

高级网络应用程序开发技术
——用于 DOS 环境的

Netware C 语言接口

北京科海培训中心

TP312
2025

高级网络应用程序开发技术
——用于 DOS 环境的

NetWare C 语言接口

北京科海培训中心

引 言

如何使用本书

本书详尽描述了在 DOS 环境中开发网络应用程序的 NetWare C 接口库函数。所有的库函数按功能的不同被编在不同的章节里，每一章中的函数按字母顺序排列。

在每页的最上方是函数名；接下来是函数的语法和对每个参数简要的说明；然后是该函数的返回值；关于函数更详细的描述放在“说明”一节。与被说明函数相关的其它函数放在“参见”一节。每个函数都有一个简单的例子来说明函数的用法。除非经过特别说明，本指南列出的函数都是用软中断 21h 访问的。

数据类型

NetWare C 接口接受下面类型的数据

.BYTE	unsigned char
.WORD	unsigned int
.DWORD	unsigned long
.ASCIIZ	(非空字符串)

标志

一些函数需要设置或清除某些存取或封锁标志来开启或关闭某些功能。

安装 NIT 文件

如果不能确定安装步骤的话，请详细阅读开发系统盘中第一张盘上的 README.DOC 文件，上面有安装 C 接口库 NIT 文件的详细步骤。

网络通用说明符

在网络中，文件名和目录名可以用含有句号的字符串来表示，其中的句号被认为是一个一般的字符。这一点使得网络能支持不同的操作系统命名语法，象 Macintosh、DOS 和 UNIX。

应用程序应当把 FCB 格式的名字转换为句号在文件名和扩展名之间合适位置的一个字符串（通常这由 shell 完成）。一个 FCB 格式的名字按下面的步骤转换为相应的网络表示：

- 空格从名字中滤掉。
- 如果存在扩展名，句号被放在文件名和扩展名之间。

虽然网络区分象“FILENAME.”和“FILENAME.”这样的文件名，而 DOS 在以 FCB 格式存入文件名时却不加以区分。

网络对通配符的处理与 DOS 略有不同，网络通配替代算法实现如下：

- 除了五个特殊字符，其它所有字符都按普通字符处理。
- 在搜索模式下，这五个字符必须用网络上记录的文件名和目录名来匹配。
- 所有的文件名和目录名在网络上用大写字母来记录。

这五个特殊字符定义如下：

- * (星号)
- ? (问号)
- ^ (句号和上箭头)
- ^? (问号和上箭头)
- ^* (星号和上箭头)

星号可以匹配零个或更多的字符，所以“*”可以匹配任何一个字符串。

模式“*. *”可以匹配任何一个带有句号的名字。（它不会匹配一个没有句号的名字。句号与其它字符作同样处理。）

模式“A * B.D * E”会匹配象下面这样的名字：

- " AB.DE "
- " ACCB.DOE "
- " AXLE-REB.DIE "
- " AX.CE.B.D..E "

上面最后一个例子显示了网络通配符比 DOS 通配符更灵活。

星号可以匹配一个句号。在一个较模糊的情况下，例如用“*. *”来匹配“A.B.C”，星号将匹配它能匹配的最长串，所以第一个星号匹配“A.B.”，第二个星号匹配“C”。

问号只用来匹配一个字符，包括句号。因此模式“A? C”能进行下面的名字：

- "ABC"
- "ARC"
- "A.C"

但该模式不能匹配“AC”。除此之外，模式“AC?”不能匹配“AC”名字。问号必须匹配一个字符，否则匹配就失败。

“*”和“?”能匹配把句号当做普通字符的任何字符串。但对于 DOS 来讲，句号不能做为普通字符使用，因为 DOS 不允许句号嵌入名字中。因此在网络环境中必须有其它通配符来满足 DOS 这种情况。

满足上面条件的第一个通配符是扩展星号“^*”，它可以匹配除句号外的任何一个字符，因此，模式“^*”可以进行下面的匹配：

- "ABC"
- "A"
- "FILENAME"

但是模式“^*”不能进行下面的匹配：

- "A.B"
- "FILE.EXT"

