



NQ Library Reference

YNQ
YNQ-1.5.1

Contents

1	Module Index	1
1.1	Modules	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Module Documentation	6
4.1	Authentication	6
4.1.1	Detailed Description	6
4.1.2	Macro Definition Documentation	6
4.1.2.1	AM_MAXSECURITYLEVEL	6
4.1.3	Function Documentation	7
4.1.3.1	amStart	7
4.1.3.2	amShutdown	7
4.1.3.3	amClientDefineLevel	7
4.1.3.4	amSetNonSecureAuthentication	8
4.2	Encryption Algorithms	9
4.2.1	Detailed Description	9
4.2.2	Macro Definition Documentation	9
4.2.2.1	AM_CRYPTER_NONE	9
4.2.2.2	AM_CRYPTER_LM	9
4.2.2.3	AM_CRYPTER_NTLM	9
4.2.2.4	AM_CRYPTER_LM2	9
4.2.2.5	AM_CRYPTER_NTLM2	9
4.3	Security Mechanisms	10
4.3.1	Detailed Description	10
4.3.2	Macro Definition Documentation	10
4.3.2.1	AM_MECH_NTLMSSP	10
4.3.2.2	AM_MECH_KERBEROS	10
4.4	Security Password Type	11
4.4.1	Detailed Description	11
4.4.2	Macro Definition Documentation	11
4.4.2.1	AM_CREDENTIALS_PASSWORD_PLAINTEXT	11
4.4.2.2	AM_CREDENTIALS_PASSWORD_MD4	11
4.5	Client	12
4.5.1	Detailed Description	12
4.6	Credential fields length	13
4.6.1	Detailed Description	13
4.6.2	Macro Definition Documentation	13
4.6.2.1	CM_USERNAMELENGTH	13
4.6.2.2	PASSWORD_LENGTH	13
4.6.2.3	DOMAIN_LENGTH	13
4.7	Driver Notification Events	14

4.7.1	Detailed Description	14
4.7.2	Macro Definition Documentation	14
4.7.2.1	NOTIFY_SRCH_HANDLE_ERROR	14
4.7.2.2	NOTIFY_FILE_HANDLE_ERROR	14
4.8	File attributes	15
4.8.1	Detailed Description	15
4.8.2	Macro Definition Documentation	15
4.8.2.1	CIFS_ATTR_READONLY	15
4.8.2.2	CIFS_ATTR_HIDDEN	15
4.8.2.3	CIFS_ATTR_SYSTEM	15
4.8.2.4	CIFS_ATTR_VOLUME	15
4.8.2.5	CIFS_ATTR_DIR	15
4.8.2.6	CIFS_ATTR_ARCHIVE	15
4.8.2.7	CIFS_ATTR_ALL	15
4.9	File access mode	16
4.9.1	Detailed Description	16
4.9.2	Macro Definition Documentation	16
4.9.2.1	FILE_AM_READ	16
4.9.2.2	FILE_AM_WRITE	16
4.9.2.3	FILE_AM_READ_WRITE	16
4.9.2.4	FILE_AM_SPECIAL_MASK	16
4.10	File share mode	17
4.10.1	Detailed Description	17
4.10.2	Macro Definition Documentation	17
4.10.2.1	FILE_SM_COMPAT	17
4.10.2.2	FILE_SM_EXCLUSIVE	17
4.10.2.3	FILE_SM_DENY_WRITE	17
4.10.2.4	FILE_SM_DENY_READ	17
4.10.2.5	FILE_SM_DENY_NONE	17
4.11	File locality	18
4.11.1	Detailed Description	18
4.11.2	Macro Definition Documentation	18
4.11.2.1	FILE_LCL_UNKNOWN	18
4.11.2.2	FILE_LCL_SEQUENTIAL	18
4.11.2.3	FILE_LCL_RANDOM	18
4.11.2.4	FILE_LCL_MIXED	18
4.12	File create action	19
4.12.1	Detailed Description	19
4.12.2	Macro Definition Documentation	19
4.12.2.1	FILE_CA_FAIL	19
4.12.2.2	FILE_CA_CREATE	19
4.13	File open action	20
4.13.1	Detailed Description	20
4.13.2	Macro Definition Documentation	20
4.13.2.1	FILE_OA_FAIL	20
4.13.2.2	FILE_OA_OPEN	20
4.13.2.3	FILE_OA_TRUNCATE	20
4.14	File seek constants	21
4.14.1	Detailed Description	21
4.14.2	Macro Definition Documentation	21
4.14.2.1	SEEK_FILE_BEGIN	21
4.14.2.2	SEEK_FILE_CURRENT	21
4.14.2.3	SEEK_FILE_END	21
4.15	Authentication Level	22
4.15.1	Detailed Description	22
4.15.2	Macro Definition Documentation	22
4.15.2.1	NQ_CC_AUTH_LM_AND_NTLM	22
4.15.2.2	NQ_CC_AUTH_NTLM	22

4.15.2.3	NQ_CC_AUTH_NTLM_V2	22
4.15.2.4	NQ_CC_AUTH_SPNEGO_KERBEROS	22
4.16	User logon type	23
4.16.1	Detailed Description	23
4.16.2	Macro Definition Documentation	23
4.16.2.1	CC_LOGON_TYPE_REGULAR	23
4.16.2.2	CC_LOGON_TYPE_ANONYMOUS	23
4.16.2.3	CC_LOGON_TYPE_GUEST	23
4.17	Server capabilities	24
4.17.1	Detailed Description	24
4.17.2	Macro Definition Documentation	24
4.17.2.1	CC_CAPABILITY_DFS	24
4.17.2.2	CC_CAPABILITY_LEASING	24
4.17.2.3	CC_CAPABILITY_LARGE_MTU	24
4.17.2.4	CC_CAPABILITY_MULTI_CHANNEL	24
4.17.2.5	CC_CAPABILITY_PERSISTENT_HANDLES	24
4.17.2.6	CC_CAPABILITY_DIRECTORY_LEASING	24
4.17.2.7	CC_CAPABILITY_ENCRYPTION	24
4.18	signing policy	25
4.18.1	Detailed Description	25
4.18.2	Macro Definition Documentation	25
4.18.2.1	CC_SIGNING_ENABLED	25
4.18.2.2	CC_SIGNING_REQUIRED	25
4.19	Initialization and Shutdown	26
4.19.1	Detailed Description	26
4.19.2	Function Documentation	26
4.19.2.1	ccInit	26
4.19.2.2	ccShutdown	26
4.19.2.3	ccIsInitialized	26
4.20	Mount and Server	27
4.20.1	Detailed Description	27
4.20.2	Function Documentation	27
4.20.2.1	ccAddMountA	27
4.20.2.2	ccAddMount	27
4.20.2.3	ccAddMountWithCredsA	28
4.20.2.4	ccAddMountWithCreds	28
4.20.2.5	ccRemoveMountA	29
4.20.2.6	ccRemoveMount	29
4.20.2.7	ccGetServerIdA	30
4.20.2.8	ccGetServerId	30
4.21	Mount information	31
4.21.1	Detailed Description	31
4.21.2	Function Documentation	31
4.21.2.1	ccGetMountInformationA	31
4.21.2.2	ccGetMountInformation	31
4.22	Search Operations	33
4.22.1	Detailed Description	33
4.22.2	Typedef Documentation	33
4.22.2.1	FindFileDataA_t	33
4.22.3	Function Documentation	33
4.22.3.1	ccFindFirstFileA	33
4.22.3.2	ccFindFirstFile	34
4.22.3.3	ccFindNextFileA	34
4.22.3.4	ccFindNextFile	35
4.22.3.5	ccFindClose	35
4.23	Directory Operations	36
4.23.1	Detailed Description	36
4.23.2	Function Documentation	36

4.23.2.1	ccCreateDirectoryA	36
4.23.2.2	ccCreateDirectory	36
4.23.2.3	ccRemoveDirectoryA	36
4.23.2.4	ccRemoveDirectory	37
4.24	File Operations	38
4.24.1	Detailed Description	38
4.24.2	Typedef Documentation	38
4.24.2.1	CCServerSideCopyCallback	38
4.24.3	Function Documentation	39
4.24.3.1	ccCreateFileA	39
4.24.3.2	ccCreateFile	39
4.24.3.3	ccDeleteFileA	40
4.24.3.4	ccDeleteFile	40
4.24.3.5	ccMoveFileA	40
4.24.3.6	ccMoveFile	41
4.24.3.7	ccReadFile	42
4.24.3.8	ccReadFileAsync	42
4.24.3.9	ccWriteFile	43
4.24.3.10	ccWriteFileAsync	43
4.24.3.11	ccFlushFile	44
4.24.3.12	ccCloseHandle	44
4.25	File Info Operations	45
4.25.1	Detailed Description	46
4.25.2	Typedef Documentation	46
4.25.2.1	FileInfo_t	46
4.25.3	Function Documentation	46
4.25.3.1	ccGetFilePointer	46
4.25.3.2	ccSetFilePointer	46
4.25.3.3	ccServerSideDataCopyA	46
4.25.3.4	ccServerSideDataCopy	47
4.25.3.5	ccGetDiskFreeSpaceA	47
4.25.3.6	ccGetDiskFreeSpace	48
4.25.3.7	ccGetDiskFreeSpaceExA	48
4.25.3.8	ccGetDiskFreeSpaceEx	49
4.25.3.9	ccGetFileAttributesA	49
4.25.3.10	ccGetFileAttributes	49
4.25.3.11	ccSetFileAttributesA	49
4.25.3.12	ccSetFileAttributes	50
4.25.3.13	ccGetFileInformationByNameA	50
4.25.3.14	ccGetFileInformationByName	50
4.25.3.15	ccSetFileSizeByNameA	51
4.25.3.16	ccSetFileSizeByName	51
4.25.3.17	ccSetFileSizeByHandle	51
4.25.3.18	ccGetFileTime	52
4.25.3.19	ccSetFileTime	53
4.25.3.20	ccGetFileSize	53
4.25.3.21	ccGetFileInformationByHandle	53
4.26	Timeout Operations	55
4.26.1	Detailed Description	55
4.26.2	Function Documentation	55
4.26.2.1	ccConfigSetTimeout	55
4.26.2.2	ccConfigGetTimeout	55
4.26.2.3	ccSetGetBackupListTimeout	55
4.27	Network Browsing	56
4.27.1	Detailed Description	56
4.27.2	Function Documentation	56
4.27.2.1	ccNetworkSetDefaultDomainA	56
4.27.2.2	ccNetworkSetDefaultDomain	57

4.27.2.3	ccNetworkCacheSet	58
4.27.2.4	ccNetworkGetDefaultDomainA	58
4.27.2.5	ccNetworkGetDefaultDomain	58
4.27.2.6	ccNetworkEnumerateDomains	59
4.27.2.7	ccNetworkGetNextDomainItem	59
4.27.2.8	ccNetworkEnumerateServersA	59
4.27.2.9	ccNetworkEnumerateServers	60
4.27.2.10	ccNetworkGetNextServerItem	60
4.27.2.11	ccNetworkEnumerateSharesA	60
4.27.2.12	ccNetworkEnumerateShares	61
4.27.2.13	ccNetworkGetNextItemNameA	61
4.27.2.14	ccNetworkGetNextItemName	62
4.27.2.15	ccNetworkGetNextShareItem	62
4.27.2.16	ccNetworkGetShareInfoA	62
4.27.2.17	ccNetworkGetShareInfo	63
4.27.2.18	ccNetworkCloseHandle	63
4.27.2.19	ccNetworkResetHandle	64
4.28	Domain Operations	65
4.28.1	Detailed Description	65
4.28.2	Function Documentation	65
4.28.2.1	ccDomainJoinA	65
4.28.2.2	ccDomainJoin	66
4.28.2.3	ccDomainJoinPersistent	66
4.28.2.4	ccDomainUpdateSecretA	66
4.28.2.5	ccDomainUpdateSecret	67
4.28.2.6	ccDomainLeaveA	67
4.28.2.7	ccDomainLeave	68
4.28.2.8	ccDomainLogonA	68
4.28.2.9	ccDomainLogon	69
4.29	Special Operations	70
4.29.1	Detailed Description	70
4.29.2	Function Documentation	70
4.29.2.1	ccGetSigningPolicy	70
4.29.2.2	ccSetSigningPolicy	70
4.29.2.3	ccSetMinSecurityLevel	70
4.29.2.4	ccGetMinSecurityLevel	71
4.29.2.5	ccSetMaxSecurityLevel	71
4.29.2.6	ccGetMaxSecurityLevel	71
4.29.2.7	ccSetExclusiveAccessToFileA	71
4.29.2.8	ccSetExclusiveAccessToFile	72
4.29.2.9	cclsExclusiveAccessToFileA	72
4.29.2.10	cclsExclusiveAccessToFile	73
4.30	Dialects Operations	74
4.30.1	Detailed Description	74
4.30.2	Enumeration Type Documentation	74
4.30.2.1	CCSmbDialect	74
4.30.3	Function Documentation	74
4.30.3.1	ccSetSmbDialect	74
4.30.3.2	ccCifsGetSMB1Support	74
4.30.3.3	ccCifsSetSMB1Support	75
4.31	User Credentials	76
4.31.1	Detailed Description	76
4.31.2	Function Documentation	76
4.31.2.1	ccResetCredentailsA	76
4.31.2.2	ccResetCredentails	76
4.31.2.3	ccValidateUserA	77
4.31.2.4	ccValidateUser	77
4.31.2.5	ccValidateUserByCredentialsA	77

4.31.2.6	ccValidateUserByCredentials	78
4.32	Errors	79
4.32.1	Detailed Description	79
4.32.2	Function Documentation	79
4.32.2.1	ccGetNTStatus	79
4.32.2.2	ccErrorToSmbStatus	79
4.33	Auxiliary methods	80
4.33.1	Detailed Description	80
4.33.2	Function Documentation	80
4.33.2.1	ccTransportRestartRecieveThread	80
4.33.2.2	ccDump	80
4.33.2.3	ccCloseAllConnections	80
4.33.2.4	ccCloseHiddenConnections	80
4.33.2.5	ccThreadSubscribe	81
4.33.2.6	ccThreadUnsubscribe	81
4.34	Common	82
4.34.1	Detailed Description	82
4.35	Initialization and Shutdown	83
4.35.1	Detailed Description	83
4.35.2	Macro Definition Documentation	83
4.35.2.1	NQ_CLIENT	83
4.35.2.2	NQ_SERVER	83
4.35.2.3	NQ_NDDAEMON	83
4.35.3	Function Documentation	83
4.35.3.1	cmInit	83
4.35.3.2	cmExit	83
4.36	Resolver	85
4.36.1	Detailed Description	85
4.36.2	Macro Definition Documentation	85
4.36.2.1	NQ_RESOLVER_IPV4	85
4.36.2.2	NQ_RESOLVER_IPV6	85
4.36.2.3	NQ_RESOLVER_NONE	86
4.36.2.4	NQ_RESOLVER_DNS	86
4.36.2.5	NQ_RESOLVER_NETBIOS	86
4.36.2.6	NQ_RESOLVER_EXTERNAL_METHOD	86
4.36.2.7	NQ_RESOLVER_DNS_DC	86
4.36.2.8	NQ_RESOLVER_NETBIOS_DC	86
4.36.2.9	NQ_RESOLVER_WSD	86
4.36.3	Typedef Documentation	86
4.36.3.1	CMResolverNameToIpA	86
4.36.3.2	CMResolverNameToIp	87
4.36.3.3	CMResolverIpToNameA	87
4.36.3.4	CMResolverIpToName	87
4.36.4	Function Documentation	88
4.36.4.1	cmResolverSetExternalA	88
4.36.4.2	cmResolverSetExternal	88
4.36.4.3	cmResolverGetHostName	89
4.36.4.4	cmResolverGetHostIps	89
4.36.4.5	cmResolverGetDCName	90
4.36.4.6	cmResolverEnableMethod	90
4.36.4.7	cmResolverRegisterExternalMethod	91
4.36.4.8	cmResolverUpdateExternalMethodsPriority	91
4.37	Product Information	92
4.37.1	Detailed Description	92
4.37.2	Macro Definition Documentation	92
4.37.2.1	CM_PRODUCT_STRING_LEN	92
4.37.3	Function Documentation	92
4.37.3.1	cmGetCurrentProductInformation	92

4.38	Codepage	93
4.38.1	Detailed Description	93
4.38.2	Function Documentation	93
4.38.2.1	cmCodepageAdd	93
4.38.2.2	cmCodepageRemove	93
4.38.2.3	cmWideCharToMultiByte	94
4.38.2.4	cmMultiByteToWideChar	95
4.39	Cryptographic	96
4.39.1	Detailed Description	96
4.39.2	Typedef Documentation	96
4.39.2.1	CMAbstractHasher	96
4.39.2.2	CMAbstractCipher	96
4.39.2.3	CMAbstractIOCipher	97
4.39.2.4	CMAbstractHasher512	97
4.39.2.5	CMAbstractCcmEncryption	98
4.39.2.6	CMAbstractCcmDecryption	98
4.39.2.7	CMAbstractGcmEncryption	99
4.39.2.8	CMAbstractGcmDecryption	99
4.39.3	Function Documentation	100
4.39.3.1	cmSetExternalCrypters	100
4.39.3.2	cmResetExternalCrypters	101
4.40	DNS and WINS	102
4.40.1	Detailed Description	102
4.40.2	Function Documentation	102
4.40.2.1	cmDnsSetServersA	102
4.40.2.2	cmDnsSetServers	102
4.40.2.3	cmDnsGetServer	102
4.40.2.4	cmDnsGetNumDnsServers	103
4.40.2.5	cmNetBiosSetWinsA	103
4.40.2.6	cmNetBiosSetWins	103
4.40.2.7	cmNetBiosGetWins	104
4.40.2.8	cmNetBiosGetNumWinsServers	104
4.40.2.9	cmDnsSetDomainA	104
4.40.2.10	cmDnsSetDomain	104
4.41	Auxiliary	105
4.41.1	Detailed Description	105
4.41.2	Function Documentation	105
4.41.2.1	cmNetBiosNameReload	105
4.42	Server	106
4.42.1	Detailed Description	106
4.43	Initialization and Shutdown	107
4.43.1	Detailed Description	107
4.43.2	Function Documentation	107
4.43.2.1	csStart	107
4.43.2.2	csStop	107
4.44	Shares Management	108
4.44.1	Detailed Description	108
4.44.2	Function Documentation	108
4.44.2.1	csAddShareA	108
4.44.2.2	csAddShare	108
4.44.2.3	csRemoveShareA	109
4.44.2.4	csRemoveShare	109
4.44.2.5	csSetShareEncryptionA	110
4.44.2.6	csSetShareEncryption	110
4.45	Open Files Enumeration	112
4.45.1	Detailed Description	112
4.45.2	Function Documentation	112
4.45.2.1	csEnumerateOpenFilesA	112

4.45.2.2	csEnumerateOpenFiles	112
4.46	Cleanup	114
4.46.1	Detailed Description	114
4.46.2	Function Documentation	114
4.46.2.1	csCleanUserServerConnectionsA	114
4.46.2.2	csCleanUserServerConnections	114
4.47	Anonymous	115
4.47.1	Detailed Description	115
4.47.2	Function Documentation	115
4.47.2.1	csSetAnonymousAllowed	115
4.48	Event Log	116
4.48.1	Detailed Description	116
4.48.2	Function Documentation	116
4.48.2.1	evcsEnumerateConnectedShares	116
4.48.2.2	evcsEnumerateOpenFiles	116
4.49	NetBIOS Daemon	117
4.49.1	Detailed Description	117
4.49.2	Function Documentation	117
4.49.2.1	ndStart	117
4.49.2.2	ndStop	117
4.49.2.3	ndNotifyConfigurationChange	117
4.50	NQ Common	118
4.50.1	Detailed Description	118
4.51	Initialization and Shutdown	119
4.51.1	Detailed Description	119
4.51.2	Function Documentation	119
4.51.2.1	nqStart	119
4.51.2.2	nqStop	119
4.52	NetBIOS Name Registration	120
4.52.1	Detailed Description	120
4.52.2	Macro Definition Documentation	120
4.52.2.1	CM_NB_GROUP	120
4.52.2.2	CM_NB_UNIQUE	120
4.52.2.3	CM_NB_POSTFIX_WORKSTATION	120
4.52.2.4	CM_NB_POSTFIX_SERVER	120
4.52.2.5	CM_NB_POSTFIX_DOMAINMASTERBROWSER	120
4.52.2.6	CM_NB_POSTFIX_DOMAINCONTROLLER	120
4.52.2.7	CM_NB_POSTFIX_MASTERBROWSER	120
4.52.3	Function Documentation	120
4.52.3.1	nqRegisterNetBiosName	120
4.52.3.2	nqReleaseNetBiosName	121
4.52.3.3	nqRegisterNetBiosNameEx	122
4.52.3.4	nqReleaseNetBiosNameEx	122
4.53	Utils	123
4.53.1	Detailed Description	123
4.53.2	Function Documentation	123
4.53.2.1	nqEnableTraceLog	123
4.54	System API	124
4.54.1	Detailed Description	124
4.54.2	Function Documentation	124
4.54.2.1	syInit	124
4.54.2.2	syStop	124
4.55	System Common API	125
4.55.1	Detailed Description	125
4.56	File Information	126
4.56.1	Detailed Description	126
4.57	File Attributes	127
4.57.1	Detailed Description	127

4.57.2	Macro Definition Documentation	127
4.57.2.1	SY_ATTR_READONLY	127
4.57.2.2	SY_ATTR_HIDDEN	127
4.57.2.3	SY_ATTR_SYSTEM	127
4.57.2.4	SY_ATTR_VOLUME	127
4.57.2.5	SY_ATTR_DIRECTORY	127
4.57.2.6	SY_ATTR_ARCHIVE	127
4.57.2.7	SY_ATTR_DEVICE	127
4.57.2.8	SY_ATTR_NORMAL	128
4.57.2.9	SY_ATTR_TEMPORARY	128
4.57.2.10	SY_ATTR_SPARSE_FILE	128
4.57.2.11	SY_ATTR_REPARSE_POINT	128
4.57.2.12	SY_ATTR_COMPRESSED	128
4.57.2.13	SY_ATTR_OFFLINE	128
4.57.2.14	SY_ATTR_NOT_CONTENT_INDEXED	128
4.57.2.15	SY_ATTR_ENCRYPTED	128
4.58	Printer	129
4.58.1	Detailed Description	129
4.58.2	Macro Definition Documentation	129
4.58.2.1	SY_PRINTEROSVERSION_WIN	129
4.59	Printer Flags	130
4.59.1	Detailed Description	130
4.59.2	Macro Definition Documentation	130
4.59.2.1	SY_DEVICEMODE_ORIENTATION	130
4.59.2.2	SY_DEVICEMODE_PAPERSIZE	130
4.59.2.3	SY_DEVICEMODE_PAPERLENGTH	130
4.59.2.4	SY_DEVICEMODE_PAPERWIDTH	130
4.59.2.5	SY_DEVICEMODE_SCALE	130
4.59.2.6	SY_DEVICEMODE_POSITION	130
4.59.2.7	SY_DEVICEMODE_NUP	130
4.59.2.8	SY_DEVICEMODE_COPIES	131
4.59.2.9	SY_DEVICEMODE_DEFAULTSOURCE	131
4.59.2.10	SY_DEVICEMODE_PRINTQUALITY	131
4.59.2.11	SY_DEVICEMODE_COLOR	131
4.59.2.12	SY_DEVICEMODE_DUPLEX	131
4.59.2.13	SY_DEVICEMODE_YRESOLUTION	131
4.59.2.14	SY_DEVICEMODE_TTOPTION	131
4.59.2.15	SY_DEVICEMODE_COLLATE	131
4.59.2.16	SY_DEVICEMODE_FORMNAME	131
4.59.2.17	SY_DEVICEMODE_LOGPIXELS	131
4.59.2.18	SY_DEVICEMODE_BITSPERPEL	131
4.59.2.19	SY_DEVICEMODE_PELWIDTH	131
4.59.2.20	SY_DEVICEMODE_PELHEIGHT	131
4.59.2.21	SY_DEVICEMODE_DISPLAYFLAGS	131
4.59.2.22	SY_DEVICEMODE_DISPLAYFREQUENCY	131
4.59.2.23	SY_DEVICEMODE_ICMMETHOD	131
4.59.2.24	SY_DEVICEMODE_ICMINTENT	131
4.59.2.25	SY_DEVICEMODE_MEDIATYPE	131
4.59.2.26	SY_DEVICEMODE_DITHERTYPE	131
4.59.2.27	SY_DEVICEMODE_PANNINGWIDTH	131
4.59.2.28	SY_DEVICEMODE_PANNINGHEIGHT	131
4.60	Printer Status Flags	132
4.60.1	Detailed Description	132
4.60.2	Macro Definition Documentation	132
4.60.2.1	SY_PRINTERSTATUS_PAUSED	132
4.60.2.2	SY_PRINTERSTATUS_ERROR	132
4.60.2.3	SY_PRINTERSTATUS_PENDINGDELETION	132
4.60.2.4	SY_PRINTERSTATUS_PAPERJAM	132

4.60.2.5	SY_PRINTERSTATUS_PAPEROUT	132
4.60.2.6	SY_PRINTERSTATUS_MANUALFEED	132
4.60.2.7	SY_PRINTERSTATUS_PAPERPROBLEM	132
4.60.2.8	SY_PRINTERSTATUS_OFFLINE	132
4.60.2.9	SY_PRINTERSTATUS_IOACTIVE	132
4.60.2.10	SY_PRINTERSTATUS_BUSY	133
4.60.2.11	SY_PRINTERSTATUS_PRINTING	133
4.60.2.12	SY_PRINTERSTATUS_OUTPUTBINFULL	133
4.60.2.13	SY_PRINTERSTATUS_NOTAVAILABLE	133
4.60.2.14	SY_PRINTERSTATUS_WAITING	133
4.60.2.15	SY_PRINTERSTATUS_PROCESSING	133
4.60.2.16	SY_PRINTERSTATUS_INITIALIZING	133
4.60.2.17	SY_PRINTERSTATUS_WARMINGUP	133
4.60.2.18	SY_PRINTERSTATUS_TONERLOW	133
4.60.2.19	SY_PRINTERSTATUS_NOTONER	133
4.60.2.20	SY_PRINTERSTATUS_PAGEPUNT	133
4.60.2.21	SY_PRINTERSTATUS_USERINTERVENTION	133
4.60.2.22	SY_PRINTERSTATUS_OUTOFMEMORY	133
4.60.2.23	SY_PRINTERSTATUS_DOOROPEN	133
4.60.2.24	SY_PRINTERSTATUS_SERVERUNKNOWN	133
4.60.2.25	SY_PRINTERSTATUS_POWERSAVE	133
4.61	Printer Attribute Flags	134
4.61.1	Detailed Description	134
4.61.2	Macro Definition Documentation	134
4.61.2.1	SY_PRINTERATTR_QUEUED	134
4.61.2.2	SY_PRINTERATTR_DIRECT	134
4.61.2.3	SY_PRINTERATTR_DEFAULT	134
4.61.2.4	SY_PRINTERATTR_SHARED	134
4.61.2.5	SY_PRINTERATTR_NETWORK	135
4.61.2.6	SY_PRINTERATTR_RESERVED	135
4.61.2.7	SY_PRINTERATTR_LOCAL	135
4.61.2.8	SY_PRINTERATTR_DEVQUERY	135
4.61.2.9	SY_PRINTERATTR_KEEPCJOBS	135
4.61.2.10	SY_PRINTERATTR_LIFO	135
4.61.2.11	SY_PRINTERATTR_OFFLINE	135
4.61.2.12	SY_PRINTERATTR_BIDI	135
4.61.2.13	SY_PRINTERATTR_RAWONLY	135
4.61.2.14	SY_PRINTERATTR_PUBLISHED	135
4.61.2.15	SY_PRINTERATTR_EXPAND	135
4.61.2.16	SY_PRINTERATTR_CONTAINER	135
4.61.2.17	SY_PRINTERATTR_ICON1	135
4.61.2.18	SY_PRINTERATTR_ICON2	135
4.61.2.19	SY_PRINTERATTR_ICON3	135
4.61.2.20	SY_PRINTERATTR_ICON4	135
4.61.2.21	SY_PRINTERATTR_ICON5	136
4.61.2.22	SY_PRINTERATTR_ICON6	136
4.61.2.23	SY_PRINTERATTR_ICON7	136
4.61.2.24	SY_PRINTERATTR_ICON8	136
4.62	Printer Status Job	137
4.62.1	Detailed Description	137
4.62.2	Macro Definition Documentation	137
4.62.2.1	SY_PRINTJOBSTATUS_PAUSED	137
4.62.2.2	SY_PRINTJOBSTATUS_ERROR	137
4.62.2.3	SY_PRINTJOBSTATUS_DELETING	137
4.62.2.4	SY_PRINTJOBSTATUS_SPOOLING	137
4.62.2.5	SY_PRINTJOBSTATUS_PRINTING	137
4.62.2.6	SY_PRINTJOBSTATUS_OFFLINE	137
4.62.2.7	SY_PRINTJOBSTATUS_PAPERPROBLEM	137

4.62.2.8	SY_PRINTJOBSTATUS_PRINTED	138
4.62.2.9	SY_PRINTJOBSTATUS_DELETED	138
4.62.2.10	SY_PRINTJOBSTATUS_BLOCKED	138
4.62.2.11	SY_PRINTJOBSTATUS_USERINTERVENTION	138
4.62.2.12	SY_PRINTJOBSTATUS_RESTART	138
4.63	Printer Priority Job	139
4.63.1	Detailed Description	139
4.63.2	Macro Definition Documentation	139
4.63.2.1	SY_PRINTJOB_MINPRIORITY	139
4.63.2.2	SY_PRINTJOB_MAXPRIORITY	139
4.63.2.3	SY_PRINTJOB_DEFPRIORITY	139
4.64	Printer Control Commands	140
4.64.1	Detailed Description	140
4.64.2	Macro Definition Documentation	140
4.64.2.1	SY_PRINTERCONTROL_PAUSE	140
4.64.2.2	SY_PRINTERCONTROL_RESUME	140
4.64.2.3	SY_PRINTERCONTROL_PURGE	140
4.64.2.4	SY_PRINTERCONTROL_SETSTATUS	140
4.65	Printer Job Control Commands	141
4.65.1	Detailed Description	141
4.65.2	Macro Definition Documentation	141
4.65.2.1	SY_PRINTJOB_COM_PAUSE	141
4.65.2.2	SY_PRINTJOB_COM_RESUME	141
4.65.2.3	SY_PRINTJOB_COM_CANCEL	141
4.65.2.4	SY_PRINTJOB_COM_RESTART	141
4.65.2.5	SY_PRINTJOB_COM_DELETE	141
4.65.2.6	SY_PRINTJOB_COM_SENDTOPRINTER	141
4.65.2.7	SY_PRINTJOB_COM_LASTPAGEEJECTED	141
4.66	Printer Form Flags	142
4.66.1	Detailed Description	142
4.66.2	Macro Definition Documentation	142
4.66.2.1	SY_PRINTFORMFLAG_USER	142
4.66.2.2	SY_PRINTFORMFLAG_BUILTIN	142
4.66.2.3	SY_PRINTFORMFLAG_PRINTER	142
4.67	User Defined	143
4.67.1	Detailed Description	143
4.68	SMB Errors	144
4.68.1	Detailed Description	145
4.68.2	Macro Definition Documentation	145
4.68.2.1	SMB_DOSERR	145
4.68.2.2	CLASS_ERR_MODULE	145
4.68.2.3	IS_DOS_ERROR	145
4.68.2.4	SMB_ERRsuccess	145
4.68.2.5	DOS_ERRbadfunc	145
4.68.2.6	DOS_ERRbadfile	145
4.68.2.7	DOS_ERRbadpath	146
4.68.2.8	DOS_ERRnofids	146
4.68.2.9	DOS_ERRnoaccess	146
4.68.2.10	DOS_ERRbadfid	146
4.68.2.11	DOS_ERRbadmcb	146
4.68.2.12	DOS_ERRnomem	146
4.68.2.13	DOS_ERRbadmem	146
4.68.2.14	DOS_ERRbadenv	146
4.68.2.15	DOS_ERRbadformat	146
4.68.2.16	DOS_ERRbadaccess	146
4.68.2.17	DOS_ERRbaddata	146
4.68.2.18	DOS_ERRbaddrive	146
4.68.2.19	DOS_ERRremcd	146

4.68.2.20	DOS_ERRdiffdevice	146
4.68.2.21	DOS_ERRnofiles	146
4.68.2.22	DOS_ERRbadshare	146
4.68.2.23	DOS_ERRlock	146
4.68.2.24	DOS_ERRdentsupportipc	146
4.68.2.25	DOS_ERRnoshare	146
4.68.2.26	DOS_ERRfileexists	146
4.68.2.27	DOS_ERRbaddirory	146
4.68.2.28	DOS_ERRinsufficientbuffer	146
4.68.2.29	DOS_ERRinvalidname	146
4.68.2.30	DOS_ERRdirnotempty	146
4.68.2.31	DOS_ERRalreadyexists	146
4.68.2.32	DOS_ERRbadpipe	146
4.68.2.33	DOS_ERRpipebusy	146
4.68.2.34	DOS_ERRpipeclosing	146
4.68.2.35	DOS_ERRnotconnected	147
4.68.2.36	DOS_ERRmoredata	147
4.68.2.37	IS_SRV_ERROR	147
4.68.2.38	SRV_ERRerror	147
4.68.2.39	SRV_ERRbadpw	147
4.68.2.40	SRV_ERRaccess	147
4.68.2.41	SRV_ERRinvtid	147
4.68.2.42	SRV_ERRinvnetname	147
4.68.2.43	SRV_ERRinvdevice	147
4.68.2.44	SRV_ERRqfull	147
4.68.2.45	SRV_ERRqtoobig	147
4.68.2.46	SRV_ERRqeof	147
4.68.2.47	SRV_ERRinvfid	147
4.68.2.48	SRV_ERRsmbcmd	147
4.68.2.49	SRV_ERRsrverror	147
4.68.2.50	SRV_ERRfilespecs	147
4.68.2.51	SRV_ERRbadpermits	147
4.68.2.52	SRV_ERRsetattrmode	147
4.68.2.53	SRV_ERRpaused	147
4.68.2.54	SRV_ERRmsgoff	147
4.68.2.55	SRV_ERRnoroomb	147
4.68.2.56	SRV_ERRrmuns	147
4.68.2.57	SRV_ERRtimeout	147
4.68.2.58	SRV_ERRnoresource	147
4.68.2.59	SRV_ERRtoomanyuids	147
4.68.2.60	SRV_ERRinvuid	147
4.68.2.61	SRV_ERRusempx	147
4.68.2.62	SRV_ERRusestd	147
4.68.2.63	SRV_ERRcontmpx	148
4.68.2.64	SRV_ERRnosupport	148
4.68.2.65	IS_HRD_ERROR	148
4.68.2.66	HRD_ERRnowrite	148
4.68.2.67	HRD_ERRbadunit	148
4.68.2.68	HRD_ERRnotready	148
4.68.2.69	HRD_ERRbadcmd	148
4.68.2.70	HRD_ERRdata	148
4.68.2.71	HRD_ERRbadreq	148
4.68.2.72	HRD_ERRseek	148
4.68.2.73	HRD_ERRbadmedia	148
4.68.2.74	HRD_ERRbadsector	148
4.68.2.75	HRD_ERRnopaper	148
4.68.2.76	HRD_ERRwrite	148
4.68.2.77	HRD_ERRread	148

4.68.2.78	HRD_ERRgeneral	148
4.68.2.79	HRD_ERRbadshare	148
4.68.2.80	HRD_ERRlock	148
4.68.2.81	HRD_ERRwrongdisk	148
4.68.2.82	HRD_ERRFCBUnavail	148
4.68.2.83	HRD_ERRsharebufexc	148
4.68.2.84	HRD_ERRdiskfull	148
4.69	SMB Error Codes	149
4.69.1	Detailed Description	158
4.69.2	Macro Definition Documentation	158
4.69.2.1	SMB_STATUS_OK	158
4.69.2.2	SMB_STATUS_INVALID	158
4.69.2.3	SMB_STATUS_SUCCESS	158
4.69.2.4	SMB_STATUS_ENUMDIR	158
4.69.2.5	SMB_STATUS_PENDING	158
4.69.2.6	SMB_STATUS_MORE_ENTRIES	159
4.69.2.7	SMB_STATUS_SOME_UNMAPPED	159
4.69.2.8	SMB_STATUS_NOTIFY_CLEANUP	159
4.69.2.9	SMB_STATUS_INSUFFICIENT_BUFFER	159
4.69.2.10	SMB_STATUS_UNKNOWN_PRINTER_DRIVER	159
4.69.2.11	SMB_STATUS_INVALID_DATATYPE	159
4.69.2.12	SMB_STATUS_BUFFER_OVERFLOW	159
4.69.2.13	SMB_STATUS_NO_MORE_FILES	159
4.69.2.14	SMB_STATUS_DEVICE_OFF_LINE	159
4.69.2.15	SMB_STATUS_NO_MORE_ENTRIES	159
4.69.2.16	SMB_STATUS_STOPPED_ON_SYMLINK	159
4.69.2.17	SMB_STATUS_UNSUCCESSFUL	159
4.69.2.18	SMB_STATUS_NOT_IMPLEMENTED	159
4.69.2.19	SMB_STATUS_INVALID_INFO_CLASS	159
4.69.2.20	SMB_STATUS_INFO_LENGTH_MISMATCH	159
4.69.2.21	SMB_STATUS_ACCESS_VIOLATION	159
4.69.2.22	SMB_STATUS_IN_PAGE_ERROR	159
4.69.2.23	SMB_STATUS_PAGEFILE_QUOTA	159
4.69.2.24	SMB_STATUS_INVALID_HANDLE	159
4.69.2.25	SMB_STATUS_BAD_INITIAL_STACK	159
4.69.2.26	SMB_STATUS_BAD_INITIAL_PC	159
4.69.2.27	SMB_STATUS_INVALID_CID	159
4.69.2.28	SMB_STATUS_TIMER_NOT_CANCELED	159
4.69.2.29	SMB_STATUS_INVALID_PARAMETER	159
4.69.2.30	SMB_STATUS_NO_SUCH_DEVICE	159
4.69.2.31	SMB_STATUS_NO_SUCH_FILE	159
4.69.2.32	SMB_STATUS_INVALID_DEVICE_REQUEST	159
4.69.2.33	SMB_STATUS_END_OF_FILE	159
4.69.2.34	SMB_STATUS_WRONG_VOLUME	160
4.69.2.35	SMB_STATUS_NO_MEDIA_IN_DEVICE	160
4.69.2.36	SMB_STATUS_UNRECOGNIZED_MEDIA	160
4.69.2.37	SMB_STATUS_NONEXISTENT_SECTOR	160
4.69.2.38	SMB_STATUS_MORE_PROCESSING_REQUIRED	160
4.69.2.39	SMB_STATUS_NO_MEMORY	160
4.69.2.40	SMB_STATUS_CONFLICTING_ADDRESSES	160
4.69.2.41	SMB_STATUS_NOT_MAPPED_VIEW	160
4.69.2.42	SMB_STATUS_UNABLE_TO_FREE_VM	160
4.69.2.43	SMB_STATUS_UNABLE_TO_DELETE_SECTION	160
4.69.2.44	SMB_STATUS_INVALID_SYSTEM_SERVICE	160
4.69.2.45	SMB_STATUS_ILLEGAL_INSTRUCTION	160
4.69.2.46	SMB_STATUS_INVALID_LOCK_SEQUENCE	160
4.69.2.47	SMB_STATUS_INVALID_VIEW_SIZE	160
4.69.2.48	SMB_STATUS_INVALID_FILE_FOR_SECTION	160

