

## Caveats for Non-DOS Name Space Support

Multi-OS (Name Space) support is not possible without caveats, conflicts and limitations. The greatest conflicts concern the fact that each Name Space supports different standards (or rules) for constructing file names and directory paths.

*(As the old saying goes, "The nice thing about standards is that there are so many to choose from.")*

Conflicts are inevitable when you consider the following Name Space rules:

### Long Name Space:

According to the standards (or rules) for Windows Long Name Space support (the most widely used Long Name Space implementation), directory and file names may contain up to 215 characters, including spaces but excluding the following characters: Forward slash (/), backward slash (\), colon (:), greater than sign (>), less than sign (<), vertical bar (|), double quotes ("), asterisk (\*), and question mark (?). The colon (:) is reserved for use as the drive or volume (NetWare) separator while backward slashes (\) are reserved for use as the UNC or path separator. The forward slash (/) is also reserved and considered a path separator for compatibility with NetWare and Unix NFS File Systems. All of the other reserved characters serve special purposes in DOS, Windows, and Unix command line processing so they are not supported in directory or file names.

### Mac Name Space:

According to the standards (or rules) for the Macintosh File System, the only restricted character in a file or folder (directory) name is the colon (:) which is reserved for use as a path separator. Characters normally used as wild cards in other Operating Systems, such as the asterisk (\*) and question mark (?), can be used within a file or folder (directory) name. Characters which have special meanings on many Operating Systems, such as greater than (>) and less than (<) signs often used for input or output redirection, or double quotes (") commonly used to enclose a specification with embedded spaces and the ampersand (&) widely used widely in internet applications as an options separator on the command line, are valid with a file and folder (directory) name. In addition, forward slashes (/) which are reserved as a path separator by both NetWare and NFS standards (or rules), as well as backward slashes (\) which are reserved as a path separator by DOS, Long (Windows), and NetWare, are allowed in Macintosh file and folder (directory) names. The only real limitation is that a file or folder (directory) name cannot contain more than 31 characters (Mac OS prior to OS/X), but a full path can exceed the 255 character DOS limitation.

## **Caveats for Long Name Space Support**

### NFS Name Space:

According to the standards (or rules) for the NFS File System, the only restricted character in a file name or path is the forward slash (/) which is reserved for use as a folder (path) separator. Characters normally used as wild cards in most Operating Systems, even for Unix commands, such as the asterisk (\*) and question mark (?), can be used within a file or directory name. In addition, colons (:), which are used by DOS, Long (OS/2 / Windows), and NetWare to separate Drives/Volumes from Paths, can be used in file or directory names. NFS also allows backward slashes (\), greater than (>) and less than (<) signs, as well as double quotes (") and ampersands (&) to be used in file and directory names. Moreover, both file and directory names are handled as case sensitive, treating FILE, File, and file as three separate names. The only real limitation is a maximum of 255 characters per file or directory name allowing a full path to far exceed the 255 character DOS limitation.

The NetWare File System (for both NSS and TFS volumes) is designed to use DOS as the primary Name Space with the other Name Spaces being links or place holders referencing the primary DOS Name Space Directory Entry. While most NetWare NLMs and DOS/Windows 16 bit applications (including many of NetWare's command line utilities) rely upon the DOS Name Space name, a directory or file can be accessed using any of the supported Name Space names.

The following exemplifies a single possible conflict between DOS and Long/OS2 Name Space support on NetWare File Systems. Because the file name limitations for DOS and Non-DOS Name Spaces are significantly different, the Non-DOS Name Space name is often truncated for its primary DOS directory entry which further reduces uniqueness and forces the File System to use an arbitrary, renaming convention. Parsing a Non-DOS file name to meet the DOS 8.3 file name limitation and maintain a unique file name can be a tricky proposition. Depending upon the order that files are created or copied, it is possible for the same Non-DOS Name Space file name residing in two different directories to have two different DOS Name Space file names.

Such discrepancies typically will not cause problems if only DOS/Windows 32-bit applications and NLMs which support Long Name Space names are used for accessing the file. However, many of the NLMs and DOS/Windows 16-bit programs depended upon for File System maintenance utilize APIs which rely upon the DOS 8.3 name for primary access to files. Under NetWare, a common scenario exists which can easily result in the wrong file being accessed even if the DOS name is relied upon.

Here is a set of commands that demonstrate the problem on a NetWare v4 Server:

Note: Assumes that the NetWare v4 Server does not have the MIXMODFX.NLM patch loaded and that Long Name Space support is loaded on the volume

1. Select a default directory (preferably a temporary directory)
2. Create a directory named: Src  
md Src
3. Create a directory named: Dest  
md Dest
4. Copy a file using its Long name to the Src directory as: FileName.job.ext  
copy LongName.File1 Src\FileName.job.ext
5. Copy a different file using its Long name to the Src directory as: FileName.ext  
copy LongName.File2 Src\FileName.ext
6. Copy the first file to the Dest directory as: FileName.ext  
copy LongName.File1 Dest\FileName.ext
7. Copy the second file to the Dest directory as: FileName.job.ext  
copy LongName.File2 Dest\FileName.job.ext
8. Generate a directory listing of the Src directory  
dir Src

Src files:

DOS: FILENAME.EXT	Long: FileName.job.ext
DOS: FILENAM0.EXT	Long: FileName.ext

9. Generate a directory listing of the Dest directory  
dir Dest

Dest files:

DOS: FILENAME.EXT	Long: FileName.ext
DOS: FILENAM0.EXT	Long: FileName.job.ext

Notice that the DOS name FILENAME.EXT represents a different Long Name Space name in each directory and, consequently, a different file. It should also be noted that the Src directory has two files named FileName.ext (albeit, of different cases with one a DOS Name Space name and one a Long Name Space name). This situation occurred as a result of there being a difference in the order in which the files were copied into the directories.

