How do I interpret file status 9? and other EXFS extended file status codes (e.g. 9i,105)?

Author: Veryant Support
Saved From: http://support.veryant.com/support/phpkb/question.php?ID=150

The letter after the 9 is the character representation of the extended file status using ASCII encoding. For example, a 9i is the same as file status 9 with extended status 105 (i.e. EXFS=105) because ';i'; is 105 in ASCII.

The following is from the runtime framework source code:

```
// IO ERROR
static final int E_IO_DUPL = 100; /* duplicate record */
static final int E_IO_NOTOPEN = 101; /* file not open */
static final int E_IO_BADARG = 102; /* illegal argument */
static final int E_IO_BADKEY = 103; /* illegal key desc */
static final int E_IO_TOOMANY = 104; /* too many files open */
static final int E_IO_BADFILE = 105; /* bad isam file format */
static final int E_IO_NOTEXCL = 106; /* non-exclusive access */
static final int E_IO_LOCKED = 107; /* record locked */
static final int E_IO_KEXISTS = 108; /* key already exists */
static final int E_IO_PRIMKEY = 109; /* is primary key */
static final int E_IO_ENDFILE = 110; /* end/begin of file */
static final int E_IO_NOREC = 111; /* no record found */
static final int E_IO_NOCURR = 112; /* no current record */
static final int E_IO_FLOCKED = 113; /* file locked */
static final int E_IO_FNAME = 114; /* file name too long */
static final int E_IO_BADMEM = 116; /* can';t alloc memory */
static final int E_IO_BADCOLL= 117; /* bad custom collating */
static final int E_IO_LOGREAD= 118; /* Cannot read log file record. */
static final int E_IO_BADLOG = 119; /* Record format of transaction-log file
                                    cannot be recognized. */
static final int E_IO_LOGOPEN =120; /* Cannot open transaction-log file. */
static final int E_IO_LOGWRIT =121; /*Cannot write to transaction-log file.*/
static final int E_IO_NOTRANS =122; /*Not in transaction. */
static final int E_IO_NOBEGIN =124; /* Beginning of transaction not found. */
/*
 * JISAM extended error codes
 */
static final int E_IO_BADOPENMODE=125; /*incompatible operation with
                                        open mode*/
static final int E_IO_NOTSUPP    =126; /* function not supported */
static final int E_IO_DISKFULL   =127; /* disk full */
static final int E_IO_RECCHANGED =128; /* rec changed */
static final int E_IO_NOLOCKS    =129; /* no more locks available */
static final int E_IO_MISSINGFILE=130; /* missing file */
static final int E_IO_PERMISSION =131; /* invalid permission */
static final int E_IO_FILEEXISTS =132; /* file exists */
static final int E_IO_SYSTEM     =133; /* system error */
```

```
static final int E_IO_UNKNOWN    =134; /* Unknown error */
// Cobol only errors
static final int E_IO_ALREADY_OPEN=135; /* Already open */
static final int E_IO_NOTFORINPUT =136; /* File not open for INPUT or I-O */
static final int E_IO_NOTFOROUTPUT=137; /* File not open for OUTPUT or I-O */
static final int E_IO_NOTFORIO    =138; /* File not open for I-O */
static final int E_IO_INVALIDWRITE=139; /* Invalid write */
static final int E_IO_NO_REMAPPABLE=140; /* Error non remappable in COBOL */
static final int E_IO_CLOSE_WITH_LOCK=141; /* File closed with lock */
static final int E_IO_INVALID_OPEN=142; /* Invalid open mode */
static final int E_IO_INVSEQDELREW=143; /*Invalid sequential rewrite/delete*/
static final int E_IO_NOTSTART    =144; /*Unable to start service*/
static final int E_IO_BOUNDVIOL   =145; /* Boundary violation */
```