4.69.2.49	SMB_STATUS_ALREADY_COMMITTED	160
4.69.2.50	SMB_STATUS_ACCESS_DENIED	160
4.69.2.51	SMB_STATUS_BUFFER_TOO_SMALL	160
4.69.2.52	SMB_STATUS_OBJECT_TYPE_MISMATCH	160
4.69.2.53	SMB_STATUS_NONCONTINUABLE_EXCEPTION	160
4.69.2.54	SMB_STATUS_INVALID_DISPOSITION	160
4.69.2.55	SMB_STATUS_UNWIND	160
4.69.2.56	SMB_STATUS_BAD_STACK	160
4.69.2.57	SMB_STATUS_INVALID_UNWIND_TARGET	160
4.69.2.58	SMB_STATUS_NOT_LOCKED	160
4.69.2.59	SMB_STATUS_PARITY_ERROR	160
4.69.2.60	SMB_STATUS_UNABLE_TO_DECOMMIT_VM	160
4.69.2.61	SMB_STATUS_NOT_COMMITTED	160
4.69.2.62	SMB_STATUS_INVALID_PORT_ATTRIBUTES	161
4.69.2.63	SMB_STATUS_PORT_MESSAGE_TOO_LONG	161
4.69.2.64	SMB_STATUS_INVALID_PARAMETER_MIX	161
4.69.2.65	SMB_STATUS_INVALID_QUOTA_LOWER	161
4.69.2.66	SMB_STATUS_DISK_CORRUPT_ERROR	161
4.69.2.67	SMB_STATUS_OBJECT_NAME_INVALID	161
4.69.2.68	SMB_STATUS_OBJECT_NAME_NOT_FOUND	161
4.69.2.69	SMB_STATUS_OBJECT_NAME_COLLISION	161
4.69.2.70	SMB_STATUS_HANDLE_NOT_WAITABLE	161
4.69.2.71	SMB_STATUS_PORT_DISCONNECTED	161
4.69.2.72	SMB_STATUS_DEVICE_ALREADY_ATTACHED	161
4.69.2.73	SMB_STATUS_OBJECT_PATH_INVALID	161
4.69.2.74	SMB_STATUS_OBJECT_PATH_NOT_FOUND	161
4.69.2.75	SMB_STATUS_OBJECT_PATH_SYNTAX_BAD	161
4.69.2.76	SMB_STATUS_DATA_OVERRUN	161
4.69.2.77	SMB_STATUS_DATA_LATE_ERROR	161
4.69.2.78	SMB_STATUS_DATA_ERROR	161
4.69.2.79	SMB_STATUS_CRC_ERROR	161
4.69.2.80	SMB_STATUS_SECTION_TOO_BIG	161
4.69.2.81	SMB_STATUS_PORT_CONNECTION_REFUSED	161
4.69.2.82	SMB_STATUS_INVALID_PORT_HANDLE	161
4.69.2.83	SMB_STATUS_SHARING_VIOLATION	161
4.69.2.84	SMB_STATUS_QUOTA_EXCEEDED	161
4.69.2.85	SMB_STATUS_INVALID_PAGE_PROTECTION	161
4.69.2.86	SMB_STATUS_MUTANT_NOT_OWNED	161
4.69.2.87	SMB_STATUS_SEMAPHORE_LIMIT_EXCEEDED	161
4.69.2.88	SMB_STATUS_PORT_ALREADY_SET	161
4.69.2.89	SMB_STATUS_SECTION_NOT_IMAGE	161
4.69.2.90	SMB_STATUS_SUSPEND_COUNT_EXCEEDED	162
4.69.2.91	SMB_STATUS_THREAD_IS_TERMINATING	162
4.69.2.92	SMB_STATUS_BAD_WORKING_SET_LIMIT	162
4.69.2.93	SMB_STATUS_INCOMPATIBLE_FILE_MAP	162
4.69.2.94	SMB_STATUS_SECTION_PROTECTION	162
4.69.2.95	SMB_STATUS_EAS_NOT_SUPPORTED	162
4.69.2.96	SMB_STATUS_EA_TOO_LARGE	162
4.69.2.97	SMB_STATUS_NONEXISTENT_EA_ENTRY	162
4.69.2.98	SMB_STATUS_NO_EAS_ON_FILE	162
4.69.2.99	SMB_STATUS_EA_CORRUPT_ERROR	162
4.69.2.100	SMB_STATUS_FILE_LOCK_CONFLICT	162
4.69.2.101	SMB_STATUS_LOCK_NOT_GRANTED	162
4.69.2.102	SMB_STATUS_DELETE_PENDING	162
4.69.2.103	SMB_STATUS_CTL_FILE_NOT_SUPPORTED	162
4.69.2.104	SMB_STATUS_UNKNOWN_REVISION	162
4.69.2.105	SMB_STATUS_REVISION_MISMATCH	162
4.69.2.106	SMB_STATUS_INVALID_OWNER	162

4.69.2.107SMB_STATUS_INVALID_PRIMARY_GROUP	162
4.69.2.108SMB_STATUS_NO_IMPERSONATION_TOKEN	162
4.69.2.109SMB_STATUS_CANT_DISABLE_MANDATORY	162
4.69.2.110SMB_STATUS_NO_LOGON_SERVERS	162
4.69.2.111SMB_STATUS_NO_SUCH_LOGON_SESSION	162
4.69.2.112SMB_STATUS_NO_SUCH_PRIVILEGE	162
4.69.2.113SMB_STATUS_PRIVILEGE_NOT_HELD	162
4.69.2.114SMB_STATUS_INVALID_ACCOUNT_NAME	162
4.69.2.115SMB_STATUS_USER_EXISTS	162
4.69.2.116SMB_STATUS_NO_SUCH_USER	162
4.69.2.117SMB_STATUS_GROUP_EXISTS	162
4.69.2.118SMB_STATUS_NO_SUCH_GROUP	163
4.69.2.119SMB_STATUS_MEMBER_IN_GROUP	163
4.69.2.120SMB_STATUS_MEMBER_NOT_IN_GROUP	163
4.69.2.121SMB_STATUS_LAST_ADMIN	163
4.69.2.122SMB_STATUS_WRONG_PASSWORD	163
4.69.2.123SMB_STATUS_ILL_FORMED_PASSWORD	163
4.69.2.124SMB_STATUS_PASSWORD_RESTRICTION	163
4.69.2.125SMB_STATUS_LOGON_FAILURE	163
4.69.2.126SMB_STATUS_ACCOUNT_RESTRICTION	163
4.69.2.127SMB_STATUS_INVALID_LOGON_HOURS	163
4.69.2.128SMB_STATUS_INVALID_WORKSTATION	163
4.69.2.129SMB_STATUS_PASSWORD_EXPIRED	163
4.69.2.130SMB_STATUS_ACCOUNT_DISABLED	163
4.69.2.131SMB_STATUS_NONE_MAPPED	163
4.69.2.132SMB_STATUS_TOO_MANY_LUIDS_REQUESTED	163
4.69.2.133SMB_STATUS_LUIDS_EXHAUSTED	163
4.69.2.134SMB_STATUS_INVALID_SUB_AUTHORITY	163
4.69.2.135SMB_STATUS_INVALID_ACL	163
4.69.2.136SMB_STATUS_INVALID_SID	163
4.69.2.137SMB_STATUS_INVALID_SECURITY_DESCR	163
4.69.2.138SMB_STATUS_PROCEDURE_NOT_FOUND	163
4.69.2.139SMB_STATUS_INVALID_IMAGE_FORMAT	163
4.69.2.140SMB_STATUS_NO_TOKEN	163
4.69.2.141SMB_STATUS_BAD_INHERITANCE_ACL	163
4.69.2.142SMB_STATUS_RANGE_NOT_LOCKED	163
4.69.2.143SMB_STATUS_DISK_FULL	163
4.69.2.144SMB_STATUS_SERVER_DISABLED	163
4.69.2.145SMB_STATUS_SERVER_NOT_DISABLED	163
4.69.2.146SMB_STATUS_TOO_MANY_GUIDS_REQUESTED	164
4.69.2.147SMB_STATUS_GUIDS_EXHAUSTED	164
4.69.2.148SMB_STATUS_INVALID_ID_AUTHORITY	164
4.69.2.149SMB_STATUS_AGENTS_EXHAUSTED	164
4.69.2.150SMB_STATUS_INVALID_VOLUME_LABEL	164
4.69.2.151SMB_STATUS_SECTION_NOT_EXTENDED	164
4.69.2.152SMB_STATUS_NOT_MAPPED_DATA	164
4.69.2.153SMB_STATUS_RESOURCE_DATA_NOT_FOUND	164
4.69.2.154SMB_STATUS_RESOURCE_TYPE_NOT_FOUND	164
4.69.2.155SMB_STATUS_RESOURCE_NAME_NOT_FOUND	164
4.69.2.156SMB_STATUS_ARRAY_BOUNDS_EXCEEDED	164
4.69.2.157SMB_STATUS_FLOAT_DENORMAL_OPERAND	164
4.69.2.158SMB_STATUS_FLOAT_DIVIDE_BY_ZERO	164
4.69.2.159SMB_STATUS_FLOAT_INEXACT_RESULT	164
4.69.2.160SMB_STATUS_FLOAT_INVALID_OPERATION	164
4.69.2.161SMB_STATUS_FLOAT_OVERFLOW	164
4.69.2.162SMB_STATUS_FLOAT_STACK_CHECK	164
4.69.2.163SMB_STATUS_FLOAT_UNDERFLOW	164
4.69.2.164SMB_STATUS_INTEGER_DIVIDE_BY_ZERO	164

4.69.2.165SMB_STATUS_INTEGER_OVERFLOW	164
4.69.2.166SMB_STATUS_PRIVILEGED_INSTRUCTION	164
4.69.2.167SMB_STATUS_TOO_MANY_PAGING_FILES	164
4.69.2.168SMB_STATUS_FILE_INVALID	164
4.69.2.169SMB_STATUS_ALLOTTED_SPACE_EXCEEDED	164
4.69.2.170SMB_STATUS_INSUFFICIENT_RESOURCES	164
4.69.2.171SMB_STATUS_DFS_EXIT_PATH_FOUND	164
4.69.2.172SMB_STATUS_DEVICE_DATA_ERROR	164
4.69.2.173SMB_STATUS_DEVICE_NOT_CONNECTED	164
4.69.2.174SMB_STATUS_DEVICE_POWER_FAILURE	165
4.69.2.175SMB_STATUS_FREE_VM_NOT_AT_BASE	165
4.69.2.176SMB_STATUS_MEMORY_NOT_ALLOCATED	165
4.69.2.177SMB_STATUS_WORKING_SET_QUOTA	165
4.69.2.178SMB_STATUS_MEDIA_WRITE_PROTECTED	165
4.69.2.179SMB_STATUS_DEVICE_NOT_READY	165
4.69.2.180SMB_STATUS_INVALID_GROUP_ATTRIBUTES	165
4.69.2.181SMB_STATUS_BAD_IMPERSONATION_LEVEL	165
4.69.2.182SMB_STATUS_CANT_OPEN_ANONYMOUS	165
4.69.2.183SMB_STATUS_BAD_VALIDATION_CLASS	165
4.69.2.184SMB_STATUS_BAD_TOKEN_TYPE	165
4.69.2.185SMB_STATUS_BAD_MASTER_BOOT_RECORD	165
4.69.2.186SMB_STATUS_INSTRUCTION_MISALIGNMENT	165
4.69.2.187SMB_STATUS_INSTANCE_NOT_AVAILABLE	165
4.69.2.188SMB_STATUS_PIPE_NOT_AVAILABLE	165
4.69.2.189SMB_STATUS_INVALID_PIPE_STATE	165
4.69.2.190SMB_STATUS_PIPE_BUSY	165
4.69.2.191SMB_STATUS_ILLEGAL_FUNCTION	165
4.69.2.192SMB_STATUS_PIPE_DISCONNECTED	165
4.69.2.193SMB_STATUS_PIPE_CLOSING	165
4.69.2.194SMB_STATUS_PIPE_CONNECTED	165
4.69.2.195SMB_STATUS_PIPE_LISTENING	165
4.69.2.196SMB_STATUS_INVALID_READ_MODE	165
4.69.2.197SMB_STATUS_IO_TIMEOUT	165
4.69.2.198SMB_STATUS_FILE_FORCED_CLOSED	165
4.69.2.199SMB_STATUS_PROFILING_NOT_STARTED	165
4.69.2.200SMB_STATUS_PROFILING_NOT_STOPPED	165
4.69.2.201SMB_STATUS_COULD_NOT_INTERPRET	165
4.69.2.202SMB_STATUS_FILE_IS_A_DIRECTORY	166
4.69.2.203SMB_STATUS_NOT_SUPPORTED	166
4.69.2.204SMB_STATUS_REMOTE_NOT_LISTENING	166
4.69.2.205SMB_STATUS_DUPLICATE_NAME	166
4.69.2.206SMB_STATUS_BAD_NETWORK_PATH	166
4.69.2.207SMB_STATUS_NETWORK_BUSY	166
4.69.2.208SMB_STATUS_DEVICE_DOES_NOT_EXIST	166
4.69.2.209SMB_STATUS_TOO_MANY_COMMANDS	166
4.69.2.210SMB_STATUS_ADAPTER_HARDWARE_ERROR	166
4.69.2.211SMB_STATUS_INVALID_NETWORK_RESPONSE	166
4.69.2.212SMB_STATUS_UNEXPECTED_NETWORK_ERROR	166
4.69.2.213SMB_STATUS_BAD_REMOTE_ADAPTER	166
4.69.2.214SMB_STATUS_PRINT_QUEUE_FULL	166
4.69.2.215SMB_STATUS_NO_SPOOL_SPACE	166
4.69.2.216SMB_STATUS_PRINT_CANCELLED	166
4.69.2.217SMB_STATUS_NETWORK_NAME_DELETED	166
4.69.2.218SMB_STATUS_NETWORK_ACCESS_DENIED	166
4.69.2.219SMB_STATUS_BAD_DEVICE_TYPE	166
4.69.2.220SMB_STATUS_BAD_NETWORK_NAME	166
4.69.2.221SMB_STATUS_TOO_MANY_NAMES	166
4.69.2.222SMB_STATUS_TOO_MANY_SESSIONS	166

4.69.2.223SMB_STATUS_SHARING_PAUSED	166
4.69.2.224SMB_STATUS_REQUEST_NOT_ACCEPTED	166
4.69.2.225SMB_STATUS_REDIRECTOR_PAUSED	166
4.69.2.226SMB_STATUS_NET_WRITE_FAULT	166
4.69.2.227SMB_STATUS_PROFILING_AT_LIMIT	166
4.69.2.228SMB_STATUS_NOT_SAME_DEVICE	166
4.69.2.229SMB_STATUS_FILE_RENAMED	166
4.69.2.230SMB_STATUS_VIRTUAL_CIRCUIT_CLOSED	167
4.69.2.231SMB_STATUS_NO_SECURITY_ON_OBJECT	167
4.69.2.232SMB_STATUS_CANT_WAIT	167
4.69.2.233SMB_STATUS_PIPE_EMPTY	167
4.69.2.234SMB_STATUS_CANT_ACCESS_DOMAIN_INFO	167
4.69.2.235SMB_STATUS_CANT_TERMINATE_SELF	167
4.69.2.236SMB_STATUS_INVALID_SERVER_STATE	167
4.69.2.237SMB_STATUS_INVALID_DOMAIN_STATE	167
4.69.2.238SMB_STATUS_INVALID_DOMAIN_ROLE	167
4.69.2.239SMB_STATUS_NO_SUCH_DOMAIN	167
4.69.2.240SMB_STATUS_DOMAIN_EXISTS	167
4.69.2.241SMB_STATUS_DOMAIN_LIMIT_EXCEEDED	167
4.69.2.242SMB_STATUS_OPLOCK_NOT_GRANTED	167
4.69.2.243SMB_STATUS_INVALID_OPLOCK_PROTOCOL	167
4.69.2.244SMB_STATUS_INTERNAL_DB_CORRUPTION	167
4.69.2.245SMB_STATUS_INTERNAL_ERROR	167
4.69.2.246SMB_STATUS_GENERIC_NOT_MAPPED	167
4.69.2.247SMB_STATUS_BAD_DESCRIPTOR_FORMAT	167
4.69.2.248SMB_STATUS_INVALID_USER_BUFFER	167
4.69.2.249SMB_STATUS_UNEXPECTED_IO_ERROR	167
4.69.2.250SMB_STATUS_UNEXPECTED_MM_CREATE_ERR	167
4.69.2.251SMB_STATUS_UNEXPECTED_MM_MAP_ERROR	167
4.69.2.252SMB_STATUS_UNEXPECTED_MM_EXTEND_ERR	167
4.69.2.253SMB_STATUS_NOT_LOGON_PROCESS	167
4.69.2.254SMB_STATUS_LOGON_SESSION_EXISTS	167
4.69.2.255SMB_STATUS_INVALID_PARAMETER_1	167
4.69.2.256SMB_STATUS_INVALID_PARAMETER_2	167
4.69.2.257SMB_STATUS_INVALID_PARAMETER_3	167
4.69.2.258SMB_STATUS_INVALID_PARAMETER_4	168
4.69.2.259SMB_STATUS_INVALID_PARAMETER_5	168
4.69.2.260SMB_STATUS_INVALID_PARAMETER_6	168
4.69.2.261SMB_STATUS_INVALID_PARAMETER_7	168
4.69.2.262SMB_STATUS_INVALID_PARAMETER_8	168
4.69.2.263SMB_STATUS_INVALID_PARAMETER_9	168
4.69.2.264SMB_STATUS_INVALID_PARAMETER_10	168
4.69.2.265SMB_STATUS_INVALID_PARAMETER_11	168
4.69.2.266SMB_STATUS_INVALID_PARAMETER_12	168
4.69.2.267SMB_STATUS_REDIRECTOR_NOT_STARTED	168
4.69.2.268SMB_STATUS_REDIRECTOR_STARTED	168
4.69.2.269SMB_STATUS_STACK_OVERFLOW	168
4.69.2.270SMB_STATUS_NO_SUCH_PACKAGE	168
4.69.2.271SMB_STATUS_BAD_FUNCTION_TABLE	168
4.69.2.272SMB_STATUS_DIRECTORY_NOT_EMPTY	168
4.69.2.273SMB_STATUS_FILE_CORRUPT_ERROR	168
4.69.2.274SMB_STATUS_NOT_A_DIRECTORY	168
4.69.2.275SMB_STATUS_BAD_LOGON_SESSION_STATE	168
4.69.2.276SMB_STATUS_LOGON_SESSION_COLLISION	168
4.69.2.277SMB_STATUS_NAME_TOO_LONG	168
4.69.2.278SMB_STATUS_FILES_OPEN	168
4.69.2.279SMB_STATUS_CONNECTION_IN_USE	168
4.69.2.280SMB_STATUS_MESSAGE_NOT_FOUND	168

4.69.2.281SMB_STATUS_PROCESS_IS_TERMINATING	168
4.69.2.282SMB_STATUS_INVALID_LOGON_TYPE	168
4.69.2.283SMB_STATUS_NO_GUID_TRANSLATION	168
4.69.2.284SMB_STATUS_CANNOT_IMPERSONATE	168
4.69.2.285SMB_STATUS_IMAGE_ALREADY_LOADED	168
4.69.2.286SMB_STATUS_ABIOS_NOT_PRESENT	169
4.69.2.287SMB_STATUS_ABIOS_LID_NOT_EXIST	169
4.69.2.288SMB_STATUS_ABIOS_LID_ALREADY_OWNED	169
4.69.2.289SMB_STATUS_ABIOS_NOT_LID_OWNER	169
4.69.2.290SMB_STATUS_ABIOS_INVALID_COMMAND	169
4.69.2.291SMB_STATUS_ABIOS_INVALID_LID	169
4.69.2.292SMB_STATUS_ABIOS_SELECTOR_NOT_AVAILABLE	169
4.69.2.293SMB_STATUS_ABIOS_INVALID_SELECTOR	169
4.69.2.294SMB_STATUS_NO_LDT	169
4.69.2.295SMB_STATUS_INVALID_LDT_SIZE	169
4.69.2.296SMB_STATUS_INVALID_LDT_OFFSET	169
4.69.2.297SMB_STATUS_INVALID_LDT_DESCRIPTOR	169
4.69.2.298SMB_STATUS_INVALID_IMAGE_NE_FORMAT	169
4.69.2.299SMB_STATUS_RXACT_INVALID_STATE	169
4.69.2.300SMB_STATUS_RXACT_COMMIT_FAILURE	169
4.69.2.301SMB_STATUS_MAPPED_FILE_SIZE_ZERO	169
4.69.2.302SMB_STATUS_TOO_MANY_OPENED_FILES	169
4.69.2.303SMB_STATUS_CANCELLED	169
4.69.2.304SMB_STATUS_CANNOT_DELETE	169
4.69.2.305SMB_STATUS_INVALID_COMPUTER_NAME	169
4.69.2.306SMB_STATUS_FILE_DELETED	169
4.69.2.307SMB_STATUS_SPECIAL_ACCOUNT	169
4.69.2.308SMB_STATUS_SPECIAL_GROUP	169
4.69.2.309SMB_STATUS_SPECIAL_USER	169
4.69.2.310SMB_STATUS_MEMBERS_PRIMARY_GROUP	169
4.69.2.311SMB_STATUS_FILE_CLOSED	169
4.69.2.312SMB_STATUS_TOO_MANY_THREADS	169
4.69.2.313SMB_STATUS_THREAD_NOT_IN_PROCESS	169
4.69.2.314SMB_STATUS_TOKEN_ALREADY_IN_USE	170
4.69.2.315SMB_STATUS_PAGEFILE_QUOTA_EXCEEDED	170
4.69.2.316SMB_STATUS_COMMITMENT_LIMIT	170
4.69.2.317SMB_STATUS_INVALID_IMAGE_LE_FORMAT	170
4.69.2.318SMB_STATUS_INVALID_IMAGE_NOT_MZ	170
4.69.2.319SMB_STATUS_INVALID_IMAGE_PROTECT	170
4.69.2.320SMB_STATUS_INVALID_IMAGE_WIN_16	170
4.69.2.321SMB_STATUS_LOGON_SERVER_CONFLICT	170
4.69.2.322SMB_STATUS_TIME_DIFFERENCE_AT_DC	170
4.69.2.323SMB_STATUS_SYNCHRONIZATION_REQUIRED	170
4.69.2.324SMB_STATUS_DLL_NOT_FOUND	170
4.69.2.325SMB_STATUS_OPEN_FAILED	170
4.69.2.326SMB_STATUS_IO_PRIVILEGE_FAILED	170
4.69.2.327SMB_STATUS_ORDINAL_NOT_FOUND	170
4.69.2.328SMB_STATUS_ENTRYPOINT_NOT_FOUND	170
4.69.2.329SMB_STATUS_CONTROL_C_EXIT	170
4.69.2.330SMB_STATUS_LOCAL_DISCONNECT	170
4.69.2.331SMB_STATUS_REMOTE_DISCONNECT	170
4.69.2.332SMB_STATUS_REMOTE_RESOURCES	170
4.69.2.333SMB_STATUS_LINK_FAILED	170
4.69.2.334SMB_STATUS_LINK_TIMEOUT	170
4.69.2.335SMB_STATUS_INVALID_CONNECTION	170
4.69.2.336SMB_STATUS_INVALID_ADDRESS	170
4.69.2.337SMB_STATUS_DLL_INIT_FAILED	170
4.69.2.338SMB_STATUS_MISSING_SYSTEMFILE	170

4.69.2.339SMB_STATUS_UNHANDLED_EXCEPTION	170
4.69.2.340SMB_STATUS_APP_INIT_FAILURE	170
4.69.2.341SMB_STATUS_PAGEFILE_CREATE_FAILED	170
4.69.2.342SMB_STATUS_NO_PAGEFILE	171
4.69.2.343SMB_STATUS_INVALID_LEVEL	171
4.69.2.344SMB_STATUS_WRONG_PASSWORD_CORE	171
4.69.2.345SMB_STATUS_ILLEGAL_FLOAT_CONTEXT	171
4.69.2.346SMB_STATUS_PIPE_BROKEN	171
4.69.2.347SMB_STATUS_REGISTRY_CORRUPT	171
4.69.2.348SMB_STATUS_REGISTRY_IO_FAILED	171
4.69.2.349SMB_STATUS_NO_EVENT_PAIR	171
4.69.2.350SMB_STATUS_UNRECOGNIZED_VOLUME	171
4.69.2.351SMB_STATUS_SERIAL_NO_DEVICE_INITED	171
4.69.2.352SMB_STATUS_NO_SUCH_ALIAS	171
4.69.2.353SMB_STATUS_MEMBER_NOT_IN_ALIAS	171
4.69.2.354SMB_STATUS_MEMBER_IN_ALIAS	171
4.69.2.355SMB_STATUS_ALIAS_EXISTS	171
4.69.2.356SMB_STATUS_LOGON_NOT_GRANTED	171
4.69.2.357SMB_STATUS_TOO_MANY_SECRETS	171
4.69.2.358SMB_STATUS_SECRET_TOO_LONG	171
4.69.2.359SMB_STATUS_INTERNAL_DB_ERROR	171
4.69.2.360SMB_STATUS_FULLSCREEN_MODE	171
4.69.2.361SMB_STATUS_TOO_MANY_CONTEXT_IDS	171
4.69.2.362SMB_STATUS_LOGON_TYPE_NOT_GRANTED	171
4.69.2.363SMB_STATUS_NOT_REGISTRY_FILE	171
4.69.2.364SMB_STATUS_NT_CROSS_ENCRYPTION_REQUIRED	171
4.69.2.365SMB_STATUS_DOMAIN_CTRLR_CONFIG_ERROR	171
4.69.2.366SMB_STATUS_FT_MISSING_MEMBER	171
4.69.2.367SMB_STATUS_ILL_FORMED_SERVICE_ENTRY	171
4.69.2.368SMB_STATUS_ILLEGAL_CHARACTER	171
4.69.2.369SMB_STATUS_UNMAPPABLE_CHARACTER	171
4.69.2.370SMB_STATUS_UNDEFINED_CHARACTER	172
4.69.2.371SMB_STATUS_FLOPPY_VOLUME	172
4.69.2.372SMB_STATUS_FLOPPY_ID_MARK_NOT_FOUND	172
4.69.2.373SMB_STATUS_FLOPPY_WRONG_CYLINDER	172
4.69.2.374SMB_STATUS_FLOPPY_UNKNOWN_ERROR	172
4.69.2.375SMB_STATUS_FLOPPY_BAD_REGISTERS	172
4.69.2.376SMB_STATUS_DISK_RECALIBRATE_FAILED	172
4.69.2.377SMB_STATUS_DISK_OPERATION_FAILED	172
4.69.2.378SMB_STATUS_DISK_RESET_FAILED	172
4.69.2.379SMB_STATUS_SHARED_IRQ_BUSY	172
4.69.2.380SMB_STATUS_FT_ORPHANING	172
4.69.2.381SMB_STATUS_PARTITION_FAILURE	172
4.69.2.382SMB_STATUS_INVALID_BLOCK_LENGTH	172
4.69.2.383SMB_STATUS_DEVICE_NOT_PARTITIONED	172
4.69.2.384SMB_STATUS_UNABLE_TO_LOCK_MEDIA	172
4.69.2.385SMB_STATUS_UNABLE_TO_UNLOAD_MEDIA	172
4.69.2.386SMB_STATUS_EOM_OVERFLOW	172
4.69.2.387SMB_STATUS_NO_MEDIA	172
4.69.2.388SMB_STATUS_NO_SUCH_MEMBER	172
4.69.2.389SMB_STATUS_INVALID_MEMBER	172
4.69.2.390SMB_STATUS_KEY_DELETED	172
4.69.2.391SMB_STATUS_NO_LOG_SPACE	172
4.69.2.392SMB_STATUS_TOO_MANY_SIDS	172
4.69.2.393SMB_STATUS_LM_CROSS_ENCRYPTION_REQUIRED	172
4.69.2.394SMB_STATUS_KEY_HAS_CHILDREN	172
4.69.2.395SMB_STATUS_CHILD_MUST_BE_VOLATILE	172
4.69.2.396SMB_STATUS_DEVICE_CONFIGURATION_ERROR	172

4.69.2.397SMB_STATUS_DRIVER_INTERNAL_ERROR	172
4.69.2.398SMB_STATUS_INVALID_DEVICE_STATE	173
4.69.2.399SMB_STATUS_IO_DEVICE_ERROR	173
4.69.2.400SMB_STATUS_DEVICE_PROTOCOL_ERROR	173
4.69.2.401SMB_STATUS_BACKUP_CONTROLLER	173
4.69.2.402SMB_STATUS_LOG_FILE_FULL	173
4.69.2.403SMB_STATUS_TOO_LATE	173
4.69.2.404SMB_STATUS_NO_TRUST_LSA_SECRET	173
4.69.2.405SMB_STATUS_NO_TRUST_SAM_ACCOUNT	173
4.69.2.406SMB_STATUS_TRUSTED_DOMAIN_FAILURE	173
4.69.2.407SMB_STATUS_TRUSTED_RELATIONSHIP_FAILURE	173
4.69.2.408SMB_STATUS_EVENTLOG_FILE_CORRUPT	173
4.69.2.409SMB_STATUS_EVENTLOG_CANT_START	173
4.69.2.410SMB_STATUS_TRUST_FAILURE	173
4.69.2.411SMB_STATUS_MUTANT_LIMIT_EXCEEDED	173
4.69.2.412SMB_STATUS_NETLOGON_NOT_STARTED	173
4.69.2.413SMB_STATUS_ACCOUNT_EXPIRED	173
4.69.2.414SMB_STATUS_POSSIBLE_DEADLOCK	173
4.69.2.415SMB_STATUS_NETWORK_CREDENTIAL_CONFLICT	173
4.69.2.416SMB_STATUS_REMOTE_SESSION_LIMIT	173
4.69.2.417SMB_STATUS_EVENTLOG_FILE_CHANGED	173
4.69.2.418SMB_STATUS_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT	173
4.69.2.419SMB_STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT	173
4.69.2.420SMB_STATUS_NOLOGON_SERVER_TRUST_ACCOUNT	173
4.69.2.421SMB_STATUS_DOMAIN_TRUST_INCONSISTENT	173
4.69.2.422SMB_STATUS_FS_DRIVER_REQUIRED	173
4.69.2.423SMB_STATUS_INVALID_LOCK_RANGE	173
4.69.2.424SMB_STATUS_NO_USER_SESSION_KEY	173
4.69.2.425SMB_STATUS_USER_SESSION_DELETED	173
4.69.2.426SMB_STATUS_RESOURCE_LANG_NOT_FOUND	174
4.69.2.427SMB_STATUS_INSUFF_SERVER_RESOURCES	174
4.69.2.428SMB_STATUS_INVALID_BUFFER_SIZE	174
4.69.2.429SMB_STATUS_INVALID_ADDRESS_COMPONENT	174
4.69.2.430SMB_STATUS_INVALID_ADDRESS_WILDCARD	174
4.69.2.431SMB_STATUS_TOO_MANY_ADDRESSES	174
4.69.2.432SMB_STATUS_ADDRESS_ALREADY_EXISTS	174
4.69.2.433SMB_STATUS_ADDRESS_CLOSED	174
4.69.2.434SMB_STATUS_CONNECTION_DISCONNECTED	174
4.69.2.435SMB_STATUS_CONNECTION_RESET	174
4.69.2.436SMB_STATUS_TOO_MANY_NODES	174
4.69.2.437SMB_STATUS_TRANSACTION_ABORTED	174
4.69.2.438SMB_STATUS_TRANSACTION_TIMED_OUT	174
4.69.2.439SMB_STATUS_TRANSACTION_NO_RELEASE	174
4.69.2.440SMB_STATUS_TRANSACTION_NO_MATCH	174
4.69.2.441SMB_STATUS_TRANSACTION_RESPONDED	174
4.69.2.442SMB_STATUS_TRANSACTION_INVALID_ID	174
4.69.2.443SMB_STATUS_TRANSACTION_INVALID_TYPE	174
4.69.2.444SMB_STATUS_NOT_SERVER_SESSION	174
4.69.2.445SMB_STATUS_NOT_CLIENT_SESSION	174
4.69.2.446SMB_STATUS_CANNOT_LOAD_REGISTRY_FILE	174
4.69.2.447SMB_STATUS_DEBUG_ATTACH_FAILED	174
4.69.2.448SMB_STATUS_SYSTEM_PROCESS_TERMINATED	174
4.69.2.449SMB_STATUS_DATA_NOT_ACCEPTED	174
4.69.2.450SMB_STATUS_NO_BROWSER_SERVERS_FOUND	174
4.69.2.451SMB_STATUS_VDM_HARD_ERROR	174
4.69.2.452SMB_STATUS_DRIVER_CANCEL_TIMEOUT	174
4.69.2.453SMB_STATUS_REPLY_MESSAGE_MISMATCH	174
4.69.2.454SMB_STATUS_MAPPED_ALIGNMENT	175

4.69.2.455SMB_STATUS_IMAGE_CHECKSUM_MISMATCH	175
4.69.2.456SMB_STATUS_LOST_WRITEBEHIND_DATA	175
4.69.2.457SMB_STATUS_CLIENT_SERVER_PARAMETERS_INVALID	175
4.69.2.458SMB_STATUS_PASSWORD_MUST_CHANGE	175
4.69.2.459SMB_STATUS_NOT_FOUND	175
4.69.2.460SMB_STATUS_NOT_TINY_STREAM	175
4.69.2.461SMB_STATUS_RECOVERY_FAILURE	175
4.69.2.462SMB_STATUS_STACK_OVERFLOW_READ	175
4.69.2.463SMB_STATUS_FAIL_CHECK	175
4.69.2.464SMB_STATUS_DUPLICATE_OBJECTID	175
4.69.2.465SMB_STATUS_OBJECTID_EXISTS	175
4.69.2.466SMB_STATUS_CONVERT_TO_LARGE	175
4.69.2.467SMB_STATUS_RETRY	175
4.69.2.468SMB_STATUS_FOUND_OUT_OF_SCOPE	175
4.69.2.469SMB_STATUS_ALLOCATE_BUCKET	175
4.69.2.470SMB_STATUS_PROPSET_NOT_FOUND	175
4.69.2.471SMB_STATUS_MARSHALL_OVERFLOW	175
4.69.2.472SMB_STATUS_INVALID_VARIANT	175
4.69.2.473SMB_STATUS_DOMAIN_CONTROLLER_NOT_FOUND	175
4.69.2.474SMB_STATUS_ACCOUNT_LOCKED_OUT	175
4.69.2.475SMB_STATUS_HANDLE_NOT_CLOSABLE	175
4.69.2.476SMB_STATUS_CONNECTION_REFUSED	175
4.69.2.477SMB_STATUS_GRACEFUL_DISCONNECT	175
4.69.2.478SMB_STATUS_ADDRESS_ALREADY_ASSOCIATED	175
4.69.2.479SMB_STATUS_ADDRESS_NOT_ASSOCIATED	175
4.69.2.480SMB_STATUS_CONNECTION_INVALID	175
4.69.2.481SMB_STATUS_CONNECTION_ACTIVE	175
4.69.2.482SMB_STATUS_NETWORK_UNREACHABLE	176
4.69.2.483SMB_STATUS_HOST_UNREACHABLE	176
4.69.2.484SMB_STATUS_PROTOCOL_UNREACHABLE	176
4.69.2.485SMB_STATUS_PORT_UNREACHABLE	176
4.69.2.486SMB_STATUS_REQUEST_ABORTED	176
4.69.2.487SMB_STATUS_CONNECTION_ABORTED	176
4.69.2.488SMB_STATUS_BAD_COMPRESSION_BUFFER	176
4.69.2.489SMB_STATUS_USER_MAPPED_FILE	176
4.69.2.490SMB_STATUS_AUDIT_FAILED	176
4.69.2.491SMB_STATUS_TIMER_RESOLUTION_NOT_SET	176
4.69.2.492SMB_STATUS_CONNECTION_COUNT_LIMIT	176
4.69.2.493SMB_STATUS_LOGIN_TIME_RESTRICTION	176
4.69.2.494SMB_STATUS_LOGIN_WKSTA_RESTRICTION	176
4.69.2.495SMB_STATUS_IMAGE_MP_UP_MISMATCH	176
4.69.2.496SMB_STATUS_INSUFFICIENT_LOGON_INFO	176
4.69.2.497SMB_STATUS_BAD_DLL_ENTRYPOINT	176
4.69.2.498SMB_STATUS_BAD_SERVICE_ENTRYPOINT	176
4.69.2.499SMB_STATUS_LPC_REPLY_LOST	176
4.69.2.500SMB_STATUS_IP_ADDRESS_CONFLICT1	176
4.69.2.501SMB_STATUS_IP_ADDRESS_CONFLICT2	176
4.69.2.502SMB_STATUS_REGISTRY_QUOTA_LIMIT	176
4.69.2.503SMB_STATUS_PATH_NOT_COVERED	176
4.69.2.504SMB_STATUS_NO_CALLBACK_ACTIVE	176
4.69.2.505SMB_STATUS_LICENSE_QUOTA_EXCEEDED	176
4.69.2.506SMB_STATUS_PWD_TOO_SHORT	176
4.69.2.507SMB_STATUS_PWD_TOO_RECENT	176
4.69.2.508SMB_STATUS_PWD_HISTORY_CONFLICT	176
4.69.2.509SMB_STATUS_PLUGPLAY_NO_DEVICE	176
4.69.2.510SMB_STATUS_UNSUPPORTED_COMPRESSION	177
4.69.2.511SMB_STATUS_INVALID_HW_PROFILE	177
4.69.2.512SMB_STATUS_INVALID_PLUGPLAY_DEVICE_PATH	177