模式“*. *”可以匹配有一个或多个问号的任何一个名字，而“^* . ^*”只能匹配有一个句号的字名字。

和星号一样，扩展星号也可以匹配零个字符，所以模式“A^*C”可以匹配“AC”。

扩展问号“^?”可以匹配除问号外的任何一个字符。

扩展问号可以匹配零个或1个字符。如果在匹配字符串中的下一个字符是句号或空字符(串结束符)，扩展问号被认为是匹配了零个或没有匹配任何字符，否则这个扩展问号被认为匹配了一个字符。

例如，模式“A^?C”可以匹配“ABC”或“ARC”，但不能匹配“A.C”或“AC”。

模式“A^?^?”能匹配“A”、“AB”或“ABC”，但不能匹配“A.B”或“AB.”。

在通常 DOS 环境下的通配搜索“*.*”可以在网络上表示为八个扩展问号，一个扩展句号(见后文)和三个扩展问号。

扩展句号“^.”可以匹配一个句号或一个字符串结束符。

所以模式“AB^.”可以匹配“AB”或“AB.”，但不能匹配“ABC”或“AB.C”

采用扩展句号主要是为了方便 Shell 或程序搜索忽略句号的文件名。通过使用前面介绍的通配符，网络用户可以设计复杂的检索模式。

网络通配符可以用任何方式组合在一个字符串中，在进行文件命名匹配时，应注意下面几点：

- 用户不希望匹配的句号可能会匹配到星号或问号中。
- 调用 DOS Shell 的 FCB 搜索能使用下面的通配符：
^.
^?
^*

由 Novell 提供的目录实用程序可以使用功能更强、更通用的问号和星号通配符，它们可以把目录名中的句号当作一般字符处理。除此之外，在使用通配模式检索装订库实体或属性时，网络用户只能使用“?”和“*”通配符。

序 言

Novell 公司是美国一家微机局域网公司。从 1983 年 Novell 公司推出它的第一代产品 Share Net 到 1991 年的 NetWare V3.11, 短短的六、七年时间, Novell 公司的 NetWare 已经成了局域网工业界的标准, 其销售量占全球微机局域网市场销售量的 65%。在我国随着办公室自动化和信息管理信息系统的推广, 越来越多的单位在网络选型上把目光集中在 Novell 公司的 NetWare 局域网。这不仅仅是由于它采用了高效的硬盘存贮管理技术, 完善的系统容错技术以及严密的安全保密技术, 更在于它开放的系统设计战略。由于 Novell 网的普及, 许多单位的科研人员已不能满足于简单的使用, 越来越多的用户感到开发能满足自己特殊应用需求软件的迫切性。美国 Novell 公司在设计自己网络操作系统时已充分考虑到这一点, 它为第三家厂方和应用开发人员提供了一整套完善的开发工具, 信之能满足各种开发人员的需要。其中最为常用的是用于 DOS 环境的 C 语言接口和汇编系统调用。由于 C 语言的普及性, C 语言接口不仅有助于广大用户深入学习和掌握 Novell 系统产品的原理和使用, 而且对于网络专业开发人员也是一个不可多得的有力助手。

我们有幸得到 NetWare C Interface 的 90 年最新资料, 在学习和使用的基础上, 我们编译整理了全套资料, 奉献给国内网络界同仁和广大用户。

在全部译文中, 引言、一、二、四、八、十、十一、十三、十四、十五、十六章由小华同志编译, 三、五、六、七、九、十二、十七、十八章由江君同志编译。

由于目前国内对于 Novell 网的术语尚无统一的定义, 同时由于编译者的水平, 书中不妥和错误之处在所难免, 恳请诸位批评指正。

作 者

金 秋

1991 年于北京

目 录

引 言	1
第一章 计账服务	1
第二章 DOS 环境的 AFP 服务	8
第三章 装订库服务	35
第四章 通讯服务	66
第五章 连接服务	92
第六章 诊断服务	104
第七章 目录服务	163
第八章 文件服务器环境服务	204
第九章 文件服务	254
第十章 消息服务	270
第十一章 杂项服务	281
第十二章 打印服务	289
第十三章 队列服务	314
第十四章 SAP 服务	342
第十五章 同步服务	348
第十六章 TTS 服务	368
第十七章 VAP 服务	377
第十八章 工作站环境服务	401
附录 A 错误码汇编(按字母序)	413
附录 B 错误码汇编(按数字序)	418
附录 C 符号常数定义	422
附录 D 函数调用对比表	428
附录 E 索引	441

第一章 计帐服务

函数	页码
GetAccountStatus.....	3
SubmitAccountCharge.....	4
SubmitAccountHold.....	5
SubmitAccountNote.....	6

