

Database Errors during Logging of Master File Transactions

From the application and user viewpoint, ETP activities during master file updating are absolutely transparent; however, errors resulting from database calls issued by ETP are reported as though the error had occurred for the application or user.

This chapter covers the following topics:

- Isolating Call Errors with TEST DBLOG
 - Automatic ETP Back-Out Transaction
-

Isolating Call Errors with TEST DBLOG

If you suspect that an error occurred for one of the additional database calls issued by ETP, use TEST DBLOG to track down the error. More specifically, use code S (snapshot) of TEST DBLOG to pinpoint the error. You should set up TEST DBLOG before you invoke the program or application for which the error occurs. TEST DBLOG can also be used in batch mode.

The following sample input could be used to identify the source of a response code 17, using snapshot:

```
LOGON MYLIB      Log on to the application library.
TEST DBLOG ?    Issue the TEST DBLOG command.
S,,,,,,,,17,9999 Input for TEST DBLOG.
MYPROG         The application program during which the error occurred.
```

Now, the Adabas call that is shown is the one that received a response code 17.

Using TEST DBLOG in this way may also help to identify errors that occur while running the ETP maintenance facility; however, almost all errors that occur during maintenance utility execution provide the information listed above to the ETP administrator.

Automatic ETP Back-Out Transaction

In addition to supplying the information described above for an error during logging operations, ETP also issues a Back out Transaction (BT) if a non-zero response code is returned for an ETP-issued rather than the original call. A BT is also issued if the original call results in a NAT3600 - NAT3624 error.