4.69.2.513	SMB_STATUS_DRIVER_ORDINAL_NOT_FOUND	177
4.69.2.514	SMB_STATUS_DRIVER_ENTRYPOINT_NOT_FOUND	177
4.69.2.515	SMB_STATUS_RESOURCE_NOT_OWNED	177
4.69.2.516	SMB_STATUS_TOO_MANY_LINKS	177
4.69.2.517	SMB_STATUS_QUOTA_LIST_INCONSISTENT	177
4.69.2.518	SMB_STATUS_FILE_IS_OFFLINE	177
4.69.2.519	SMB_STATUS_DFS_UNAVAILABLE	177
4.69.2.520	SMB_STATUS_VOLUME_DISMOUNTED	177
4.69.2.521	SMB_STATUS_NOT_A_REPARSE_POINT	177
4.69.2.522	SMB_STATUS_IO_REPARSE_TAG_NOT_HANDLED	177
4.69.2.523	SMB_STATUS_NETWORK_SESSION_EXPIRED	177
4.69.2.524	SMB_STATUS_INVALID_TOKEN	177
4.69.2.525	SMB_STATUS_SERVER_UNAVAILABLE	177
4.69.2.526	SMB_STATUS_FILE_NOT_AVAILABLE	177
4.69.2.527	SMB_STATUS_SHARE_UNAVAILABLE	177
4.69.2.528	SMB_STATUS_NO_SUCH_JOB	177
4.69.2.529	SMB_STATUS_NO_PREAUTH_INTEGRITY_HASH_OVERLAP	177
4.70	Internal NT Statuses	178
4.70.1	Detailed Description	178
4.70.2	Macro Definition Documentation	178
4.70.2.1	SMB_STATUS_DISCONNECT	178
4.70.2.2	SMB_STATUS_INTERNAL_RELEASE_REQUEST_RESPONSE	178
4.70.2.3	SMB_STATUS_INTERNAL_BUFFER_TOO_SMALL	178
4.70.2.4	SMB_STATUS_DONOTRELEASERESPONSE	178
4.71	NQ Error Codes	179
4.71.1	Detailed Description	181
4.71.2	Macro Definition Documentation	181
4.71.2.1	NQ_ERR_MODULE	181
4.71.2.2	IS_NQ_ERROR	181
4.71.2.3	NQ_ERR_OK	181
4.71.2.4	NQ_ERR_BADPARAM	181
4.71.2.5	NQ_ERR_GETDATA	181
4.71.2.6	NQ_ERR_INVALIDMODE	181
4.71.2.7	NQ_ERR_NOSERVERMAC	181
4.71.2.8	NQ_ERR_SIGNATUREFAIL	181
4.71.2.9	NQ_ERR_OBJEXISTS	181
4.71.2.10	NQ_ERR_MOUNTERROR	181
4.71.2.11	NQ_ERR_UNABLETODISPOSE	182
4.71.2.12	NQ_ERR_INVALIDHANDLE	182
4.71.2.13	NQ_ERR_NEGOTIATEFAILED	182
4.71.2.14	NQ_ERR_PATHNOTCOVERED	182
4.71.2.15	NQ_ERR_DFSCACHEOVERFLOW	182
4.71.2.16	NQ_ERR_ACCOUNTLOCKEDOUT	182
4.71.2.17	NQ_ERR_USEREXISTS	182
4.71.2.18	NQ_ERR_USERNOTFOUND	182
4.71.2.19	NQ_ERR_NOTFOUND	182
4.71.2.20	NQ_ERR_LOGONFAILURE	182
4.71.2.21	NQ_ERR_VOLUMEDISMOUNTED	182
4.71.2.22	NQ_ERR_NETWORKERROR	182
4.71.2.23	NQ_ERR_BADCONNECTION	183
4.71.2.24	NQ_ERR_SHARINGPAUSED	183
4.71.2.25	NQ_ERR_BADDFS	183
4.71.2.26	NQ_ERR_IOTIMEOUT	183
4.71.2.27	NQ_ERR_TRYAGAIN	183
4.71.2.28	NQ_ERR_ACCOUNTEXPIRED	183
4.71.2.29	NQ_ERR_INVALIDLOGONHOURS	183
4.71.2.30	NQ_ERR_ACCOUNTDISABLED	183
4.71.2.31	NQ_ERR_SESSIONREAUTHREQUIRED	183

4.71.2.32 NQ_ERR_USERRELOGONFAILED	183
4.71.2.33 NQ_ERR_SERVERUNAVAILABLE	183
4.71.2.34 NQ_ERR_SHAREUNAVAILABLE	183
4.71.2.35 NQ_ERR_OUTOFMEMORY	184
4.71.2.36 NQ_ERR_BADFUNC	184
4.71.2.37 NQ_ERR_BADFILE	184
4.71.2.38 NQ_ERR_BADPATH	184
4.71.2.39 NQ_ERR_NOFIDS	184
4.71.2.40 NQ_ERR_NOACCESS	184
4.71.2.41 NQ_ERR_BADFID	184
4.71.2.42 NQ_ERR_BADMCB	184
4.71.2.43 NQ_ERR_NOMEM	184
4.71.2.44 NQ_ERR_BADMEM	184
4.71.2.45 NQ_ERR_BADENV	184
4.71.2.46 NQ_ERR_BADFORMAT	184
4.71.2.47 NQ_ERR_BADACCESS	185
4.71.2.48 NQ_ERR_BADDATA	185
4.71.2.49 NQ_ERR_BADDRIVE	185
4.71.2.50 NQ_ERR_REMCD	185
4.71.2.51 NQ_ERR_DIFFDEVICE	185
4.71.2.52 NQ_ERR_NOFILES	185
4.71.2.53 NQ_ERR_WRONGDIALECT	185
4.71.2.54 NQ_ERR_BADSHARE	185
4.71.2.55 NQ_ERR_LOCK	185
4.71.2.56 NQ_ERR_DONTSUPPORTIPC	185
4.71.2.57 NQ_ERR_SHARINGVIOLATION	185
4.71.2.58 NQ_ERR_FILEEXISTS	185
4.71.2.59 NQ_ERR_BADDIRECTORY	186
4.71.2.60 NQ_ERR_INSUFFICIENTBUFFER	186
4.71.2.61 NQ_ERR_INVALIDNAME	186
4.71.2.62 NQ_ERR_DIRNOTEMPTY	186
4.71.2.63 NQ_ERR_ALREADYEXISTS	186
4.71.2.64 NQ_ERR_BADPIPE	186
4.71.2.65 NQ_ERR_PIPEBUSY	186
4.71.2.66 NQ_ERR_PIPECLOSING	186
4.71.2.67 NQ_ERR_NOTCONNECTED	186
4.71.2.68 NQ_ERR_MOREDATA	186
4.71.2.69 NQ_ERR_INVALIDNAMELEN	186
4.71.2.70 NQ_ERR_ERROR	186
4.71.2.71 NQ_ERR_BADPW	187
4.71.2.72 NQ_ERR_ACCESS	187
4.71.2.73 NQ_ERR_INVTID	187
4.71.2.74 NQ_ERR_INVNETNAME	187
4.71.2.75 NQ_ERR_INVDEVICE	187
4.71.2.76 NQ_ERR_QFULL	187
4.71.2.77 NQ_ERR_QTOOBIG	187
4.71.2.78 NQ_ERR_QEOF	187
4.71.2.79 NQ_ERR_INVFID	187
4.71.2.80 NQ_ERR_SMBCMD	187
4.71.2.81 NQ_ERR_SRVERORR	187
4.71.2.82 NQ_ERR_FILESPCS	187
4.71.2.83 NQ_ERR_BADPERMITS	188
4.71.2.84 NQ_ERR_SETATTRMODE	188
4.71.2.85 NQ_ERR_PAUSED	188
4.71.2.86 NQ_ERR_MSGOFF	188
4.71.2.87 NQ_ERR_NOROOM	188
4.71.2.88 NQ_ERR_RMUNS	188
4.71.2.89 NQ_ERR_TIMEOUT	188

4.71.2.90	NQ_ERR_NORESOURCE	188
4.71.2.91	NQ_ERR_TOOMANYUIDS	188
4.71.2.92	NQ_ERR_INVUID	188
4.71.2.93	NQ_ERR_USEMPX	188
4.71.2.94	NQ_ERR_USESTD	188
4.71.2.95	NQ_ERR_CONTMPX	189
4.71.2.96	NQ_ERR_NOSUPPORT	189
4.71.2.97	NQ_ERR_NOWRITE	189
4.71.2.98	NQ_ERR_BADUNIT	189
4.71.2.99	NQ_ERR_NOTREADY	189
4.71.2.100	NQ_ERR_BADCMD	189
4.71.2.101	NQ_ERR_DATA	189
4.71.2.102	NQ_ERR_BADREQ	189
4.71.2.103	NQ_ERR_SEEK	189
4.71.2.104	NQ_ERR_BADMEDIA	189
4.71.2.105	NQ_ERR_BADSECTOR	189
4.71.2.106	NQ_ERR_NOPAPER	189
4.71.2.107	NQ_ERR_WRITE	190
4.71.2.108	NQ_ERR_READ	190
4.71.2.109	NQ_ERR_GENERAL	190
4.71.2.110	NQ_ERR_WRONGDISK	190
4.71.2.111	NQ_ERR_FCBUNAVAIL	190
4.71.2.112	NQ_ERR_SHAREBUFEXC	190
4.71.2.113	NQ_ERR_DISKFULL	190
4.72	TCP and NetBIOS Layer Error Codes	191
4.72.1	Detailed Description	191
4.72.2	Macro Definition Documentation	191
4.72.2.1	NQ_ERR_NBILLEGALSOCKETSLLOT	191
4.72.2.2	NQ_ERR_NBNOTNETBIOSNAME	192
4.72.2.3	NQ_ERR_NBTIMEOUT	192
4.72.2.4	NQ_ERR_NBNEGATIVERESPONSE	192
4.72.2.5	NQ_ERR_NBHOSTNAMENOTRESOLVED	192
4.72.2.6	NQ_ERR_NBCANCELLISTENFAIL	192
4.72.2.7	NQ_ERR_NBSOCKETOVERFLOW	192
4.72.2.8	NQ_ERR_NBNOBINDBEFORELISTEN	192
4.72.2.9	NQ_ERR_NBILLEGALDATAGRAMSOURCE	192
4.72.2.10	NQ_ERR_NBILLEGALDATAGRAMDESTINATION	192
4.72.2.11	NQ_ERR_NBINVALIDPARAMETER	192
4.72.2.12	NQ_ERR_NBINTERNALERROR	192
4.72.2.13	NQ_ERR_NBILLEGALDATAGRAMTYPE	192
4.72.2.14	NQ_ERR_NBDDCOMMUNICATIONERROR	193
4.72.2.15	NQ_ERR_NBBUFFEROVERFLOW	193
4.72.2.16	NQ_ERR_NBRELEasenameFAIL	193
4.72.2.17	NQ_ERR_NBLISTENFAIL	193
4.72.2.18	NQ_ERR_SOCKETCREATE	193
4.72.2.19	NQ_ERR_SOCKETBIND	193
4.72.2.20	NQ_ERR_SOCKETNAME	193
4.72.2.21	NQ_ERR_SOCKETSEND	193
4.72.2.22	NQ_ERR_SOCKETSETOPTION	193
4.72.2.23	NQ_ERR_SOCKETADDRESS	193
4.72.2.24	NQ_ERR_CAPTUREZERO	193
4.72.2.25	NQ_ERR_INVALIDUINT32SIZE	193
4.72.2.26	NQ_ERR_RECONNECTREQUIRED	194
4.72.2.27	NQ_ERR_SIZEERROR	194
4.72.2.28	NQ_ERR_SEEKERROR	194
4.72.2.29	NQ_ERR_ATTRERROR	194
4.72.2.30	NQ_ERR_KERBEROSERROR	194
4.72.2.31	NQ_ERR_RAPGENERALERROR	194

4.72.2.32	NQ_ERR_RAPACCESSDENIED	194
4.72.2.33	NQ_ERR_RAPMOREDATA	194
4.72.2.34	NQ_ERR_RAPSERVERNOTSTARTED	194
4.72.2.35	NQ_ERR_RAPBADTRANSACTCONFIG	194
4.72.2.36	NQ_ERR_RAPSERVICENOTINSTALLED	194
4.72.2.37	NQ_ERR_RAPDEVICENOTCONNECTED	194
4.72.2.38	NQ_ERR_RAPNOTACTIVEINDOMAIN	195
4.72.2.39	NQ_ERR_RAPNOBROWSESERVERSFOUND	195
4.72.2.40	NQ_ERR_ICONVERROR	195
4.72.2.41	NQ_ERR_ICONVOPEN	195
4.72.2.42	UD_NQ_MAXPWDLEN	195
4.73	Authentication Statuses	196
4.73.1	Detailed Description	196
4.73.2	Macro Definition Documentation	196
4.73.2.1	NQ_CS_PWDFOUND	196
4.73.2.2	NQ_CS_PWDNOAUTH	196
4.73.2.3	NQ_CS_PWDNOUSER	196
4.73.2.4	NQ_CS_PWDLMHASH	196
4.73.2.5	NQ_CS_PWDANY	196
4.74	Functions	197
4.74.1	Detailed Description	198
4.74.2	Function Documentation	198
4.74.2.1	udInit	198
4.74.2.2	udStop	198
4.74.2.3	udResetServerParams	198
4.74.2.4	udCifsServerStarted	199
4.74.2.5	udCifsServerClosed	199
4.74.2.6	udNetBiosDaemonStarted	199
4.74.2.7	udNetBiosDaemonClosed	199
4.74.2.8	udGetScopeId	199
4.74.2.9	udGetWins	199
4.74.2.10	udGetDomain	199
4.74.2.11	udGetDnsParams	201
4.74.2.12	udGetCredentials	201
4.74.2.13	udGetPassword	201
4.74.2.14	udGetSmbError	202
4.74.2.15	udNqToSystemError	202
4.74.2.16	udGetTransportPriority	202
4.74.2.17	udGetServerComment	203
4.74.2.18	udNetBiosDataIn	203
4.74.2.19	udServerDataIn	203
4.74.2.20	udServerShareConnect	203
4.74.2.21	udServerShareDisconnect	203
4.74.2.22	udGetCodePage	204
4.74.2.23	udSetCodePage	204
4.74.2.24	udGetPort	204
4.74.2.25	udGetComputerId	204
4.74.2.26	udGetUserCount	205
4.74.2.27	udGetUserRidByName	205
4.74.2.28	udGetUserNameByRid	205
4.74.2.29	udGetUserInfo	206
4.74.2.30	udSetUserAsAdministrator	206
4.74.2.31	udSetUserInfo	206
4.74.2.32	udCreateUser	207
4.74.2.33	udDeleteUserByRid	207
4.74.2.34	udSaveShareInformation	207
4.74.2.35	udRemoveShare	208
4.74.2.36	udNetBiosError	208

4.74.2.37	udGetComputerSecret	209
4.74.2.38	udGetComputerSecretByDomain	210
4.74.2.39	udSetComputerSecret	210
4.74.2.40	udSetComputerSecretByDomain	210
4.74.2.41	udGetFileSystemName	211
4.74.2.42	udEventLog	211
4.74.2.43	udAllocateBuffer	211
4.74.2.44	udReleaseBuffer	212
4.74.2.45	udGetSecurityDescriptor	213
4.74.2.46	udSetSecurityDescriptor	213
4.74.2.47	udLoadShareSecurityDescriptor	213
4.74.2.48	udSaveShareSecurityDescriptor	213
4.74.2.49	udGetNextShareEx	214
4.74.2.50	udGetNextMount	214
4.74.2.51	udGetMessageSigningPolicy	214
4.74.2.52	udGetGlobalEncryption	214
4.74.2.53	udGetInternalCapture	215
4.74.2.54	udGetCaptureFileBaseFolder	215
4.74.2.55	udGetInternalTrace	215
4.74.2.56	udGetTaskPriorities	215
4.74.2.57	udGetDriverName	215
4.74.2.58	udGetHostName	215
4.74.2.59	udGetLogFileBaseFolder	215
4.74.2.60	udGetMaxDiskSize	216
4.75	Event Log Constants	217
4.75.1	Detailed Description	217
4.75.2	Macro Definition Documentation	217
4.75.2.1	UD_LOG_MODULE_CS	217
4.75.2.2	UD_LOG_MODULE_CC	217
4.75.2.3	UD_LOG_CLASS_GEN	217
4.75.2.4	UD_LOG_CLASS_FILE	218
4.75.2.5	UD_LOG_CLASS_SHARE	218
4.75.2.6	UD_LOG_CLASS_USER	218
4.75.2.7	UD_LOG_CLASS_CONNECTION	218
4.75.2.8	UD_LOG_GEN_START	218
4.75.2.9	UD_LOG_GEN_STOP	218
4.75.2.10	UD_LOG_GEN_NAMECONFLICT	218
4.75.2.11	UD_LOG_GEN_NAMEREGFAIL	218
4.75.2.12	UD_LOG_FILE_CREATE	218
4.75.2.13	UD_LOG_FILE_OPEN	218
4.75.2.14	UD_LOG_FILE_CLOSE	218
4.75.2.15	UD_LOG_FILE_DELETE	218
4.75.2.16	UD_LOG_FILE_RENAME	219
4.75.2.17	UD_LOG_FILE_ATTRIBGET	219
4.75.2.18	UD_LOG_FILE_ATTRIBSET	219
4.75.2.19	UD_LOG_FILE_SIZESET	219
4.75.2.20	UD_LOG_FILE_VOLUMEINFO	219
4.75.2.21	UD_LOG_FILE_QUERYDIRECTORY	219
4.75.2.22	UD_LOG_FILE_SEEK	219
4.75.2.23	UD_LOG_FILE_LOCK	219
4.75.2.24	UD_LOG_FILE_UNLOCK	219
4.75.2.25	UD_LOG_SHARE_CONNECT	219
4.75.2.26	UD_LOG_SHARE_DISCONNECT	219
4.75.2.27	UD_LOG_USER_LOGON	219
4.75.2.28	UD_LOG_USER_LOGOFF	220
4.75.2.29	UD_LOG_CONNECTION_CONNECT	220
4.75.2.30	UD_LOG_CONNECTION_DISCONNECT	220

5	Data Structure Documentation	221
5.1	AMCredentials Struct Reference	221
5.1.1	Detailed Description	221
5.1.2	Field Documentation	221
5.1.2.1	domain	221
5.1.2.2	user	221
5.1.2.3	password	221
5.1.2.4	type	221
5.2	AMCredentialsA Struct Reference	222
5.2.1	Detailed Description	222
5.2.2	Field Documentation	222
5.2.2.1	domain	222
5.2.2.2	user	222
5.2.2.3	password	222
5.2.2.4	type	222
5.3	AMDomain Struct Reference	222
5.3.1	Field Documentation	223
5.3.1.1	name	223
5.3.1.2	realm	223
5.4	AMDomainA Struct Reference	223
5.4.1	Detailed Description	223
5.4.2	Field Documentation	223
5.4.2.1	name	223
5.4.2.2	realm	223
5.5	CCFileInfo Struct Reference	223
5.5.1	Detailed Description	224
5.5.2	Field Documentation	224
5.5.2.1	creationTime	224
5.5.2.2	lastAccessTime	224
5.5.2.3	lastWriteTime	224
5.5.2.4	changeTime	224
5.5.2.5	endOfFile	224
5.5.2.6	allocationSize	224
5.5.2.7	attributes	224
5.5.2.8	numberOfLinks	224
5.5.2.9	fileIndex	225
5.5.2.10	volumeld	225
5.6	CCFileInfo_t Struct Reference	225
5.6.1	Detailed Description	225
5.6.2	Field Documentation	225
5.6.2.1	creationTimeLow	225
5.6.2.2	creationTimeHigh	225
5.6.2.3	lastAccessTimeLow	225
5.6.2.4	lastAccessTimeHigh	226
5.6.2.5	lastWriteTimeLow	226
5.6.2.6	lastWriteTimeHigh	226
5.6.2.7	attributes	226
5.6.2.8	volumeSerialNumber	226
5.6.2.9	allocationSizeLow	226
5.6.2.10	allocationSizeHigh	226
5.6.2.11	fileSizeLow	226
5.6.2.12	fileSizeHigh	226
5.6.2.13	numberOfLinks	226
5.6.2.14	fileIndexLow	226
5.6.2.15	fileIndexHigh	226
5.7	CCFileTime Struct Reference	227
5.7.1	Detailed Description	227
5.7.2	Field Documentation	227

5.7.2.1	timeLow	227
5.7.2.2	timeHigh	227
5.8	CCFindFileData Struct Reference	227
5.8.1	Detailed Description	228
5.8.2	Field Documentation	228
5.8.2.1	fileAttributes	228
5.8.2.2	creationTimeLow	228
5.8.2.3	creationTimeHigh	228
5.8.2.4	lastAccessTimeLow	228
5.8.2.5	lastAccessTimeHigh	228
5.8.2.6	lastWriteTimeLow	228
5.8.2.7	lastWriteTimeHigh	228
5.8.2.8	changeTimeLow	228
5.8.2.9	changeTimeHigh	228
5.8.2.10	fileSizeLow	228
5.8.2.11	fileSizeHigh	228
5.8.2.12	allocationSizeLow	229
5.8.2.13	allocationSizeHigh	229
5.8.2.14	fileNameLength	229
5.8.2.15	fileName	229
5.9	CCFindFileDataA Struct Reference	229
5.9.1	Detailed Description	229
5.9.2	Field Documentation	229
5.9.2.1	fileAttributes	229
5.9.2.2	creationTimeLow	230
5.9.2.3	creationTimeHigh	230
5.9.2.4	lastAccessTimeLow	230
5.9.2.5	lastAccessTimeHigh	230
5.9.2.6	lastWriteTimeLow	230
5.9.2.7	lastWriteTimeHigh	230
5.9.2.8	changeTimeLow	230
5.9.2.9	changeTimeHigh	230
5.9.2.10	fileSizeLow	230
5.9.2.11	fileSizeHigh	230
5.9.2.12	allocationSizeLow	230
5.9.2.13	allocationSizeHigh	230
5.9.2.14	fileNameLength	231
5.9.2.15	fileName	231
5.10	CCMountInfo Struct Reference	231
5.10.1	Detailed Description	231
5.10.2	Field Documentation	231
5.10.2.1	creationTime	231
5.10.2.2	userLogonType	231
5.10.2.3	serverName	231
5.10.2.4	serverInfo	231
5.10.2.5	statistics	232
5.11	CCMountInfoA Struct Reference	232
5.11.1	Detailed Description	232
5.11.2	Field Documentation	232
5.11.2.1	creationTime	232
5.11.2.2	userLogonType	232
5.11.2.3	serverName	232
5.11.2.4	serverInfo	232
5.11.2.5	statistics	232
5.12	CCNetDomainItem Struct Reference	233
5.12.1	Detailed Description	233
5.12.2	Field Documentation	233
5.12.2.1	name	233

5.12.2.2	type	233
5.12.2.3	comment	233
5.12.2.4	majorVersion	233
5.12.2.5	minorVersion	233
5.13	CCNetServerItem Struct Reference	233
5.13.1	Detailed Description	234
5.13.2	Field Documentation	234
5.13.2.1	name	234
5.13.2.2	type	234
5.13.2.3	comment	234
5.13.2.4	majorVersion	234
5.13.2.5	minorVersion	234
5.14	CCNetShareItem Struct Reference	234
5.14.1	Detailed Description	234
5.14.2	Field Documentation	234
5.14.2.1	name	234
5.14.2.2	type	234
5.14.2.3	comment	235
5.15	CCNetShareItemA Struct Reference	235
5.15.1	Detailed Description	235
5.15.2	Field Documentation	235
5.15.2.1	name	235
5.15.2.2	type	235
5.15.2.3	comment	235
5.16	CCServerInfo Struct Reference	235
5.16.1	Detailed Description	236
5.16.2	Field Documentation	236
5.16.2.1	dialect	236
5.16.2.2	serverCapabilities	236
5.16.2.3	ips	236
5.16.2.4	numIps	236
5.16.2.5	serverSigningPolicy	236
5.17	CCSStatistics Struct Reference	236
5.17.1	Detailed Description	236
5.17.2	Field Documentation	236
5.17.2.1	readAmount	236
5.17.2.2	writeAmount	237
5.18	CCVolumeInfo Struct Reference	237
5.18.1	Detailed Description	237
5.18.2	Field Documentation	237
5.18.2.1	sectorsPerCluster	237
5.18.2.2	bytesPerSector	237
5.18.2.3	freeClusters	237
5.18.2.4	totalClusters	237
5.18.2.5	fsType	237
5.18.2.6	serialNumber	237
5.19	CMCodepage Struct Reference	237
5.19.1	Detailed Description	238
5.19.2	Field Documentation	238
5.19.2.1	id	238
5.19.2.2	toAnsi	238
5.19.2.3	toUnicode	238
5.19.2.4	toUpper	238
5.19.2.5	ansiToFs	238
5.19.2.6	fsToAnsi	238
5.19.2.7	a2uTab	239
5.20	CMCrypterList Struct Reference	239
5.20.1	Detailed Description	239

5.20.2	Field Documentation	239
5.20.2.1	md4	239
5.20.2.2	md5	239
5.20.2.3	hmacmd5	239
5.20.2.4	sha256	239
5.20.2.5	aes128cmac	240
5.20.2.6	sha512	240
5.20.2.7	aes128ccmEncryption	240
5.20.2.8	aes128ccmDecryption	240
5.20.2.9	aes128gcmEncryption	240
5.20.2.10	aes128gcmDecryption	240
5.21	CMIOVecBlob Struct Reference	240
5.21.1	Detailed Description	240
5.21.2	Field Documentation	240
5.21.2.1	data	240
5.21.2.2	len	241
5.22	CMPProductInfo Struct Reference	241
5.22.1	Field Documentation	241
5.22.1.1	productName	241
5.22.1.2	productType	241
5.22.1.3	version	241
5.22.1.4	checksum	241
5.23	CMResolverRegisteredMethodDescription Struct Reference	241
5.23.1	Detailed Description	242
5.23.2	Field Documentation	242
5.23.2.1	activationPriority	242
5.23.2.2	timeout	242
5.23.2.3	serverIP	242
5.23.2.4	requestByName	242
5.23.2.5	responseByName	242
5.23.2.6	requestByIp	243
5.23.2.7	responseByIp	243
5.24	CSFileData Struct Reference	243
5.24.1	Detailed Description	244
5.24.2	Field Documentation	244
5.24.2.1	fileNamePath	244
5.24.2.2	userName	244
5.24.2.3	IP	244
5.24.2.4	isDirectory	244
5.25	EVCSOpenFileEntry Struct Reference	244
5.25.1	Detailed Description	244
5.25.2	Field Documentation	245
5.25.2.1	userName	245
5.25.2.2	fileName	245
5.25.2.3	shareName	245
5.25.2.4	ip	245
5.25.2.5	access	245
5.26	EVCSShareConnectionEntry Struct Reference	245
5.26.1	Detailed Description	245
5.26.2	Field Documentation	245
5.26.2.1	userName	245
5.26.2.2	shareName	246
5.26.2.3	ip	246
5.26.2.4	ipc	246
5.26.2.5	printQueue	246
5.27	FileDataA_t Struct Reference	246
5.27.1	Detailed Description	246
5.27.2	Field Documentation	246

5.27.2.1	fileNamePath	246
5.27.2.2	userName	246
5.27.2.3	IP	246
5.27.2.4	isDirectory	247
5.28	NQ_INT64 Struct Reference	247
5.28.1	Detailed Description	247
5.28.2	Field Documentation	247
5.28.2.1	low	247
5.28.2.2	high	247
5.28.2.3	sign	247
5.29	NQ_IOVEC Struct Reference	247
5.29.1	Field Documentation	248
5.29.1.1	iov_base	248
5.29.1.2	iov_len	248
5.30	NQ_IOVECBUF Struct Reference	248
5.30.1	Field Documentation	248
5.30.1.1	iovec	248
5.30.1.2	iovSegmentCnt	248
5.31	NQ_IOVecBuffer Struct Reference	248
5.31.1	Field Documentation	248
5.31.1.1	data	248
5.31.1.2	len	249
5.32	NQ_IPADDRESS Struct Reference	249
5.32.1	Detailed Description	249
5.32.2	Field Documentation	249
5.32.2.1	version	249
5.32.2.2	v6	249
5.32.2.3	v4	249
5.32.2.4	vr	249
5.32.2.5	addr	249
5.33	NQ_IPADDRESSVR Struct Reference	250
5.33.1	Field Documentation	250
5.33.1.1	id	250
5.33.1.2	version	250
5.33.1.3	v6	250
5.33.1.4	v4	250
5.33.1.5	addr	250
5.34	NQ_UINT64 Struct Reference	250
5.34.1	Detailed Description	250
5.34.2	Field Documentation	250
5.34.2.1	low	250
5.34.2.2	high	251
5.35	SYDeviceMode Struct Reference	251
5.35.1	Detailed Description	251
5.35.2	Field Documentation	252
5.35.2.1	specVersion	252
5.35.2.2	driverVersion	252
5.35.2.3	size	252
5.35.2.4	driverExtraLength	252
5.35.2.5	driverExtraData	252
5.35.2.6	fields	252
5.35.2.7	orientation	252
5.35.2.8	paperSize	252
5.35.2.9	paperLength	252
5.35.2.10	paperWidth	252
5.35.2.11	scale	252
5.35.2.12	copies	252
5.35.2.13	defaultSource	252

5.35.2.14	printQuality	252
5.35.2.15	color	252
5.35.2.16	duplex	252
5.35.2.17	yResolution	252
5.35.2.18	ttOption	252
5.35.2.19	collate	252
5.35.2.20	formName	252
5.35.2.21	logPixels	252
5.35.2.22	bitsPerPel	252
5.35.2.23	pelsWidth	252
5.35.2.24	pelsHeight	252
5.35.2.25	displayFlags	252
5.35.2.26	displayFrequency	252
5.35.2.27	icmMethod	252
5.35.2.28	icmIntent	253
5.35.2.29	mediaType	253
5.35.2.30	ditherType	253
5.35.2.31	reserved1	253
5.35.2.32	reserved2	253
5.35.2.33	panningWidth	253
5.35.2.34	panningHeight	253
5.36	SYFileInformation Struct Reference	253
5.36.1	Detailed Description	253
5.36.2	Field Documentation	253
5.36.2.1	creationTime	253
5.36.2.2	lastChangeTime	254
5.36.2.3	lastAccessTime	254
5.36.2.4	lastWriteTime	254
5.36.2.5	sizeLow	254
5.36.2.6	sizeHigh	254
5.36.2.7	allocSizeLow	254
5.36.2.8	allocSizeHigh	254
5.36.2.9	attributes	254
5.36.2.10	isDeleted	254
5.36.2.11	numLinks	254
5.36.2.12	fileIdLow	254
5.36.2.13	fileIdHigh	254
5.37	SYPrinterDriver Struct Reference	255
5.37.1	Detailed Description	255
5.37.2	Field Documentation	255
5.37.2.1	osVersion	255
5.37.2.2	name	255
5.37.2.3	driverPath	255
5.37.2.4	dataFile	255
5.37.2.5	configFile	255
5.37.2.6	helpFile	255
5.37.2.7	dependentFiles	255
5.37.2.8	monitorName	255
5.37.2.9	defaultDataType	255
5.37.2.10	previousNames	255
5.37.2.11	driverDate	255
5.37.2.12	driverVersions	256
5.37.2.13	manufacturer	256
5.37.2.14	manufacturerURL	256
5.37.2.15	hardwareID	256
5.37.2.16	provider	256
5.37.2.17	attributes	256
5.37.2.18	configVersion	256

5.37.2.19	driverVersion	256
5.38	SYPrinterInfo Struct Reference	256
5.38.1	Detailed Description	257
5.38.2	Field Documentation	257
5.38.2.1	flags	257
5.38.2.2	portName	257
5.38.2.3	driverName	257
5.38.2.4	comment	257
5.38.2.5	location	257
5.38.2.6	devMode	257
5.38.2.7	sepFile	257
5.38.2.8	printProcessor	257
5.38.2.9	dataType	257
5.38.2.10	parameters	257
5.38.2.11	attributes	257
5.38.2.12	priority	257
5.38.2.13	defaultPriority	257
5.38.2.14	startTime	257
5.38.2.15	untilTime	257
5.38.2.16	status	257
5.38.2.17	cJobs	257
5.38.2.18	totalJobs	257
5.38.2.19	totalBytes	257
5.38.2.20	globalCounter	257
5.38.2.21	totalPages	257
5.38.2.22	majorVersion	258
5.38.2.23	buildVersion	258
5.38.2.24	sessionCounter	258
5.38.2.25	printerErrors	258
5.38.2.26	cSetPrinter	258
5.38.2.27	averagePpm	258
5.38.2.28	deviceNotSelectedTimeout	258
5.38.2.29	transmissionRetryTimeout	258
5.38.2.30	securityDescriptor	258
5.38.2.31	securityDescriptorLength	258
5.39	SYPrintFormInfo Struct Reference	258
5.39.1	Detailed Description	258
5.39.2	Field Documentation	258
5.39.2.1	id	258
5.39.2.2	name	258
5.39.2.3	flags	259
5.39.2.4	size	259
5.39.2.5	imageableArea	259
5.40	SYPrintJobInfo Struct Reference	259
5.40.1	Detailed Description	259
5.40.2	Field Documentation	259
5.40.2.1	id	259
5.40.2.2	documentName	259
5.40.2.3	pStatus	260
5.40.2.4	status	260
5.40.2.5	priority	260
5.40.2.6	position	260
5.40.2.7	totalPages	260
5.40.2.8	pagesPrinted	260
5.40.2.9	submitTime	260
5.40.2.10	startTime	260
5.40.2.11	untilTime	260
5.40.2.12	time	260

5.40.2.13	size	260
5.40.2.14	securityDescriptor	260
5.40.2.15	securityDescriptorLength	261
5.40.2.16	user	261
5.41	SYPrintRect Struct Reference	261
5.41.1	Detailed Description	261
5.41.2	Field Documentation	261
5.41.2.1	left	261
5.41.2.2	top	261
5.41.2.3	right	261
5.41.2.4	bottom	261
5.42	SYPrintSize Struct Reference	262
5.42.1	Detailed Description	262
5.42.2	Field Documentation	262
5.42.2.1	width	262
5.42.2.2	height	262
5.43	SYTimeFragments Struct Reference	262
5.43.1	Detailed Description	262
5.43.2	Field Documentation	263
5.43.2.1	sec	263
5.43.2.2	min	263
5.43.2.3	hour	263
5.43.2.4	day	263
5.43.2.5	month	263
5.43.2.6	year	263
5.44	SYVolumeInformation Struct Reference	263
5.44.1	Detailed Description	264
5.44.2	Field Documentation	264
5.44.2.1	serialNumberLow	264
5.44.2.2	serialNumberHigh	264
5.44.2.3	creationTimeLow	264
5.44.2.4	creationTimeHigh	264
5.44.2.5	fileSystemIdLow	264
5.44.2.6	fileSystemIdHigh	264
5.44.2.7	blockSizeLow	264
5.44.2.8	blockSizeHigh	264
5.44.2.9	blocksPerUnitLow	264
5.44.2.10	blocksPerUnitHigh	264
5.44.2.11	totalUnitsLow	265
5.44.2.12	totalUnitsHigh	265
5.44.2.13	freeUnitsLow	265
5.44.2.14	freeUnitsHigh	265
5.45	UDFileAccessEvent Struct Reference	265
5.45.1	Detailed Description	265
5.45.2	Field Documentation	265
5.45.2.1	fileName	265
5.45.2.2	newName	266
5.45.2.3	access	266
5.45.2.4	before	266
5.45.2.5	rid	266
5.45.2.6	tid	266
5.45.2.7	sizeLow	266
5.45.2.8	sizeHigh	266
5.45.2.9	offsetLow	266
5.45.2.10	offsetHigh	266
5.45.2.11	infoLevel	266
5.46	UDShareAccessEvent Struct Reference	266
5.46.1	Detailed Description	267

5.46.2	Field Documentation	267
5.46.2.1	shareName	267
5.46.2.2	ipc	267
5.46.2.3	printQueue	267
5.46.2.4	rid	267
5.46.2.5	tid	267
5.47	UDUserAccessEvent Struct Reference	267
5.47.1	Detailed Description	267
5.47.2	Field Documentation	268
5.47.2.1	rid	268
6	File Documentation	269
6.1	amapi.h File Reference	269
6.2	ccapi.h File Reference	270
6.3	cmapi.h File Reference	274
6.4	csapi.h File Reference	276
6.5	ndapi.h File Reference	277
6.6	nqapi.h File Reference	277
6.7	syapi.h File Reference	277
6.8	syccommon.h File Reference	278
6.9	udapi.h File Reference	280
6.9.1	Macro Definition Documentation	297
6.9.1.1	UD_DNS_SERVERSTRINGSIZE	297
6.9.1.2	NQ_IOBuf	297
6.9.1.3	NQ_IOBuffer	297
6.9.1.4	CMIOBufBlob	298
6.9.2	Typedef Documentation	298
6.9.2.1	NQ_CHAR	298
6.9.2.2	NQ_BYTE	298
6.9.2.3	NQ_INT	298
6.9.2.4	NQ_UINT	298
6.9.2.5	NQ_BOOL	298
6.9.2.6	NQ_INDEX	298
6.9.2.7	NQ_COUNT	298
6.9.2.8	NQ_INT16	298
6.9.2.9	NQ_INT32	298
6.9.2.10	NQ_UINT16	298
6.9.2.11	NQ_UINT32	298
6.9.2.12	NQ_WCHAR	298
6.9.2.13	NQ_HANDLE	298
6.9.2.14	NQ_STATUS	298
6.9.2.15	NQ_PORT	298
6.9.2.16	NQ_IPADDRESS4	298
6.9.2.17	NQ_IPADDRESS6	298
6.9.2.18	NQ_ULONG	298
6.9.2.19	NQ_FLOAT	298
6.9.2.20	NQ_TIME	298

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Authentication	6
Encryption Algorithms	9
Security Mechanisms	10
Security Password Type	11
Client	12
Initialization and Shutdown	26
Mount and Server	27
Mount information	31
User logon type	23
Server capabilities	24
signing policy	25
Search Operations	33
Directory Operations	36
File Operations	38
File attributes	15
File access mode	16
File share mode	17
File locality	18
File create action	19
File open action	20
File seek constants	21
File Info Operations	45
Timeout Operations	55
Network Browsing	56
Domain Operations	65
Special Operations	70
Authentication Level	22
Dialects Operations	74
User Credentials	76
Credential fields length	13
Errors	79
Driver Notification Events	14
Auxiliary methods	80
Common	82
Initialization and Shutdown	83
Resolver	85
Product Information	92

Codepage	93
Cryptographic	96
DNS and WINS	102
Auxiliary	105
Server	106
Initialization and Shutdown	107
Shares Management	108
Open Files Enumeration	112
Cleanup	114
Anonymous	115
Event Log	116
NetBIOS Daemon	117
NQ Common	118
Initialization and Shutdown	119
NetBIOS Name Registration	120
Utils	123
System API	124
System Common API	125
File Information	126
File Attributes	127
Printer	129
Printer Flags	130
Printer Status Flags	132
Printer Attribute Flags	134
Printer Status Job	137
Printer Priority Job	139
Printer Control Commands	140
Printer Job Control Commands	141
Printer Form Flags	142
User Defined	143
SMB Errors	144
SMB Error Codes	149
Internal NT Statuses	178
NQ Error Codes	179
TCP and NetBIOS Layer Error Codes	191
Authentication Statuses	196
Functions	197
Event Log Constants	217

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

AMCredentials	221
AMCredentialsA	222
AMDomain	222
AMDomainA	223
CCFileInfo	223
CCFileInfo_t	225
CCFileTime	227
CCFindFileData	227
CCFindFileDataA	229
CCMountInfo	231
CCMountInfoA	232
CCNetDomainItem	233
CCNetServerItem	233
CCNetShareItem	234
CCNetShareItemA	235
CCServerInfo	235
CCStatistics	236
CCVolumeInfo	237
CMCodepage	237
CMCrypterList	239
CMIOVecBlob	240
CMProductInfo	241
CMResolverRegisteredMethodDescription	241
CSFileData	243
EVCSOpenFileEntry	244
EVCSShareConnectionEntry	245
FileDataA_t	246
NQ_INT64	247
NQ_IOVEC	247
NQ_IOVECBUF	248
NQ_IOVecBuffer	248
NQ_IPADDRESS	249
NQ_IPADDRESSVR	250
NQ_UINT64	250
SYDeviceMode	251
SYFileInformation	253
SYPrinterDriver	255
SYPrinterInfo	256

SYPrintFormInfo	258
SYPrintJobInfo	259
SYPrintRect	261
SYPrintSize	262
SYTimeFragments	262
SYVolumeInformation	263
UDFileAccessEvent	265
UDShareAccessEvent	266
UDUserAccessEvent	267

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

amapi.h	269
ccapi.h	270
cmapi.h	274
csapi.h	276
ndapi.h	277
nqapi.h	277
syapi.h	277
sycommon.h	278
udapi.h	280

Chapter 4

Module Documentation

4.1 Authentication

Modules

- [Encryption Algorithms](#)
- [Security Mechanisms](#)
- [Security Password Type](#)

Data Structures

- struct [AMDomainA](#)
- struct [AMDomain](#)
- struct [AMCredentialsA](#)
- struct [AMCredentials](#)

Macros

- `#define` [AM_MAXSECURITYLEVEL](#) 4

Functions

- [NQ_BOOL](#) [amStart](#) (void)
- void [amShutdown](#) (void)
- void [amClientDefineLevel](#) ([NQ_UINT](#) level, [NQ_UINT](#) crypter1, [NQ_UINT](#) crypter2, [NQ_UINT32](#) mechanisms)
- void [amSetNonSecureAuthentication](#) ([NQ_BOOL](#) enableNonSecureAuthentication)

4.1.1 Detailed Description

4.1.2 Macro Definition Documentation

4.1.2.1 `#define` [AM_MAXSECURITYLEVEL](#) 4

Maximum security level. Security levels are counted from zero to this value. Used by [amClientDefineLevel\(\)](#)

4.1.3 Function Documentation

4.1.3.1 NQ_BOOL amStart (void)

This function initializes this module.

