Path Services

The Path Services calls enable an application program to allocate and deallocate directory handles.

Tables Accessed By Path Services

File servers and workstations maintain several tables used by Path Service APIs. They are the following: Directory Table and Directory Handle Table.

Directory Table

To record information about directories, a file server maintains a Directory Table. Entries in the Directory Table define a file server's directory structure; all
Directory Services calls access the Directory Table. The Directory Table contains 3 kinds of entries: directory nodes, file nodes and trustee nodes.

A directory node includes the following information about a directory: directory name, attribute byte, maximum rights mask, creation date, creator's object ID, a link to the parent directory and a link to a trustee node (if one exists).

A file node includes the following information about a file: file name, attribute byte, file size, creation date, last-accessed date, last-updated date and time, last archive date and time, the file owner's object ID and a link to a directory.

A trustee node includes the following information: the object IDs of one to five trustees of a directory linked to the trustee node, one to five corresponding trustee rights masks, a link to a directoryDirectory Handle Table and a link to the next trustee node (if one exists).

Directory Handle Table

The file server maintains a Directory Handle Table for each workstation logged into it. The Directory Handle Table has 256 entries (0x00...0xFF), each of which can be set to point to a volume or directory path. When a workstation requests a directory handle, the file server enters the volume number and directory entry number for the specified directory into the Directory Handle Table for the requesting workstation. Applications running on the workstation can then refer to a directory using a directory handle, which is actually an index into the Directory Handle Table.

Directory Handles

Once a directory handle is set, you can use the directory handle to specify a volume or a directory path on the file server. NetWare APIs that refer to directories permit you to specify a directory path in three different ways.

- Allocate a directory handle to point to the target directory.
- Use both a directory handle and a complementary directory path to specify the complete path. The directory handle must lead part of the way to the target directory, and the directory path must lead the rest of the way.
- Specify the entire directory path.

When an application begins executing, the application must allocate all the directory handles that it needs.

The path services uses the NWPath_t structure below to specify the location of a NetWare file or directory.

NWPath_t Structure

```
typedef struct {
        NWDirHandle_ts          dirHandle;
        uint16                  serverConnID;
        char                    *pathName;
} NWPath_t;
```

dirHandle. This field represents the directory handle allocated by the client pointing to a particular place in the directory structure. The dirHandle is specific to the file server connection (serverConnID) and can only be used to access directories or files on that file server.

serverConnID. This field represents the file server connection which contains the file system being accessed.

pathName. This field is a pointer which points to a character string which the client must allocate and fill in with a path name.

The NWPath_t structure can be used in one of three ways:

1) The application can pass a 0 in the dirHandle field and then pass a full path (of the target directory or file) in the pathName field. In the following example, the path to be accessed is the SYS:APPS/WP directory on file server connection 0 (first file server attached to).

```
NWPath_t        path;
char            fullPath[NWMAX_DIR_PATH_LENGTH];

path.serverConnID = 0;
path.dirHandle = 0;
strcpy( fullPath, "SYS:APPS/WP" );
path.pathName = fullPath;
```

2) The application can pass a previously allocated directory handle in the dirHandle field (see NWAllocTemporaryDirHandle or NWAllocPermanentDirHandle) and then can pass a path in the pathName field which is relative to the directory that the dirHandle points to. In the following example, the path to be accessed is the SYS:APPS/WP directory on file server connection 0 (first file server attached to). A dirHandle of 3 has already been allocated for SYS:APPS.

```
NWPath_t        path;
char            partialPath[NWMAX_DIR_PATH_LENGTH];

path.serverConnID = 0;
path.dirHandle = 3;
strcpy( partialPath, "WP" );
path.pathName = partialPath;
```

3) The application can allocate a directory handle to point to the target directory. Note that the pathName is null when the dirHandle being passed already references the full path. In the following example, the path to be accessed is the SYS:APPS/WP directory on file server connection 0 (first file server attached to). A dirHandle of 4 has already been allocated for SYS:APPS/WP.

```
NWPath_t        path;
char            noPath[NWMAX_DIR_PATH_LENGTH];

path.serverConnID = 0;
path.dirHandle = 4;
strcpy( noPath, "" );
path.pathName = noPath;
```

Path Services provides six calls to allow the manipulation of directory handles:

- NWAllocPermanentDirHandle
- NWAllocTemporaryDirHandle

- NWDeallocateDirHandle
- NWGetDirPath
- NWParseFullPath
- NWSetDirHandle

NWAllocPermanentDirHandle. This function allows an application to assign a directory handle and permanently maps a workstation drive to a network directory.

NWAllocTemporaryDirHandle. This function temporarily assigns a directory handle and maps a workstation drive to a network directory.

NWDeallocateDirHandle. This function deallocates a previously allocated directory handle. However, the file server automatically deallocates directory handles under the following situations:

- Temporary directory handles are deallocated when the application loses its connection (logs out).
- Permanent directory handles are deallocated when another permanent handle is allocated to the same path.

The full path that a directory handle points to can be seen using the NWGetDirPath function call. To further see the different parts of a full path, the NWParseFullPath is used.

NWSetDirHandle. This function assigns a directory handle to a file server directory path (relative to an existing directory handle).

Permanent and Temporary Directory Handles

Temporary directory handles exist until the application that allocated them exits or calls the EndOfJob function. The mapping of a workstation drive letter to a permanent directory handle continues after the application that created the mapping exits. Temporary directory handles also differ from permanent directory handles in that several temporary directory handles can be assigned the same drive letter, but still be separate handles mapped to different directory paths.

Applications allocate permanent directory handles by calling NWAllocPermanentDirHandle, which takes the same parameters as NWAllocTemporaryDirHandle. Permanent directory handles are not automatically deallocated when the application that allocated them exits. The NWDeallocateDirHandle function is provided to deallocate directory handles. Even though NetWare automatically deallocates temporary directory handles when the application that allocated them exits, some application developers prefer to explicitly deallocate temporary directory handles as a matter of programming style.

File Paths

When specifying a file to a file services call, an application can use either a full or a partial file path. A full file path has the following format:

Volume:\Directory\...\Directory\File

Because a full file path completely specifies a file, file services calls ignore the value of their directory handle parameter when an application passes a full file path.

A partial file path includes a file name and optionally one or more antecedent directory names. The file service routine then combines the directory handle setting and the partial file path to obtain a full file specification. For example, suppose an application calls a file services routine and passes a directory handle mapped to the directory WORK:\HOME\MARY and a file path parameter containing the partial file path ACCTS\ACTIVE\ZZYZX.DAT. The routine would use the directory handle and partial file path to identify the file WORK:\HOME\MARY\ ACCTS\ACTIVE\ZZYZX.DAT.

A full path, including a server name, has the following format:

Server/Volume:\Directory\...\Directory\File

The NWParseFullPath function call uses the full path, including server name.