计帐服务介绍

本章介绍的四个函数可以使软件开发人员开启服务器计帐功能，使服务器按它们提供的服务收费。例如，一个数据库服务器可以按用户浏览的记录数、用户请求服务的次数或者用户连接时间收费；而一个打印服务器可以按打印的页数收费。

注：

在阅读本书之前，开发者必须熟悉与装订库有关的一些概念：实体，属性和值。关于这些概念的定义，见“装订库服务”介绍

下面的情况说明了计帐调用的使用情况：FS₁ 是网络上许多文件服务器中的一个。FS₁ 的装订库包括几个实体（带有不同的属性），它们的情况如下：

Object Name	Object DD	Properties
FS1	00030011h	ACCOUNT_SERVERS
BILL	00060025h	ACCOUNT_BALANCE ACCOUNT_HOLDS
PSERVER	5C2701F1h	

在这个例子中，BILL 是一个顾客，PSERVER 是一个打印服务器，它是 FS₁ 装订库中的一个装订库实体。FS₁ 的超级用户通过把 PSERVER 的实体 ID 加到 FS₁ 的 128 字节的 ACCOUNT_SERVERS 属性中使之成为一个计帐服务器。这使得 FS₁ 装订库中的任何一个实体如果使用 PSERVER 提供的打印服务器，PSERVER 均向其收费。

工作站 WS₁ 上的一个用户以 BILL 身份登录到 FS₁。ACCOUNT_BALANCE 属性是超级用户级用户才能修改的属性。ACCOUNT_BALANCE 属性定义如下：

OFFesc	Content	Type	Order	
0	Balance	long	signed hi-lo	5000
4	Credit Limit	long	signed hi-lo	
8	Reserved	byte[120]		

Balance 域指定 BILL 可以用来支付收费服务的金额。由网络管理员选择货币单位。在本例中，单位是 1 美分，所以 BILL 的计帐余额是 \$ 50.00 (1388h)。

Credit Limit 域表示用户 BILL 能够进行赊帐服务的金额。这个域含有一个有符号整数，表示最多赊帐额。当用户的赊帐额低于这个数时，系统不提供收费的服务。这个域的值可以是正数（要求用户保留一定的基金）。80000000h 是最大赊帐值。

BILL 的 ACCOUNT_HOLDS 定义如下：

Offset	Content	Type	Order
0	Server1(Object ID)	long	hi-lo
4	Server1(Amount)	long	signed hi-lo
	Server2(Object ID)	long	hi-lo
	Server2(Amount)	long	signed hi-lo
120	Server16(Object ID)	long	hi-lo
124	Server16(Amount)	long	signed hi-lo

每个服务器的实体 ID 域指定将要为 BILL 提供一个服务的某个实体 ID，每个服务器的 Amount 域表示该服务的估算收费额。

如果一个顾客请求一个在 PSERVER 上打印 10 页文件，PSERVER 将完成下面几步：

- 1) PSERVER 按每页 7 美分收费，BILL 的打印作业将需 70 美分。
- 2) PSERVER 调用 Submit Account Hold 来向 BILL 收取 70 美分的打印费用。
- 3) PSERVER 将打印作业放入打印队列等待打印。

Get Account Status

本函数返回一个装订库实体的计帐情况。

语法

```
int GetAccountStatus(WORD binderyObjectType, char * binderyObjectName, long * balance, long
* limit, long holds,)
```

binderyObjectType 要查询的装订库实体的类型。(见“装订库服务”介绍中的实体类型)。

binderyObjectName 含有要查询计帐状态的装订库实体名 (最长48个字符，包括零结束符)。

balance 该实体可以用来支付收费服务的计帐结余额。(选择项：如果本参数无需返回，则在调用时置为空 (null))

limit 该实体的允许赊帐额。(选项：如果不需要返回它的值，应将它置为空)。

holds 一个保存了账目结算的实体表 (最大为16)。

返回值

-1 OUT_OF_MEMORY

0(0x00) SUCCESSFUL
 192(0xC0) NO__ACCOUNT__PRIVILEGES
 193(0xC1) NO__ACCOUNT__BALANCE

说明

本函数通过传递 binderyObjectType 和 binderyObjectName 参数查询文件服务器装订库中某一装订库实体的当前计帐状态，它们在 balance、limit 和 holds 参数中返回。

binderyObjectType 和 binderyObjectName 参数必须能唯一确定一个装订库实体，它们中不能含任何通配符。binderyObjectName 是以空字符结束的 1 到 47 个字符长的字符串，这些字符只能是可打印字符而且禁止使用斜线、反斜线、冒号、分号、逗号、星号、空格和问号。

balance 参数中的值为实体的帐目结余。

holds 参数列出向实体发出 SubmitAccountHold 调用的服务器。holds 参数也是向该实体发出 SubmitAccountHold 调用的增值服务器的实体 ID 号。