Returns

TRUE on success and FALSE on failure.

4.1.3.2 void amShutdown (void)

This function disposes resources used by this module.

4.1.3.3 void amClientDefineLevel (NQ_UINT level, NQ_UINT crypter1, NQ_UINT crypter2, NQ_UINT32 mehanisms)

NQ Authentication module conveys authentication according to the required level of security.

This function assigns parameters for one authentication level:

- NQ combines two encryption algorithms in one authentication blob. User can choose between LM, NTLM and NTLMv2 encryptions.
- User can choose security mechanisms for extended security negotiations. Currently, the available mechanisms are: NTLMSSP and Kerberos.

Parameters

<i>level</i>	Authentication level to define. This value should be greater or equal to zero and it should not exceed the maximum security level as defined in AM_MAXSECURITYLEVEL . An illegal value will have no effect.
<i>crypter1</i>	The first encryption algorithm. Available values are: <ul style="list-style-type: none"> • AM_CRYPTER_LM • AM_CRYPTER_NTLM • AM_CRYPTER_LM2 • AM_CRYPTER_NTLM2 • AM_CRYPTER_NONE
<i>crypter2</i>	The second encryption algorithm. Available values are: <ul style="list-style-type: none"> • AM_CRYPTER_LM • AM_CRYPTER_NTLM • AM_CRYPTER_LM2 • AM_CRYPTER_NTLM2 • AM_CRYPTER_NONE

<i>mechanisms</i>	<p>Available security mechanisms. This value is a bit mask of the following:</p> <ul style="list-style-type: none"> • AM_MECH_KERBEROS • AM_MECH_NTLMSSP User can specify one of those masks, both or none. When two masks are specified, Kerberos is considered first.
-------------------	---

Note

- Level parameters are assigned globally so that two concurrent applications using the same level are sharing the same parameters. If one of them assigns new parameters this also affects the second one.
- This function is not thread-safe.
- The default per-level parameters are:

Level	Crypter 1	Crypter 2	NTLMSSP	KERBEROS
0	LM	none	-	-
1	LM	NTLM	-	-
2	LM	NTLM	Yes	Yes
3	LMv2	NTLMv2	Yes	-
4	LMv2	NTLMv2	Yes	Yes

4.1.3.4 void amSetNonSecureAuthentication (**NQ_BOOL** *enableNonSecureAuthentication*)

Some old authentication methods - used in SMB 1 only - are considered today non secure. By default those non secure methods will be disabled. Use this function to enable / disable non secure methods.

Parameters

<i>enableNon-Secure-Authentication</i>	TRUE - non secure authentication will be used. FALSE - non secure authentication will not be used.
--	--

Note

- This function uses [amClientDefineLevel\(\)](#) to set authentication algorithms. The [amClientDefineLevel\(\)](#) call can be used to further modify the levels set by this function.
- See also notes in [amClientDefineLevel\(\)](#).

4.2 Encryption Algorithms

Macros

- #define [AM_CRYPTER_NONE](#) 0
- #define [AM_CRYPTER_LM](#) 1
- #define [AM_CRYPTER_NTLM](#) 2
- #define [AM_CRYPTER_LM2](#) 3
- #define [AM_CRYPTER_NTLM2](#) 4

4.2.1 Detailed Description

Available encryption algorithms

4.2.2 Macro Definition Documentation

4.2.2.1 #define [AM_CRYPTER_NONE](#) 0

Encryption is skipped.

4.2.2.2 #define [AM_CRYPTER_LM](#) 1

Fixed-length 128-bit LM response.

4.2.2.3 #define [AM_CRYPTER_NTLM](#) 2

Fixed-length 192-bit NTLM response.

4.2.2.4 #define [AM_CRYPTER_LM2](#) 3

Fixed-length 128-bit LMv2 response.

4.2.2.5 #define [AM_CRYPTER_NTLM2](#) 4

Variable-length NTLMv2 response.

4.3 Security Mechanisms

Macros

- #define [AM_MECH_NTLMSSP](#) 1
- #define [AM_MECH_KERBEROS](#) 2

4.3.1 Detailed Description

4.3.2 Macro Definition Documentation

4.3.2.1 #define AM_MECH_NTLMSSP 1

NTLMSSP security mechanism. This value is used in a security mechanism mask (see [amClientDefineLevel\(\)](#)).

4.3.2.2 #define AM_MECH_KERBEROS 2

Kerberos security mechanism. This value is used in a security mechanism mask (see [amClientDefineLevel\(\)](#)).

4.4 Security Password Type

Macros

- #define [AM_CREDENTIALS_PASSWORD_PLAINTEXT](#) 0
- #define [AM_CREDENTIALS_PASSWORD_MD4](#) 1

4.4.1 Detailed Description

4.4.2 Macro Definition Documentation

4.4.2.1 #define [AM_CREDENTIALS_PASSWORD_PLAINTEXT](#) 0

4.4.2.2 #define [AM_CREDENTIALS_PASSWORD_MD4](#) 1

4.5 Client

Modules

- [Initialization and Shutdown](#)
- [Mount and Server](#)
- [Mount information](#)
- [Search Operations](#)
- [Directory Operations](#)
- [File Operations](#)
- [File Info Operations](#)
- [Timeout Operations](#)
- [Network Browsing](#)
- [Domain Operations](#)
- [Special Operations](#)
- [Dialects Operations](#)
- [User Credentials](#)
- [Errors](#)
- [Auxiliary methods](#)

4.5.1 Detailed Description

4.6 Credential fields length

Macros

- #define `CM_USERNAMELENGTH` 256
- #define `PASSWORD_LENGTH UD_NQ_MAXPWDLEN`
- #define `DOMAIN_LENGTH` (CM_NQ_HOSTNAMESIZE)

4.6.1 Detailed Description

4.6.2 Macro Definition Documentation

4.6.2.1 #define `CM_USERNAMELENGTH` 256

maximum user name length

4.6.2.2 #define `PASSWORD_LENGTH UD_NQ_MAXPWDLEN`

maximum password length

4.6.2.3 #define `DOMAIN_LENGTH` (CM_NQ_HOSTNAMESIZE)

maximum domain name length

4.7 Driver Notification Events

Macros

- `#define NOTIFY_SRCH_HANDLE_ERROR 1`
- `#define NOTIFY_FILE_HANDLE_ERROR 2`

4.7.1 Detailed Description

4.7.2 Macro Definition Documentation

4.7.2.1 `#define NOTIFY_SRCH_HANDLE_ERROR 1`

An error occurred in Find File handle. Application should clean the related resources.

4.7.2.2 `#define NOTIFY_FILE_HANDLE_ERROR 2`

An error occurred in File handle. Application should clean the related resources.

4.8 File attributes

Macros

- #define CIFS_ATTR_READONLY 0x01
- #define CIFS_ATTR_HIDDEN 0x02
- #define CIFS_ATTR_SYSTEM 0x04
- #define CIFS_ATTR_VOLUME 0x08
- #define CIFS_ATTR_DIR 0x10
- #define CIFS_ATTR_ARCHIVE 0x20
- #define CIFS_ATTR_ALL (CIFS_ATTR_HIDDEN | CIFS_ATTR_SYSTEM | CIFS_ATTR_DIR)

4.8.1 Detailed Description

4.8.2 Macro Definition Documentation

4.8.2.1 #define CIFS_ATTR_READONLY 0x01

file is read-only

4.8.2.2 #define CIFS_ATTR_HIDDEN 0x02

file is hidden

4.8.2.3 #define CIFS_ATTR_SYSTEM 0x04

file is a system file

4.8.2.4 #define CIFS_ATTR_VOLUME 0x08

this is a volume, not a file

4.8.2.5 #define CIFS_ATTR_DIR 0x10

the file is directory

4.8.2.6 #define CIFS_ATTR_ARCHIVE 0x20

file is scheduled for archiving

4.8.2.7 #define CIFS_ATTR_ALL (CIFS_ATTR_HIDDEN | CIFS_ATTR_SYSTEM | CIFS_ATTR_DIR)

4.9 File access mode

Macros

- #define `FILE_AM_READ` 0
- #define `FILE_AM_WRITE` 1
- #define `FILE_AM_READ_WRITE` 2
- #define `FILE_AM_SPECIAL_MASK` 0x80000000

4.9.1 Detailed Description

4.9.2 Macro Definition Documentation

4.9.2.1 #define `FILE_AM_READ` 0

read-only

4.9.2.2 #define `FILE_AM_WRITE` 1

write-only

4.9.2.3 #define `FILE_AM_READ_WRITE` 2

read and write

4.9.2.4 #define `FILE_AM_SPECIAL_MASK` 0x80000000

When this bit is set in the *access* value of `ccCreateFile()` it has special meaning. After stripping off this bit, NQ uses it as *DesiredAccess* in the `NtCreateAndX(SMB)` command or in the `Create(SMB2)` command.

4.10 File share mode

Macros

- #define `FILE_SM_COMPAT` 0
- #define `FILE_SM_EXCLUSIVE` 1
- #define `FILE_SM_DENY_WRITE` 2
- #define `FILE_SM_DENY_READ` 3
- #define `FILE_SM_DENY_NONE` 4

4.10.1 Detailed Description

4.10.2 Macro Definition Documentation

4.10.2.1 #define `FILE_SM_COMPAT` 0

compatibility mode

4.10.2.2 #define `FILE_SM_EXCLUSIVE` 1

deny read/write/execute (exclusive)

4.10.2.3 #define `FILE_SM_DENY_WRITE` 2

deny write

4.10.2.4 #define `FILE_SM_DENY_READ` 3

deny read/execute

4.10.2.5 #define `FILE_SM_DENY_NONE` 4

deny none

4.11 File locality

Macros

- #define `FILE_LCL_UNKNOWN` 0
- #define `FILE_LCL_SEQUENTIAL` 1
- #define `FILE_LCL_RANDOM` 2
- #define `FILE_LCL_MIXED` 3

4.11.1 Detailed Description

4.11.2 Macro Definition Documentation

4.11.2.1 #define `FILE_LCL_UNKNOWN` 0

locality of reference is unknown

4.11.2.2 #define `FILE_LCL_SEQUENTIAL` 1

mainly sequential access

4.11.2.3 #define `FILE_LCL_RANDOM` 2

mainly random access

4.11.2.4 #define `FILE_LCL_MIXED` 3

random access with some locality

4.12 File create action

Macros

- #define [FILE_CA_FAIL](#) 0
- #define [FILE_CA_CREATE](#) 1

4.12.1 Detailed Description

4.12.2 Macro Definition Documentation

4.12.2.1 #define FILE_CA_FAIL 0

if file does not exist - fail

4.12.2.2 #define FILE_CA_CREATE 1

if file does not exist - create it

4.13 File open action

Macros

- #define `FILE_OA_FAIL` 0
- #define `FILE_OA_OPEN` 1
- #define `FILE_OA_TRUNCATE` 2

4.13.1 Detailed Description

4.13.2 Macro Definition Documentation

4.13.2.1 #define `FILE_OA_FAIL` 0

if file exists - fail

4.13.2.2 #define `FILE_OA_OPEN` 1

if file exists - open it

4.13.2.3 #define `FILE_OA_TRUNCATE` 2

if file exists - open it and truncate

4.14 File seek constants

Macros

- #define [SEEK_FILE_BEGIN](#) 0
- #define [SEEK_FILE_CURRENT](#) 1
- #define [SEEK_FILE_END](#) 2

4.14.1 Detailed Description

4.14.2 Macro Definition Documentation

4.14.2.1 #define SEEK_FILE_BEGIN 0

seek from start of file

4.14.2.2 #define SEEK_FILE_CURRENT 1

seek from the current position in file

4.14.2.3 #define SEEK_FILE_END 2

seek from the file end

4.15 Authentication Level

Macros

- #define [NQ_CC_AUTH_LM_AND_NTLM](#) 1
- #define [NQ_CC_AUTH_NTLM](#) 2
- #define [NQ_CC_AUTH_NTLM_V2](#) 3
- #define [NQ_CC_AUTH_SPNEGO_KERBEROS](#) 4

4.15.1 Detailed Description

4.15.2 Macro Definition Documentation

4.15.2.1 #define NQ_CC_AUTH_LM_AND_NTLM 1

forces CIFS Client to send both LM and NTLM encrypted password

4.15.2.2 #define NQ_CC_AUTH_NTLM 2

forces CIFS Client to send NTLM encrypted password only

4.15.2.3 #define NQ_CC_AUTH_NTLM_V2 3

forces CIFS Client to use NTLMv2 authentication

4.15.2.4 #define NQ_CC_AUTH_SPNEGO_KERBEROS 4

forces CIFS Client to use SPNEGO authentication with an extended security mechanism

4.16 User logon type

Macros

- #define [CC_LOGON_TYPE_REGULAR](#) 0
- #define [CC_LOGON_TYPE_ANONYMOUS](#) 1
- #define [CC_LOGON_TYPE_GUEST](#) 2

4.16.1 Detailed Description

4.16.2 Macro Definition Documentation

4.16.2.1 #define CC_LOGON_TYPE_REGULAR 0

regular user

4.16.2.2 #define CC_LOGON_TYPE_ANONYMOUS 1

anonymous user

4.16.2.3 #define CC_LOGON_TYPE_GUEST 2

guest user

4.17 Server capabilities

Macros

- #define `CC_CAPABILITY_DFS` 0x00000001
- #define `CC_CAPABILITY_LEASING` 0x00000002
- #define `CC_CAPABILITY_LARGE_MTU` 0x00000004
- #define `CC_CAPABILITY_MULTI_CHANNEL` 0x00000008
- #define `CC_CAPABILITY_PERSISTENT_HANDLES` 0x00000010
- #define `CC_CAPABILITY_DIRECTORY_LEASING` 0x00000020
- #define `CC_CAPABILITY_ENCRYPTION` 0x00000040

4.17.1 Detailed Description

4.17.2 Macro Definition Documentation

4.17.2.1 #define `CC_CAPABILITY_DFS` 0x00000001

DFS capability

4.17.2.2 #define `CC_CAPABILITY_LEASING` 0x00000002

leasing capability

4.17.2.3 #define `CC_CAPABILITY_LARGE_MTU` 0x00000004

multi credit operations

4.17.2.4 #define `CC_CAPABILITY_MULTI_CHANNEL` 0x00000008

multi channel capability

4.17.2.5 #define `CC_CAPABILITY_PERSISTENT_HANDLES` 0x00000010

persistent handle capability

4.17.2.6 #define `CC_CAPABILITY_DIRECTORY_LEASING` 0x00000020

directory leasing capability

4.17.2.7 #define `CC_CAPABILITY_ENCRYPTION` 0x00000040

encryption capability

4.18 signing policy

Macros

- #define [CC_SIGNING_ENABLED](#) 0x0001
- #define [CC_SIGNING_REQUIRED](#) 0x0002

4.18.1 Detailed Description

4.18.2 Macro Definition Documentation

4.18.2.1 #define CC_SIGNING_ENABLED 0x0001

signing enabled

4.18.2.2 #define CC_SIGNING_REQUIRED 0x0002

signing required

4.19 Initialization and Shutdown

Functions

- [NQ_BOOL ccInit](#) (void(*fsdNotify)([NQ_INT](#) eventId, [NQ_ULONG](#) param))
- void [ccShutdown](#) (void)
- [NQ_BOOL ccIsInitialized](#) (void)

4.19.1 Detailed Description

4.19.2 Function Documentation

4.19.2.1 [NQ_BOOL ccInit](#) (void(*)([NQ_INT](#) eventId, [NQ_ULONG](#) param) *fsdNotify*)

Initialization functions. Application should call this function at its startup to initialize NQ Client.

The *fsNotify* parameter is used for callback notification. Application can provide this function to become informed of important events inside NQ Notification. It is useful when, for instance, NQ Client is being wrapped as a local file system.

Parameters

<i>fsdNotify</i>	Pointer to the notification function. This value can be NULL.
------------------	---

Returns

None Application can examine the error code for the failure reason.

4.19.2.2 void [ccShutdown](#) (void)

This function stops NQ Client and releases all its resources.

4.19.2.3 [NQ_BOOL ccIsInitialized](#) (void)

This function checks whether NQ Client has started.

Returns

TRUE when NQ Client was initialized and FALSE otherwise.

4.20 Mount and Server

Functions

- [NQ_INT ccAddMountA](#) (const [NQ_CHAR](#) *mountPoint, const [NQ_CHAR](#) *remotePath, [NQ_BOOL](#) connect)
- [NQ_INT ccAddMount](#) (const [NQ_WCHAR](#) *mountPoint, const [NQ_WCHAR](#) *remotePath, [NQ_BOOL](#) connect)
- [NQ_INT ccAddMountWithCredsA](#) (const [NQ_CHAR](#) *mountPoint, const [NQ_CHAR](#) *remotePath, [NQ_BOOL](#) connect, const [AMCredentialsA](#) *userCredentials)
- [NQ_INT ccAddMountWithCreds](#) (const [NQ_WCHAR](#) *mountPoint, const [NQ_WCHAR](#) *remotePath, [NQ_BOOL](#) connect, const [AMCredentials](#) *userCredentials)
- [NQ_INT ccRemoveMountA](#) (const [NQ_CHAR](#) *mountPoint)
- [NQ_INT ccRemoveMount](#) (const [NQ_WCHAR](#) *mountPoint)
- [NQ_GUID ccGetServerIdA](#) (const [NQ_CHAR](#) *mountpoint)
- [NQ_GUID ccGetServerId](#) (const [NQ_WCHAR](#) *mountpoint)

4.20.1 Detailed Description

4.20.2 Function Documentation

4.20.2.1 [NQ_INT ccAddMountA](#) (const [NQ_CHAR](#) * *mountPoint*, const [NQ_CHAR](#) * *remotePath*, [NQ_BOOL](#) *connect*)

This function mounts the remote share as a local sub directory under the local virtual network file system.

Parameters

<i>mountPoint</i>	Local mount point for the share. The leading backslash is required. Example: <code>\\remoteShare1.</code>
<i>remotePath</i>	Path to the remote share, starting from host name. Path syntax is <code>\\<host> <share>.</code>
<i>connect</i>	A value of TRUE means that NQ will immediately connect to the remote share. Use FALSE to postpone connection to the first operation over this mount point.

Returns

This function returns 0 if the mount point was connected successfully or -1 otherwise. Application can examine the error code for the failure reason.

See Also

[ccRemoveMount\(\)](#)

Note

This is an ASCII version of [ccAddMount\(\)](#)

4.20.2.2 [NQ_INT ccAddMount](#) (const [NQ_WCHAR](#) * *mountPoint*, const [NQ_WCHAR](#) * *remotePath*, [NQ_BOOL](#) *connect*)

This function mounts the remote share as a local sub directory under the local virtual network file system.

Parameters

<i>mountPoint</i>	Local mount point for the share. The leading backslash is required. Example: <code>\\remoteShare1</code> .
<i>remotePath</i>	Path to the remote share, starting from host name. Path syntax is <code>\\<host> <share></code> .
<i>connect</i>	A value of TRUE means that NQ will immediately connect to the remote share. Use FALSE to postpone connection to the first operation over this mount point.

Returns

This function returns 0 if the mount point was connected successfully or -1 otherwise. Application can examine the error code for the failure reason.

See Also

[ccRemoveMount\(\)](#)

4.20.2.3 `NQ_INT ccAddMountWithCredsA (const NQ_CHAR * mountPoint, const NQ_CHAR * remotePath, NQ_BOOL connect, const AMCredentialsA * userCredentials)`

This function mounts the remote share as a local sub directory under the local virtual network file system.

Parameters

<i>mountPoint</i>	Local mount point for the share. The leading backslash is required. Example: <code>\\remoteShare1</code> .
<i>remotePath</i>	Path to the remote share, starting from host name. Path syntax is <code>\\<host> <share></code> .
<i>connect</i>	A value of TRUE means that NQ will immediately connect to the remote share. Use FALSE to postpone connection to the first operation over this mount point.
<i>userCredentials</i>	Pointer to a structure, which stores the user credentials. See AMCredentialsA structure in amapi.h file. If NULL is supplied anonymous credentials will be used. To initialize that structure use <code>amCredentialsInitA()</code> function from <code>amcredentials</code> file. In order to use anonymous credentials pass empty strings to <code>amCredentialsInitA()</code> : <code>amCredentialsInitA(credentials, "", "", "", 0)</code> . In order to use anonymous credentials with password pass empty strings and password to <code>amCredentialsInitA()</code> : <code>amCredentialsInitA(credentials, "", "", passwordString, 0)</code> . It is the caller responsibility to release that structure.

Returns

This function returns 0 if the mount point was connected successfully or -1 otherwise. Application can examine the error code for the failure reason.

See Also

[ccRemoveMount\(\)](#)

Note

This is an ASCII version of [ccAddMountWithCreds\(\)](#)

4.20.2.4 `NQ_INT ccAddMountWithCreds (const NQ_WCHAR * mountPoint, const NQ_WCHAR * remotePath, NQ_BOOL connect, const AMCredentials * userCredentials)`

This function mounts the remote share as a local sub directory under the local virtual network file system.

Parameters

<i>mountPoint</i>	Local mount point for the share. The leading backslash is required. Example: <code>\\remoteShare1.</code>
<i>remotePath</i>	Path to the remote share, starting from host name. Path syntax is <code>\\<host> <share>.</code>
<i>connect</i>	A value of TRUE means that NQ will immediately connect to the remote share. Use FALSE to postpone connection to the first operation over this mount point.
<i>userCredentials</i>	Pointer to a structure, which stores the user credentials. See AMCredentials structure in amapi.h file. If NULL is supplied anonymous credentials will be used. To initialize that structure use <code>amCredentialsInitW()</code> from <code>amcredentials</code> file. In order to use anonymous credentials pass empty strings to <code>amCredentialsInitW()</code> : <code>amCredentialsInitW(credentials, emptyWideCharString, emptyWideCharString, emptyWideCharString, 0)</code> . In order to use anonymous credentials with password pass empty strings and password to <code>amCredentialsInitW()</code> : <code>amCredentialsInitW(credentials, emptyWideCharString, emptyWideCharString, passwordString, 0)</code> . It is the caller responsibility to release that structure.

Returns

This function returns 0 if the mount point was connected successfully or -1 otherwise. Application can examine the error code for the failure reason.

See Also

[ccRemoveMount\(\)](#)

4.20.2.5 NQ_INT ccRemoveMountA (const NQ_CHAR * mountPoint)

This function makes mount point unavailable. It does not close the files opened over this mount point.

Parameters

<i>mountPoint</i>	The name of the mount point to be removed.
-------------------	--

Returns

This function returns 0 if the mount point has been successfully removed or -1 otherwise. Application can examine the error code for the failure reason.

See Also

[ccAddMount\(\)](#)

Note

This is an ASCII version of [ccRemoveMount\(\)](#)

4.20.2.6 NQ_INT ccRemoveMount (const NQ_WCHAR * mountPoint)

This function makes mount point unavailable. It does not close the files opened over this mount point.

Parameters

<i>mountPoint</i>	The name of the mount point to be removed.
-------------------	--

Returns

This function returns 0 if the mount point has been successfully removed or -1 otherwise. Application can examine the error code for the failure reason.

See Also[ccAddMount\(\)](#)**4.20.2.7 NQ_GUID ccGetServerIdA (const NQ_CHAR * *mountpoint*)**

This function returns globally unique server identifier (GUID) associated with the given mount point.

Parameters

<i>mountpoint</i>	The name of the mount point.
-------------------	------------------------------

Returns

Globally unique server identifier 16 bytes long if available, zero identifier otherwise.

Note

- Server GUID must not be used by a client as a secure method of identifying a server. Windows-based servers generate a new GUID each time they are started.
- This is an ASCII version of [ccGetServerId\(\)](#)

4.20.2.8 NQ_GUID ccGetServerId (const NQ_WCHAR * *mountpoint*)

This function returns globally unique server identifier (GUID) associated with the given mount point.

Parameters

<i>mountpoint</i>	The name of the mount point.
-------------------	------------------------------

Returns

Globally unique server identifier 16 bytes long if available, zero identifier otherwise.

Note

Server GUID must not be used by a client as a secure method of identifying a server. Windows-based servers generate a new GUID each time they are started.

4.21 Mount information

Modules

- [User logon type](#)
- [Server capabilities](#)
- [signing policy](#)

Data Structures

- struct [CCSStatistics](#)
- struct [CCServerInfo](#)
- struct [CCMountInfoA](#)
- struct [CCMountInfo](#)

Functions

- [NQ_BOOL ccGetMountInformationA](#) (const [NQ_CHAR](#) *mountPoint, [CCMountInfoA](#) *mountPointInfo)
- [NQ_BOOL ccGetMountInformation](#) (const [NQ_WCHAR](#) *mountPoint, [CCMountInfo](#) *mountPointInfo)

4.21.1 Detailed Description

4.21.2 Function Documentation

4.21.2.1 [NQ_BOOL ccGetMountInformationA](#) (const [NQ_CHAR](#) * *mountPoint*, [CCMountInfoA](#) * *mountPointInfo*)

This function is called by application to get information about specific mount point.

Parameters

<i>mountPoint</i>	The name of the mount point.
<i>mountPointInfo</i>	pointer to CCMountInfoA structure to fill.

Returns

TRUE if the information was written successfully and FALSE otherwise.

See Also

[CCMountInfoA](#) structure.

Note

This is an ASCII version of [ccGetMountInformation\(\)](#). Memory for ips (field of [CCServerInfo](#)) and serverName (field of [CCMountInfoA](#)) is allocated by this function and it is the caller responsibility to release it

4.21.2.2 [NQ_BOOL ccGetMountInformation](#) (const [NQ_WCHAR](#) * *mountPoint*, [CCMountInfo](#) * *mountPointInfo*)

This function is called by application to get information about specific mount point.

Parameters

<i>mountPoint</i>	The name of the mount point.
<i>mountPointInfo</i>	pointer to CCMountInfo structure to fill.

Returns

None.

See Also

[CCMountInfo](#) structure.

Returns

TRUE if the information was written successfully and FALSE otherwise.

Note

Memory for ips (field of [CCServerInfo](#)) and serverName (field of [CCMountInfoA](#)) is allocated by this function and it is the caller responsibility to release it

4.22 Search Operations

Data Structures

- struct [CCFindFileDataA](#)
- struct [CCFindFileData](#)

Typedefs

- typedef struct [CCFindFileDataA](#) [FindFileDataA_t](#)

Functions

- [NQ_HANDLE](#) [ccFindFirstFileA](#) (const [NQ_CHAR](#) **srchPath*, [FindFileDataA_t](#) **findFileData*, [NQ_BOOL](#) *extractFirst*)
- [NQ_HANDLE](#) [ccFindFirstFile](#) (const [NQ_WCHAR](#) **srchPath*, [CCFindFileData](#) **findFileData*, [NQ_BOOL](#) *extractFirst*)
- [NQ_BOOL](#) [ccFindNextFileA](#) ([NQ_HANDLE](#) *handle*, [FindFileDataA_t](#) **findFileData*)
- [NQ_BOOL](#) [ccFindNextFile](#) ([NQ_HANDLE](#) *handle*, [CCFindFileData](#) **findFileData*)
- [NQ_BOOL](#) [ccFindClose](#) ([NQ_HANDLE](#) *handle*)

4.22.1 Detailed Description

4.22.2 Typedef Documentation

4.22.2.1 typedef struct [CCFindFileDataA](#) [FindFileDataA_t](#)

This structure is used when calling [ccFindFirstFile\(\)](#) and [ccFindNextFile\(\)](#) calls.

4.22.3 Function Documentation

4.22.3.1 [NQ_HANDLE](#) [ccFindFirstFileA](#) (const [NQ_CHAR](#) * *srchPath*, [FindFileDataA_t](#) * *findFileData*, [NQ_BOOL](#) *extractFirst*)

This function is called by application to find a first file matching the specified search criteria. On success, this function returns a search handle which can be used by application for getting more search results ([ccFindNextFile\(\)](#)). At the end of the search the application should close this handle by calling [ccFindClose\(\)](#).

The *srchPath* argument (see below) designates either a local or a remote path. A remote path starts with a mount point name. Any other path, which does not start from a mount point is considered local. A local path, regardless of its form, designates a list of mount points.

Parameters

<i>srchPath</i>	Path and the filename to search on the remote share. Wildcard characters are applicable for the filename.
<i>findFileData</i>	Pointer to a structure, where NQ places the search result data in CCFindFileData structure.
<i>extractFirst</i>	This flag specifies whether NQ should place the search results data for this call (if TRUE), or it should only open the search handle for the specified path for future search data retrieval (if FALSE) by calling ccFindNextFile() .

Returns

This function returns NULL if it cannot create a handle or a valid handle otherwise. The application can inspect the error code for the failure reason. Error code NQ_ERR_OK means that there are no files on a remote share matching the search criteria [CCFindFileData](#) structure

See Also

[ccFindNextFile\(\)](#) [ccFindClose\(\)](#)

Note

- When application calls this function with a local path, NQ Client will lock the list of mount points until the [ccFindClose\(\)](#) will be called. During that period any other application thread trying to call a function from NQ Client API may remain locked.
- This is an ASCII version of [ccFindFirstFile\(\)](#)

4.22.3.2 NQ_HANDLE ccFindFirstFile (const NQ_WCHAR * *srchPath*, CCFindFileData * *findFileData*, NQ_BOOL *extractFirst*)

This function is called by application to find a first file matching the specified search criteria. On success, this function returns a search handle which can be used by application for getting more search results ([ccFindNextFile\(\)](#)). At the end of the search the application should close this handle by calling [ccFindClose\(\)](#).

The *srchPath* argument (see below) designates either a local or a remote path. A remote path starts with a mount point name. Any other path, which does not start from a mount point is considered local. Specifying a mount point name for example '\mnt' or '*' or '*.*' will return a list of mount points currently mounted.

Parameters

<i>srchPath</i>	Path and the filename to search on the remote share. Wildcard characters are applicable for the filename.
<i>findFileData</i>	Pointer to a structure, where NQ places the search result data in CCFindFileData structure.
<i>extractFirst</i>	This flag specifies whether NQ should place the search results data for this call (if TRUE), or it should only open the search handle for the specified path for future search data retrieval (if FALSE) by calling ccFindNextFile() .

Returns

This function returns NULL if it cannot create a handle or a valid handle otherwise. The application can inspect the error code for the failure reason. Error code NQ_ERR_OK means that there are no files on a remote share matching the search criteria.

See Also

[CCFindFileData](#) structure [ccFindNextFile\(\)](#) [ccFindClose\(\)](#)

Note

When application calls this function with a local path, NQ Client will lock the list of mount points until the [ccFindClose\(\)](#) will be called. During that period any other application thread trying to call a function from NQ Client API may remain locked.

4.22.3.3 NQ_BOOL ccFindNextFileA (NQ_HANDLE *handle*, FindFileDataA_t * *findFileData*)

This function is called by application to find a next file matching the specified wildcard.

Parameters

<i>handle</i>	Handle value returned by calling ccFindFirstFile() .
<i>findFileData</i>	Pointer to a structure, where NQ places the search result data in CCFindFileData structure.

Returns

This function returns TRUE if the next file was found successfully or FALSE otherwise. The application can inspect the error code for the failure reason. Error code NQ_ERR_OK means that there are no more files on a remote share matching the search criteria.

See Also

[CCFindFileData](#) structure [ccFindFirstFile\(\)](#) [ccFindClose\(\)](#)

Note

This is an ASCII version of [ccFindNextFile\(\)](#)

4.22.3.4 NQ_BOOL ccFindNextFile (NQ_HANDLE handle, CCFindFileData * findFileData)

This function is called by application to find a next file matching the specified wildcard.

Parameters

<i>handle</i>	Handle value returned by calling ccFindFirstFile() .
<i>findFileData</i>	Pointer to a structure, where NQ places the search result data in CCFindFileData structure.

Returns

This function returns TRUE if the next file was found successfully or FALSE otherwise. The application can inspect the error code for the failure reason. Error code NQ_ERR_OK means that there are no more files on a remote share matching the search criteria.

See Also

[CCFindFileData](#) structure [ccFindFirstFile\(\)](#) [ccFindClose\(\)](#)

4.22.3.5 NQ_BOOL ccFindClose (NQ_HANDLE handle)

This function is called by application to close the search handle

Parameters

<i>handle</i>	Handle value returned by calling the ccFindFirstFile()
---------------	--

Returns

This function returns TRUE if the search handle is closed successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.23 Directory Operations

Functions

- [NQ_BOOL ccCreateDirectoryA](#) (const [NQ_CHAR](#) *pathName)
- [NQ_BOOL ccCreateDirectory](#) (const [NQ_WCHAR](#) *pathName)
- [NQ_BOOL ccRemoveDirectoryA](#) (const [NQ_CHAR](#) *pathName)
- [NQ_BOOL ccRemoveDirectory](#) (const [NQ_WCHAR](#) *pathName)

4.23.1 Detailed Description

4.23.2 Function Documentation

4.23.2.1 [NQ_BOOL ccCreateDirectoryA](#) (const [NQ_CHAR](#) * *pathName*)

This function is called by application to create a directory.

Parameters

<i>pathName</i>	Path of the directory to be created
-----------------	-------------------------------------

Returns

This function returns TRUE if directory is created successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccCreateDirectory\(\)](#)

4.23.2.2 [NQ_BOOL ccCreateDirectory](#) (const [NQ_WCHAR](#) * *pathName*)

This function is called by application to create a directory.

Parameters

<i>pathName</i>	Path of the directory to be created
-----------------	-------------------------------------

Returns

This function returns TRUE if directory is created successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.23.2.3 [NQ_BOOL ccRemoveDirectoryA](#) (const [NQ_CHAR](#) * *pathName*)

This function is called by application to remove a directory.

Parameters

<i>pathName</i>	Path of the directory to be removed.
-----------------	--------------------------------------

Returns

This function returns TRUE if the specified directory is removed successfully or FALSE otherwise. The application can inspect the error code for the failure reason. The directory should be empty, otherwise error condition will occur.

Note

This is an ASCII version of [ccRemoveDirectory\(\)](#)

4.23.2.4 NQ_BOOL ccRemoveDirectory (const NQ_WCHAR * *pathName*)

This function is called by application to remove a directory.

Parameters

<i>pathName</i>	Path of the directory to be removed.
-----------------	--------------------------------------

Returns

This function returns TRUE if the specified directory is removed successfully or FALSE otherwise. The application can inspect the error code for the failure reason. The directory should be empty, otherwise error condition will occur.

4.24 File Operations

Modules

- [File attributes](#)
- [File access mode](#)
- [File share mode](#)
- [File locality](#)
- [File create action](#)
- [File open action](#)
- [File seek constants](#)

Typedefs

- typedef void(* [CCServerSideCopyCallback](#))(NQ_STATUS status, NQ_UINT32 totalBytesWritten, NQ_UINT64 totalFileSize, void *context)

Functions

- [NQ_HANDLE ccCreateFileA](#) (const NQ_CHAR *fileName, NQ_INT access, NQ_INT shareMode, NQ_INT locality, NQ_BOOL writeThrough, NQ_UINT16 attributes, NQ_INT createAction, NQ_INT openAction)
- [NQ_HANDLE ccCreateFile](#) (const NQ_WCHAR *fileName, NQ_INT access, NQ_INT shareMode, NQ_INT locality, NQ_BOOL writeThrough, NQ_UINT16 attributes, NQ_INT createAction, NQ_INT openAction)
- [NQ_BOOL ccDeleteFileA](#) (const NQ_CHAR *fileName)
- [NQ_BOOL ccDeleteFile](#) (const NQ_WCHAR *fileName)
- [NQ_BOOL ccMoveFileA](#) (const NQ_CHAR *oldFilename, const NQ_CHAR *newFilename)
- [NQ_BOOL ccMoveFile](#) (const NQ_WCHAR *oldFilename, const NQ_WCHAR *newFilename)
- [NQ_BOOL ccReadFile](#) (NQ_HANDLE hndl, NQ_IOBufPos buffer, NQ_UINT count, NQ_UINT *readSize)
- [NQ_BOOL ccReadFileAsync](#) (NQ_HANDLE hndl, NQ_IOBufPos buffer, NQ_UINT count, void *context, void(*callback)(NQ_STATUS, NQ_UINT, void *))
- [NQ_BOOL ccWriteFile](#) (NQ_HANDLE hndl, NQ_IOBufPos buffer, NQ_UINT count, NQ_UINT *writtenSize)
- [NQ_BOOL ccWriteFileAsync](#) (NQ_HANDLE hndl, NQ_IOBufPos buffer, NQ_UINT count, void *context, void(*callback)(NQ_STATUS, NQ_UINT, void *))
- [NQ_BOOL ccFlushFile](#) (NQ_HANDLE handle)
- [NQ_BOOL ccCloseHandle](#) (NQ_HANDLE handle)

4.24.1 Detailed Description

4.24.2 Typedef Documentation

- 4.24.2.1 typedef void(* [CCServerSideCopyCallback](#))(NQ_STATUS status, NQ_UINT32 totalBytesWritten, NQ_UINT64 totalFileSize, void *context)

Prototype for a callback function for server side copy chunk response. This function is called after each response from the server arrives.

Parameters

<i>status</i>	The response status
<i>totalBytesWritten</i>	Total bytes that was written in the file

<i>totalFileSize</i>	Total file size
<i>context</i>	A context pointer supplied by the application. By using context, an application can distinguish between different copy chunk operations.

4.24.3 Function Documentation

4.24.3.1 **NQ_HANDLE** ccCreateFileA (const **NQ_CHAR** * *fileName*, **NQ_INT** *access*, **NQ_INT** *shareMode*, **NQ_INT** *locality*, **NQ_BOOL** *writeThrough*, **NQ_UINT16** *attributes*, **NQ_INT** *createAction*, **NQ_INT** *openAction*)

This function is called by application to create or open a file.

Parameters

<i>fileName</i>	Path of the file to be created/opened.
<i>access</i>	Desired file access mode (see File access mode module).
<i>shareMode</i>	Desired file share mode (see File share mode module).
<i>locality</i>	Desired file locality (see File locality module).
<i>writeThrough</i>	If TRUE then no read ahead or write behind allowed on this file or device. When the response is returned, data is expected to be on the disk or device.
<i>attributes</i>	File attributes for a newly created file (see File attributes module)
<i>createAction</i>	Desired file create action (see File create action module).
<i>openAction</i>	Desired file open action (see File open action module).

Returns

This function returns NULL if it cannot create/open a file or a valid handle otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccCreateFile\(\)](#)

4.24.3.2 **NQ_HANDLE** ccCreateFile (const **NQ_WCHAR** * *fileName*, **NQ_INT** *access*, **NQ_INT** *shareMode*, **NQ_INT** *locality*, **NQ_BOOL** *writeThrough*, **NQ_UINT16** *attributes*, **NQ_INT** *createAction*, **NQ_INT** *openAction*)

This function is called by application to create or open a file.