If you use an NLM or DOS/Window 16-bit utility and the DOS name to access the data in the two files, you will be mixing two different sets of data. Worse yet, if you use an NLM or DOS/Windows 16 bit utility and the DOS name to copy (or backup) the data between the

## Caveats for Long Name Space Support

Src and Dest directories, you will cross the file data between the files.

The problem is that Novell's default support for DOS-to-Long name parsing in NetWare v4 is non-standard and does not guarantee that two separate files in the same directory will not share the same name. NetWare v5+ does not have this problem because Novell implemented a methodology similar to that found in Windows 32 bit platforms for Long-to-DOS file name parsing.

Fortunately, Novell has provided a patch for NetWare v4 which modifies the Long-to-DOS file name parsing logic to that used in NetWare v5 and Windows 32 bit platforms:

For more information on the patch, visit the Novell Support Connection WEB Site:

- Select link -> Knowledgebase
- Select product categories -> NetWare option
- Select a document set -> Technical Information (TIDs)
- Enter the Search phrase -> MIXMODFX Long Directory Names
- Click on the Search Now button

To download the patch directly, visit the Novell Support Connection WEB Site:

- Select link -> Patches and Files
- Enter the Search phrase -> MIXMODFX
- Click on the Search button

Note: As always, review the Installation Instructions included in the download and note any caveats before actually installing the patch.

While this worst case scenario is specific to NetWare v4 Servers without the MIXMODFX patch applied, a patched NetWare v4 Server is not completely immune to name space conflicts. In fact, NetWare v5+ Servers and Windows 32 bit platforms can also incur name space conflicts.

Executing the first seven of the previously listed commands on a patched NetWare v4 Server, a NetWare v5+ Server, or a Windows 32 bit platform will generate the following results:

8. Generate a directory listing of the Src directory  
    dir Src

Src files:

DOS: FILENA~1.EXT	Long: FileName.job.ext
DOS: FILENAME.EXT	Long: FileName.ext

9. Generate a directory listing of the Dest directory  
dir Dest

Dest files:

DOS: FILENAME.EXT	Long: FileName.ext
DOS: FILENA~1.EXT	Long: FileName.job.ext

Even though the files were copied in the same order as in the previous test, the DOS and Long file names remain consistent between the two directories and neither directory has two files which share the same name. However, that does not mean that name discrepancies cannot occur under these patched and more recent file system.

To demonstrate the problems which can occur, execute the following commands in the same default directory:

1. Delete the files in the Src directory  
del Src
2. Delete the files in the Dest directory  
del Dest
3. Execute the following copy commands in the order specified:  
copy LongName.File1 Src\FileName.job.ext  
copy LongName.File1 Src\FileName.bak.ext  
copy LongName.File2 Src\FileName.ext  
  
copy LongName.File2 Dest\FileName.ext  
copy LongName.File1 Dest\FileName.bak.ext  
copy LongName.File1 Dest\FileName.job.ext
4. Generate a directory listing of the Src directory  
dir Src

Src files:

DOS: FILENA~1.EXT	Long: FileName.job.ext
DOS: FILENA~2.EXT	Long: FileName.bak.ext
DOS: FILENAME.EXT	Long: FileName.ext

## Caveats for Long Name Space Support

5. Generate a directory listing of the Dest directory  
dir Dest

Dest files:

DOS: FILENAME.EXT	Long: FileName.ext
DOS: FILENA~1.EXT	Long: FileName.bak.ext
DOS: FILENA~2.EXT	Long: FileName.job.ext

Note that the DOS names are mixed between the two extended Long name files in the different directories. However, no two files in either directory share the same name.

Though still a caveat, this potential conflict is greatly minimized as long as the only commands or programs used to access the files support the Long Name Space name.

While Name Space conflicts are bound to occur on Multi-OS (Name Space) supported platforms, they can be reduced to a minimum if Multi-OS compatible rules are implemented for directory and file naming. Ultimately, in any realm of conflict, it is often best to avoid such problems by defining rules which embrace the path of least resistance (i.e., the lowest common denominator, in this case the Long Name Space).

The following simple rules should provide a reasonable compromise that also insures directories and files are fully accessible and manageable across a heterogeneous network:

- S Avoid using any characters in directory or file names which may have special application in other Operating Systems or Name Spaces supported by the network, such as the asterisk (\*), question mark (?), colon (:), forward slash (/), backward slash (\), less than sign (<), greater than sign (>), and ampersand (&) in particular. Being descriptive is good but remember that a good file name is also easy to remember and access so keep it brief and simple. It is also important to note that longer names consume more disk space, require more resources to process, complicate backups, waste network bandwidth, and may be incompatible with some applications.
- S Avoid using spaces in directory and file names. While spaces within directory and file names are supported by Long, Mac, and NFS Name Spaces, such habits can create conflicts for parsers which recognize spaces as command line option separators. A more preferable method is to use the underscore (\_) or a period (.) to separate words within directory or file names, such as my\_file.doc.

Note: While the Macintosh and NFS Operating Systems allow directory and file names to have leading spaces, not all commands and utilities are able to properly process such entries due to limitations in the NetWare API set. Moreover, such names are not supported by either

the DOS or Windows Operating Systems. Therefore, the use of leading spaces in directory and file names is not recommended.

- S One final suggestion: Try to use the proper common or widely used extensions. Without a proper extension, some applications and browsers may not be able to recognize the file type.

Users should be advised of the importance of adhering to multi-OS compatible rules both for file system management (i.e., backup, restore, disaster recovery, maintenance, etc.) and for transportability of information between systems as networks become more and more heterogeneous.