Identification

Appendix to BY.2.01
K. J. Martin

Purpose

This appendix documents a discrepancy between the BY.2.01
document and current implementation of the file system
interface procedures and corrects mistakes in the current
document.

Discussion

When BY.2.01 was published, it was optimistically thought
that the command system error handling mechanism (BY.11)
would be installed shortly. Therefore, BY.2.01 was written
to conform with the final implementation of the procedures.
The procedures, however, were written to be usable now;
each has an undocumented fixed bin(17) error code as its
last argument and merely passes back to the caller whatever
code the file system returned to it.

When these procedures are changed to use the command system
standard error handling, these last arguments will disappear.
This will not be fatal to any calling procedures which
either ignore the error code returned or are clever enough
to set it to zero before calling the file system interface
procedures. If all goes well in the file system's handling
of a call, the calling procedure will see no difference
in the interface procedure's activity.

If all is not well, the interface procedure will signal
a condition (append_branch_err, for example). Since the
calling program is not prepared for this signal, a handler
established elsewhere will deal with the signal.

In most cases unclaimed_signal (by.11.05) will handle it.
Unclaimed_signal places the user at a secondary command
level where he has many options. 'If he chooses to continue,
the procedure which called the file system interface procedure
will see no sign of irregularity and will proceed as though
all had gone well. (Note that the user -personally at
a console- was informed of the nature of the problem and
presumably made an intelligent decision.)

The calling procedure may be changed at the author's
convenience to conform with the command system standard
error handling.

References

The reader is referred to BY.2.02 where file system errors
are documented and to BY.11 for an explanation of command
system standard error handling.

Interim Calling Sequences

The final argument is declared:

    dcl code fixed bin(17);

The other arguments are described in BY.2.01.

call change_name (dir, entry, oldname, newname, code);

call delete_entry (dir, entry, courtesy_flag, code);

call append_branch (dir, name, type, mode, optsw, max1, code);

call append_link (dir, name, pathname, code);

call set_max_length (dir, entry, max1, code);

call move_file_ (dir, entry, courtesy_flag, new_dir, new_entry,
    code);

call set_copy (dirname, entry, copysw, code);

call set_relate (dirname, entry, relatesw, code);

call set_retention_date (dirname, entry, retdate, code);

call set_consistency (dirname, entry, constsw, code);

call truncate_seg (addr_ptr, code);

mode = check_access (segptr, ringno, code);

Errata

page 2, courtesy_flag setting:

the three values 0,1,2 should be "00"b, "01"b, "10"b.

page 3, third paragraph from the bottom should read:

If type is off (i.e., name is a nondirectory branch), the optsw
argument is meaningful. If type is on, it is ignored.