Parameters

<i>fileName</i>	Path of the file to be created/opened.
<i>access</i>	Desired file access mode (see File access mode module: FILE_AM_READ , FILE_AM_SPECIAL_MASK , FILE_AM_READ_WRITE , FILE_AM_WRITE).
<i>shareMode</i>	Desired file share mode (see File share mode module: FILE_SM_COMPAT , FILE_SM_DENY_NONE , FILE_SM_DENY_READ , FILE_SM_DENY_WRITE , FILE_SM_EXCLUSIVE).
<i>locality</i>	Desired file locality (see File locality module: FILE_LCL_MIXED , FILE_LCL_RANDOM , FILE_LCL_SEQUENTIAL , FILE_LCL_UNKNOWN).
<i>writeThrough</i>	If TRUE then no read ahead or write behind allowed on this file or device. When the response is returned, data is expected to be on the disk or device.
<i>attributes</i>	File attributes for a newly created file (see File attributes module: CIFS_ATTR_ALL , CIFS_ATTR_ARCHIVE , CIFS_ATTR_DIR , CIFS_ATTR_HIDDEN , CIFS_ATTR_READONLY , CIFS_ATTR_SYSTEM , CIFS_ATTR_VOLUME).

<i>createAction</i>	Desired file create action, can be one of the following values: FILE_CA_CREATE , FILE_CA_FAIL .
<i>openAction</i>	Desired file open action, can be one of the following values: FILE_OA_FAIL , FILE_OA_OPEN , FILE_OA_TRUNCATE .

Returns

This function returns NULL if it cannot create/open a file or a valid handle otherwise. The application can inspect the error code for the failure reason.

4.24.3.3 **NQ_BOOL** ccDeleteFileA (const **NQ_CHAR** * *fileName*)

This function is called by application to delete a file.

Parameters

<i>fileName</i>	Path of the directory to be created
-----------------	-------------------------------------

Returns

This function returns TRUE if the specified file is deleted successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccDeleteFile\(\)](#)

4.24.3.4 **NQ_BOOL** ccDeleteFile (const **NQ_WCHAR** * *fileName*)

This function is called by application to delete a file.

Parameters

<i>fileName</i>	Path of the directory to be created
-----------------	-------------------------------------

Returns

This function returns TRUE if the specified file is deleted successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.24.3.5 **NQ_BOOL** ccMoveFileA (const **NQ_CHAR** * *oldFilename*, const **NQ_CHAR** * *newFilename*)

This function is called by application to move (rename) a file/directory to a different location. note, that a file cannot be moved between different remote shares.

Parameters

<i>oldFilename</i>	Path of the file/directory to be moved.
<i>newFilename</i>	Path of the file/directory to be moved to.

Returns

This function returns TRUE if the specified file is moved successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccMoveFile\(\)](#)

4.24.3.6 **NQ_BOOL** ccMoveFile (**const NQ_WCHAR *** *oldFilename*, **const NQ_WCHAR *** *newFilename*)

This function is called by application to move (rename) a file/directory to a different location. note, that a file cannot be moved between different remote shares.

Parameters

<i>oldFilename</i>	Path of the file/directory to be moved.
<i>newFilename</i>	Path of the file/directory to be moved to.

Returns

This function returns TRUE if the specified file is moved successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.24.3.7 NQ_BOOL ccReadFile (NQ_HANDLE hndl, NQ_IOBufPos buffer, NQ_UINT count, NQ_UINT * readSize)

This function is called by application to read data from the opened file.

NQ attempts to use the biggest available payload. However, if the *count* parameter is big enough, calling this function results in transmitting multiple SMB requests. NQ Client attempts to send these requests concurrently. Since this call is synchronous, NQ will wait for all responses to come back from the server.

NQ Client reads file bytes from its current position. For random access, use function [ccSetFilePointer\(\)](#).

Parameters

<i>hndl</i>	Handle value returned by calling ccCreateFile()
<i>buffer</i>	Pointer to a buffer to read the file data to
<i>count</i>	Number of bytes to read from the file
<i>readSize</i>	The pointer to a variable which on exit receives the number of bytes actually read. This value can be NULL.

Returns

This function returns TRUE if the data is read successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.24.3.8 NQ_BOOL ccReadFileAsync (NQ_HANDLE hndl, NQ_IOBufPos buffer, NQ_UINT count, void * context, void (*)(NQ_STATUS, NQ_UINT, void *) callback)

This function is called by application to read data from the opened file.

NQ attempts to use the biggest available payload. However, if the *count* parameter is big enough, calling this function results in transmitting multiple SMB requests. NQ Client attempts to send these requests concurrently. Since this call is asynchronous, NQ will not wait for all responses to come back from the server but will rather call the *callback* function after receiving the last response.

NQ places the actual number of data read into the first argument of the call to the *callback* function.

NQ Client reads bytes starting from the current position in the file. For random read use function [ccSetFilePointer\(\)](#).

This function is not thread-protected over the same file. It assumes that there is just one thread reading file through the given handle. Otherwise, the result is undefined. It is possible, however, to read the same file over two different handles simultaneously.

Parameters

<i>hndl</i>	Handle value returned by calling ccCreateFile() .
<i>buffer</i>	Pointer to a buffer to use for reading.
<i>count</i>	Number of bytes to write to the file.

<i>context</i>	A context pointer supplied by the application. By using context, an application can distinguish between different write operations. For casting purposes first structure field must be of type NQ_BOOL representing whether response is postponed by the server, used internally by NQ Client.
<i>callback</i>	Pointer to a callback function supplied by application. This function accepts operation status, the actual number of bytes read and an abstract context pointer supplied by the application.

Returns

This function returns TRUE if the read operations were successfully queued or FALSE otherwise.

The application can inspect the error code for the failure reason.

Because of the asynchronous character of this operation the value does not reflect read results.

Application should use *callback* to analyze read results.

4.24.3.9 NQ_BOOL ccWriteFile (NQ_HANDLE hndl, NQ_IOBufPos buffer, NQ_UINT count, NQ_UINT * writtenSize)

This function is called by application to write data to an open file.

Upon successful completion NQ places the actual number of data written into the memory location pointed by parameter writtenSize. On an error, this value is left unmodified.

NQ Client writes bytes starting from the current position in the file. For random write use function [ccSetFilePointer\(\)](#).

The [ccWriteFile\(\)](#) function can be also called for file truncation. Use [ccSetFilePointer\(\)](#) function first to set file pointer into the desired end-of-file position. Then call the current function with zero data size. This will truncate the file.

Parameters

<i>hndl</i>	Handle value returned by calling ccCreateFile() .
<i>buffer</i>	Pointer to a buffer with bytes to be written.
<i>count</i>	Number of bytes to write to the file.
<i>writtenSize</i>	Pointer to a variable which will receive the number of bytes actually written. This value can be NULL.

Returns

This function returns TRUE if the data is written successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.24.3.10 NQ_BOOL ccWriteFileAsync (NQ_HANDLE hndl, NQ_IOBufPos buffer, NQ_UINT count, void * context, void (*)(NQ_STATUS, NQ_UINT, void *) callback)

This function is called by application to write data to an open file using asynchronous operations.

Upon successful completion NQ schedules necessary write operations. When all writes will be complete, NQ will call the *callback* function.

When calling the *callback* function, NQ places the actual number of data written into the first argument of this call.

NQ Client writes bytes starting from the current position in the file. For random write use function [ccSetFilePointer\(\)](#).

The [ccWriteFile\(\)](#) function can be also called for file truncation. Use [ccSetFilePointer\(\)](#) function first to set file pointer into the desired end-of-file position. Then call the current function with zero data size. This will truncate the file.

This function is not thread-protected over the same file. It assumes that there is just one thread writing to the file through the given handle. Otherwise, the result is undefined. It is possible, however, to write the same file over two different handles simultaneously. The permission to open the same file twice for read depends on the server's policies.

Parameters

<i>hdl</i>	Handle value returned by calling ccCreateFile() .
<i>buffer</i>	Pointer to a buffer with bytes to be written.
<i>count</i>	Number of bytes to write to the file.
<i>context</i>	A context pointer supplied by the application. By using context, an application can distinguish between different write operations.
<i>callback</i>	Pointer to a callback function supplied by application. This function accepts the actual number of bytes written and an abstract context pointer supplied by the application. For casting purposes first structure field must be of type NQ_BOOL representing whether response is postponed by the server, used internally by NQ Client.

Returns

This function returns TRUE if the write operations where successfully queued or FALSE otherwise. The application can inspect the error code for the failure reason.

Because of the asynchronous character of this operation the value does not reflect write results. Application should use *callback* to analyze write results.

4.24.3.11 NQ_BOOL ccFlushFile (NQ_HANDLE handle)

This function is called by application to force server to synchronize its local buffers with the file contents.

Parameters

<i>handle</i>	Handle value returned by calling ccCreateFile()
---------------	---

Returns

This function returns TRUE if the next file was found successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.24.3.12 NQ_BOOL ccCloseHandle (NQ_HANDLE handle)

This function is called by application to close the file handle

Returns

This function returns TRUE if the file handle is closed successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

Parameters

<i>handle</i>	Handle value returned by calling ccCreateFile()
---------------	---

4.25 File Info Operations

Data Structures

- struct [CCFileTime](#)
- struct [CCFileInfo_t](#)
- struct [CCFileInfo](#)
- struct [CCVolumeInfo](#)

Typedefs

- typedef struct [CCFileInfo_t](#) [FileInfo_t](#)

Functions

- [NQ_BOOL](#) [ccGetFilePointer](#) ([NQ_HANDLE](#) handle, [NQ_UINT64](#) *offset)
- [NQ_BOOL](#) [ccSetFilePointer](#) ([NQ_HANDLE](#) handle, [NQ_INT64](#) *offset, [NQ_INT](#) method)
- [NQ_BOOL](#) [ccServerSideDataCopyA](#) ([NQ_CHAR](#) *srcPath, [NQ_CHAR](#) *dstPath, void *context, [CCServerSideCopyCallback](#) *callback)
- [NQ_BOOL](#) [ccServerSideDataCopy](#) ([NQ_WCHAR](#) *srcPath, [NQ_WCHAR](#) *dstPath, void *context, [CCServerSideCopyCallback](#) *callback)
- [NQ_BOOL](#) [ccGetDiskFreeSpaceA](#) (const [NQ_CHAR](#) *pathName, [NQ_UINT](#) *sectorsPerCluster, [NQ_UINT](#) *bytesPerSector, [NQ_UINT](#) *freeClusters, [NQ_UINT](#) *totalClusters, [NQ_UINT](#) *fsType, [NQ_UINT](#) *serialNumber)
- [NQ_BOOL](#) [ccGetDiskFreeSpace](#) (const [NQ_WCHAR](#) *pathName, [NQ_UINT](#) *sectorsPerCluster, [NQ_UINT](#) *bytesPerSector, [NQ_UINT](#) *freeClusters, [NQ_UINT](#) *totalClusters, [NQ_UINT](#) *fsType, [NQ_UINT](#) *serialNumber)
- [NQ_BOOL](#) [ccGetDiskFreeSpaceExA](#) (const [NQ_CHAR](#) *pathName, [NQ_UINT](#) *sectorsPerCluster, [NQ_UINT](#) *bytesPerSector, [NQ_UINT64](#) *freeClusters, [NQ_UINT64](#) *totalClusters, [NQ_UINT](#) *fsType, [NQ_UINT](#) *serialNumber)
- [NQ_BOOL](#) [ccGetDiskFreeSpaceEx](#) (const [NQ_WCHAR](#) *pathName, [CCVolumeInfo](#) *volumeInfo)
- [NQ_UINT32](#) [ccGetFileAttributesA](#) (const [NQ_CHAR](#) *fileName)
- [NQ_BOOL](#) [ccGetFileAttributes](#) (const [NQ_WCHAR](#) *fileName, [NQ_UINT32](#) *attributes)
- [NQ_BOOL](#) [ccSetFileAttributesA](#) (const [NQ_CHAR](#) *fileName, [NQ_UINT32](#) attributes)
- [NQ_BOOL](#) [ccSetFileAttributes](#) (const [NQ_WCHAR](#) *fileName, [NQ_UINT32](#) attributes)
- [NQ_BOOL](#) [ccGetFileInformationByNameA](#) (const [NQ_CHAR](#) *fileName, [FileInfo_t](#) *fileInfo)
- [NQ_BOOL](#) [ccGetFileInformationByName](#) (const [NQ_WCHAR](#) *fileName, [CCFileInfo](#) *fileInfo)
- [NQ_BOOL](#) [ccSetFileSizeByNameA](#) (const [NQ_CHAR](#) *fileName, [NQ_UINT32](#) sizeLow, [NQ_UINT32](#) sizeHigh)
- [NQ_BOOL](#) [ccSetFileSizeByName](#) (const [NQ_WCHAR](#) *fileName, [NQ_UINT64](#) size)
- [NQ_BOOL](#) [ccSetFileSizeByHandle](#) ([NQ_HANDLE](#) handle, [NQ_UINT64](#) size)
- [NQ_BOOL](#) [ccGetFileTime](#) ([NQ_HANDLE](#) handle, [CCFileTime](#) *pCreation, [CCFileTime](#) *pLastAccess, [CCFileTime](#) *pLastWrite)
- [NQ_BOOL](#) [ccSetFileTime](#) ([NQ_HANDLE](#) handle, [CCFileTime](#) *pCreation, [CCFileTime](#) *pLastAccess, [CCFileTime](#) *pLastWrite)
- [NQ_BOOL](#) [ccGetFileSize](#) ([NQ_HANDLE](#) handle, [NQ_UINT64](#) *fileSize)
- [NQ_BOOL](#) [ccGetFileInformationByHandle](#) ([NQ_HANDLE](#) handle, [CCFileInfo](#) *fileInfo)

4.25.1 Detailed Description

4.25.2 Typedef Documentation

4.25.2.1 typedef struct CCFileInfo_t FileInfo_t

This structure is used when calling File Information related CIFS Client API calls, it is used by [ccGetFileInformation-ByNameA\(\)](#) for backward compatibility.

4.25.3 Function Documentation

4.25.3.1 NQ_BOOL ccGetFilePointer (NQ_HANDLE handle, NQ_UINT64 * offset)

This function is called by application to withdraw the current position in the file. The result is returned as a 64-bit number.

Parameters

<i>handle</i>	Handle value returned by calling ccCreateFile() .
<i>offset</i>	Pointer to 64 bit value of the file offset.

Returns

This function returns TRUE if the specified file information was retrieved successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.25.3.2 NQ_BOOL ccSetFilePointer (NQ_HANDLE handle, NQ_INT64 * offset, NQ_INT method)

This function is called by application to set a file pointer of the file read/write location in the opened file.

Parameters

<i>handle</i>	Handle value returned by calling ccCreateFile() .
<i>offset</i>	Pointer to 64 bit value of the file offset. Upon successful return this variable contains new offset.
<i>method</i>	File seek method (see SEEK_FILE_BEGIN , SEEK_FILE_CURRENT and SEEK_FILE_END Macros in File seek constants module)

Returns

This function returns TRUE if the specified file information was set successfully or FALSE otherwise. The application can inspect the error code for the failure reason. Upon successful return offset variable contains new offset. Use [cmS64Assign\(\)](#) for initializing offset parameter.

4.25.3.3 NQ_BOOL ccServerSideDataCopyA (NQ_CHAR * srcPath, NQ_CHAR * dstPath, void * context, CCServerSideCopyCallback * callBack)

This function is called by application to perform server side copy to a new location. The source and the destination files must be on the same server. Some servers do not support server-side data copy operation such as SMB1 servers, old versions of SAMBA and NetApp. In this case the error code that will be set is NQ_ERR_NOTSUPPORTED. Some servers do not support server-side data copy across different shares such as old versions of SAMBA and lower or equal to SMB 3.0.0 versions of MAC. In this case the error code that will be set is NQ_ERR_NOTSUPPORTED.

Parameters

<i>srcPath</i>	Path to the source file.
<i>dstPath</i>	Path to the destination file.
<i>context</i>	A pointer to a context for the callBack function, if needed.
<i>callBack</i>	User defined method that will be called at a completion of an asynchronous server-side data copy operation, can be null. Currently not in use.

Returns

This function returns TRUE if file was copy successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccServerSideDataCopy\(\)](#)

4.25.3.4 NQ_BOOL ccServerSideDataCopy (NQ_WCHAR * *srcPath*, NQ_WCHAR * *dstPath*, void * *context*, CCServerSideCopyCallback * *callBack*)

This function is called by application to perform server side copy to a new location. The source and the destination files must be on the same server. Some servers do not support server-side data copy operation such as SMB1 servers, old versions of SAMBA and NetApp. In this case the error code that will be set is NQ_ERR_NOTSUPPORTED. Some servers do not support server-side data copy across different shares such as old versions of SAMBA and lower or equal to SMB 3.0.0 versions of MAC. In this case the error code that will be set is NQ_ERR_NOTSUPPORTED.

Parameters

<i>srcPath</i>	Path to the source file.
<i>dstPath</i>	Path to the destination file.
<i>context</i>	A pointer to a context for the callBack function, if needed.
<i>callBack</i>	User defined method that will be called at a completion of an asynchronous server-side data copy operation, can be null. Currently not in use.

Returns

This function returns TRUE if file was copy successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.25.3.5 NQ_BOOL ccGetDiskFreeSpaceA (const NQ_CHAR * *pathName*, NQ_UINT * *sectorsPerCluster*, NQ_UINT * *bytesPerSector*, NQ_UINT * *freeClusters*, NQ_UINT * *totalClusters*, NQ_UINT * *fsType*, NQ_UINT * *serialNumber*)

This function is called by application to get the free disk space information for the specified remote share.

Parameters

<i>pathName</i>	Path of any object on the remote share the free disk space information is requested for.
<i>sectorsPerCluster</i>	Pointer to a variable which will receive the number of sectors per cluster on the remote share.
<i>bytesPerSector</i>	Pointer to a variable which will receive the number of bytes per sector on the remote share.

<i>freeClusters</i>	Pointer to a variable which will receive the number of free clusters on the remote share.
<i>totalClusters</i>	Pointer to a variable which will receive the number of total clusters on the remote share.
<i>fsType</i>	Pointer to a variable which will receive the type of the file system.
<i>serialNumber</i>	Pointer to a variable which will receive the serial number of the file system

Returns

This function returns TRUE if the remote disk information is received successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccGetDiskFreeSpace\(\)](#)

4.25.3.6 `NQ_BOOL ccGetDiskFreeSpace (const NQ_WCHAR * pathName, NQ_UINT * sectorsPerCluster, NQ_UINT * bytesPerSector, NQ_UINT * freeClusters, NQ_UINT * totalClusters, NQ_UINT * fsType, NQ_UINT * serialNumber)`

This function is called by application to get the free disk space information for the specified remote share.

Parameters

<i>pathName</i>	Path of any object on the remote share the free disk space information is requested for.
<i>sectorsPerCluster</i>	Pointer to a variable which will receive the number of sectors per cluster on the remote share.
<i>bytesPerSector</i>	Pointer to a variable which will receive the number of bytes per sector on the remote share.
<i>freeClusters</i>	Pointer to a variable which will receive the number of free clusters on the remote share.
<i>totalClusters</i>	Pointer to a variable which will receive the number of total clusters on the remote share.
<i>fsType</i>	Pointer to a variable which will receive the type of the file system.
<i>serialNumber</i>	Pointer to a variable which will receive the serial number of the file system

Returns

This function returns TRUE if the remote disk information is received successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.25.3.7 `NQ_BOOL ccGetDiskFreeSpaceExA (const NQ_CHAR * pathName, NQ_UINT * sectorsPerCluster, NQ_UINT * bytesPerSector, NQ_UINT64 * freeClusters, NQ_UINT64 * totalClusters, NQ_UINT * fsType, NQ_UINT * serialNumber)`

Extended version of [ccGetDiskFreeSpace\(\)](#). When disk is larger than 4 giga clusters - only the extended version gives the correct result.

Parameters

<i>pathName</i>	Path of any object on the remote share the free disk space information is requested for.
<i>sectorsPerCluster</i>	Pointer to a variable which will receive the number of sectors per cluster on the remote share.
<i>bytesPerSector</i>	Pointer to a variable which will receive the number of bytes per sector on the remote share.
<i>freeClusters</i>	Pointer to a variable which will receive the number of free clusters on the remote share.
<i>totalClusters</i>	Pointer to a variable which will receive the number of total clusters on the remote share.

<i>fsType</i>	Pointer to a variable which will receive the type of the file system.
<i>serialNumber</i>	Pointer to a variable which will receive the serial number of the file system

Returns

This function returns TRUE if the remote disk information is received successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.25.3.8 NQ_BOOL ccGetDiskFreeSpaceEx (const NQ_WCHAR * *pathName*, CCVolumeInfo * *volumeInfo*)

Extended version of [ccGetDiskFreeSpace\(\)](#). When disk is larger than 4 giga clusters - only the extended version gives the correct result.

Parameters

<i>pathName</i>	Path of any object on the remote share the free disk space information is requested for.
<i>volumeInfo</i>	Pointer to a variable which will receive the volume information, see CCVolumeInfo .

Returns

This function returns TRUE if the remote disk information is received successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.25.3.9 NQ_UINT32 ccGetFileAttributesA (const NQ_CHAR * *fileName*)

This function is called by application to get attributes of the specified file.

Parameters

<i>fileName</i>	Path to the file.
-----------------	-------------------

Returns

This function returns file attributes if successful (see [File attributes](#) module) or NQ_ERR_ATTRERROR in case of failure. The application can inspect the error code for the failure reason.

4.25.3.10 NQ_BOOL ccGetFileAttributes (const NQ_WCHAR * *fileName*, NQ_UINT32 * *attributes*)

This function is called by application to get attributes of the specified file.

Parameters

<i>fileName</i>	Path to the file.
<i>attributes</i>	Pointer to attributes value.

Returns

This function returns TRUE if successful and the file attributes in attributes parameter (see [File attributes](#) module) otherwise FALSE. The application can inspect the error code for the failure reason.

4.25.3.11 NQ_BOOL ccSetFileAttributesA (const NQ_CHAR * *fileName*, NQ_UINT32 *attributes*)

This function is called by application to set attributes of the specified file.

Parameters

<i>fileName</i>	Path to the file.
<i>attributes</i>	Attributes to set. See File attributes module for values.

Returns

This function returns TRUE if the specified attributes are set successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccSetFileAttributes\(\)](#)

4.25.3.12 NQ_BOOL ccSetFileAttributes (const NQ_WCHAR * fileName, NQ_UINT32 attributes)

This function is called by application to set attributes of the specified file.

Parameters

<i>fileName</i>	Path to the file.
<i>attributes</i>	Attributes to set. See File attributes module for values.

Returns

This function returns TRUE if the specified attributes are set successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[File attributes](#) module

4.25.3.13 NQ_BOOL ccGetFileInformationByNameA (const NQ_CHAR * fileName, FileInfo_t * fileInfo)

This function is called by application to get the information of the specified file.

Parameters

<i>fileName</i>	Path to the file.
<i>fileInfo</i>	Pointer to a structure which will receive the file information (see CCFileInfo Structure).

Returns

This function returns TRUE if the specified file information is obtained successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccGetFileInformationByName\(\)](#)

4.25.3.14 NQ_BOOL ccGetFileInformationByName (const NQ_WCHAR * fileName, CCFileInfo * fileInfo)

This function is called by application to get the information of the specified file.

Parameters

<i>fileName</i>	Path to the file.
<i>fileInfo</i>	Pointer to a structure which will receive the file information (see CCFileInfo Structure).

Returns

This function returns TRUE if the specified file information is obtained successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.25.3.15 **NQ_BOOL** ccSetFileSizeByNameA (const **NQ_CHAR** * *fileName*, **NQ_UINT32** *sizeLow*, **NQ_UINT32** *sizeHigh*)

This function is called by application to set file size when file is specified by its name.

Parameters

<i>fileName</i>	Path to file.
<i>sizeLow</i>	Low 32 bit desired file size.
<i>sizeHigh</i>	High 32 bit desired file size.

Returns

This function returns TRUE if the file size was modified successfully or FALSE otherwise.

When the specified file size is less than the current file size, this operation truncates the file to the specified size.

4.25.3.16 **NQ_BOOL** ccSetFileSizeByName (const **NQ_WCHAR** * *fileName*, **NQ_UINT64** *size*)

This function is called by application to set file size when file is specified by its name.

Parameters

<i>fileName</i>	Path to file.
<i>size</i>	64 bit desired file size.

Returns

This function returns TRUE if the file size was modified successfully or FALSE otherwise.

When the specified file size is less than the current file size, this operation truncates the file to the specified size.

4.25.3.17 **NQ_BOOL** ccSetFileSizeByHandle (**NQ_HANDLE** *handle*, **NQ_UINT64** *size*)

This function is called by application to apply size to an open file. When the specified file size is less than the current file size, this operation truncates the file to the specified size.

Parameters

<i>handle</i>	Handle value returned by calling ccCreateFile() .
<i>size</i>	64 bit desired file size.

Returns

This function returns TRUE if the file size was modified successfully or FALSE otherwise.

4.25.3.18 NQ_BOOL ccGetFileTime (NQ_HANDLE *handle*, CCFileTime * *pCreation*, CCFileTime * *pLastAccess*, CCFileTime * *pLastWrite*)

This function is called by application to get the time information of the specified file.

Parameters

<i>handle</i>	Handle value returned by calling ccCreateFile()
<i>pCreation</i>	Pointer to a buffer for file creation time. This buffer should hold the time structure (see CC-FileTime Structure)
<i>pLastAccess</i>	Pointer to a buffer for file last access time. This buffer should hold the time structure (see CCFileTime Structure)
<i>pLastWrite</i>	Pointer to a buffer for file last write time. This buffer should hold the time structure (see CCFileTime Structure)

Returns

This function returns TRUE if the specified file time information is obtained successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.25.3.19 NQ_BOOL ccSetFileTime (NQ_HANDLE handle, CCFileTime * pCreation, CCFileTime * pLastAccess, CCFileTime * pLastWrite)

This function is called by application to set the time information of the specified file.

NQ Client sends time modification request using the provided time values but the server interpretation of this request may apply additional calculations, so that the resulted file times may differ. Null can be passed when no change is required in one of the times.

Parameters

<i>handle</i>	Handle value returned by calling ccCreateFile() .
<i>pCreation</i>	Pointer to a structure holding the file creation time (see CCFileTime Structure)
<i>pLastAccess</i>	Pointer to a structure holding the file last access time (see CCFileTime Structure)
<i>pLastWrite</i>	Pointer to a structure holding the file last write time (see CCFileTime Structure)

Returns

This function returns TRUE if the specified file time information is set successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.25.3.20 NQ_BOOL ccGetFileSize (NQ_HANDLE handle, NQ_UINT64 * fileSize)

This function is called by application to get the size of the specified file.

Parameters

<i>handle</i>	Handle value returned by calling ccCreateFile()
<i>fileSize</i>	Pointer to a buffer for the NQ_UINT64 value

Returns

This function returns TRUE if the specified file information is obtained successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.25.3.21 NQ_BOOL ccGetFileInformationByHandle (NQ_HANDLE handle, CCFileInfo * fileInfo)

This function is called by application to get the information of an open file

Parameters

<i>handle</i>	Handle value returned by calling ccCreateFile()
<i>fileInfo</i>	Pointer to a structure holding the file information (see CCFileInfo Structure)

Returns

This function returns TRUE if the specified file information is obtained successfully or FALSE otherwise. The application can inspect the error code for the failure reason.

4.26 Timeout Operations

Functions

- void [ccConfigSetTimeout](#) (NQ_UINT32 secs)
- NQ_UINT32 [ccConfigGetTimeout](#) (void)
- NQ_BOOL [ccSetGetBackupListTimeout](#) (NQ_UINT32 seconds)

4.26.1 Detailed Description

4.26.2 Function Documentation

4.26.2.1 void [ccConfigSetTimeout](#) (NQ_UINT32 secs)

Set SMB timeout for waiting a response.

Parameters

<i>secs</i>	Number of seconds to wait for an SMB response. The default value is 15 seconds.
-------------	---

4.26.2.2 NQ_UINT32 [ccConfigGetTimeout](#) (void)

Get the current value of SMB timeout.

Returns

Number of seconds to wait for an SMB response. The default value is 15 seconds.

4.26.2.3 NQ_BOOL [ccSetGetBackupListTimeout](#) (NQ_UINT32 seconds)

This function allows to modify the default timeout value for the GetBackupList operation.

This operation is sent as a broadcast. NQ issues this operation when starting network browsing.

The default value is defined in the UD_CC_CLIENTRESPONSETIMEOUT parameter.

Parameters

<i>seconds</i>	Timeout in seconds.
----------------	---------------------

See Also

[ccNetworkEnumerateDomains\(\)](#).

Returns

This function returns TRUE if seconds is valid parameter or FALSE otherwise.

4.27 Network Browsing

Data Structures

- struct [CCNetShareItemA](#)
- struct [CCNetShareItem](#)
- struct [CCNetServerItem](#)
- struct [CCNetDomainItem](#)

Functions

- [NQ_BOOL](#) [ccNetworkSetDefaultDomainA](#) (const [NQ_CHAR](#) *domain)
- [NQ_BOOL](#) [ccNetworkSetDefaultDomain](#) (const [NQ_WCHAR](#) *domain)
- void [ccNetworkCacheSet](#) ([NQ_UINT32](#) timeout)
- const [NQ_CHAR](#) * [ccNetworkGetDefaultDomainA](#) ()
- const [NQ_WCHAR](#) * [ccNetworkGetDefaultDomain](#) ()
- [NQ_HANDLE](#) [ccNetworkEnumerateDomains](#) (void)
- [CCNetDomainItem](#) * [ccNetworkGetNextDomainItem](#) ([NQ_HANDLE](#) handle)
- [NQ_HANDLE](#) [ccNetworkEnumerateServersA](#) (const [NQ_CHAR](#) *domain)
- [NQ_HANDLE](#) [ccNetworkEnumerateServers](#) (const [NQ_WCHAR](#) *domain)
- [CCNetServerItem](#) * [ccNetworkGetNextServerItem](#) ([NQ_HANDLE](#) handle)
- [NQ_HANDLE](#) [ccNetworkEnumerateSharesA](#) (const [NQ_CHAR](#) *server)
- [NQ_HANDLE](#) [ccNetworkEnumerateShares](#) (const [NQ_WCHAR](#) *server)
- const [NQ_CHAR](#) * [ccNetworkGetNextItemNameA](#) ([NQ_HANDLE](#) handle)
- const [NQ_WCHAR](#) * [ccNetworkGetNextItemName](#) ([NQ_HANDLE](#) handle)
- [CCNetShareItem](#) * [ccNetworkGetNextShareItem](#) ([NQ_HANDLE](#) handle)
- [NQ_BOOL](#) [ccNetworkGetShareInfoA](#) (const [NQ_CHAR](#) *server, const [NQ_CHAR](#) *share, [NQ_UINT16](#) *type, [NQ_CHAR](#) *remarkBuffer, [NQ_INT](#) bufferSize)
- [NQ_BOOL](#) [ccNetworkGetShareInfo](#) (const [NQ_WCHAR](#) *server, const [NQ_WCHAR](#) *share, [NQ_UINT16](#) *type, [NQ_WCHAR](#) *remarkBuffer, [NQ_INT](#) bufferSize)
- [NQ_BOOL](#) [ccNetworkCloseHandle](#) ([NQ_HANDLE](#) handle)
- [NQ_BOOL](#) [ccNetworkResetHandle](#) ([NQ_HANDLE](#) handle)

4.27.1 Detailed Description

4.27.2 Function Documentation

4.27.2.1 [NQ_BOOL](#) [ccNetworkSetDefaultDomainA](#) (const [NQ_CHAR](#) * *domain*)

This function defines the name of domain/workgroup that will be used as the default in subsequent calls to cc-
Network methods.

Parameters

<i>domain</i>	The name of the default domain/workgroup to be used in subsequent calls.
---------------	--

Returns

This function returns TRUE if domain is valid parameter or FALSE otherwise.

Note

- This function is only available when NQ supports the NetBIOS transport.
- This is an ASCII version of [ccNetworkSetDefaultDomain\(\)](#)

4.27.2.2 **NQ_BOOL** ccNetworkSetDefaultDomain (**const NQ_WCHAR** * *domain*)

This function defines the name of domain/workgroup that will be used as the default in subsequent calls to cc-Network methods.

Parameters

<i>domain</i>	The name of the default domain/workgroup to be used in subsequent calls.
---------------	--

Returns

This function returns TRUE if domain is valid parameter or FALSE otherwise.

Note

This function is only available when NQ supports the NetBIOS transport.

4.27.2.3 void ccNetworkCacheSet (NQ_UINT32 timeout)

This function sets network browsing cache state.

Network browsing cache is enabled by default.

Parameters

<i>timeout</i>	Timeout/ttl value for entry cache (0 to disable cache)
----------------	--

Note

When cache is enabled domains, servers and shares per server are cached. This function is part of the new browser API.

4.27.2.4 const NQ_CHAR* ccNetworkGetDefaultDomainA ()

This function is called by application to get the name of the workgroup/domain that CIFS Client will use by default. This default will be used in those CIFS Client functions where workgroup/domain is not explicitly specified.

Returns

Buffer pointer to the name of the default domain/workgroup.

Note

- This function is relevant when NQ supports the NetBIOS transport.
- This function is part of the new Browser API. The functions of the new Browser API are designated with a *ccNetwork...* prefix.
- This an ASCII version of [ccNetworkGetDefaultDomain\(\)](#)

4.27.2.5 const NQ_WCHAR* ccNetworkGetDefaultDomain ()

This function is called by application to get the name of the workgroup/domain that CIFS Client will use by default. This default will be used in those CIFS Client functions where workgroup/domain is not explicitly specified.

Returns

Buffer pointer to the name of the default domain/workgroup.

Note

- This function is relevant when NQ supports the NetBIOS transport.
- This function is part of the new Browser API. The functions of the new Browser API are designated with a *ccNetwork...* prefix.

4.27.2.6 NQ_HANDLE ccNetworkEnumerateDomains (void)

This function is called by application to start enumerating domains/workgroups. It executes necessary transactions to withdraw domains/workgroups announced on the network. After successful return from this call, NQ Client creates a list of domains/workgroups designated by an abstract handle.

Application may withdraw domain/workgroup names one by one by calling [ccNetworkGetNextDomainItem\(\)](#) with the handle obtained in the current call. When finished with domain enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Returns

This function returns an enumeration handle. Application should use this abstract handle in subsequent calls to [ccNetworkGetNextDomainItem\(\)](#). On error this function returns a NULL handle. Application can examine the error code for the failure reason.

Note

- The resulted handle is not thread safe.
- This function is only available when NQ supports the NetBIOS transport.

4.27.2.7 CCNetDomainItem* ccNetworkGetNextDomainItem (NQ_HANDLE handle)

Application calls this function after successfully starting a domain enumeration ([ccNetworkEnumerateDomains\(\)](#)).

Application may withdraw [CCNetDomainItem](#) Items one by one by calling [ccNetworkGetNextDomainItem\(\)](#) with the handle obtained in the current call. When finished with domain enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Parameters

<i>handle</i>	Enumeration handle as obtained in one of Browser enumeration calls.
---------------	---

Returns

This function returns a pointer to the next [CCNetDomainItem](#) structure in the enumeration. This function returns a NULL pointer when NQ Client reaches the end of enumeration.

4.27.2.8 NQ_HANDLE ccNetworkEnumerateServersA (const NQ_CHAR * domain)

This function is called by application to start enumerating servers of a particular domain/workgroups. It executes necessary transactions to withdraw the server list. After successful return from this call, NQ Client creates a list of servers designated by an abstract handle.

Application may withdraw server names one by one by calling [ccNetworkGetNextItemName\(\)](#) with the handle obtained in the current call. When finished with domain enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Parameters

<i>domain</i>	Domain/workgroup name. This call will return a list of servers for this domain/workgroup. When domain name not supplied the default one will be used see ccNetworkSetDefaultDomain() , in addition all hosts discoverable over WS-Discovery will be returned.
---------------	---

Returns

This function returns an enumeration handle. Application should use this abstract handle in subsequent calls to [ccNetworkGetNextItemName\(\)](#). On error this function returns a NULL handle. Application can examine the error code for the failure reason.

Note

- The resulted handle is not thread-safe.
- This is an ASCII version of [ccNetworkEnumerateServers\(\)](#)

4.27.2.9 NQ_HANDLE ccNetworkEnumerateServers (const NQ_WCHAR * domain)

This function is called by application to start enumerating servers of a particular domain/workgroups. It executes necessary transactions to withdraw the server list. After successful return from this call, NQ Client creates a list of servers designated by an abstract handle.

Application may withdraw server names one by one by calling [ccNetworkGetNextItemName\(\)](#) with the handle obtained in the current call. When finished with domain enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Parameters

<i>domain</i>	Domain/workgroup name. This call will return a list of servers for this domain/workgroup. When domain name not supplied the default one will be used see ccNetworkSetDefaultDomain() , in addition all hosts discoverable over WS-Discovery will be returned.
---------------	---

Returns

This function returns an enumeration handle. Application should use this abstract handle in subsequent calls to [ccNetworkGetNextItemName\(\)](#). On error this function returns a NULL handle. Application can examine the error code for the failure reason.

Note

The resulted handle is not thread-safe.

4.27.2.10 CCNetServerItem* ccNetworkGetNextServerItem (NQ_HANDLE handle)

Application calls this function after successfully starting a server enumeration ([ccNetworkEnumerateServers\(\)](#)).

Application may withdraw [CCNetServerItem](#) Items one by one by calling [ccNetworkGetNextServerItem\(\)](#) with the handle obtained in the current call. When finished with server enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Parameters

<i>handle</i>	Enumeration handle as obtained in one of Browser enumeration calls.
---------------	---

Returns

This function returns a pointer to the next [CCNetServerItem](#) structure in the enumeration. This function returns a NULL pointer when NQ Client reaches the end of enumeration.

4.27.2.11 NQ_HANDLE ccNetworkEnumerateSharesA (const NQ_CHAR * server)

This function is called by application to start enumerating shares on a particular server. It executes necessary transactions to withdraw the list of shares. After successful return from this call, NQ Client creates a list of shares designated by an abstract handle.

Application may withdraw share names one by one by calling [ccNetworkGetNextItemName\(\)](#) with the handle obtained in the current call. When finished with domain enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Parameters

<i>server</i>	Server name. This call will return a list of shares for this server.
---------------	--

Returns

This function returns an enumeration handle. Application should use this abstract handle in subsequent calls to [ccNetworkGetNextItemName\(\)](#). On error this function returns a NULL handle. Application can examine the error code for the failure reason.

Note

- The resulted handle is not thread-safe.
- This is an ASCII version of [ccNetworkEnumerateShares\(\)](#)

4.27.2.12 NQ_HANDLE ccNetworkEnumerateShares (const NQ_WCHAR * server)

This function is called by application to start enumerating shares on a particular server. It executes necessary transactions to withdraw the list of shares. After successful return from this call, NQ Client creates a list of shares designated by an abstract handle.

Application may withdraw share names one by one by calling [ccNetworkGetNextItemName\(\)](#) with the handle obtained in the current call. When finished with domain enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Parameters

<i>server</i>	Server name. This call will return a list of shares for this server.
---------------	--

Returns

This function returns an enumeration handle. Application should use this abstract handle in subsequent calls to [ccNetworkGetNextItemName\(\)](#). On error this function returns a NULL handle. Application can examine the error code for the failure reason.

Note

The resulted handle is not thread-safe.

4.27.2.13 const NQ_CHAR* ccNetworkGetNextItemNameA (NQ_HANDLE handle)

Application calls this function after successfully starting a browser enumeration ([ccNetworkEnumerateDomains\(\)](#), [ccNetworkEnumerateServers\(\)](#) or [ccNetworkEnumerateShares\(\)](#)). Application may withdraw share names one by one by calling [ccNetworkGetNextItemName\(\)](#) with the handle obtained in the current call. When finished with domain enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Parameters

<i>handle</i>	Enumeration handle as obtained in one of Browser enumeration calls.
---------------	---

Returns

This function returns a pointer to the next item name in the enumeration. Item type depends on the handle source - one of the browser enumeration calls. This function returns a NULL pointer when NQ Client reaches the end of enumeration.