本函数不把审计记录记录到 SYS: SYSTEM\NET\$ ACCT.DAT 审计文件中。

调用范例

```
#include <nit.h>
#include <niterror.h>
#include <nwacct.h>
#define maxHoldingServers 16

int completionCode;
WORD binderyObjectType;
char binderyObjectName[48];
long balance;
long limit;
long holds[32];

completionCode = GetAccountStatus(binderyObjectType,
                                   binderyObjectName, balance, limit, holds);
```

SubmitAccountCharge

本函数改变一个装订库实体的帐目结余。

语法

```
int SubmitAccountCharge(WORD binderyObjectType,
                        char * binderyObjectName, WORD serviceType,
                        long chargeAmount, long cancelHoldAmount,
                        WORD commentType, char * comment)

binderyObjectType      将要修改帐目结余的装订库实体（见“装订库服务介绍”中的  

                       实体类型）。

binderyObjectName      含有将要修改计帐结余的装订库实体名（名字可以为1到47  

                       个字符长）。
```

ServiceType	表示收费服务的类型。
ChargeAmount	计帐服务器对它提供服务的收费额。
CancelHoldAmount	将要从该文件服务器过去保留的计帐额中减去的值。如果过去没有调用过 SubmitAccountHolds, 那么这个域的值应当为零。
CommentType	表示写在审计报告上的注释类型。
Comment	与实体计帐额有关的注释。

返回值

0(0x00)	SUCCESSFUL
192(0xC0)	NO__ACCOUNT__PRIVILEGES
193(0xC1)	NO__ACCOUNT__BALANCE
194(0xC2)	CREDIT__LIMIT__EXCEEDED

说明

本函数向一个实体收费并改变它的计帐余额。本函数也在审计记录中加一个有关计帐事务的注释。

binderyObjectType 和 bindery Object Name 参数必须能唯一确定一个装订库实体。它们中不能含任何通配符。binderyObjectName 是以空字符结束的 1 到 47 个字符长的字符串。这些字符只能是可打印字符而且禁止使用斜线、反斜线、冒号、分号、逗号、星号和问号。

ServiceType 参数含有某一个收费服务的类型。外部服务器必须使用他们自己的实体类型。因为一个服务器可能提供几种不同的收费服务, 所以用户需指明服务类型。

Comment 是增值服务器在审计记录中加入的注释。审计记录包含在文件 SYS:SYSTEM\NET\$ACCT.DAT 文件中。

调用范例

```
#include <nit.h>
#include <niterror.h>
```

```
int      completionCode;
WORD     binderyObjectType;
char     binderyObjectName[48];
WORD     serviceType;
long     chargeAmount;
long     cancelHoldAmount;
WORD     commentType;
char     comment[48];
```

```
completionCode = SubmitAccountCharge(binderyObjectType,
                                     binderyObjectName, serviceType, chargeAmount,
                                     cancelHoldAmount, commentType, comment)
```

SubmitAccountHold

本函数通过调用 `SubmitAccountCharge` 存入一定数目的实体的计帐结余。

语法

```
int SubmitAccountHold(WORD binderyObjectType,
    char * binderyObjectName, long reserveAmount)

binderyObjectType      将要部分保留帐目结余的装订库实体类型（见“装订库服务介绍”中的实体类型）。

binderyObjectName      装订库实体名（长为1至47个字符，包括零结束符）。

reserveAmount          将要存入顾客计账余额中的余额。
```

返回值

```
0(0x00)    SUCCESSFUL
192(0xC0)   NO__ACCOUNT__PRIVILEGES
193(0xC1)   NO__ACCOUNT__BALANCE
194(0xC2)   CREDIT__LIMIT__BXCEEDED
195(0xC3)   TOO__MANY__HOLDS
```

