ETP Setup

This document gives hints on how to define the ETP environment for the maximum performance.

This chapter covers the following topics:

- Getting Started

- General Considerations when Defining Files

- Performance and Maintenance Considerations

Getting Started

To perform the first steps with ETP, you can use the demonstration files `EMPLOYEES` and `VEHICLES` that are supplied with every Natural installation, and the corresponding sample programs described in the Natural *Programming Guide*.

You can also use the application program that you plan on using with ETP by copying the necessary files for test purposes. For easier maintenance, Software AG recommends starting with small or empty files; this makes it easier to see which changes ETP has performed.

It is possible and recommended to initially install both master and replicate files on the same database. This eases the first steps, and avoids problems that might occur when data is transmitted over a network.

General Considerations when Defining Files

You should ensure that user exits do not change either the database ID or the file number specified in the Adabas control block. If required, use the Natural macro `NTTF` or the Natural profile parameter `TF`. The following exits are a potential source of such changes:

- `NATUEX5` (Natural, called before an Adabas call)

- `NATUEX6` (Natural, called after an Adabas call)

- User exit B (`ADALNK`)

- User exit 1 (Adabas)

When accessing an Adabas database that requires the first command to be an `OPEN` command, a NAT3009 error will occur during replication if Natural has not been set up so that an `OPEN` command is issued. When your system is set up to require an `OPEN` command, heed the hints in the following sections when defining replicate files.

Specifying Replicate Files on OpenVMS, UNIX or Windows

When defining replicate files on diverse platforms and systems, you should be aware of the points described in the following topics.

Generating an Automatic OPEN (OP) Command

If an Adabas database requires that an OPEN (OP) command to be issued before any other database call, the Natural system variable *ETID must be a non-blank value; if *ETID is blank, Natural does not issue the OP command. You can also use the dynamic Natural profile parameter ETID to specify a non-blank ETID.

If Natural Security is not installed, the default for *ETID is the user ID. If Natural Security *is* installed, Software AG recommends letting Natural Security generate the *ETID value automatically; RESTART=YES should also be specified for the library SYSETP. For more information, see the *Natural Security* documentation.

Specifying the Record Buffer Value for the OPEN (OP) Command

You can use either the Natural profile parameter OPRB or the NTOPRB macro to specify a Record Buffer for the OPEN (OP) command issued to the replicate databases. See the Natural *Parameter Reference* documentation for parameter or macro details.

Data Representation Inconsistencies

Alphanumeric and binary data is represented differently on mainframe and OpenVMS, UNIX and Windows systems. This can lead to unexpected results if:

- You use Natural REDEFINE statements in your application to redefine a field defined as alphanumeric in Natural and in the Adabas Field Definition Table (FDT), so that the alphanumeric data is overlaid with numeric or binary data-or vice versa;

- Binary superdescriptors are used that contain alphanumeric data;

- Alphanumeric superdescriptors are used that contain binary data (i.e., by means of a REDEFINE).

In the above cases, strange results can occur because:

- Binary data that is part of an alphanumeric field is converted from EBCDIC to ASCII, producing useless result data;

- Alphanumeric data that is part of a binary field is *not* converted from EBCDIC to ASCII;

- The sequence obtained by means of a READ LOGICAL statement may be different due to the fact that Windows and some UNIX machines have a byte-swap architecture especially affecting alphanumeric data that is contained in binary fields.

Superdescriptor Conversion

Particularly for superdescriptors, Entire Net-Work for mainframe systems provide the superconverter feature that allows for an exact specification of the translation process for all parts of a superdescriptor. Refer to the appropriate Entire Net-work documents for a detailed description.

Performance and Maintenance Considerations

ETP performance is affected by the definitions of master and replicate files. If you plan to replicate large amounts of transactions, you should specify the definitions carefully to avoid performance problems. The CHECK DEFINITIONS command of the ETP maintenance utility is useful for examining your definitions for potential performance problems. If a potential problem is reported, it will in many cases be a problem only if large numbers of transactions are to be replicated.

You should also take into account that not only the use of ETP itself, but also other resources are likely to be potential sources of data distribution problems; for example:

- Usage of the machine hosting the master file database or performing the ETP replication process may be short of resources (CPU time, etc.);

- The process performing the ETP replication task may run at low priority;

- The Adabas nucleus hosting the master and log files may run at low priority;

- The Net-work processes on either the master or the replicate site may be overloaded or run at low priority;

- The transmission rate may be low or the usage high on the network link connecting the master and replicate databases;

- The Adabas nucleus hosting the replicate file may run at low priority.

When replicating a large number of master file updates, you should examine all of the above possibilities to ensure that the updates are replicated as you intended. When performing only a small number of master file updates, the danger of performance-related problems is usually very small.

The following are hints and tips for setting up master and replicate files:

- **Avoid defining distribution keys for a master file unless you actually need them.**
  Distribution keys cause overhead for both logging and replication of master file updates. During replication, any defined distribution key ranges are kept in storage for performance reasons. If you have defined a number of large distribution key ranges for replicate files and those replicate files are on the same database for which the master file share a common log file (see the next point), then some key ranges have to be loaded from the database. To improve performance in this case, assign different log files to the corresponding master files.

- **For all master files with replicate files on common databases, use a common log file.**
  For example, consider the following relationship of master and replicate files:

Master File DBID/File no.:	Replicate File DBID/File no.:
10/20	12/20
10/30	12/30
10/40	14/40
10/42	14/44

In this case, master files 20 and 30 could share a common log file and master files 40 and 42 could share a second log file. In fact, if the number of changes to all the master files are small, all four master files can share a single log file. Note, however, the exception when using distribution key ranges as described in the point above, and the last point on the following page.

- **Run multiple replication tasks from different log files or to different databases, in parallel.**
  Running parallel multiple tasks that perform replication from different log files and/or to replicate databases improves system throughput and is highly recommended if many updates are to be replicated or if the number of replicate databases is high. For a network link with a high transmission rate, running multiple tasks that replicate from different log files to a common replicate database may also improve the system throughput.

- **Short self-restart intervals are unproductive.**
  The checking necessary to determine if any new transactions in a log file are to be replicated requires a fixed number of database calls; therefore, using a self-restart interval that is too small results in unnecessary overhead. You can use the report written by ETP following every replication to choose an appropriate self-restart interval.

- **Use batch mode to define a large number of files.**
  When defining a large number of master and replicate files, Software AG recommends doing this in batch mode because changes to all definitions are more easily achieved by deleting the existing definitions and then adding the modified definitions again.

- **An error when replicating to one replicate file affects all replicate files served by that log file.**
  When an error occurs during replication of an update, replication is terminated immediately for all replicate files on that database that are served by the related log file. This includes transactions to replicate files from other master files that share the log file. File-specific errors (indicated by, for example, Adabas response code 113) should not occur under normal conditions; however, to ensure that replication continues for as many log files and replicate databases as possible, assign different log files to the master files.