Note

This is an ASCII version of [ccNetworkGetNextItemName\(\)](#)

4.27.2.14 `const NQ_WCHAR* ccNetworkGetNextItemName (NQ_HANDLE handle)`

Application calls this function after successfully starting a browser enumeration ([ccNetworkEnumerateDomains\(\)](#), [ccNetworkEnumerateServers\(\)](#) or [ccNetworkEnumerateShares\(\)](#)). Application may withdraw share names one by one by calling [ccNetworkGetNextItemName\(\)](#) with the handle obtained in the current call. When finished with domain enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Parameters

<i>handle</i>	Enumeration handle as obtained in one of Browser enumeration calls.
---------------	---

Returns

This function returns a pointer to the next item name in the enumeration. Item type depends on the handle source - one of the browser enumeration calls. This function returns a NULL pointer when NQ Client reaches the end of enumeration.

4.27.2.15 `CCNetShareItem* ccNetworkGetNextShareItem (NQ_HANDLE handle)`

Application calls this function after successfully starting a share enumeration ([ccNetworkEnumerateShares\(\)](#)).

Application may withdraw [CCNetShareItem](#) Items one by one by calling [ccNetworkGetNextShareItem\(\)](#) with the handle obtained in the current call. When finished with domain enumeration, application must call [ccNetworkCloseHandle\(\)](#) to release resources associated with this handle.

Parameters

<i>handle</i>	Enumeration handle as obtained in one of Browser enumeration calls.
---------------	---

Returns

This function returns a pointer to the next item name in the enumeration. Item type depends on the handle source - one of the browser enumeration calls. This function returns a NULL pointer when NQ Client reaches the end of enumeration.

4.27.2.16 `NQ_BOOL ccNetworkGetShareInfoA (const NQ_CHAR * server, const NQ_CHAR * share, NQ_UINT16 * type, NQ_CHAR * remarkBuffer, NQ_INT bufferSize)`

This function is called by application to get information about a particular share on the specified host.

Parameters

<i>server</i>	Server name.
<i>share</i>	Share name.
<i>type</i>	Share types as: <ul style="list-style-type: none"> • directory tree • print queue • serial device • IPC 0x80000000 • hidden share

<i>remarkBuffer</i>	Buffer to place the share remark into.
<i>bufferSize</i>	Size of the remark buffer. The recommended value is 256 characters.

Returns

This function returns TRUE if the share information is obtained successfully or FALSE otherwise. Application can examine the error code for the failure reason.

Note

This is an ASCII version of [ccNetworkGetShareInfo\(\)](#)

4.27.2.17 `NQ_BOOL ccNetworkGetShareInfo (const NQ_WCHAR * server, const NQ_WCHAR * share, NQ_UINT16 * type, NQ_WCHAR * remarkBuffer, NQ_INT bufferSize)`

This function is called by application to get information about a particular share on the specified host. Parameters

Parameters

<i>server</i>	Server name.
<i>share</i>	Share name.
<i>type</i>	Share types as: <ul style="list-style-type: none"> • directory tree • print queue • serial device • IPC 0x80000000 • hidden share
<i>remarkBuffer</i>	Buffer to place the share remark into.
<i>bufferSize</i>	Size of the remark buffer. The recommended value is 256 characters.

Returns

This function returns TRUE if the share information is obtained successfully or FALSE otherwise. Application can examine the error code for the failure reason.

4.27.2.18 `NQ_BOOL ccNetworkCloseHandle (NQ_HANDLE handle)`

This function is called by application to finish a browser enumeration and release all resources (e.g., - item names) associated with that enumeration.

Parameters

<i>handle</i>	An abstract handle as obtained from one of the browser enumeration calls.
---------------	---

Returns

TRUE on success or FALSE on error.

Note

Currently, this function always return TRUE.

See Also

[ccNetworkEnumerateDomains\(\)](#) [ccNetworkEnumerateServers\(\)](#) [ccNetworkEnumerateShares\(\)](#)

4.27.2.19 NQ_BOOL ccNetworkResetHandle (NQ_HANDLE *handle*)

This function is called by application to restart a browser enumeration designated by an abstract handle. If this handle was already used in one or more of the [ccNetworkGetNextItemName\(\)](#) calls, NQ Client will revert it to the state it has right after the respective Browser enumeration call (see below).

Parameters

<i>handle</i>	An abstract handle as obtained from one of the browser enumeration calls.
---------------	---

Returns

TRUE on success or FALSE on error.

Note

Currently, this function always return TRUE.

See Also

[ccNetworkEnumerateDomains\(\)](#) [ccNetworkEnumerateServers\(\)](#) [ccNetworkEnumerateShares\(\)](#)

4.28 Domain Operations

Functions

- [NQ_BOOL ccDomainJoinA](#) (const [NQ_CHAR](#) *domain, const [NQ_CHAR](#) *computer, const [AMCredentialsA](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainJoin](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [AM-Credentials](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainJoinPersistent](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [A-M-Credentials](#) *admin, const [NQ_WCHAR](#) *dns)
- [NQ_BOOL ccDomainUpdateSecretA](#) (const [NQ_CHAR](#) *domain, const [NQ_CHAR](#) *computer, const [AM-CredentialsA](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainUpdateSecret](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [AM-Credentials](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainLeaveA](#) (const [NQ_CHAR](#) *domain, const [NQ_CHAR](#) *computer, const [AMCredentialsA](#) *admin)
- [NQ_BOOL ccDomainLeave](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [AM-Credentials](#) *admin)
- [NQ_BOOL ccDomainLogonA](#) (const [NQ_CHAR](#) *domain, const [NQ_CHAR](#) *computer, const [AM-CredentialsA](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainLogon](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [AM-Credentials](#) *admin, [NQ_BYTE](#) secret[16])

4.28.1 Detailed Description

4.28.2 Function Documentation

4.28.2.1 [NQ_BOOL ccDomainJoinA](#) (const [NQ_CHAR](#) * domain, const [NQ_CHAR](#) * computer, const [AMCredentialsA](#) * admin, [NQ_BYTE](#) secret[16])

This function joins target computer to the domain.

After successfully joining domain, this function returns a computer password obtained from domain controller to be used for further logging into domain. This operation requires domain administrative credentials.

Parameters

<i>domain</i>	Name of the domain to join. This may be a fully qualified name.
<i>computer</i>	Name of the computer (host) that joins.
<i>admin</i>	Administrative credentials for its domain.
<i>secret</i>	Buffer for computer password. This buffer should accommodate at least 16 bytes .

Returns

This function returns TRUE if computer succeeded to join the domain or FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[ccDomainLogon\(\)](#) [ccDomainLeave\(\)](#) [AMCredentials](#) structure in [amapi.h](#) File Operations.

Note

This is an ASCII version of [ccDomainJoin\(\)](#)

4.28.2.2 NQ_BOOL ccDomainJoin (const NQ_WCHAR * domain, const NQ_WCHAR * computer, const AMCredentials * admin, NQ_BYTE secret[16])

This function joins target computer to the domain.

After successfully joining domain, this function returns a computer password obtained from domain controller to be used for further logging into domain. This operation requires domain administrative credentials.

Parameters

<i>domain</i>	Name of the domain to join. This may be a fully qualified name.
<i>computer</i>	Name of the computer (host) that joins.
<i>admin</i>	Administrative credentials for its domain.
<i>secret</i>	Buffer for computer password. This buffer should accommodate at least 16 bytes .

Returns

This function returns TRUE if computer succeeded to join the domain or FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[ccDomainLogon\(\)](#) [ccDomainLeave\(\)](#) [AMCredentials](#) structure in [amapi.h File Operations](#).

4.28.2.3 NQ_BOOL ccDomainJoinPersistent (const NQ_WCHAR * domain, const NQ_WCHAR * computer, const AMCredentials * admin, const NQ_WCHAR * dns)

This function joins target computer to the domain.

After successfully joining domain, this function returns a computer password obtained from domain controller to be used for further logging into domain. This operation requires domain administrative credentials.

Parameters

<i>domain</i>	Name of the domain to join. This may be a fully qualified name.
<i>computer</i>	Name of the computer (host) that joins.
<i>admin</i>	Administrative credentials for its domain.
<i>dns</i>	A semicolon delimited list of DNS servers, can be NULL.

Returns

This function returns TRUE if computer succeeded to join the domain or FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[ccDomainLogon\(\)](#) [ccDomainLeave\(\)](#) [AMCredentials](#) structure in [amapi.h File Operations](#).

4.28.2.4 NQ_BOOL ccDomainUpdateSecretA (const NQ_CHAR * domain, const NQ_CHAR * computer, const AMCredentialsA * admin, NQ_BYTE secret[16])

This function joins target computer to the domain or updates the password (secret) for previously joined computer.

On success this function returns a computer password obtained from domain controller to be used for further logging into domain. This operation requires domain administrative credentials.

Parameters

<i>domain</i>	Name of the domain to join. This may be a fully qualified name.
<i>computer</i>	Name of the computer (host) that joins.
<i>admin</i>	Administrative credentials for its domain.
<i>secret</i>	Buffer for computer password. This buffer should accommodate at least 16 bytes.

Returns

This function returns TRUE if computer succeeded to join the domain or when computer account password was updated (in case when account already existed on domain), FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[ccDomainJoin\(\)](#) [ccDomainLogon\(\)](#) [ccDomainLeave\(\)](#) [AMCredentials](#) structure in [amapi.h](#) File Operations.

Note

This is an ASCII version of [ccDomainUpdateSecret\(\)](#)

4.28.2.5 `NQ_BOOL ccDomainUpdateSecret (const NQ_WCHAR * domain, const NQ_WCHAR * computer, const AMCredentials * admin, NQ_BYTE secret[16])`

This function joins target computer to the domain or updates the password (secret) for previously joined computer. On success this function returns a computer password obtained from domain controller to be used for further logging into domain. This operation requires domain administrative credentials.

Parameters

<i>domain</i>	Name of the domain to join. This may be a fully qualified name.
<i>computer</i>	Name of the computer (host) that joins.
<i>admin</i>	Administrative credentials for its domain.
<i>secret</i>	Buffer for computer password. This buffer should accommodate at least 16 bytes.

Returns

This function returns TRUE if computer succeeded to join the domain or when computer account password was updated (in case when account already existed on domain), FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[ccDomainJoin\(\)](#) [ccDomainLogon\(\)](#) [ccDomainLeave\(\)](#) [AMCredentials](#) structure in [amapi.h](#) File Operations.

4.28.2.6 `NQ_BOOL ccDomainLeaveA (const NQ_CHAR * domain, const NQ_CHAR * computer, const AMCredentialsA * admin)`

This function cancels domain membership of the target computer.

Parameters

<i>domain</i>	Name of the domain to leave.
<i>computer</i>	Name of the computer (host) that leaves.
<i>admin</i>	Administrative credentials for its domain.

Returns

This function returns TRUE if computer successfully leaves the domain or FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[ccDomainJoin\(\)](#) [AMCredentials](#) structure in [amapi.h](#) File Operations.

Note

This is an ASCII version of [ccDomainLeave\(\)](#)

4.28.2.7 NQ_BOOL ccDomainLeave (const NQ_WCHAR * domain, const NQ_WCHAR * computer, const AMCredentials * admin)

This function cancels domain membership of the target computer.

Parameters

<i>domain</i>	Name of the domain to leave.
<i>computer</i>	Name of the computer (host) that leaves.
<i>admin</i>	Administrative credentials for its domain.

Returns

This function returns TRUE if computer successfully leaves the domain or FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[ccDomainJoin\(\)](#) [AMCredentials](#) structure in [amapi.h](#) File Operations.

4.28.2.8 NQ_BOOL ccDomainLogonA (const NQ_CHAR * domain, const NQ_CHAR * computer, const AMCredentialsA * admin, NQ_BYTE secret[16])

This function performs domain logon operation supplying a password obtained from domain controller by calling [ccDomainJoin\(\)](#).

Parameters

<i>domain</i>	Name of the domain to log on.
<i>computer</i>	Computer (host) name.
<i>admin</i>	User credentials.
<i>secret</i>	Password as obtained when joining domain.

Returns

This function returns TRUE if the computer has successfully logged with the domain controller or FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[AMCredentials](#) structure in [amapi.h File Operations](#).

Note

This is an ASCII version of [ccDomainLogon\(\)](#)

4.28.2.9 `NQ_BOOL ccDomainLogon (const NQ_WCHAR * domain, const NQ_WCHAR * computer, const AMCredentials * admin, NQ_BYTE secret[16])`

This function performs domain logon operation supplying a password obtained from domain controller by calling [ccDomainJoin\(\)](#).

Parameters

<i>domain</i>	Name of the domain to log on.
<i>computer</i>	Computer (host) name.
<i>admin</i>	User credentials.
<i>secret</i>	Password as obtained when joining domain.

Returns

This function returns TRUE if the computer has successfully logged with the domain controller or FALSE otherwise. The application can inspect the error code for the failure reason.

See Also

[AMCredentials](#) structure in [amapi.h File Operations](#).

4.29 Special Operations

Modules

- [Authentication Level](#)

Functions

- [NQ_BOOL ccGetSigningPolicy](#) (void)
- [void ccSetSigningPolicy](#) (NQ_BOOL enable)
- [NQ_STATUS ccSetMinSecurityLevel](#) (NQ_INT level)
- [NQ_INT ccGetMinSecurityLevel](#) (void)
- [NQ_STATUS ccSetMaxSecurityLevel](#) (NQ_INT level)
- [NQ_INT ccGetMaxSecurityLevel](#) (void)
- [NQ_BOOL ccSetExclusiveAccessToFileA](#) (NQ_CHAR *fileName, NQ_BOOL makeExclusive)
- [NQ_BOOL ccSetExclusiveAccessToFile](#) (NQ_WCHAR *fileName, NQ_BOOL makeExclusive)
- [NQ_BOOL ccIsExclusiveAccessToFileA](#) (NQ_CHAR *fileName)
- [NQ_BOOL ccIsExclusiveAccessToFile](#) (NQ_WCHAR *fileName)

4.29.1 Detailed Description

4.29.2 Function Documentation

4.29.2.1 NQ_BOOL ccGetSigningPolicy (void)

This function is called by application to get the current Message Signing mode.

Returns

TRUE if Message Signing is enabled and FALSE otherwise

4.29.2.2 void ccSetSigningPolicy (NQ_BOOL enable)

This function is called by application to set the current Message Signing mode.

Parameters

<i>enable</i>	If set to TRUE NQ Client will sign messages. if set to FALSE NQ Client will not sign messages.
---------------	--

Note

Default is TRUE. It's not recommended to set to FALSE unless the server shows unusual behavior.

4.29.2.3 NQ_STATUS ccSetMinSecurityLevel (NQ_INT level)

This function is called by application to set the NQ CIFS Client security minimal level. Before this function is called NQ CIFS Client uses the default security configuration (see AM_MAXSECURITYLEVEL in NQ Library Reference - Authentication)

Parameters

<i>level</i>	LAN Manager authentication level. The value of this parameter corresponds to table see amClientDefineLevel in NQ Library Reference - Authentication
--------------	---

Returns

NQ_SUCCESS or NQ_FAIL when level requested is invalid.

4.29.2.4 NQ_INT ccGetMinSecurityLevel (void)

This function is called by application to get the NQ CIFS Client security minimal level.

Returns

Current NQ CIFS Client security minimal level. Value returned corresponds to table see amClientDefineLevel in NQ Library Reference - Authentication

4.29.2.5 NQ_STATUS ccSetMaxSecurityLevel (NQ_INT level)

This function is called by application to set the NQ CIFS Client security maximal level. Before this function is called NQ CIFS Client uses the default security configuration (see AM_MAXSECURITYLEVEL in NQ Library Reference - Authentication)

Parameters

<i>level</i>	LAN Manager authentication level. The value of this parameter corresponds to table see amClientDefineLevel in NQ Library Reference - Authentication
--------------	---

Returns

NQ_SUCCESS or NQ_FAIL when level requested is invalid.

4.29.2.6 NQ_INT ccGetMaxSecurityLevel (void)

This function is called by application to get the NQ CIFS Client security maximal level.

Returns

Current NQ CIFS Client security maximal level. Value returned corresponds to table see amClientDefineLevel in NQ Library Reference - Authentication

4.29.2.7 NQ_BOOL ccSetExclusiveAccessToFileA (NQ_CHAR * fileName, NQ_BOOL makeExclusive)

This function sets or removes exclusive access rights or restores default access rights.

This function applies access rights to a file. When makeExclusive parameter is TRUE the following access rights will be applied:

- Owner - full access
- Administrators (group) - full access
- All others - no access

- When applied to a folder, access rights will be propagated down to subfolders/files, unless they have non-inheriting security descriptors
- The ownership is changed to the user authenticated.
- Access rights of the parent directory will be no more inherited. When the second parameter is FALSE file descriptor is modified so that all access rights are inherited from the parent directory.

When this function returns TRUE it means that the remote server has accepted new access rights but it does not guarantee that those rights were effectively modified. To check the latter use [ccIsExclusiveAccessToFile\(\)](#).

Parameters

<i>fileName</i>	Path to the file.
<i>makeExclusive</i>	TRUE to set exclusive access, FALSE to revert to default access rights.

Returns

TRUE when operation succeeded, FALSE on error. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccSetExclusiveAccessToFile\(\)](#)

4.29.2.8 NQ_BOOL ccSetExclusiveAccessToFile (NQ_WCHAR * fileName, NQ_BOOL makeExclusive)

This function sets or removes exclusive access rights or restores default access rights.

This function applies access rights to a file. When makeExclusive parameter is TRUE the following access rights will be applied:

- Owner - full access
- Administrators (group) - full access
- All others - no access
- When applied to a folder, access rights will be propagated down to subfolders/files, unless they have non-inheriting security descriptors
- The ownership is changed to the user authenticated.
- Access rights of the parent directory will be no more inherited. When the second parameter is FALSE file descriptor is modified so that all access rights are inherited from the parent directory.

When this function returns TRUE it means that the remote server has accepted new access rights but it does not guarantee that those rights were effectively modified. To check the latter use [ccIsExclusiveAccessToFile\(\)](#).

Parameters

<i>fileName</i>	Path to the file.
<i>makeExclusive</i>	TRUE to set exclusive access, FALSE to revert to default access rights.

Returns

TRUE when operation succeeded, FALSE on error. The application can inspect the error code for the failure reason.

4.29.2.9 NQ_BOOL ccIsExclusiveAccessToFileA (NQ_CHAR * fileName)

This function is called by application to determine whether the file/directory specified by the parameter has exclusive access rights.

Parameters

<i>fileName</i>	Path to the file.
-----------------	-------------------

Returns

TRUE if the file specified by the parameter has exclusive access rights or FALSE otherwise.

See Also

[ccSetExclusiveAccessToFile\(\)](#)

Note

This is an ASCII version of [ccIsExclusiveAccessToFile\(\)](#)

4.29.2.10 NQ_BOOL ccIsExclusiveAccessToFile (NQ_WCHAR * fileName)

This function is called by application to determine whether the file/directory specified by the parameter has exclusive access rights.

Parameters

<i>fileName</i>	Path to the file.
-----------------	-------------------

Returns

TRUE if the file specified by the parameter has exclusive access rights or FALSE otherwise.

See Also

[ccSetExclusiveAccessToFile\(\)](#)

4.30 Dialects Operations

Enumerations

- enum [CCSmbDialect](#) {
[CC_SMB100](#), [CC_SMB202](#), [CC_SMB210](#), [CC_SMB300](#),
[CC_SMB302](#), [CC_SMB311](#) }

Functions

- [NQ_BOOL](#) [ccSetSmbDialect](#) ([CCSmbDialect](#) dialect, [NQ_BOOL](#) setActive)
- [NQ_BOOL](#) [ccCifsGetSMB1Support](#) (void)
- void [ccCifsSetSMB1Support](#) ([NQ_BOOL](#) isSupport)

4.30.1 Detailed Description

4.30.2 Enumeration Type Documentation

4.30.2.1 enum [CCSmbDialect](#)

Available SMB dialects.

See [ccSetSmbDialect\(\)](#)

Enumerator

[CC_SMB100](#) SMB 1
[CC_SMB202](#) SMB 2.0.2
[CC_SMB210](#) SMB 2.1.0
[CC_SMB300](#) SMB 3.0.0
[CC_SMB302](#) SMB 3.0.2
[CC_SMB311](#) SMB 3.1.1

4.30.3 Function Documentation

4.30.3.1 [NQ_BOOL](#) [ccSetSmbDialect](#) ([CCSmbDialect](#) *dialect*, [NQ_BOOL](#) *setActive*)

Activates or deactivates selected SMB dialect.

This function affects new connections and it does not affect the existing connections.

Parameters

<i>dialect</i>	SMB dialect value, see CCSmbDialect
<i>setActive</i>	<i>TRUE</i> to enable this dialect or <i>FALSE</i> to disable it.

Returns

TRUE or *FALSE* .

4.30.3.2 [NQ_BOOL](#) [ccCifsGetSMB1Support](#) (void)

Determine whether SMB1 dialect is supported.

Returns*TRUE* or *FALSE***4.30.3.3 void ccCifsSetSMB1Support (NQ_BOOL *isSupport*)**

Activates or deactivates of SMB1 dialect.

Parameters

<i>isSupport</i>	<i>TRUE</i> to enable SMB1 dialect, <i>FALSE</i> to disable SMB1 dialect.
------------------	---

4.31 User Credentials

Modules

- [Credential fields length](#)

Functions

- [NQ_BOOL ccResetCredentailsA](#) (const [NQ_CHAR](#) *mountpoint)
- [NQ_BOOL ccResetCredentails](#) (const [NQ_WCHAR](#) *mountpoint)
- [NQ_BOOL ccValidateUserA](#) ([NQ_CHAR](#) *server)
- [NQ_BOOL ccValidateUser](#) ([NQ_WCHAR](#) *server)
- [NQ_BOOL ccValidateUserByCredentialsA](#) ([NQ_CHAR](#) *server, const [AMCredentialsA](#) *userCredentials)
- [NQ_BOOL ccValidateUserByCredentials](#) ([NQ_WCHAR](#) *server, const [AMCredentials](#) *userCredentials)

4.31.1 Detailed Description

4.31.2 Function Documentation

4.31.2.1 [NQ_BOOL ccResetCredentailsA](#) (const [NQ_CHAR](#) * *mountpoint*)

This function clears user credentials associated with the given mount point.

After this call, NQ will query application in any of the following cases:

- Reconnecting one of the shares, associated with this mount point as a result of a temporary server disconnect.
- Establishing a connection with additional servers as a result of DFS redirection.

This call does not affect those shares that were connected before this call.

Parameters

<i>mountpoint</i>	The name of the mount point to clear credentials.
-------------------	---

Returns

TRUE on success, FALSE on failure.

Note

This is an ASCII version of [ccResetCredentails\(\)](#)

4.31.2.2 [NQ_BOOL ccResetCredentails](#) (const [NQ_WCHAR](#) * *mountpoint*)

This function clears user credentials associated with the given mount point.

After this call, NQ will query application in any of the following cases:

- Reconnecting one of the shares, associated with this mount point as a result of a temporary server disconnect.
- Establishing a connection with additional servers as a result of DFS redirection. This call does not affect those shares that were connected before this call.

Parameters

<i>mountpoint</i>	The name of the mount point to clear credentials.
-------------------	---

Returns

TRUE on success, FALSE on failure.

4.31.2.3 NQ_BOOL ccValidateUserA (NQ_CHAR * server)

This function is called by application to verify that the user credentials supplied by [udGetCredentials\(\)](#) are sufficient on the specified server. Those credentials can be set by [udSetCredentials\(\)](#) before calling this function.

Parameters

<i>server</i>	Server name the credentials should be verified against. If this parameter is NULL then N-Q Client will try to automatically discover the Domain Controller and verify the credentials against it.
---------------	---

Returns

This function returns TRUE if the user is successfully authenticated or FALSE otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccValidateUser\(\)](#)

4.31.2.4 NQ_BOOL ccValidateUser (NQ_WCHAR * server)

This function is called by application to verify that the user credentials supplied by [udGetCredentials\(\)](#) are sufficient on the specified server. Those credentials can be set by [udSetCredentials\(\)](#) before calling this function.

Parameters

<i>server</i>	Server name the credentials should be verified against. If this parameter is NULL then N-Q Client will try to automatically discover the Domain Controller and verify the credentials against it.
---------------	---

Returns

This function returns TRUE if the user is successfully authenticated or FALSE otherwise. The application can inspect the error code for the failure reason

4.31.2.5 NQ_BOOL ccValidateUserByCredentialsA (NQ_CHAR * server, const AMCredentialsA * userCredentials)

This function is called by application to verify that the supplied user credentials sufficient on the specified server.

Parameters

<i>server</i>	Server name the credentials should be verified against. If this parameter is NULL then N-Q Client will try to automatically discover the Domain Controller and verify the credentials against it.
---------------	---

<i>userCredentials</i>	Pointer to a structure, which stores the user credentials. See AMCredentialsA structure. If NULL is supplied the credentials will be queried by udGetCredentials() . To initialize that structure use amCredentialsInitA() . It is the caller responsibility to release that structure.
------------------------	---

Returns

This function returns TRUE if the user is successfully authenticated or FALSE otherwise. The application can inspect the error code for the failure reason.

Note

This is an ASCII version of [ccValidateUserByCredentials\(\)](#)

4.31.2.6 NQ_BOOL ccValidateUserByCredentials (NQ_WCHAR * server, const AMCredentials * userCredentials)

This function is called by application to verify that the supplied user credentials sufficient on the specified server.

Parameters

<i>server</i>	Server name the credentials should be verified against. If this parameter is NULL then N-Q Client will try to automatically discover the Domain Controller and verify the credentials against it.
<i>userCredentials</i>	Pointer to a structure, which stores the user credentials. See AMCredentials structure. If NULL is supplied the credentials will be queried by udGetCredentials() . To initialize that structure use amCredentialsInitW() . It is the caller responsibility to release that structure.

Returns

This function returns TRUE if the user is successfully authenticated or FALSE otherwise. The application can inspect the error code for the failure reason

4.32 Errors

Modules

- [Driver Notification Events](#)

Functions

- [NQ_UINT32 ccGetNTStatus](#) (void)
- [NQ_UINT32 ccErrorToSmbStatus](#) (NQ_STATUS nqError)

4.32.1 Detailed Description

4.32.2 Function Documentation

4.32.2.1 NQ_UINT32 ccGetNTStatus (void)

Returns last SMB NT error status for application thread.

Status returned is for last erroneous SMB command response from server.

Note

That last erroneous NT status code may not reflect the real problem. Use syGetLastError() for proper NQ error code.

Returns

Error code.

4.32.2.2 NQ_UINT32 ccErrorToSmbStatus (NQ_STATUS nqError)

This function is called by the application when some returned NQ error value should be translated to a SMB error.

Parameters

<i>nqError</i>	The error code returned from some nq function.
----------------	--

Returns

SMB error code.

4.33 Auxiliary methods

Functions

- [NQ_BOOL ccTransportRestartRecieveThread \(\)](#)
- void [ccDump](#) (void)
- void [ccCloseAllConnections](#) (void)
- void [ccCloseHiddenConnections](#) (void)
- void [ccThreadSubscribe](#) (void)
- void [ccThreadUnsubscribe](#) (void)

4.33.1 Detailed Description

4.33.2 Function Documentation

4.33.2.1 [NQ_BOOL ccTransportRestartRecieveThread \(\)](#)

This function must be called only by the driver. When the FUSE driver runs in regular mode (not debug) it creates a hard fork and closes parent process. This causes the receiving thread to be killed. This function is used to restart the receiving thread.

Returns

TRUE on success or FALSE on error.

Note

Currently, this function always return TRUE.

4.33.2.2 [void ccDump \(void \)](#)

This function is called by application to print all the CIFS Client tables. It is used for debugging purposes only.

4.33.2.3 [void ccCloseAllConnections \(void \)](#)

This function is called by application to close all open connections and to release resources associated with them. In this function call NQ will do the following:

- Close all files;
- Release outstanding searches;
- Disconnect from shares (trees);
- Logoff users;
- Disconnect from servers.

4.33.2.4 [void ccCloseHiddenConnections \(void \)](#)

This function is called by application to those connections that were used implicitly and are not associated with any mount. For instance, upon this call NQ will close those connections that were open solely for network browsing. NQ will release all resources associated with the connections being closed, namely:

- Close open files;
- Release outstanding searches;
- Disconnect from shares (trees);
- Logoff users;
- Disconnect from servers.

4.33.2.5 void ccThreadSubscribe (void)

Register the current system thread.

This function creates an NQ object describing the current system (application) thread. NQ expects the application to call [ccThreadUnsubscribe\(\)](#) to release NQ resources associated with this thread. If this thread was already registered, nothing will happen.

4.33.2.6 void ccThreadUnsubscribe (void)

Release the resources associated with the current system thread.

NQ assumes that the current system (application) thread was registered with a [ccThreadSubscribe\(\)](#) call. If this is not true, nothing will happen.

4.34 Common

Modules

- [Initialization and Shutdown](#)
- [Resolver](#)
- [Product Information](#)
- [Codepage](#)
- [Cryptographic](#)
- [DNS and WINS](#)
- [Auxiliary](#)

4.34.1 Detailed Description

4.35 Initialization and Shutdown

Macros

- `#define NQ_CLIENT 0x1`
- `#define NQ_SERVER 0x2`
- `#define NQ_NDDAEMON 0x4`

Functions

- `NQ_STATUS cmInit` (`NQ_UINT32` component)
- `void cmExit` (`NQ_UINT32` component)

4.35.1 Detailed Description

4.35.2 Macro Definition Documentation

4.35.2.1 `#define NQ_CLIENT 0x1`

Client component

4.35.2.2 `#define NQ_SERVER 0x2`

Server component

4.35.2.3 `#define NQ_NDDAEMON 0x4`

NetBIOS daemon component

4.35.3 Function Documentation

4.35.3.1 `NQ_STATUS cmInit (NQ_UINT32 component)`

This function starts the Common Module

Parameters

<i>component</i>	One of the following: NQ_CLIENT , NQ_NDDAEMON , NQ_SERVER
------------------	---

Returns

This function return `NQ_SUCCESS` if module was started successfully or `NQ_FAIL` otherwise.

See Also

[cmExit\(\)](#)

4.35.3.2 `void cmExit (NQ_UINT32 component)`

This function performs the shutdown of the Common Module

Parameters

<i>component</i>	One of the following: NQ_CLIENT , NQ_NDDAEMON , NQ_SERVER
------------------	---

See Also[cmInit\(\)](#)

4.36 Resolver

Data Structures

- struct [CMResolverRegisteredMethodDescription](#)

Macros

- `#define NQ_RESOLVER_IPV4 4`
- `#define NQ_RESOLVER_IPV6 6`
- `#define NQ_RESOLVER_NONE 0`
- `#define NQ_RESOLVER_DNS 1`
- `#define NQ_RESOLVER_NETBIOS 2`
- `#define NQ_RESOLVER_EXTERNAL_METHOD 5`
- `#define NQ_RESOLVER_DNS_DC 8`
- `#define NQ_RESOLVER_NETBIOS_DC 10`
- `#define NQ_RESOLVER_WSD 12`

Typedefs

- `typedef NQ_INT(* CMResolverNameToIpA)(const NQ_CHAR *name, void *ip, NQ_COUNT index)`
- `typedef NQ_INT(* CMResolverNameToIp)(const NQ_WCHAR *name, void *ip, NQ_COUNT index)`
- `typedef NQ_BOOL(* CMResolverIpToNameA)(NQ_CHAR *name, const void *ip, NQ_INT ipType)`
- `typedef NQ_BOOL(* CMResolverIpToName)(NQ_WCHAR *name, const void *ip, NQ_INT ipType)`

Functions

- `void cmResolverSetExternalA (CMResolverNameToIpA nameToIp, CMResolverIpToNameA ipToName)`
- `void cmResolverSetExternal (CMResolverNameToIp nameToIp, CMResolverIpToName ipToName)`
- `const NQ_WCHAR * cmResolverGetHostName (const NQ_IPADDRESS *ip)`
- `const NQ_IPADDRESS * cmResolverGetHostIps (const NQ_WCHAR *dnsList, const NQ_WCHAR *host, NQ_INT *numIps)`
- `const NQ_WCHAR * cmResolverGetDCName (const NQ_WCHAR *domain, const NQ_WCHAR *dnsList, NQ_INT *numDCs)`
- `void cmResolverEnableMethod (NQ_INT type, NQ_BOOL unicast, NQ_BOOL multicast)`
- `NQ_BOOL cmResolverRegisterExternalMethod (const CMResolverRegisteredMethodDescription *pMethod)`
- `NQ_BOOL cmResolverUpdateExternalMethodsPriority (NQ_INT requiredPriority)`

4.36.1 Detailed Description

4.36.2 Macro Definition Documentation

4.36.2.1 `#define NQ_RESOLVER_IPV4 4`

The resolved address is IPV4.

4.36.2.2 `#define NQ_RESOLVER_IPV6 6`

The resolved address is IPV6.

4.36.2.3 #define NQ_RESOLVER_NONE 0

No address resolved. This value designates a resolution failure.

4.36.2.4 #define NQ_RESOLVER_DNS 1

Mechanism type is DNS.

4.36.2.5 #define NQ_RESOLVER_NETBIOS 2

Mechanism type is NetBIOS.

4.36.2.6 #define NQ_RESOLVER_EXTERNAL_METHOD 5

Mechanism type not known, external method set by the user.

4.36.2.7 #define NQ_RESOLVER_DNS_DC 8

Mechanism type is DNS DC.

4.36.2.8 #define NQ_RESOLVER_NETBIOS_DC 10

Mechanism type is NetBIOS DC.

4.36.2.9 #define NQ_RESOLVER_WSD 12

Mechanism type is WS-Discovery.

4.36.3 Typedef Documentation

4.36.3.1 typedef NQ_INT(* CMResolverNameToIpA)(const NQ_CHAR *name, void *ip, NQ_COUNT index)

This function prototype designates a callback for resolving host IP address by its name. NQ uses this callback for external resolution.

Parameters

<i>name</i>	The name of the host to resolve.
<i>index</i>	Index of the resolved IP address. External resolver may resolve multiple IP addresses for the same name. When external resolution is required, NQ subsequently calls this function with ' <i>index</i> ' values incrementing, starting from zero. External resolver should return the appropriate IP address or it should fail the call when an ' <i>index</i> ' result is not available.

Returns

One of the following values:

- [NQ_RESOLVER_IPV4](#). This means that '*index*' designates an IPv4 address.
- [NQ_RESOLVER_IPV6](#). This means that '*index*' designates an IPv6 address.
- [NQ_RESOLVER_NONE](#). This means that either resolution failed or the '*index*' value is out of range.

Note

This is an ASCII version of [CMResolverNameToIp\(\)](#)

4.36.3.2 typedef NQ_INT(* CMResolverNameToIp)(const NQ_WCHAR *name, void *ip, NQ_COUNT index)

This function prototype designates a callback for resolving host IP address by its name. NQ uses this callback for external resolution.

Parameters

<i>name</i>	The name of the host to resolve.
<i>index</i>	Index of the resolved IP address. External resolver may resolve multiple IP addresses for the same name. When external resolution is required, NQ subsequently calls this function with ' <i>index</i> ' values incrementing, starting from zero. External resolver should return the appropriate IP address or it should fail the call when an ' <i>index</i> ' result is not available.

Returns

One of the following values:

- [NQ_RESOLVER_IPV4](#) This means that '*index*' designates an IPv4 address.
- [NQ_RESOLVER_IPV6](#) This means that '*index*' designates an IPv6 address.
- [NQ_RESOLVER_NONE](#) This means that either resolution failed or the '*index*' value is out of range.

4.36.3.3 typedef NQ_BOOL(* CMResolverIpToNameA)(NQ_CHAR *name, const void *ip, NQ_INT ipType)

This function prototype designates a callback for resolving host name by its IP address. NQ uses this callback for external resolution.

Parameters

<i>name</i>	Buffer for the resolved name.
<i>ip</i>	Pointer to IP address. This pointer designates either NQ_IPADDRESS4, NQ_IPADDRESS4 type or NQ_IPADDRESS6, NQ_IPADDRESS6 type depending on the ' <i>type</i> ' argument below.
<i>ipType</i>	IP address type. This value can be either NQ_RESOLVER_IPV4 or NQ_RESOLVER_IPV6 . It defines the format of the ' <i>ip</i> ' argument.

Returns

TRUE on success, FALSE on failure.

Note

This is an ASCII version of [CMResolverIpToName\(\)](#)

4.36.3.4 typedef NQ_BOOL(* CMResolverIpToName)(NQ_WCHAR *name, const void *ip, NQ_INT ipType)

This function prototype designates a callback for resolving host name by its IP address. NQ uses this callback for external resolution.

Parameters

<i>name</i>	Buffer for the resolved name.
<i>ip</i>	Pointer to IP address. This pointer designates either NQ_IPADDRESS4, NQ_IPADDRESS4 type or NQ_IPADDRESS6, NQ_IPADDRESS6 type depending on the ' <i>type</i> ' argument below.

<i>ipType</i>	IP address type. This value can be either NQ_RESOLVER_IPV4 or NQ_RESOLVER_IPV6 . It defines the format of the ' <i>ip</i> ' argument.
---------------	---

Returns

TRUE on success, FALSE on failure.

4.36.4 Function Documentation

4.36.4.1 void cmResolverSetExternalA (CMResolverNameToIpA nameToIp, CMResolverIpToNameA ipToName)

This function extends NQ's name-to-IP and IP-to-name resolver. NQ attempts to use its internal resolution methods first:

- NetBIOS Naming Service;
- DNS;
- LLMNR. If none of the above succeeds, NQ attempts an external resolver. By calling this function application can enable or disable an external resolving mechanism.

NQ uses callback pointers as follows:

- First, NQ attempts internal methods. If at least one of them succeeded, the external method is not used.
- If a Unicode callback was specified, it is used.
- If a Unicode callback pointer was not specified or its pointer was set to NULL, NQ uses ASCII callback.
- If neither Unicode nor ASCII callback were specified or both pointers are NULL, NQ skips external resolution.
- The approach above applies independently to each of name-to-IP and IP-to-name resolution.

Parameters

<i>nameToIp</i>	Pointer to a function that resolves host IP by its name. This function signature should conform to the CMResolverNameToIp prototype. This value can be NULL. In this case, NQ will fail external name-to-IP resolution.
<i>ipToName</i>	Pointer to a function that resolves host name by its IP. This function signature should conform to the CMResolverIpToName prototype. This value can be NULL. In this case, NQ will fail external IP-to-name resolution.

4.36.4.2 void cmResolverSetExternal (CMResolverNameToIp nameToIp, CMResolverIpToName ipToName)

This function extends NQ's name-to-IP and IP-to-name resolver. NQ attempts to use its internal resolution methods first:

- NetBIOS Naming Service;
- DNS;
- LLMNR. If none of the above succeeds, NQ attempts an external resolver. By calling this function application can enable or disable an external resolving mechanism.

NQ uses callback pointers as follows:

- First, NQ attempts internal methods. If at least one of them succeeded, the external method is not used.
- If a Unicode callback was specified, it is used.

- If a Unicode callback pointer was not specified or its pointer was set to NULL, NQ uses ASCII callback.
- If neither Unicode nor ASCII callback were specified or both pointers are NULL, NQ skips external resolution.
- The approach above applies independently to each of name-to-IP and IP-to-name resolution.

Parameters

<i>nameToIp</i>	Pointer to a function that resolves host IP by its name. This function signature should conform to the CMResolverNameToIp prototype. This value can be NULL. In this case, NQ will fail external name-to-IP resolution.
<i>ipToName</i>	Pointer to a function that resolves host name by its IP. This function signature should conform to the CMResolverIpToName prototype. This value can be NULL. In this case, NQ will fail external IP-to-name resolution.

4.36.4.3 `const NQ_WCHAR* cmResolverGetHostName (const NQ_IPADDRESS * ip)`

This function resolves host by its IP and return its name.

Host name is created in allocated memory so that it is caller's responsibility to free this memory.

NQ uses several unicast and multicast methods for this resolution. First, it attempts all unicast methods concurrently. Depending on compilation parameters those methods may be:

- DNS queries to one or more DNS servers;
- NetBIOS query to one or more WINS. If none of the above succeeded, NQ concurrently attempts multicast methods as:
- LLMNR;
- NetBIOS broadcasts. If those methods did not succeed, NQ calls external method if this method has been installed in [cmResolverRegisterExternalMethod\(\)](#) call.

Parameters

<i>ip</i>	Pointer to host IP address.
-----------	-----------------------------

Returns

Pointer to newly created host name or NULL on failure.

Note

It is caller's responsibility to release this name.

4.36.4.4 `const NQ_IPADDRESS* cmResolverGetHostIps (const NQ_WCHAR * dnsList, const NQ_WCHAR * host, NQ_INT * numIps)`

This function resolves host by its name and return its IP addresses.

The array of IP addresses is created in allocated memory so that it is caller's responsibility to free this memory.

NQ uses several unicast and multicast methods for this resolution. First, it attempts all unicast methods concurrently. Depending on compilation parameters those methods may be:

- DNS queries to one or more DNS servers;
- NetBIOS query to one or more WINS. If none of the above succeeded, NQ concurrently attempts multicast methods as:
- LLMNR;
- NetBIOS broadcasts. If those methods did not succeed, NQ calls external method if this method has been installed in [cmResolverRegisterExternalMethod\(\)](#) call.

Parameters

<i>host</i>	Pointer to host name to resolve.
<i>numIps</i>	Pointer to variable that on exit gets the number of resolved IP addresses.
<i>dnsList</i>	DNS ips to register and prepare methods.

Returns

Pointer to an array of IP addresses or NULL on error. The size of array is placed into the variable pointed by *numIps*.

Note

It is caller's responsibility to release this array. To release this array the caller need to call the *cmMemoryFree()* function.

4.36.4.5 `const NQ_WCHAR* cmResolverGetDCName (const NQ_WCHAR * domain, const NQ_WCHAR * dnsList, NQ_INT * numDCs)`

This function resolves domain's DC and return its name.

DC name is created in allocated memory so that it is caller's responsibility to free this memory.

NQ uses several unicast and multicast methods for this resolution. First, it attempts all unicast methods concurrently. Depending on compilation parameters those methods may be:

- DNS queries to one or more DNS servers;
- NetBIOS query to one or more WINS. If none of the above succeeded, NQ concurrently attempts multicast methods as:
- NetBIOS broadcasts. If those methods did not succeed, NQ calls external method if this method has been installed in [cmResolverRegisterExternalMethod\(\)](#) call.

Parameters

<i>domain</i>	Pointer to domain name.
<i>dnsList</i>	Semicolon delimited list of DNS servers.
<i>numDCs</i>	Pointer to variable that on exit gets the number of resolved DC names.

Returns

Pointer to newly created DC name or NULL on failure.

Note

It is caller's responsibility to release this name. To release the name call *cmMemoryFree()*

4.36.4.6 `void cmResolverEnableMethod (NQ_INT type, NQ_BOOL unicast, NQ_BOOL multicast)`

Resolver uses different methods (mechanisms) to resolve host IP(s) by name or to resolve host name by IP. Each mechanism may use either unicasts or multicasts. NQ attempts all possible unicasts first. Then, if unicasts failed, it attempts multicasts. Currently, the following mechanisms are available:

Method	DNS	NetBIOS
Unicast	DNS server query	WINS query
Multicast	LLMNR	local broadcast

All methods are initially enabled on NQ startup. This function allows to enable or disable a particular one in run-time.

Parameters

<i>type</i>	Resolution type. This may be one of the following: NQ_RESOLVER_DNS , NQ_RESOLVER_NETBIOS , NQ_RESOLVER_EXTERNAL_METHOD , NQ_RESOLVER_DNS_DC , NQ_RESOLVER_NETBIOS_DC , NQ_RESOLVER_WSD .
<i>unicast</i>	When this parameter is <i>TRUE</i> , NQ enables unicast method (s) of the given type. When this parameter is <i>FALSE</i> ., NQ disables unicast method(s).
<i>multicast</i>	When this parameter is <i>TRUE</i> , NQ enables multicast method(s) of the given type. When this parameter is <i>FALSE</i> ., NQ disables multicast method(s).

4.36.4.7 **NQ_BOOL** cmResolverRegisterExternalMethod (const **CMResolverRegisteredMethodDescription** * *pMethod*)

Resolver uses DNS and NET BIOS to resolve host IP(s) by name or to resolve host name by IP. This function enables adding another resolution method. A method is defined using the [CMResolverRegisteredMethodDescription](#) Structure. See structure description with different members. Not all functions have to be defined, but notice following guidelines:

1. Must define at least one request function - request by name or request by IP
2. If a request function is defined, then the corresponding response function must be defined as well.
3. If defined timeout value is too long it might cause a delay in setup process.
4. Resolver methods are not executed all at once. Registered method will be executed according to defined priority. See the [CMResolverRegisteredMethod](#) structure.

Parameters

<i>pMethod</i>	resolution method.
----------------	--------------------

Returns

TRUE - on registration success. *FALSE* otherwise.

4.36.4.8 **NQ_BOOL** cmResolverUpdateExternalMethodsPriority (**NQ_INT** *requiredPriority*)

This function enables run time control on the order of resolver methods execution. In resolver point of view three groups of methods exist: unicast methods, multicast methods, external methods. External methods are all methods that were registered by the user. Each group of methods is executed separately and if no reply is received the next group is executed.

Parameters

<i>requiredPriority</i>	This parameter takes the following values: <ol style="list-style-type: none"> 1. Execute external methods before all other methods. 2. Execute external methods after unicast methods and before multicast methods. 3. Execute external methods only if all other resolution methods failed.
-------------------------	---

Returns

TRUE - on success. *FALSE* otherwise.

4.37 Product Information

Data Structures

- struct [CMProductInfo](#)

Macros

- #define [CM_PRODUCT_STRING_LEN](#) 20

Functions

- [CMProductInfo](#) * [cmGetCurrentProductInformation](#) (void)

4.37.1 Detailed Description

4.37.2 Macro Definition Documentation

4.37.2.1 #define [CM_PRODUCT_STRING_LEN](#) 20

4.37.3 Function Documentation

4.37.3.1 [CMProductInfo](#)* [cmGetCurrentProductInformation](#) (void)

This function provides information of the current compiled product.

Returns

Pointer to a static [CMProductInfo](#) struct filled with the current product information.

4.38 Codepage

Data Structures

- struct [CMCodepage](#)

Functions

- [NQ_BOOL cmCodepageAdd](#) (const [CMCodepage](#) *codePage)
- [NQ_BOOL cmCodepageRemove](#) (const [CMCodepage](#) *codePage)
- void [cmWideCharToMultiByte](#) ([NQ_CHAR](#) *strMultiByte, const [NQ_WCHAR](#) *strWideChar)
- void [cmMultiByteToWideChar](#) ([NQ_WCHAR](#) *strWideChar, const [NQ_CHAR](#) *strMultiByte)

4.38.1 Detailed Description

4.38.2 Function Documentation

4.38.2.1 [NQ_BOOL cmCodepageAdd](#) (const [CMCodepage](#) * *codePage*)

This function installs code page for another language.

Code page is identified by the '*id*' value of its descriptor. Duplicate installation overwrite the previous one.

Once successfully installed, the code page becomes available by its '*id*' number through the [udGetCodePage\(\)](#) function call.

This function may be used for one the following purposes:

- Installing new code page.
- Replacing one of previously installed code pages.
- Replacing one of precompiled code pages.

Parameters

<i>codePage</i>	Pointer to codepage descriptor
-----------------	--------------------------------

Returns

[NQ_TRUE](#) on success and [NQ_FALSE](#) on error. Application may analyze error reason by reading the last system error. It may be [NQ_ERR_NOMEM](#) on codepage table overflow.

4.38.2.2 [NQ_BOOL cmCodepageRemove](#) (const [CMCodepage](#) * *codePage*)

This function removes code page.

This call will either remove a code page previously installed by calling [cmCodepageAdd\(\)](#) or it may remove a pre-compiled page.

Parameters

<i>codePage</i>	Pointer to codepage descriptor as CMCodepage structure. NQ uses only the ' <i>id</i> ' field and ignores others.
-----------------	--

Returns

[NQ_TRUE](#) on success and [NQ_FALSE](#) on error. Application may analyze error reason by reading the last system error. It may be [NQ_ERR_NOTFOUND](#) when this codepage does not exist.

4.38.2.3 void cmWideCharToMultiByte (**NQ_CHAR** * *strMultiByte*, const **NQ_WCHAR** * *strWideChar*)

Conversion of Wide Characters string into Multi Byte.

Parameters

<i>strMultiByte</i>	Pointer to result multi byte string
<i>strWideChar</i>	Pointer to wide char string to convert

4.38.2.4 void cmMultiByteToWideChar (NQ_WCHAR * *strWideChar*, const NQ_CHAR * *strMultiByte*)

Conversion of Multi Byte string into Wide Characters.

Parameters

<i>strWideChar</i>	Pointer to result wide char string
<i>strMultiByte</i>	Pointer to multi byte string to convert

4.39 Cryptographic

Data Structures

- struct [CMCrypterList](#)

Typedefs

- typedef void(* [CMAbstractHasher](#))(const [NQ_BYTE](#) *dataIn, [NQ_BYTE](#) *dataOut, [NQ_COUNT](#) length)
- typedef void(* [CMAbstractCipher](#))(const [CMBlob](#) *key, const [CMBlob](#) *key1, const [CMIOBlob](#) dataFragments[], [NQ_COUNT](#) numFragments, [NQ_BYTE](#) *buffer, [NQ_COUNT](#) bufferSize)
- typedef void(* [CMAbstractIOCipher](#))(const [CMBlob](#) *key, const [CMBlob](#) *key1, const [CMIOBlob](#) dataFragments[], [NQ_COUNT](#) numFragments, [NQ_BYTE](#) *buffer, [NQ_COUNT](#) bufferSize)
- typedef void(* [CMAbstractHasher512](#))(const [CMBlob](#) *key, const [CMBlob](#) *key1, const [CMIOBlob](#) dataFragments[], [NQ_COUNT](#) numFragments, [NQ_IOBufPos](#) buffer, [NQ_COUNT](#) bufferSize, [NQ_BYTE](#) *ctx-Buff)
- typedef void(* [CMAbstractCcmEncryption](#))(const [CMBlob](#) *key, const [CMBlob](#) *key1, const [CMIOBlob](#) *prefix, [CMIOBlob](#) *message, [NQ_BYTE](#) *auth)
- typedef [NQ_BOOL](#)(* [CMAbstractCcmDecryption](#))(const [CMBlob](#) *key, const [CMBlob](#) *key1, const [CMIOBlob](#) *prefix, [CMIOBlob](#) *message, const [NQ_BYTE](#) *auth)
- typedef void(* [CMAbstractGcmEncryption](#))(const [CMBlob](#) *key, const [CMBlob](#) *key1, const [CMIOBlob](#) *prefix, [CMIOBlob](#) *message, [NQ_BYTE](#) *auth, [NQ_BYTE](#) *keyBuffer, [NQ_BYTE](#) *encMsgBuffer)
- typedef [NQ_BOOL](#)(* [CMAbstractGcmDecryption](#))(const [CMBlob](#) *key, const [CMBlob](#) *key1, const [CMIOBlob](#) *prefix, [CMIOBlob](#) *message, const [NQ_BYTE](#) *auth, [NQ_BYTE](#) *keyBuffer, [NQ_BYTE](#) *msgBuffer)

Functions

- void [cmSetExternalCrypters](#) (const [CMCrypterList](#) *crypters)
- void [cmResetExternalCrypters](#) (void)

4.39.1 Detailed Description

4.39.2 Typedef Documentation

4.39.2.1 typedef void(* [CMAbstractHasher](#))(const [NQ_BYTE](#) *dataIn, [NQ_BYTE](#) *dataOut, [NQ_COUNT](#) length)

This function prototype designates an abstract cryptographic hash algorithm. It is used to replace internal NQ encryption with an external one.

Parameters

<i>dataIn</i>	Pointer to the data to encrypt.
<i>dataOut</i>	Pointer to the buffer for encrypted.
<i>length</i>	Length of the incoming data and also the length of the encrypted data.

4.39.2.2 typedef void(* [CMAbstractCipher](#))(const [CMBlob](#) *key, const [CMBlob](#) *key1, const [CMIOBlob](#) dataFragments[], [NQ_COUNT](#) numFragments, [NQ_BYTE](#) *buffer, [NQ_COUNT](#) bufferSize)

This function prototype designates an abstract cryptographic algorithm using a cryptographic key. It is used to replace internal NQ encryption with an external one.

Parameters

<i>key</i>	Pointer to the encryption key descriptor. Some algorithms do not use this value.
<i>key1</i>	Pointer to the auxiliary key descriptor. This value may be NULL. Also the data pointer may be null as well as the blob length may be zero. In either of those cases the auxiliary key is ignored. Some algorithms may ignore this value anyway.
<i>dataFragments</i>	A pointer to the array of data fragment descriptors. For encryption algorithms which allow encrypting multiple fragments, this array may contain more than one element. For other algorithms, only the first element is used. An element may be NULL. Also a data pointer in an element may be NULL as well as blob length may be zero. In either of those cases the respective fragment is ignored.
<i>numFragments</i>	Number of data fragments (see above).
<i>buffer</i>	Place holder for hash result.
<i>bufferSize</i>	The expected length of the encrypted data

4.39.2.3 `typedef void(* CMAbstractOCipher)(const CMBlob *key, const CMBlob *key1, const CMIOBlob dataFragments[], NQ_COUNT numFragments, NQ_BYTE *buffer, NQ_COUNT bufferSize)`

This function prototype designates an abstract cryptographic algorithm using a cryptographic key. It is used to replace internal NQ encryption with an external one.

Parameters

<i>key</i>	Pointer to the encryption key descriptor. Some algorithms do not use this value.
<i>key1</i>	Pointer to the auxiliary key descriptor. This value may be NULL. Also the data pointer may be null as well as the blob length may be zero. In either of those cases the auxiliary key is ignored. Some algorithms may ignore this value anyway.
<i>dataFragments</i>	A pointer to the array of data fragment descriptors. For encryption algorithms which allow encrypting multiple fragments, this array may contain more than one element. For other algorithms, only the first element is used. An element may be NULL. Also a data pointer in an element may be NULL as well as blob length may be zero. In either of those cases the respective fragment is ignored.
<i>numFragments</i>	Number of data fragments (see above).
<i>buffer</i>	Place holder for hash result.
<i>bufferSize</i>	The expected length of the encrypted data

4.39.2.4 `typedef void(* CMAbstractHasher512)(const CMBlob *key, const CMBlob *key1, const CMIOBlob dataFragments[], NQ_COUNT numFragments, NQ_IOBufPos buffer, NQ_COUNT bufferSize, NQ_BYTE *ctxBuff)`

This function prototype designates an abstract cryptographic algorithm for both encryption and producing authentication data (CCM) . It is used to replace internal NQ encryption with an external one.

This algorithm assumes that there are two blocks of data:

1. A prefix that is preserved as is
2. The messages itself that is to be encrypted The authentication is a production of both blocks.

Parameters

<i>key</i>	Pointer to the encryption key descriptor. Some algorithms do not use this value.
<i>key1</i>	Pointer to the auxiliary key descriptor. This value may be NULL. Also the data pointer may be null as well as the blob length may be zero. In either of those cases the auxiliary key is ignored. Some algorithms may ignore this value anyway.

<i>prefix</i>	A pointer to the message prefix descriptor. This data participate in authentication but remains as is, without encryption.
<i>message</i>	Pointer to the message descriptor. This message will be encrypted in-place.
<i>auth</i>	Pointer to the authentication data. It will be filled as the result of the algorithm.

4.39.2.5 `typedef void(* CMAbstractCcmEncryption)(const CMBlob *key, const CMBlob *key1, const CMIOBlob *prefix, CMIOBlob *message, NQ_BYTE *auth)`

This function prototype designates an abstract cryptographic algorithm for producing authentication data. It is used to replace internal NQ encryption with an external one. The authentication (hash result) is a product of all data blocks.

Parameters

<i>key</i>	Pointer to the encryption key descriptor. Some algorithms do not use this value.
<i>key1</i>	Pointer to the auxiliary key descriptor. This value may be NULL. Also the data pointer may be null as well as the blob length may be zero. In either of those cases the auxiliary key is ignored. Some algorithms may ignore this value anyway.
<i>dataFragments</i>	A pointer to the array of data fragment descriptors. For encryption algorithms which allow encrypting multiple fragments, this array may contain more than one element. For other algorithms, only the first element is used. An element may be NULL. Also a data pointer in an element may be NULL as well as blob length may be zero. In either of those cases the respective fragment is ignored.
<i>numFragments</i>	Number of data fragments (see above).
<i>buffer</i>	Place holder for hash result.
<i>bufferSize</i>	The expected length of the encrypted data
<i>ctxBuff</i>	Buffer for context data. if this buffer is null, context buffer will be allocated in function.

4.39.2.6 `typedef NQ_BOOL(* CMAbstractCcmDecryption)(const CMBlob *key, const CMBlob *key1, const CMIOBlob *prefix, CMIOBlob *message, const NQ_BYTE *auth)`

This function prototype designates an abstract cryptographic algorithm for both decryption and authentication (CCM). It is used to replace internal NQ encryption with an external one.

This algorithm assumes that there are two blocks of data:

1. A prefix that is preserved as is
2. The messages itself that is to be encrypted The authentication is a production of both blocks.

Parameters

<i>key</i>	Pointer to the encryption key descriptor. Some algorithms do not use this value.
<i>key1</i>	Pointer to the auxiliary key descriptor. This value may be NULL. Also the data pointer may be null as well as the blob length may be zero. In either of those cases the auxiliary key is ignored. Some algorithms may ignore this value anyway.
<i>prefix</i>	A pointer to the message prefix descriptor. This data participate in authentication but remains as is, without encryption.
<i>message</i>	Pointer to the message descriptor. This message will be encrypted in-place.
<i>auth</i>	Pointer to the authentication data. This value is be used to authenticate.

Returns

TRUE if authenticated, FALSE if not.

4.39.2.7 `typedef void(* CMAbstractGcmEncryption)(const CMBlob *key, const CMBlob *key1, const CMIOBlob *prefix, CMIOBlob *message, NQ_BYTE *auth, NQ_BYTE *keyBuffer, NQ_BYTE *encMsgBuffer)`

This function prototype designates an abstract cryptographic algorithm for both encryption and producing authentication data (GCM) . It is used to replace internal NQ encryption with an external one.

This algorithm assumes that there are two blocks of data:

1. A prefix that is preserved as is
2. The messages itself that is to be encrypted The authentication is a production of both blocks.

Parameters

<i>key</i>	Pointer to the encryption key descriptor. Some algorithms do not use this value.
<i>key1</i>	Pointer to the auxiliary key descriptor. This value may be NULL. Also the data pointer may be null as well as the blob length may be zero. In either of those cases the auxiliary key is ignored. Some algorithms may ignore this value anyway.
<i>prefix</i>	A pointer to the message prefix descriptor. This data participate in authentication but remains as is, without encryption.
<i>message</i>	Pointer to the message descriptor. This message will be encrypted in-place.
<i>auth</i>	Pointer to the authentication data. It will be filled with the result of the algorithm.
<i>keyBuffer,:</i>	Optional buffer for usage in key calculations. If NULL a buffer will be allocated per function call. size - AES_PRIV_SIZE
<i>ncMsgBuffer</i>	Optional buffer for the message encryption which is not done in place. If NULL a buffer will be allocated per function call. size: message size.

4.39.2.8 `typedef NQ_BOOL(* CMAbstractGcmDecryption)(const CMBlob *key, const CMBlob *key1, const CMIOBlob *prefix, CMIOBlob *message, const NQ_BYTE *auth, NQ_BYTE *keyBuffer, NQ_BYTE *msgBuffer)`

This function prototype designates an abstract cryptographic algorithm for both decryption and authentication (GCM). It is used to replace internal NQ encryption with an external one.

This algorithm assumes that there are two blocks of data:

1. A prefix that is preserved as is
2. The messages itself that is to be encrypted The authentication is a production of both blocks.

Parameters

<i>key</i>	Pointer to the encryption key descriptor. Some algorithms do not use this value.
<i>key1</i>	Pointer to the auxiliary key descriptor. This value may be NULL. Also the data pointer may be null as well as the blob length may be zero. In either of those cases the auxiliary key is ignored. Some algorithms may ignore this value anyway.
<i>prefix</i>	A pointer to the message prefix descriptor. This data participate in authentication but remains as is, without encryption.
<i>message</i>	Pointer to the message descriptor. This message will be encrypted in-place.
<i>auth</i>	Pointer to the authentication data. This value is be used to authenticate.
<i>keyBuffer,:</i>	Optional buffer for usage in key calculations. If NULL a buffer will be allocated per function call. size - AES_PRIV_SIZE
<i>msgBuffer,:</i>	Optional buffer for the message decryption which is not done in place. If NULL a buffer will be allocated per function call. Size - message size.

Returns

TRUE if authenticated, FALSE if not.

4.39.3 Function Documentation

4.39.3.1 `void cmSetExternalCrypters (const CMCrypterList * crypters)`

This function replaces internal NQ cryptographic algorithms with external ones.

Parameters

<i>crypters</i>	A pointer to the list of internal ciphers. Only non-NULL values are applied.
-----------------	--

4.39.3.2 void cmResetExternalCrypters (void)

This function reverts the list of cryptographic algorithms to internal algorithms only.

4.40 DNS and WINS

Functions

- void [cmDnsSetServersA](#) (const [NQ_CHAR](#) *servers)
- void [cmDnsSetServers](#) (const [NQ_WCHAR](#) *servers)
- [NQ_IPADDRESS](#) * [cmDnsGetServer](#) ([NQ_COUNT](#) dnsID)
- [NQ_COUNT](#) [cmDnsGetNumDnsServers](#) (void)
- void [cmNetBiosSetWinsA](#) (const [NQ_CHAR](#) *servers)
- void [cmNetBiosSetWins](#) (const [NQ_WCHAR](#) *servers)
- [NQ_IPADDRESS4](#) [cmNetBiosGetWins](#) ([NQ_COUNT](#) winsID)
- [NQ_COUNT](#) [cmNetBiosGetNumWinsServers](#) (void)
- [NQ_STATUS](#) [cmDnsSetDomainA](#) (const [NQ_CHAR](#) *domainName)
- [NQ_STATUS](#) [cmDnsSetDomain](#) (const [NQ_WCHAR](#) *domainName)

4.40.1 Detailed Description

4.40.2 Function Documentation

4.40.2.1 void [cmDnsSetServersA](#) (const [NQ_CHAR](#) * *servers*)

After startup NQ uses the list of DNS servers as defined in the NQ configuration. By calling this function application replaces the initial list with a new one.

Parameters

<i>servers</i>	Pointer to the new list of DNS servers. This list is a string of IP addresses, delimited by a semicolon. Each address may have a form of IPv4 or IPv6 address. This pointer can be NULL. In this case NQ will not query DNS servers.
----------------	--

Note

This is an ASCII version of [cmDnsSetServers\(\)](#)

4.40.2.2 void [cmDnsSetServers](#) (const [NQ_WCHAR](#) * *servers*)

After startup NQ uses the list of DNS servers as defined in the NQ configuration. By calling this function application replaces the initial list with a new one.

Parameters

<i>servers</i>	Pointer to the new list of DNS servers. This list is a string of IP addresses, delimited by a semicolon. Each address may have a form of IPv4 or IPv6 address. This pointer can be NULL. In this case NQ will not query DNS servers.
----------------	--

4.40.2.3 [NQ_IPADDRESS](#)* [cmDnsGetServer](#) ([NQ_COUNT](#) *dnsID*)

After startup NQ uses the DNS servers as defined in the NQ configuration or by calling [cmDnsSetServersA\(\)](#) or [cmDnsSetServers\(\)](#). By calling this function with a specific DNS index application returns the specific DNS address.

Parameters

<i>dnsID</i>	DNS server index.
--------------	-------------------

Returns

the pointer of the ip address of the specific DNS server or 0.

Note

The pointer should be used as NQ_IPADDRESS4 or NQ_IPADDRESS6 depends on the current system configurations.

4.40.2.4 NQ_COUNT cmDnsGetNumDnsServers (void)

After startup NQ uses the DNS servers as defined in the NQ configuration or by calling [cmDnsSetServersA\(\)](#) or [cmDnsSetServers\(\)](#). By calling this function application returns the number of current defined DNS addresses.

Returns

Number of defined DNS addresses

4.40.2.5 void cmNetBiosSetWinsA (const NQ_CHAR * servers)

After startup NQ uses the WINS IP as defined in the NQ configuration. By calling this function application replaces the initial WINS with a list of new ones.

Parameters

<i>servers</i>	Pointer to the list of new WINS servers. This list is a string of IP addresses, delimited by a semicolon. Each address should have a form of IPv4 address. This pointer can be NULL, in this case NQ will not query WINS.
----------------	---

Note

- WINS addresses set over this function affect only name-to-IP and IP-to-name resolution. They do not affect name registration, which is always performed against the default WINS. We assume that in a case of multiple WINS they replicate name registered in just one of them.
- This is an ASCII version of [cmNetBiosSetWins\(\)](#)

4.40.2.6 void cmNetBiosSetWins (const NQ_WCHAR * servers)

After startup NQ uses the WINS IP as defined in the NQ configuration. By calling this function application replaces the initial WINS with a list of new ones.

Parameters

<i>servers</i>	Pointer to the list of new WINS servers. This list is a string of IP addresses, delimited by a semicolon. Each address should have a form of IPv4 address. This pointer can be NULL, in this case NQ will not query WINS.
----------------	---

Note

WINS addresses set over this function affect only name-to-IP and IP-to-name resolution. They do not affect name registration, which is always performed against the default WINS. We assume that in a case of multiple WINS they replicate name registered in just one of them.

4.40.2.7 NQ_IPADDRESS4 cmNetBiosGetWins (NQ_COUNT winsID)

After startup NQ uses the WINS servers as defined in the NQ configuration or by calling [cmNetBiosSetWinsA\(\)](#) or [cmNetBiosSetWins\(\)](#). By calling this function with a specific WINS index application returns the specific WINS address.

Parameters

<i>winsID</i>	WINS server index.
---------------	--------------------

Returns

the IPV4 address of the specific WINS server or 0.

4.40.2.8 NQ_COUNT cmNetBiosGetNumWinsServers (void)

After startup NQ uses the WINS servers as defined in the NQ configuration or by calling [cmNetBiosSetWinsA\(\)](#) or [cmNetBiosSetWins\(\)](#). By calling this function application returns the number of current defined WINS addresses.

Returns

Number of defined WINS addresses

4.40.2.9 NQ_STATUS cmDnsSetDomainA (const NQ_CHAR * domainName)

After startup NQ uses the domain name as defined in the NQ configuration. By calling this function application replaces the initial domain name with a new ones. This name becomes available for subsequent operations.

Parameters

<i>domainName</i>	New domain name. It should contain more than one symbol and it should not be longer than UD_NQ_HOSTNAMESIZE
-------------------	---

Returns

NQ_SUCCESS on success or NQ_FAIL when the domain name is NULL or it is too long or too short.

Note

This is an ASCII version of [cmDnsSetDomain\(\)](#)

4.40.2.10 NQ_STATUS cmDnsSetDomain (const NQ_WCHAR * domainName)

After startup NQ uses the domain name as defined in the NQ configuration. By calling this function application replaces the initial domain name with a new ones.

This name becomes available for subsequent operations.

Parameters

<i>domainName</i>	New domain name. It should contain more than one symbol and it should not be longer than UD_NQ_HOSTNAMESIZE
-------------------	---

Returns

NQ_SUCCESS on success or NQ_FAIL when the domain name is NULL or it is too long or too short.

4.41 Auxiliary

Functions

- [NQ_STATUS cmNetBiosNameReload](#) (void)

4.41.1 Detailed Description

4.41.2 Function Documentation

4.41.2.1 **NQ_STATUS** cmNetBiosNameReload (void)

Reload cmnbyname.c various name resources in the run-time by calling UD methods, can be used to modify host-name or domain name in the run-time.

Returns

NQ_SUCCESS or NQ_FAIL.

4.42 Server

Modules

- [Initialization and Shutdown](#)
- [Shares Management](#)
- [Open Files Enumeration](#)
- [Cleanup](#)
- [Anonymous](#)
- [Event Log](#)

4.42.1 Detailed Description

4.43 Initialization and Shutdown

Functions

- [NQ_STATUS csStart](#) (void)
- void [csStop](#) (void)

4.43.1 Detailed Description

4.43.2 Function Documentation

4.43.2.1 [NQ_STATUS csStart](#) (void)

This function starts the CIFS Server, must be called after NetBIOS daemon startup. See [ndStart\(\)](#) in [ndapi.h](#) file.

Returns

This function returns NQ_SUCCESS if the server was started successfully or NQ_FAIL otherwise.

See Also

[csStop\(\)](#)

4.43.2.2 void [csStop](#) (void)

This function stops the CIFS Server.

See Also

[csStart\(\)](#)

4.44 Shares Management

Functions

- `NQ_INT csAddShareA` (const `NQ_CHAR` *name, const `NQ_CHAR` *path, `NQ_BOOL` printQueue, const `NQ_CHAR` *comment, const `NQ_CHAR` *reserved)
- `NQ_INT csAddShare` (const `NQ_WCHAR` *name, const `NQ_WCHAR` *path, `NQ_BOOL` printQueue, const `NQ_WCHAR` *comment, const `NQ_CHAR` *reserved)
- `NQ_INT csRemoveShareA` (const `NQ_CHAR` *name)
- `NQ_INT csRemoveShare` (const `NQ_WCHAR` *name)
- `NQ_INT csSetShareEncryptionA` (const `NQ_CHAR` *name, `NQ_BOOL` isEncrypted)
- `NQ_INT csSetShareEncryption` (const `NQ_WCHAR` *name, `NQ_BOOL` isEncrypted)

4.44.1 Detailed Description

4.44.2 Function Documentation

4.44.2.1 `NQ_INT csAddShareA` (const `NQ_CHAR` * *name*, const `NQ_CHAR` * *path*, `NQ_BOOL` *printQueue*, const `NQ_CHAR` * *comment*, const `NQ_CHAR` * *reserved*)

This function adds another share to NQ Server.

Parameters

<i>name</i>	Name for new share. This name must not contain the '\$' character.
<i>path</i>	Local path to the share root.
<i>printQueue</i>	FALSE for file system, TRUE for print queue.
<i>comment</i>	Share description as a free text.
<i>reserved</i>	For future use. This value is currently ignored and may be any, including NULL.

Returns

value	meaning
0	Success
-1	the DB was not initialized
-2	parameter error (string too long)
-3	share table full
-6	share already exists

Note

- This function should be called from the context of NQ Server thread.
- It is recommended to sue it on NQ Server startup only.
- This is an ASCII version of `csAddShare()`.

4.44.2.2 `NQ_INT csAddShare` (const `NQ_WCHAR` * *name*, const `NQ_WCHAR` * *path*, `NQ_BOOL` *printQueue*, const `NQ_WCHAR` * *comment*, const `NQ_CHAR` * *reserved*)

This function adds another share to NQ Server. Parameters

Parameters

<i>name</i>	Name for new share. This name must not contain the '\$' character.
<i>path</i>	Local path to the share root.
<i>printQueue</i>	FALSE for file system, TRUE for print queue.
<i>comment</i>	Share description as a free text.
<i>reserved</i>	For future use. This value is currently ignored and may be any, including NULL.

Returns

value	meaning
0	Success
-1	the DB was not initialized
-2	parameter error (string too long)
-3	share table full
-6	share already exists

Note

- This function should be called from the context of NQ Server thread.
- It is recommended to sue it on NQ Server startup only.

4.44.2.3 NQ_INT csRemoveShareA (const NQ_CHAR * name)

This function removes an existing share from NQ Server.

Parameters

<i>name</i>	Name for the share to remove. This name must not contain the '\$' character.
-------------	--

Returns

value	meaning
0	Success
-1	NQ Server was not initialized
-2	parameter error (share not found)

Note

- This function should be called from the context of NQ Server thread.
- It is recommended to sue it on NQ Server startup only.
- This is an ASCII version of [csRemoveShare\(\)](#).

4.44.2.4 NQ_INT csRemoveShare (const NQ_WCHAR * name)

This function removes an existing share from NQ Server.

Parameters

<i>name</i>	Name for the share to remove. This name must not contain the '\$' character.
-------------	--

Returns

value	meaning
0	Success
-1	NQ Server was not initialized
-2	parameter error (share not found)

Note

- This function should be called from the context of NQ Server thread.
- It is recommended to sue it on NQ Server startup only.

4.44.2.5 NQ_INT csSetShareEncryptionA (const NQ_CHAR * name, NQ_BOOL isEncrypted)

This function enables or disables encryption for a particular share. An encrypted share may be only accessed over SMB3.0.

This function is applicable to any share regardless of way it was created. When called with NULL share name parameter global encryption is applied.

Parameters

<i>name</i>	Share name or NULL.
<i>isEncrypted</i>	TRUE to enable encryption, FALSE to disable encryption. NQ does not consider the previous encryption status.

Returns

value	meaning
0	Success
-1	NQ Server was not initialized
-2	parameter error (share not found)

Note

- To enable or disable global encryption use NULL as share name parameter
- This function should be called from the context of NQ Server thread. It is recommended to issue it on NQ Server startup only.
- This function is available when NQ was compiled with SMB3 support as designated by the UD_NQ_INCLUDESMB3 parameter.
- This is an ASCII version of [csSetShareEncryption\(\)](#)

4.44.2.6 NQ_INT csSetShareEncryption (const NQ_WCHAR * name, NQ_BOOL isEncrypted)

This function enables or disables encryption for a particular share. An encrypted share may be only accessed over SMB3.0.

This function is applicable to any share regardless of way it was created. When called with NULL share name parameter global encryption is applied.

Parameters

<i>name</i>	Share name or NULL.
<i>isEncrypted</i>	TRUE to enable encryption, FALSE to disable encryption. NQ does not consider the previous encryption status.

Returns

value	meaning
0	Success
-1	NQ Server was not initialized
-2	parameter error (share not found)

Note

- To enable or disable global encryption use NULL as share name parameter
- This function should be called from the context of NQ Server thread. It is recommended to issue it on NQ Server startup only.
- This function is available when NQ was compiled with SMB3 support as designated by the UD_NQ_INCLUDESMB3 parameter.

4.45 Open Files Enumeration

Data Structures

- struct [FileDataA_t](#)
- struct [CSFileData](#)

Functions

- [NQ_BOOL](#) [csEnumerateOpenFilesA](#) ([NQ_UINT](#) index, [FileDataA_t](#) *fileData)
- [NQ_BOOL](#) [csEnumerateOpenFiles](#) ([NQ_UINT](#) index, [CSFileData](#) *fileData)

4.45.1 Detailed Description

4.45.2 Function Documentation

4.45.2.1 [NQ_BOOL](#) [csEnumerateOpenFilesA](#) ([NQ_UINT](#) index, [FileDataA_t](#) * fileData)

This function is used to retrieve all open files and directories on NQ Server. Call this function inside a loop and continue till False result is returned. On first call index should be 0, advance index by one per call to this function.

File data will be filled in function parameter fileData, unless returned value is false.

If a file is opened or closed during the loop calling this function, the returned values might not exactly reflect the open files situation.

Parameters

<i>index</i>	Index is indicating which open file data to return. Upon 2nd call the 2nd open file in the internal DB will be returned. On call x, the x'th open file will be returned. For NQ Storage zero index is signaling function to start iteration. Next calls must have any index > 0.
<i>fileData</i>	Pointer to a structure, where NQ places the file data for this index. See FileDataA_t structure.

Returns

This function returns TRUE if the requested file index exists. False - if the requested file index doesn't exist.

Note

- Be sure to start any loop call to this function with index 0.
- This function is non reentrant
- This is an ASCII version of [csEnumerateOpenFiles\(\)](#)

4.45.2.2 [NQ_BOOL](#) [csEnumerateOpenFiles](#) ([NQ_UINT](#) index, [CSFileData](#) * fileData)

This function is used to retrieve all open files and directories on NQ Server. Call this function inside a loop and continue till False result is returned. On first call index should be 0, advance index by one per call to this function.

File data will be filled in function parameter fileData, unless returned value is false.

If a file is opened or closed during the loop calling this function, the returned values might not exactly reflect the open files situation.

Parameters

<i>index</i>	Index is indicating which open file data to return. Upon 2nd call the 2nd open file in the internal DB will be returned. On call x, the x'th open file will be returned. For NQ Storage zero index is signaling function to start iteration. Next calls must have any index > 0.
<i>fileData</i>	Pointer to a structure, where NQ places the file data for this index. See CSFileData structure.

Returns

This function returns TRUE if the requested file index exists. False - if the requested file index doesn't exist.

See Also

[CSFileData](#) structure.

Note

- Be sure to start any loop call to this function with index 0.
- This function is non reentrant

4.46 Cleanup

Functions

- [NQ_STATUS csCleanUserServerConnectionsA](#) (const [NQ_CHAR](#) *name, [NQ_BOOL](#) isDomainUser)
- [NQ_STATUS csCleanUserServerConnections](#) (const [NQ_WCHAR](#) *name, [NQ_BOOL](#) isDomainUser)

4.46.1 Detailed Description

4.46.2 Function Documentation

4.46.2.1 [NQ_STATUS csCleanUserServerConnectionsA](#) (const [NQ_CHAR](#) * name, [NQ_BOOL](#) isDomainUser)

This function closes all connections initiated by a user with the given name close user's existing connections to NQ server

Parameters

<i>name</i>	User name.
<i>isDomainUser</i>	TRUE for a domain or FALSE for a local user.

Returns

This function returns [NQ_SUCCESS](#) if the function succeed or [NQ_FAIL](#) otherwise.

Note

This is an ASCII version of [csCleanUserServerConnections\(\)](#)

4.46.2.2 [NQ_STATUS csCleanUserServerConnections](#) (const [NQ_WCHAR](#) * name, [NQ_BOOL](#) isDomainUser)

This function closes all connections initiated by a user with the given name close user's existing connections to NQ server

Parameters

<i>name</i>	User name.
<i>isDomainUser</i>	TRUE for a domain or FALSE for a local user.

Returns

This function returns [NQ_SUCCESS](#) if the function succeed or [NQ_FAIL](#) otherwise.

4.47 Anonymous

Functions

- void [csSetAnonymousAllowed](#) (NQ_BOOL on)

4.47.1 Detailed Description

4.47.2 Function Documentation

4.47.2.1 void csSetAnonymousAllowed (NQ_BOOL on)

This function sets Anonymous Access to allowed/restricted

Parameters

<i>on</i>	TRUE or FALSE.
-----------	----------------

4.48 Event Log

Data Structures

- struct [EVCSShareConnectionEntry](#)
- struct [EVCSOpenFileEntry](#)

Functions

- [NQ_COUNT](#) [evcsEnumerateConnectedShares](#) ([NQ_COUNT](#) maxEntries, [EVCSShareConnectionEntry](#) *buffer)
- [NQ_COUNT](#) [evcsEnumerateOpenFiles](#) ([NQ_COUNT](#) maxEntries, [EVCSOpenFileEntry](#) *buffer)

4.48.1 Detailed Description

4.48.2 Function Documentation

4.48.2.1 [NQ_COUNT](#) [evcsEnumerateConnectedShares](#) ([NQ_COUNT](#) maxEntries, [EVCSShareConnectionEntry](#) *buffer)

This function reads share connection entries

Parameters

<i>maxEntries</i>	Buffer capacity.
<i>buffer</i>	Buffer pointer.