说明

本函数给一个实体的帐目结余存入一定数量的钱额。

`binderyObject` 和 `binderyObjectName` 参数必须能唯一确定一个装订库实体。它们中不能含任何通配符。`binderyObjectName` 是以空字符结束的 1 到 47 个字符长的字符串。这些字符只能是可打印字符而且禁止使用斜线、反斜线、冒号、分号、逗号、星号和问号。

`reserveAmount` 域取得服务器提供给某实体的帐目结余额。

调用范例

```
#include    <nit.h>
#include    <niterror.h>

int         completionCode;
WORD        binderyObjectType;
char        binderyObjectName[48];
long        reserveAmount;

completionCode = SubmitAccountHold(binderyObjectType,
    binderyObjectName, reserveAmount);
```

SubmitAccountNote

本函数将有关一个实体帐目的注释加到一个审计记录中。

语法

```
int SubmitAccountNote(WORD binderyObjectType,
    char * binderyObjectName, WORD serviceType,
    WORD commentType, char * comment)

binderyObjectType      被请求的装订库实体类型（见“装订库服务介绍”）

binderyObjectName      含有被请求的装订库实体的名字（可以为1到47个字符长，包括空结束符）。

ServiceType            收费服务的类型（通常是计帐服务器的实体类型）。
```

CommentType 表示在Comment参数中的注释类型。

Comment 与一个实体计帐有关的注释。

返回值

0(0x00) **SUCCESSFUL**

192(0xC0) **NO_ACCOUNT_PRIVILEGES**

说明

本函数将一个计帐事务的注释加入到一个审计记录中。

binderyObjectType 和 **binderyObjectName** 参数必须能唯一确定一个装订库实体。它们中不能含任何通配符。**binderyObjectName** 是以空字符结束的 1 到 47 个字符长的字符串。这些字符只能是可打印字符而且禁止使用斜线、反斜线、冒号、分号、逗号、星号和问号。

ServiceType 参数含有计帐服务器的实体类型。

CommentType 参数含有 **Comment** 参数的注释类型。注释类型由 Novell 管理，它们是：

- Connect Time Charge
- Disk Storage Charge
- Log In Note
- log Out Note
- Account Locked Note
- Server Time Modified Note

Comment 参数含有服务器在审计记录中的入口。审计记录在文件 **SYS:SYSTEM\NET\$ACCT.DAT** 文件中。

调用范例

```
#include <nit.h>
#include <niterror.h>

int      completionCode;
WORD     binderyObjectType;
char     binderyObjectName[48];
WORD     serviceType;
WORD     commentType;
char     comment[48];

completionCode = SubmitAccountNote(binderyObjectType,
                                   binderyObjectName, serviceType, commentType, comment);
```

第二章 DOS 环境的 AFP 服务

函数

页码

AFPAllocTemporaryDirHandle	9
AFPCreateDirectory	11
AFPCreateFile	13
AFPDelete	14
AFPDirectoryEntry	15
AFPGetEntryIDFromName	16
AFPGetEntryIDFromNetWareHandle	17
AFPGetEntryIDFromPathName	18
AFPGetFileInformation	19
AFPOpenFileFork	23
AFPRename	25
AFPScanFileInformation	26
AFPSetFileInformation	31
AFPSupported	34

DOS 环境 AFP 服务介绍

Apple Talk 文件服务协议 (AFP) 工具允许应用程序创建、访问和删除文件服务器上 Macintosh 格式的目录和文件。下面是 Netware 环境中用于 AFP 的命名约定。

AFP 目录名和文件名长度可以为一个字符到三十一个字符。较长的名字前有一个字节说明名字的长度。名字由除了冒号(:) 和空字符 (null) 的 ASCII 码字符组成。文件服务器根据创建的 AFP 目录名和文件名自动生成与其对应的 DOS 格式的名字 (短名字)。所以, 文件服务器上的 AFP 目录名和文件名都有一个长名字和一个短名字。为了把一个不含句号的长名字转换为一个短名字, 文件服务器选择了长名字的前八个字符作为它的短名字。

长名字	短名字
THISISANAME	THISISAN

如果长名字的前九个字符中含有句号, 文件服务器就选择句号前的八个字符和句号后的三个字符同时包括第一个句号:

长名字	短名字
THIS.IS.A.NAME	THIS.IS

这种双名字机制允许 AFP 和非 AFP 工作站访问 AFP 目录和文件。上面的名字机制也可以引起下面的情况。假设一个应用在它的父目录里创建了下面二个文件:

