

Native Files

General	
<code>R ← □NNAMES</code>	Lists the names of all tied native files
<code>R ← □NNUMS</code>	Lists the tie numbers of all tied native files
File operations	
<code>{R} ← X □NCREATE Y</code>	Creates a new native file with name <code>X</code> and file tie number <code>Y</code> ; a tie number of 0 allocates the next available tie number to the file
<code>{R} ← X □NERASE Y</code>	Erases the tied native file that has name <code>X</code> and file tie number <code>Y</code>
<code>{R} ← X □NRENAME Y</code>	Renames the tied native file that has file tie number <code>Y</code> to have name <code>X</code>
<code>{R} ← X □NRESIZE Y</code>	Changes the size of the native file that has file tie number <code>Y</code> to size <code>X</code> (either by truncating the file or by extending it with undefined additional bytes)
<code>R ← □NSIZE Y</code>	Returns the size in bytes of the native file that has file tie number <code>Y</code>
<code>{R} ← X □NTIE Y</code>	Ties the native file that has name <code>X</code> using file tie number <code>Y[1]</code> ; optionally, <code>Y[2]</code> can specify the type of access needed by other users (see <i>Access Codes</i>)
<code>{R} ← □NUNTIE Y</code>	Unties all native files that have a tie number in vector <code>Y</code> (<code>□NUNTIE 0</code> does not untie any files but flushes all file caches to disk) and returns the number of native files that have been untied
Data Transfer	
<code>{R} ← X □NAPPEND Y</code>	Appends the ravel of array <code>X</code> to the end of the native file that has tie number <code>Y[1]</code> ; optionally, <code>Y[2]</code> can specify the conversion code to use to convert array <code>X</code> (by default, 80 is assumed when using the Unicode version – see <i>Conversion Codes</i>)
<code>R ← □NREAD Y</code>	Reads the content of the native file identified by file tie number <code>Y[1]</code> ; <code>Y[2]</code> specifies the conversion code to use (see <i>Conversion Codes</i>), <code>Y[3]</code> specifies the count (see <i>Conversion Codes</i>) and, optionally, <code>Y[4]</code> can define the offset from 0 of the first byte to read
<code>{R} ← X □NREPLACE Y</code>	Replaces content in a native file identified by file tie number <code>Y[1]</code> with <code>X</code> ; <code>Y[2]</code> defines the offset from 0 of the first byte to replace and, optionally, <code>Y[3]</code> specifies the conversion code to use (by default, 80 is assumed when using the Unicode version) (see <i>Conversion Codes</i>)
<code>{R} ← {X} □NXLATE Y</code>	Associates the native file that has tie number <code>Y</code> with character translation vector <code>X</code> . Note that: <ul style="list-style-type: none"> if <code>X</code> is not specified then the currently-associated translation vector is returned if <code>X</code> has the value <code>(ι256)–□IO</code> then the translation process is bypassed and raw input/output is provided if <code>Y</code> is set to 0, then the translate vector used by <code>□DR</code> is used Unicode version only: This is only relevant when processing native files that contain characters expressed as indices into <code>□AV</code>
Manipulating access to a file	
<code>{R} ← X □NLOCK Y</code>	Changes the lock status (as defined by <code>X</code>) of part of the native file that has file tie number <code>Y[1]</code> ; optionally, <code>Y[2]</code> can define the offset from 0 of the first byte to apply the lock change to (defaults to 0) and <code>Y[3]</code> can specify the number of bytes impacted by the lock change (defaults to the maximum possible file size) (see <i>File Locking</i>)

Access Codes

The access codes used by `NTIE` are integer values calculated as the sum of:

- the type of access needed from users who have already tied the native file
- the type of access to grant to users who subsequently try to open the file while you have it open

Needed from existing users	
0	read access
1	write access
2	read and write access

Granted to subsequent users	
0	compatibility mode
16	no access (exclusive)
32	read access
48	write access
64	read and write access

Conversion Codes

The conversion codes used by `NAPPEND`, `NREAD` and `NREPLACE` vary according to the installation of Dyalog APL that is used to read the native file; the following two tables show the conversion codes for the Unicode version and Classic version respectively.

Value	Number of Bytes	Result Type	Result Shape
11	count	1 bit Boolean	8 x count
80	count	8 bit character	count
82*	count	8 bit character	count
83	count	8 bit integer	count
160	2 x count	16 bit character	count
163	2 x count	16 bit integer	count
320	4 x count	32 bit character	count
323	4 x count	32 bit integer	count
645	8 x count	64 bit floating	count

Value	Number of Bytes	Result Type	Result Shape
11	count	1 bit Boolean	8 x count
-	-	-	-
82	count	8 bit character	count
83	count	8 bit integer	count
-	-	-	-
163	2 x count	16 bit integer	count
-	-	-	-
323	4 x count	32 bit integer	count
645	8 x count	64 bit floating	count

* Conversion code 82 is permitted in the Unicode Edition for compatibility and causes 1-byte data on file to be *translated* (according to `NXLATE`) from `AV` indices into normal (Unicode) characters of type 80, 160 or 320.

File Locking

Unlike component files, which can be tied with an exclusive tie or a share tie, native files cannot be tied in different ways. Instead, `NLOCK` is used to lock byte ranges within files, thereby managing access between users. There are three possible lock statuses:

- 1 means unlock
- 2 means read (share) lock – multiple read locks can exist over the same byte-range. The presence of a read lock prevents a write lock being obtained
- 3 means write lock – only one write lock can exist for a specific byte-range of a native file. The presence of a write lock prevents a read lock being obtained

The lock status can also, optionally, define a timeout period in seconds; if this period is exceeded before the lock status change has occurred, then a TIMEOUT error is displayed (defaults to no timeout limit).

Different file servers can follow different locking standards – `NLOCK` does not standardise this.