends applications should use the same ETID in all databases for one client. When this recommendation is followed, there can be no confusion about which ETID should be used when `ET` data is to be stored or read.

For information about:

- controlling the storage of `ET` data, see the ET data storage location client runtime control and ET data storage location TM control.

- establishing the current `ET` data of your applications in the `TM` recovery file before they execute, see ET Data Management.

Continuous operation mode and ET Data

When a client session is temporarily running in continuous operation mode, `ET` data requests are directed to the database specified in the Adabas control block even though the client is configured to use the TM recovery file. This could result in

- incorrect `ET` data being read.

- `ET` data being written to a database from which Adabas Transaction Manager is not able to retrieve it.

- incorrect results when Adabas Transaction Manager attempts to recover a transaction on behalf of the user

For these reasons, it is strongly recommended that the client control *Continuous operation mode* is set to NO when `ET` data is configured to be maintained in the TM recovery file.

External transaction coordinators and ET Data

When running with the CICS Syncpoint Manager or Recoverable Resource Management Services (RRMS), it is not possible to synchronize the storage of `ET` data when an unsolicited syncpoint occurs, because CICS and RRMS syncpoints have no knowledge of `ET` data.

If an application stores `ET` data and runs in a CICS/RMI or RRMS environment, you can ensure that the storing of its `ET` data is synchronized with the two-phase commit process by conforming to the following rules:

- Any syncpoint for which `ET` data is to be stored must be triggered by an `ET` or `CL` command.

- The `ET` or `CL` command that triggers the syncpoint must also supply the `ET` data; that is, if the application issues a series of `ET` commands to different databases, the first `ET` must supply the `ET` data.

In an IMS TM system whose transactions are coordinated by RRMS, it is not possible to store `ET` data synchronously with an RRMS syncpoint. IMS allows an RRMS commit syncpoint to take place only at the successful completion of message processing, and this syncpoint cannot be triggered by an `ET` or `CL` command.