长名字
THIS IS THE FIRST FILE
THIS IS THE SECOND FILE

因为它们的短名字都是一样的 (THISISTH), 文件服务器在第二个文件短名后加一个递增的十进制数: THISIST1. 现在假设一台 Macintosh 工作站把文件 THIS IS THE SECOND FILE (THISIST1) 拷贝到软盘上, 然后把它从软盘拷贝到文件服务器上另一个目录中, 这时文件服务器将文件短名变为 THISISTH。程序员应该记住这一点。

与 Netware 目录路径包括斜线和反斜线来区别目录和文件不同, AFP 路径用空字符 (十进制零) 来做到这一点:

Netware 路径: VOLUME: DIRECTORY / DIRECTORY / FILE

AFP 路径: VOLUME: DIRECTORY DIRECTORY FILE

应用程序经常将目录和文件的短名字与 Netware 目录句柄以及 Netware 格式的目录和文件路径相对应。同样, 应用程序也应将目录和文件的长名字与 AFP 入口 ID 和 AFP 格式的目录和文件路径相对应。应用程序不应将 Netware 目录句柄与长名字对应起来, 也不应将 AFP 入口 ID 与短名对应起来。(后面的函数功能说明中将介绍有关 AFP 入口 ID 的内容)

AFPAllocTemporaryDirHandle

本函数将一个 Netware 目录句柄映射到一个 AFP 目录。

语法

```
int AFPAllocTemporaryDirHandle(WORD connectionID, BYTE volumeNum, long AFPEntID,  
                                char * AFPPathString,
```


ConnectionID	标明目标文件服务器的连接 ID (1 到 8)。
VolumeNum	标明目标卷 (0 到 31)。
AFPEnttyID	注明一个卷或目录, 功能与 NetWare 目录句柄类似。
AFPPathString	长名字格式, 与 AFPEnttyID 有关。
NetWareDirectoryHandle	返回一个字节的索引号 (0 到 255), 指向文件服务器上的一个卷或目录。
AccessRights	返回用户在目标目录的有效权限。

返回值

0	(0x00)	成功
156	(0x9c)	无效路径

说明

connectionID 是服务器地址值在工作站 Shell 中 ConnectionID 表中的位置。

VolumeNumber 标明文件服务器卷表中的一个卷, 卷表含有文件服务器上每个卷的信息。Netware 文件服务器最多可以包括 32 (0 到 31) 个卷。应用可以通过执行 GetVolume Number (见目录服务) 返回一个卷号。

4 字节的 AFPEnttyID 是与 1 字节 NetWare 目录句柄等价的 AFP 入口标识, 但只有一个例外: NetWare 目录句柄指向入文件服务器上的一个卷或目录, 而 AFP 入口 ID 指向文件服务器上的一个卷、目录或文件。

AFPFilePathS 域至少要注明目标 AFP 文件名, 也可选择使用一个或更多的双亲目录名。这个域可以为 1 到 255 个字节长。AFP 文件名是 1 到 31 个字符长, 其中只能有一个冒号(:)。AFP 文件的路径格式如下:

VOLUME:DIRECTORY DIRECTORY...DIRECTORY FILE

与 NetWare 目录路径中包含斜线 (/) 或反斜线 (\) 来分离目录和文件名不同, AFP 路径用零字符 (null)。应用经常使用 AFP 入口 ID 和 AFP 文件路径来注明一个特定的 AFP 文件。例如, 在下面的假设下:

- 注明的 AFP 入口 ID 指向 SYS;
- 注明的 AFP 目录路径 = PUBLIC WORDP FILE1.

在这种情况下, 指定的文件是 SYS: PUBLIC WORDP FILE1.

当一个应用通过只使用 AFP 入口 ID 定义了一个目标文件时, 它必须将 0x00 传到 AFP 文件路径长度域。当一个应用只通过 AFP 文件路径定义一个 AFP 文件时, 它必须将 0x00 传到 AFP 入口 ID 域。

NetWareDirectotyHandle 是一个索引号 (1 到 255)。文件服务器为每个登录的工作站维护一个目录句柄表。目录句柄指向文件服务器上的一个卷或一个目录。

AccessRights 返回用户在目标目录的有效权限, 其中的每一位定义如下: