

Programming ETP

Natural application programs written for Adabas can use the ETP master files without change. Therefore, it is assumed that you are familiar with Adabas programming techniques. If you have questions regarding Adabas programming, refer to the Adabas manuals listed on the Overview page of the Entire Transaction Propagator documentation.

This document describes the few considerations required when creating Natural programs for ETP, and describes the optional ETP user exits and driver control using a user-written ET program.

Note:

ETP also provides an interface for 3GL (COBOL, PL/I, etc.) programs. See *Using ETP with 3GL Programs* (in the ETP Installation documentation) for more information. An additional product for 3GL programs running under CICS, ETP Interface for CICS (ETC), can also be ordered. For more information, see the ETP CICS Interface documentation.

This chapter covers the following topics:

- ETP Programming Principles
 - ET-Time User Program
-

ETP Programming Principles

- Limits and Restrictions
- Updating Master Files
- Updating Replicate Files
- Using Distribution Keys
- Updating the Administration File
- Processing of Asynchronous Task Messages

Limits and Restrictions

ETP is a distributed database environment that provides high reliability without the overhead needed for absolute data integrity. However, there are some limits and restrictions that you should consider when creating programs for the ETP master/replicate file environment.

ETP requires that update operations be performed to the master files only. In addition, read operations, where possible, should be addressed to the nearest replicate copy. Although the latter is not a requirement, reading the nearest replicate file is recommended to reduce network traffic.

ETP cannot handle the new control block format as available with Adabas Version 8; if such a control block is encountered, an error is returned to the application that issued the database call.

Updating Master Files

All updates to the ETP environment must be performed *only* on master files. If an End Transaction (ET) call to a master file contains ET user data, the ET data are included in the resynchronization changes made to the replicate files.

Natural Updates to Master Files

A Natural program can update all master files defined in a single ETP administration file. The reason is, only one administration file can be assigned using an `NTLFILE` or `LFILE` definition. Also, the master files updated by a single transaction must all be in the same database. Note, however, that master files under control of the same administration file can be in different databases.

To update files on different databases in a *single* transaction, at most one of the databases affected may be defined as an ETP database, that is, a database that contains master files.

Multi-Processor Updates

When updating master files which use the same log file from more than one processor, the processor clocks must be synchronized to ensure that the transaction time stamps in the log file are not inconsistent. Software AG therefore recommends that you make all updates to master files using the same log file from a single processor only.

Updating Replicate Files

Programs should read replicate files only. Replicate files may not be changed by application programs. An Adabas Security password can be defined for a replicate file to protect it from accidental updates.

Using Distribution Keys

A distribution key is the control mechanism used by ETP to manage replicate files that contain only a subset of the master file records. You create a distribution key by defining one field in the master file's record as the key field. When you later define a replicate file having only a subset of the master file records, you can specify a value, a range of values, or a combination of values and/or value ranges for the key field that determine which records are in the replicate file.

A master/replicate file set can have only one distribution key field. The field can contain meaningful data (part number, employee number, family/last name, etc.) or it can be an "invisible" field used only to control the ETP files. In addition, a distribution key can be any Adabas field except a periodic (PE) group or a field within a PE group, a multiple-value field, or a phonetic descriptor.

Checking the Key with User Exit 2

The ETP optional user exit is `WADUSER2` (see *ETP Installation*). This user exit must be located in library `SYSETP`, or in a steplib when a replication task is executed. See the `SYSETP` library, which contains an example of the user exit.

Storing a Record with No Key

If you store a record in a master file that has a distribution key, but the Natural view does not define the distribution key, ETP sets the key for the new field to a default value. Depending on the data format of the field, the default value is blank for a format A field, binary zeros for a format B field, or zero for all numeric types.

Updating the Distribution Key

You cannot simply UPDATE a field in the master file that is defined as a distribution key. To perform an update on the field, the program must perform an explicit DELETE/STORE sequence. This is due to the fact that the information not included in the view used for the UPDATE process would otherwise be lost.

Updating the Administration File

If a master file definition in the administration file is changed while logging is being done on the database of the master file, the changed definition is not used immediately. The new definition is first used after either a transaction has been logged for another database, or after Natural has been restarted - whichever occurs first.



Warning:

Therefore, you should avoid master file definition changes while a Natural application using ETP logging facilities is active. For the same reasons, you should also avoid changing master or replicate file definitions while a task is executing.

Processing of Asynchronous Task Messages

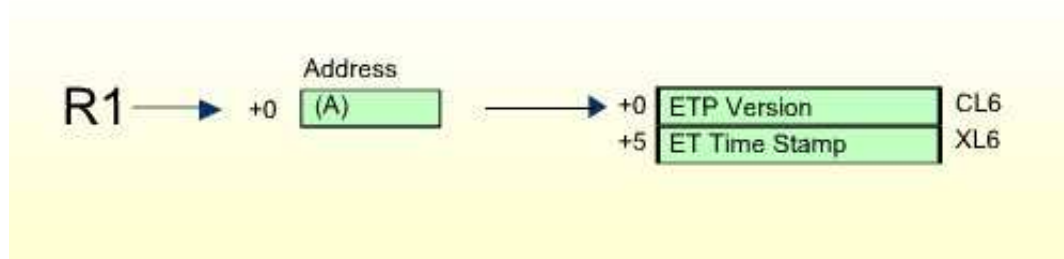
The subprogram WADUSER3 is used to display all messages issued by the replication and clean-up tasks. WADUSER3 can be modified to filter any resulting messages and, if desired, send them directly to the operator console. (For more information about the WADUSER3 subprogram, which is delivered in source form with ETP, see *ETP Installation*.)

ET-Time User Program

After each update to an ETP master file, you can invoke your own program to perform such functions as starting a replication task to resynchronize the replicate file(s) with their master file. When called, the program is passed a pointer to an address of a parameter list in General Register 1, as shown in the figure below.

ETP uses normal OS conventions to pass the parameter list.

Parameters Passed to the ET User Program



The user program can be for any purpose; however, the ET program is intended for starting the replication task(s) to resynchronize replicate files with the master file. The program that receives control after an ET is typically a CICS or Com-plete transaction which starts a Natural nucleus. That nucleus must be provided with appropriate parameters to log on to SYSETP (the ETP maintenance utility) and to start a replication task.

If any of the master files which are updated in a transaction have enabled ET program control, the user-defined program is started after the ET command has been successfully executed.

Note:

The ET Program Control mode for resynchronizing master and replicate files described here is presently limited to only those systems using the CICS or Com-plete TP monitors. The controlling program must be a transaction which is defined to the TP system on which it executes. For CICS, the messages issued by the replication task must be routed to the operator console by means of the subprogram WADUSER3. (For more information about the WADUSER3 subprogram, which is delivered in source form with ETP, see *ETP Installation*.)