Returns

Number of entries read.

Note

- This function is only available when NQ was compiled with Event Log support as indicated by the UD_ - NQ_INCLUDEEVENTLOG parameter.

4.48.2.2 [NQ_COUNT](#) [evcsEnumerateOpenFiles](#) ([NQ_COUNT](#) maxEntries, [EVCSOpenFileEntry](#) *buffer)

This function reads entries Parameters

Parameters

<i>maxEntries</i>	Buffer capacity.
<i>buffer</i>	Buffer pointer.

Returns

Number of entries read.

Note

- This function is only available when NQ was compiled with Event Log support as indicated by the UD_ - NQ_INCLUDEEVENTLOG parameter.

4.49 NetBIOS Daemon

Functions

- [NQ_STATUS ndStart](#) (SYSemaphore *sem)
- void [ndStop](#) (void)
- void [ndNotifyConfigurationChange](#) (NDAdapterInfo *adapter)

4.49.1 Detailed Description

4.49.2 Function Documentation

4.49.2.1 [NQ_STATUS ndStart](#) (SYSemaphore * *sem*)

Startup point for NetBIOS Daemon.

Parameters

<i>sem</i>	Pointer to semaphore
------------	----------------------

Returns

NQ_SUCCESS when the daemon has started and *NQ_FAIL* when it failed to start.

4.49.2.2 void [ndStop](#) (void)

Shutdown point for NetBIOS Daemon.

4.49.2.3 void [ndNotifyConfigurationChange](#) (NDAdapterInfo * *adapter*)

Notification about a change in the network configuration. The application may call this function when the list of adapters changes or when wins servers are updated. This call causes NetBIOS Daemon to reload the list of adapters and send new registrations to WINS servers.

4.50 NQ Common

Modules

- [Initialization and Shutdown](#)
- [NetBIOS Name Registration](#)
- [Utils](#)

4.50.1 Detailed Description

4.51 Initialization and Shutdown

Functions

- [NQ_INT nqStart](#) (void)
- [NQ_INT nqStop](#) (void)

4.51.1 Detailed Description

4.51.2 Function Documentation

4.51.2.1 [NQ_INT nqStart](#) (void)

This function is called to start NQ system. This function must initialize UD, SY and NS modules prior to calling to [csStart\(\)](#) or [ndStart\(\)](#) or [cclnit\(\)](#).

Returns

[NQ_SUCCESS](#) or [NQ_FAIL](#)

See Also

[udlnit\(\)](#) in [File Operations udapi.h](#), [sylnit\(\)](#) in [File Operations syapi.h](#), [nslnitGuard\(\)](#) in [File Operations nsapi.h](#), [csStart\(\)](#) in [File Operations csapi.h](#), [ndStart\(\)](#) in [File Operations ndapi.h](#).

4.51.2.2 [NQ_INT nqStop](#) (void)

This function is called to stop NQ system started by [nqStart\(\)](#).

Returns

[NQ_SUCCESS](#) or [NQ_FAIL](#)

See Also

[nqStart\(\)](#)

4.52 NetBIOS Name Registration

Macros

- `#define CM_NB_GROUP 0x80`
- `#define CM_NB_UNIQUE 0x40`
- `#define CM_NB_POSTFIX_WORKSTATION 0x0`
- `#define CM_NB_POSTFIX_SERVER 0x20`
- `#define CM_NB_POSTFIX_DOMAINMASTERBROWSER 0x1b`
- `#define CM_NB_POSTFIX_DOMAINCONTROLLER 0x1c`
- `#define CM_NB_POSTFIX_MASTERBROWSER 0x1d`

Functions

- `NQ_STATUS nqRegisterNetBiosName` (const `NQ_CHAR` *name)
- `NQ_STATUS nqReleaseNetBiosName` (const `NQ_CHAR` *name)
- `NQ_STATUS nqRegisterNetBiosNameEx` (const `NQ_CHAR` *name, const `NQ_BYTE` flags)
- `NQ_STATUS nqReleaseNetBiosNameEx` (const `NQ_CHAR` *name, const `NQ_BYTE` flags)

4.52.1 Detailed Description

4.52.2 Macro Definition Documentation

4.52.2.1 `#define CM_NB_GROUP 0x80`

4.52.2.2 `#define CM_NB_UNIQUE 0x40`

4.52.2.3 `#define CM_NB_POSTFIX_WORKSTATION 0x0`

4.52.2.4 `#define CM_NB_POSTFIX_SERVER 0x20`

4.52.2.5 `#define CM_NB_POSTFIX_DOMAINMASTERBROWSER 0x1b`

4.52.2.6 `#define CM_NB_POSTFIX_DOMAINCONTROLLER 0x1c`

4.52.2.7 `#define CM_NB_POSTFIX_MASTERBROWSER 0x1d`

4.52.3 Function Documentation

4.52.3.1 `NQ_STATUS nqRegisterNetBiosName (const NQ_CHAR * name)`

This function is called to register an abstract NetBIOS name. An abstract name is registered in addition to the host name and is served as its alias.

Parameters

<i>name</i>	Pointer to the name to register as a NetBIOS name.
-------------	--

Returns

`NQ_SUCCESS` or `NQ_FAIL`

See Also

[nqReleaseNetBiosName\(\)](#)

4.52.3.2 NQ_STATUS `nqReleaseNetBiosName (const NQ_CHAR * name)`

This function is called to unregister an abstract NetBIOS name.

Parameters

<i>name</i>	Pointer to the name to release as a NetBIOS name.
-------------	---

Returns

NQ_SUCCESS or NQ_FAIL

Note

No check is made that the name to release was registered with a [nqRegisterNetBiosName\(\)](#) call. It is possible to release the basic host name with this call.

See Also

[nqRegisterNetBiosName\(\)](#)

4.52.3.3 NQ_STATUS nqRegisterNetBiosNameEx (const NQ_CHAR * name, const NQ_BYTE flags)

This function is called to register an abstract NetBIOS name. An abstract name is registered in addition to the host name and is served as its alias. This is an extended version of [nqRegisterNetBiosName\(\)](#).

Parameters

<i>name</i>	Pointer to the name to register as a NetBIOS name.
<i>flags</i>	Combination of flags to designate unique or group name and postfix (e.g. CM_NB_UNIQUE CM_NB_POSTFIX_SERVER)

Returns

NQ_SUCCESS or NQ_FAIL

See Also

[nqReleaseNetBiosNameEx\(\)](#)

4.52.3.4 NQ_STATUS nqReleaseNetBiosNameEx (const NQ_CHAR * name, const NQ_BYTE flags)

This function is called to unregister an abstract NetBIOS name. This is an extended version of [nqReleaseNetBiosName\(\)](#).

Parameters

<i>name</i>	Pointer to the name to release as a NetBIOS name.
<i>flags</i>	Combination of flags to designate unique or group name and postfix (e.g. CM_NB_UNIQUE CM_NB_POSTFIX_SERVER)

Returns

NQ_SUCCESS or NQ_FAIL

Note

No check is made that the name to release was registered with a [nqRegisterNetBiosNameEx\(\)](#) call. It is possible to release the basic host name with this call.

See Also

[nqRegisterNetBiosNameEx\(\)](#)

4.53 Utils

Functions

- void [nqEnableTraceLog](#) ([NQ_BOOL](#) on)

4.53.1 Detailed Description

4.53.2 Function Documentation

4.53.2.1 void [nqEnableTraceLog](#) ([NQ_BOOL](#) on)

This defines trace availability

Parameters

<i>on</i>	Trace availability (TRUE for On, FALSE for off).
-----------	--

Note

- This function is only available when NQ was compiled with Internal Trace support as indicated by the `UD_NQ_INCLUDETRACE` parameter.

4.54 System API

Functions

- [NQ_BOOL syInit](#) (void)
- void [syStop](#) (void)

4.54.1 Detailed Description

4.54.2 Function Documentation

4.54.2.1 [NQ_BOOL syInit](#) (void)

Initialize SY module.

Returns

TRUE on success or *FALSE* on failure.

4.54.2.2 void [syStop](#) (void)

Shutdown SY module.

4.55 System Common API

Modules

- [File Information](#)
- [Printer](#)

4.55.1 Detailed Description

4.56 File Information

Modules

- [File Attributes](#)

Data Structures

- struct [SYFileInformation](#)
- struct [SYVolumeInformation](#)

4.56.1 Detailed Description

4.57 File Attributes

Macros

- #define SY_ATTR_READONLY 0x001
- #define SY_ATTR_HIDDEN 0x002
- #define SY_ATTR_SYSTEM 0x004
- #define SY_ATTR_VOLUME 0x008
- #define SY_ATTR_DIRECTORY 0x010
- #define SY_ATTR_ARCHIVE 0x020
- #define SY_ATTR_DEVICE 0x040
- #define SY_ATTR_NORMAL 0x080
- #define SY_ATTR_TEMPORARY 0x100
- #define SY_ATTR_SPARSE_FILE 0x200
- #define SY_ATTR_REPARSE_POINT 0x400
- #define SY_ATTR_COMPRESSED 0x800
- #define SY_ATTR_OFFLINE 0x1000
- #define SY_ATTR_NOT_CONTENT_INDEXED 0x2000
- #define SY_ATTR_ENCRYPTED 0x4000

4.57.1 Detailed Description

4.57.2 Macro Definition Documentation

4.57.2.1 #define SY_ATTR_READONLY 0x001

File attributes bits: the file is read-only.

4.57.2.2 #define SY_ATTR_HIDDEN 0x002

File attributes bits: the file is hidden.

4.57.2.3 #define SY_ATTR_SYSTEM 0x004

File attributes bits: the file is a system file.

4.57.2.4 #define SY_ATTR_VOLUME 0x008

File attributes bits: this is a volume.

4.57.2.5 #define SY_ATTR_DIRECTORY 0x010

File attributes bits: the file is a directory.

4.57.2.6 #define SY_ATTR_ARCHIVE 0x020

File attributes bits: the file is archived.

4.57.2.7 #define SY_ATTR_DEVICE 0x040

File attributes bits: this is a device.

4.57.2.8 #define SY_ATTR_NORMAL 0x080

File attributes bits: the file is a regular file.

4.57.2.9 #define SY_ATTR_TEMPORARY 0x100

File attributes bits: the file is temporary.

4.57.2.10 #define SY_ATTR_SPARSE_FILE 0x200

File attributes bits: the file is distributed.

4.57.2.11 #define SY_ATTR_REPARSE_POINT 0x400

File attributes bits: the file is a DFS reparse point.

4.57.2.12 #define SY_ATTR_COMPRESSED 0x800

File attributes bits: the file is compressed.

4.57.2.13 #define SY_ATTR_OFFLINE 0x1000

File attributes bits: the file is offline.

4.57.2.14 #define SY_ATTR_NOT_CONTENT_INDEXED 0x2000

File attributes bits: the file has indexed contents.

4.57.2.15 #define SY_ATTR_ENCRYPTED 0x4000

File attributes bits: the file is encrypted.

4.58 Printer

Modules

- [Printer Flags](#)
- [Printer Status Flags](#)
- [Printer Attribute Flags](#)
- [Printer Status Job](#)
- [Printer Priority Job](#)
- [Printer Control Commands](#)
- [Printer Job Control Commands](#)
- [Printer Form Flags](#)

Data Structures

- struct [SYTimeFragments](#)
- struct [SYDeviceMode](#)
- struct [SYPrinterDriver](#)
- struct [SYPrinterInfo](#)
- struct [SYPrintJobInfo](#)
- struct [SYPrintSize](#)
- struct [SYPrintRect](#)
- struct [SYPrintFormInfo](#)

Macros

- `#define SY_PRINTEROSVERSION_WIN 3`

4.58.1 Detailed Description

4.58.2 Macro Definition Documentation

4.58.2.1 `#define SY_PRINTEROSVERSION_WIN 3`

4.59 Printer Flags

Macros

- #define SY_DEVICEMODE_ORIENTATION 0x00000001
- #define SY_DEVICEMODE_PAPERSIZE 0x00000002
- #define SY_DEVICEMODE_PAPERLENGTH 0x00000004
- #define SY_DEVICEMODE_PAPERWIDTH 0x00000008
- #define SY_DEVICEMODE_SCALE 0x00000010
- #define SY_DEVICEMODE_POSITION 0x00000020
- #define SY_DEVICEMODE_NUP 0x00000040
- #define SY_DEVICEMODE_COPIES 0x00000080
- #define SY_DEVICEMODE_DEFAULTSOURCE 0x00000100
- #define SY_DEVICEMODE_PRINTQUALITY 0x00000400
- #define SY_DEVICEMODE_COLOR 0x00000800
- #define SY_DEVICEMODE_DUPLEX 0x00001000
- #define SY_DEVICEMODE_YRESOLUTION 0x00002000
- #define SY_DEVICEMODE_TTOPTION 0x00004000
- #define SY_DEVICEMODE_COLLATE 0x00008000
- #define SY_DEVICEMODE_FORMNAME 0x00010000
- #define SY_DEVICEMODE_LOGPIXELS 0x00020000
- #define SY_DEVICEMODE_BITSPERPEL 0x00040000
- #define SY_DEVICEMODE_PELWIDTH 0x00080000
- #define SY_DEVICEMODE_PELHEIGHT 0x00100000
- #define SY_DEVICEMODE_DISPLAYFLAGS 0x00200000
- #define SY_DEVICEMODE_DISPLAYFREQUENCY 0x00400000
- #define SY_DEVICEMODE_ICMMETHOD 0x00800000
- #define SY_DEVICEMODE_ICMINTENT 0x01000000
- #define SY_DEVICEMODE_MEDIATYPE 0x02000000
- #define SY_DEVICEMODE_DITHERTYPE 0x04000000
- #define SY_DEVICEMODE_PANNINGWIDTH 0x08000000
- #define SY_DEVICEMODE_PANNINGHEIGHT 0x10000000

4.59.1 Detailed Description

Field flags in [SYPrinterInfo](#) struct

4.59.2 Macro Definition Documentation

- 4.59.2.1 #define SY_DEVICEMODE_ORIENTATION 0x00000001
- 4.59.2.2 #define SY_DEVICEMODE_PAPERSIZE 0x00000002
- 4.59.2.3 #define SY_DEVICEMODE_PAPERLENGTH 0x00000004
- 4.59.2.4 #define SY_DEVICEMODE_PAPERWIDTH 0x00000008
- 4.59.2.5 #define SY_DEVICEMODE_SCALE 0x00000010
- 4.59.2.6 #define SY_DEVICEMODE_POSITION 0x00000020
- 4.59.2.7 #define SY_DEVICEMODE_NUP 0x00000040

4.59.2.8 #define SY_DEVICEMODE_COPIES 0x00000080

4.59.2.9 #define SY_DEVICEMODE_DEFAULTSOURCE 0x00000100

4.59.2.10 #define SY_DEVICEMODE_PRINTQUALITY 0x00000400

4.59.2.11 #define SY_DEVICEMODE_COLOR 0x00000800

4.59.2.12 #define SY_DEVICEMODE_DUPLEX 0x00001000

4.59.2.13 #define SY_DEVICEMODE_YRESOLUTION 0x00002000

4.59.2.14 #define SY_DEVICEMODE_TTOPTION 0x00004000

4.59.2.15 #define SY_DEVICEMODE_COLLATE 0x00008000

4.59.2.16 #define SY_DEVICEMODE_FORMNAME 0x00010000

4.59.2.17 #define SY_DEVICEMODE_LOGPIXELS 0x00020000

4.59.2.18 #define SY_DEVICEMODE_BITSPERPEL 0x00040000

4.59.2.19 #define SY_DEVICEMODE_PELWIDTH 0x00080000

4.59.2.20 #define SY_DEVICEMODE_PELHEIGHT 0x00100000

4.59.2.21 #define SY_DEVICEMODE_DISPLAYFLAGS 0x00200000

4.59.2.22 #define SY_DEVICEMODE_DISPLAYFREQUENCY 0x00400000

4.59.2.23 #define SY_DEVICEMODE_ICMMETHOD 0x00800000

4.59.2.24 #define SY_DEVICEMODE_ICMINTENT 0x01000000

4.59.2.25 #define SY_DEVICEMODE_MEDIATYPE 0x02000000

4.59.2.26 #define SY_DEVICEMODE_DITHERTYPE 0x04000000

4.59.2.27 #define SY_DEVICEMODE_PANNINGWIDTH 0x08000000

4.59.2.28 #define SY_DEVICEMODE_PANNINGHEIGHT 0x10000000

4.60 Printer Status Flags

Macros

- #define SY_PRINTERSTATUS_PAUSED 0x00000001
- #define SY_PRINTERSTATUS_ERROR 0x00000002
- #define SY_PRINTERSTATUS_PENDINGDELETION 0x00000004
- #define SY_PRINTERSTATUS_PAPERJAM 0x00000008
- #define SY_PRINTERSTATUS_PAPEROUT 0x00000010
- #define SY_PRINTERSTATUS_MANUALFEED 0x00000020
- #define SY_PRINTERSTATUS_PAPERPROBLEM 0x00000040
- #define SY_PRINTERSTATUS_OFFLINE 0x00000080
- #define SY_PRINTERSTATUS_IOACTIVE 0x00000100
- #define SY_PRINTERSTATUS_BUSY 0x00000200
- #define SY_PRINTERSTATUS_PRINTING 0x00000400
- #define SY_PRINTERSTATUS_OUTPUTBINFULL 0x00000800
- #define SY_PRINTERSTATUS_NOTAVAILABLE 0x00001000
- #define SY_PRINTERSTATUS_WAITING 0x00002000
- #define SY_PRINTERSTATUS_PROCESSING 0x00004000
- #define SY_PRINTERSTATUS_INITIALIZING 0x00008000
- #define SY_PRINTERSTATUS_WARMINGUP 0x00010000
- #define SY_PRINTERSTATUS_TONERLOW 0x00020000
- #define SY_PRINTERSTATUS_NOTONER 0x00040000
- #define SY_PRINTERSTATUS_PAGEPUNT 0x00080000
- #define SY_PRINTERSTATUS_USERINTERVENTION 0x00100000
- #define SY_PRINTERSTATUS_OUTOFMEMORY 0x00200000
- #define SY_PRINTERSTATUS_DOOROPEN 0x00400000
- #define SY_PRINTERSTATUS_SERVERUNKNOWN 0x00800000
- #define SY_PRINTERSTATUS_POWERSAVE 0x01000000

4.60.1 Detailed Description

Status flags

4.60.2 Macro Definition Documentation

- 4.60.2.1 #define SY_PRINTERSTATUS_PAUSED 0x00000001
- 4.60.2.2 #define SY_PRINTERSTATUS_ERROR 0x00000002
- 4.60.2.3 #define SY_PRINTERSTATUS_PENDINGDELETION 0x00000004
- 4.60.2.4 #define SY_PRINTERSTATUS_PAPERJAM 0x00000008
- 4.60.2.5 #define SY_PRINTERSTATUS_PAPEROUT 0x00000010
- 4.60.2.6 #define SY_PRINTERSTATUS_MANUALFEED 0x00000020
- 4.60.2.7 #define SY_PRINTERSTATUS_PAPERPROBLEM 0x00000040
- 4.60.2.8 #define SY_PRINTERSTATUS_OFFLINE 0x00000080
- 4.60.2.9 #define SY_PRINTERSTATUS_IOACTIVE 0x00000100

4.60.2.10 #define SY_PRINTERSTATUS_BUSY 0x00000200

4.60.2.11 #define SY_PRINTERSTATUS_PRINTING 0x00000400

4.60.2.12 #define SY_PRINTERSTATUS_OUTPUTBINFULL 0x00000800

4.60.2.13 #define SY_PRINTERSTATUS_NOTAVAILABLE 0x00001000

4.60.2.14 #define SY_PRINTERSTATUS_WAITING 0x00002000

4.60.2.15 #define SY_PRINTERSTATUS_PROCESSING 0x00004000

4.60.2.16 #define SY_PRINTERSTATUS_INITIALIZING 0x00008000

4.60.2.17 #define SY_PRINTERSTATUS_WARMINGUP 0x00010000

4.60.2.18 #define SY_PRINTERSTATUS_TONERLOW 0x00020000

4.60.2.19 #define SY_PRINTERSTATUS_NOTONER 0x00040000

4.60.2.20 #define SY_PRINTERSTATUS_PAGEPUNT 0x00080000

4.60.2.21 #define SY_PRINTERSTATUS_USERINTERVENTION 0x00100000

4.60.2.22 #define SY_PRINTERSTATUS_OUTOFMEMORY 0x00200000

4.60.2.23 #define SY_PRINTERSTATUS_DOOROPEN 0x00400000

4.60.2.24 #define SY_PRINTERSTATUS_SERVERUNKNOWN 0x00800000

4.60.2.25 #define SY_PRINTERSTATUS_POWERSAVE 0x01000000

4.61 Printer Attribute Flags

Macros

- #define SY_PRINTERATTR_QUEUED 0x0001
- #define SY_PRINTERATTR_DIRECT 0x0002
- #define SY_PRINTERATTR_DEFAULT 0x0004
- #define SY_PRINTERATTR_SHARED 0x0008
- #define SY_PRINTERATTR_NETWORK 0x0010
- #define SY_PRINTERATTR_RESERVED 0x0020 /* */
- #define SY_PRINTERATTR_LOCAL 0x0040
- #define SY_PRINTERATTR_DEVQUERY 0x0080
- #define SY_PRINTERATTR_KEEPCJOBS 0x0100
- #define SY_PRINTERATTR_LIFO 0x0200
- #define SY_PRINTERATTR_OFFLINE 0x0400
- #define SY_PRINTERATTR_BIDI 0x0800
- #define SY_PRINTERATTR_RAWONLY 0x1000
- #define SY_PRINTERATTR_PUBLISHED 0x2000
- #define SY_PRINTERATTR_EXPAND 0x4000
- #define SY_PRINTERATTR_CONTAINER 0x8000
- #define SY_PRINTERATTR_ICON1 0x10000
- #define SY_PRINTERATTR_ICON2 0x20000
- #define SY_PRINTERATTR_ICON3 0x40000
- #define SY_PRINTERATTR_ICON4 0x80000
- #define SY_PRINTERATTR_ICON5 0x100000
- #define SY_PRINTERATTR_ICON6 0x200000
- #define SY_PRINTERATTR_ICON7 0x400000
- #define SY_PRINTERATTR_ICON8 0x800000

4.61.1 Detailed Description

Attribute flags

4.61.2 Macro Definition Documentation

4.61.2.1 #define SY_PRINTERATTR_QUEUED 0x0001

Spools first, then prints

4.61.2.2 #define SY_PRINTERATTR_DIRECT 0x0002

Jobs are printed directly, without spooling

4.61.2.3 #define SY_PRINTERATTR_DEFAULT 0x0004

9x/ME only - Default printer

4.61.2.4 #define SY_PRINTERATTR_SHARED 0x0008

Shared printer

4.61.2.5 #define SY_PRINTERATTR_NETWORK 0x0010

Network printer

4.61.2.6 #define SY_PRINTERATTR_RESERVED 0x0020 /* */**4.61.2.7 #define SY_PRINTERATTR_LOCAL 0x0040**

Local printer

4.61.2.8 #define SY_PRINTERATTR_DEVQUERY 0x0080

Enable DevQueryPrint

4.61.2.9 #define SY_PRINTERATTR_KEEPJOBS 0x0100

Do not delete jobs after printing

4.61.2.10 #define SY_PRINTERATTR_LIFO 0x0200

Jobs are printed in back order

4.61.2.11 #define SY_PRINTERATTR_OFFLINE 0x0400

9x/ME only

4.61.2.12 #define SY_PRINTERATTR_BIDI 0x0800

9x/ME only - Bidirectional operations

4.61.2.13 #define SY_PRINTERATTR_RAWONLY 0x1000

Only raw data may be spooled

4.61.2.14 #define SY_PRINTERATTR_PUBLISHED 0x2000

Printer is published in the directory

4.61.2.15 #define SY_PRINTERATTR_EXPAND 0x4000**4.61.2.16 #define SY_PRINTERATTR_CONTAINER 0x8000****4.61.2.17 #define SY_PRINTERATTR_ICON1 0x10000****4.61.2.18 #define SY_PRINTERATTR_ICON2 0x20000****4.61.2.19 #define SY_PRINTERATTR_ICON3 0x40000****4.61.2.20 #define SY_PRINTERATTR_ICON4 0x80000**

4.61.2.21 #define SY_PRINTERATTR_ICON5 0x100000

4.61.2.22 #define SY_PRINTERATTR_ICON6 0x200000

4.61.2.23 #define SY_PRINTERATTR_ICON7 0x400000

4.61.2.24 #define SY_PRINTERATTR_ICON8 0x800000

4.62 Printer Status Job

Macros

- #define SY_PRINTJOBSTATUS_PAUSED 0x00000001
- #define SY_PRINTJOBSTATUS_ERROR 0x00000002
- #define SY_PRINTJOBSTATUS_DELETING 0x00000004
- #define SY_PRINTJOBSTATUS_SPOOLING 0x00000008
- #define SY_PRINTJOBSTATUS_PRINTING 0x00000010
- #define SY_PRINTJOBSTATUS_OFFLINE 0x00000020
- #define SY_PRINTJOBSTATUS_PAPERPROBLEM 0x00000040
- #define SY_PRINTJOBSTATUS_PRINTED 0x00000080
- #define SY_PRINTJOBSTATUS_DELETED 0x00000100
- #define SY_PRINTJOBSTATUS_BLOCKED 0x00000200
- #define SY_PRINTJOBSTATUS_USERINTERVENTION 0x00000400
- #define SY_PRINTJOBSTATUS_RESTART 0x00000800

4.62.1 Detailed Description

Values in the "status" field of [SYPrintJobInfo](#) structure

4.62.2 Macro Definition Documentation

4.62.2.1 #define SY_PRINTJOBSTATUS_PAUSED 0x00000001

Job is paused.

4.62.2.2 #define SY_PRINTJOBSTATUS_ERROR 0x00000002

An error is associated with the job.

4.62.2.3 #define SY_PRINTJOBSTATUS_DELETING 0x00000004

Job is being deleted.

4.62.2.4 #define SY_PRINTJOBSTATUS_SPOOLING 0x00000008

Paper jam on printer.

4.62.2.5 #define SY_PRINTJOBSTATUS_PRINTING 0x00000010

Job is being printed.

4.62.2.6 #define SY_PRINTJOBSTATUS_OFFLINE 0x00000020

Printer is offline.

4.62.2.7 #define SY_PRINTJOBSTATUS_PAPERPROBLEM 0x00000040

Paper jam or out of paper.

4.62.2.8 #define SY_PRINTJOBSTATUS_PRINTED 0x00000080

Job has printed.

4.62.2.9 #define SY_PRINTJOBSTATUS_DELETED 0x00000100

Job has been deleted.

4.62.2.10 #define SY_PRINTJOBSTATUS_BLOCKED 0x00000200

The driver cannot print the job.

4.62.2.11 #define SY_PRINTJOBSTATUS_USERINTERVENTION 0x00000400

User intervention required.

4.62.2.12 #define SY_PRINTJOBSTATUS_RESTART 0x00000800

Job has been restarted.

4.63 Printer Priority Job

Macros

- #define [SY_PRINTJOB_MINPRIORITY](#) 1
- #define [SY_PRINTJOB_MAXPRIORITY](#) 99
- #define [SY_PRINTJOB_DEFPPRIORITY](#) 1

4.63.1 Detailed Description

Values in the "priority" field of [SYPrintJobInfo](#) structure

4.63.2 Macro Definition Documentation

4.63.2.1 #define [SY_PRINTJOB_MINPRIORITY](#) 1

Minimum priority.

4.63.2.2 #define [SY_PRINTJOB_MAXPRIORITY](#) 99

Maximum priority.

4.63.2.3 #define [SY_PRINTJOB_DEFPPRIORITY](#) 1

Default priority.

4.64 Printer Control Commands

Macros

- #define SY_PRINTERCONTROL_PAUSE 1
- #define SY_PRINTERCONTROL_RESUME 2
- #define SY_PRINTERCONTROL_PURGE 3
- #define SY_PRINTERCONTROL_SETSTATUS 4

4.64.1 Detailed Description

Printer control commands

4.64.2 Macro Definition Documentation

4.64.2.1 #define SY_PRINTERCONTROL_PAUSE 1

Pause printer

4.64.2.2 #define SY_PRINTERCONTROL_RESUME 2

Resume printer

4.64.2.3 #define SY_PRINTERCONTROL_PURGE 3

Purge all jobs

4.64.2.4 #define SY_PRINTERCONTROL_SETSTATUS 4

Force printer status

4.65 Printer Job Control Commands

Macros

- #define SY_PRINTJOB_COM_PAUSE 1
- #define SY_PRINTJOB_COM_RESUME 2
- #define SY_PRINTJOB_COM_CANCEL 3
- #define SY_PRINTJOB_COM_RESTART 4
- #define SY_PRINTJOB_COM_DELETE 5
- #define SY_PRINTJOB_COM_SENDDTOPRINTER 6
- #define SY_PRINTJOB_COM_LASTPAGEEJECTED 7

4.65.1 Detailed Description

Job control commands

4.65.2 Macro Definition Documentation

4.65.2.1 #define SY_PRINTJOB_COM_PAUSE 1

Pause a job

4.65.2.2 #define SY_PRINTJOB_COM_RESUME 2

Resume a paused job

4.65.2.3 #define SY_PRINTJOB_COM_CANCEL 3

Cancel a job (NT4.0 and later only)

4.65.2.4 #define SY_PRINTJOB_COM_RESTART 4

Restart the print job. A job can only be restarted if it was printing

4.65.2.5 #define SY_PRINTJOB_COM_DELETE 5

Windows NT 4.0 and later: Delete the print job

4.65.2.6 #define SY_PRINTJOB_COM_SENDDTOPRINTER 6

Windows NT 4.0 and later: Used by port monitors to end the print job.

4.65.2.7 #define SY_PRINTJOB_COM_LASTPAGEEJECTED 7

Windows NT 4.0 and later: Used by language monitors to end the print job.

4.66 Printer Form Flags

Macros

- #define [SY_PRINTFORMFLAG_USER](#) 0x00000000
- #define [SY_PRINTFORMFLAG_BUILTIN](#) 0x00000001
- #define [SY_PRINTFORMFLAG_PRINTER](#) 0x00000002

4.66.1 Detailed Description

4.66.2 Macro Definition Documentation

4.66.2.1 #define SY_PRINTFORMFLAG_USER 0x00000000

User-defined

4.66.2.2 #define SY_PRINTFORMFLAG_BUILTIN 0x00000001

Spooler level

4.66.2.3 #define SY_PRINTFORMFLAG_PRINTER 0x00000002

Printer level

4.67 User Defined

Modules

- [SMB Errors](#)
- [SMB Error Codes](#)
- [Internal NT Statuses](#)
- [NQ Error Codes](#)
- [TCP and NetBIOS Layer Error Codes](#)
- [Authentication Statuses](#)
- [Functions](#)
- [Event Log Constants](#)

4.67.1 Detailed Description

4.68 SMB Errors

Macros

```

• #define SMB_DOSERR(_class, _code) (((NQ_UINT32)_code << 16) | _class)
• #define CLASS_ERR_MODULE(_code) ((NQ_UINT32)_code << 16)
• #define IS_DOS_ERROR(err) ((err & CLASS_ERR_MODULE(1)) == CLASS_ERR_MODULE(1))
• #define SMB_ERRsuccess 0x00000000
• #define DOS_ERRbadfunc SMB_DOSERR(1, 1)
• #define DOS_ERRbadfile SMB_DOSERR(1, 2)
• #define DOS_ERRbadpath SMB_DOSERR(1, 3)
• #define DOS_ERRnofids SMB_DOSERR(1, 4)
• #define DOS_ERRnoaccess SMB_DOSERR(1, 5)
• #define DOS_ERRbadfid SMB_DOSERR(1, 6)
• #define DOS_ERRbadmcb SMB_DOSERR(1, 7)
• #define DOS_ERRnomem SMB_DOSERR(1, 8)
• #define DOS_ERRbadmem SMB_DOSERR(1, 9)
• #define DOS_ERRbadenv SMB_DOSERR(1, 10)
• #define DOS_ERRbadformat SMB_DOSERR(1, 11)
• #define DOS_ERRbadaccess SMB_DOSERR(1, 12)
• #define DOS_ERRbaddata SMB_DOSERR(1, 13)
• #define DOS_ERRbaddrive SMB_DOSERR(1, 15)
• #define DOS_ERRremcd SMB_DOSERR(1, 16)
• #define DOS_ERRdiffdevice SMB_DOSERR(1, 17)
• #define DOS_ERRnofiles SMB_DOSERR(1, 18)
• #define DOS_ERRbadshare SMB_DOSERR(1, 32)
• #define DOS_ERRlock SMB_DOSERR(1, 33)
• #define DOS_ERRdentsupportipc SMB_DOSERR(1, 66)
• #define DOS_ERRnoshare SMB_DOSERR(1, 67)
• #define DOS_ERRfileexists SMB_DOSERR(1, 80)
• #define DOS_ERRbaddirectory SMB_DOSERR(1, 87)
• #define DOS_ERRinsufficientbuffer SMB_DOSERR(1, 122)
• #define DOS_ERRinvalidname SMB_DOSERR(1, 123)
• #define DOS_ERRdirnotempty SMB_DOSERR(1, 145)
• #define DOS_ERRalreadyexists SMB_DOSERR(1, 183)
• #define DOS_ERRbadpipe SMB_DOSERR(1, 230)
• #define DOS_ERRpipebusy SMB_DOSERR(1, 231)
• #define DOS_ERRpipeclosing SMB_DOSERR(1, 232)
• #define DOS_ERRnotconnected SMB_DOSERR(1, 233)
• #define DOS_ERRmoredata SMB_DOSERR(1, 234)
• #define IS_SRV_ERROR(err) ((err & CLASS_ERR_MODULE(2)) == CLASS_ERR_MODULE(2))
• #define SRV_ERRerror SMB_DOSERR(2, 1)
• #define SRV_ERRbadpw SMB_DOSERR(2, 2)
• #define SRV_ERRaccess SMB_DOSERR(2, 4)
• #define SRV_ERRinvtid SMB_DOSERR(2, 5)
• #define SRV_ERRinvnetname SMB_DOSERR(2, 6)
• #define SRV_ERRinvdevice SMB_DOSERR(2, 7)
• #define SRV_ERRqfull SMB_DOSERR(2, 49)
• #define SRV_ERRqtoobig SMB_DOSERR(2, 50)
• #define SRV_ERRqeof SMB_DOSERR(2, 51)
• #define SRV_ERRinvfid SMB_DOSERR(2, 52)
• #define SRV_ERRsmbcmd SMB_DOSERR(2, 64)
• #define SRV_ERRsrverror SMB_DOSERR(2, 65)
• #define SRV_ERRfilespecs SMB_DOSERR(2, 67)

```

- `#define SRV_ERRbadpermits SMB_DOSERR(2, 69)`
- `#define SRV_ERRsetattrmode SMB_DOSERR(2, 71)`
- `#define SRV_ERRpaused SMB_DOSERR(2, 81)`
- `#define SRV_ERRmsgoff SMB_DOSERR(2, 82)`
- `#define SRV_ERRnoroom SMB_DOSERR(2, 83)`
- `#define SRV_ERRrmuns SMB_DOSERR(2, 87)`
- `#define SRV_ERRtimeout SMB_DOSERR(2, 88)`
- `#define SRV_ERRnoresource SMB_DOSERR(2, 89)`
- `#define SRV_ERRtoomanyuids SMB_DOSERR(2, 90)`
- `#define SRV_ERRinvuid SMB_DOSERR(2, 91)`
- `#define SRV_ERRusempx SMB_DOSERR(2, 250)`
- `#define SRV_ERRusestd SMB_DOSERR(2, 251)`
- `#define SRV_ERRcontmpx SMB_DOSERR(2, 252)`
- `#define SRV_ERRnosupport SMB_DOSERR(2, 65535)`
- `#define IS_HRD_ERROR(err) ((err & CLASS_ERR_MODULE(3)) == CLASS_ERR_MODULE(3))`
- `#define HRD_ERRnowrite SMB_DOSERR(3, 19)`
- `#define HRD_ERRbadunit SMB_DOSERR(3, 20)`
- `#define HRD_ERRnotready SMB_DOSERR(3, 21)`
- `#define HRD_ERRbadcmd SMB_DOSERR(3, 22)`
- `#define HRD_ERRdata SMB_DOSERR(3, 23)`
- `#define HRD_ERRbadreq SMB_DOSERR(3, 24)`
- `#define HRD_ERRseek SMB_DOSERR(3, 25)`
- `#define HRD_ERRbadmedia SMB_DOSERR(3, 26)`
- `#define HRD_ERRbadsector SMB_DOSERR(3, 27)`
- `#define HRD_ERRnopaper SMB_DOSERR(3, 28)`
- `#define HRD_ERRwrite SMB_DOSERR(3, 29)`
- `#define HRD_ERRread SMB_DOSERR(3, 30)`
- `#define HRD_ERRgeneral SMB_DOSERR(3, 31)`
- `#define HRD_ERRbadshare SMB_DOSERR(3, 32)`
- `#define HRD_ERRlock SMB_DOSERR(3, 33)`
- `#define HRD_ERRwrongdisk SMB_DOSERR(3, 34)`
- `#define HRD_ERRFCBUnavail SMB_DOSERR(3, 35)`
- `#define HRD_ERRsharebufexc SMB_DOSERR(3, 36)`
- `#define HRD_ERRdiskfull SMB_DOSERR(3, 39)`

4.68.1 Detailed Description

SMB error in "DOS" format consists of an error class and error code

4.68.2 Macro Definition Documentation

4.68.2.1 `#define SMB_DOSERR(_class, _code) (((NQ_UINT32)_code << 16) | _class)`

4.68.2.2 `#define CLASS_ERR_MODULE(_code) ((NQ_UINT32)_code << 16)`

4.68.2.3 `#define IS_DOS_ERROR(err) ((err & CLASS_ERR_MODULE(1)) == CLASS_ERR_MODULE(1))`

4.68.2.4 `#define SMB_ERRsuccess 0x00000000`

4.68.2.5 `#define DOS_ERRbadfunc SMB_DOSERR(1, 1)`

4.68.2.6 `#define DOS_ERRbadfile SMB_DOSERR(1, 2)`

4.68.2.7 #define DOS_ERRbadpath SMB_DOSERR(1, 3)

4.68.2.8 #define DOS_ERRnofids SMB_DOSERR(1, 4)

4.68.2.9 #define DOS_ERRnoaccess SMB_DOSERR(1, 5)

4.68.2.10 #define DOS_ERRbadfid SMB_DOSERR(1, 6)

4.68.2.11 #define DOS_ERRbadmcb SMB_DOSERR(1, 7)

4.68.2.12 #define DOS_ERRnomem SMB_DOSERR(1, 8)

4.68.2.13 #define DOS_ERRbadmem SMB_DOSERR(1, 9)

4.68.2.14 #define DOS_ERRbadenv SMB_DOSERR(1, 10)

4.68.2.15 #define DOS_ERRbadformat SMB_DOSERR(1, 11)

4.68.2.16 #define DOS_ERRbadaccess SMB_DOSERR(1, 12)

4.68.2.17 #define DOS_ERRbaddata SMB_DOSERR(1, 13)

4.68.2.18 #define DOS_ERRbaddrive SMB_DOSERR(1, 15)

4.68.2.19 #define DOS_ERRremcd SMB_DOSERR(1, 16)

4.68.2.20 #define DOS_ERRdiffdevice SMB_DOSERR(1, 17)

4.68.2.21 #define DOS_ERRnofiles SMB_DOSERR(1, 18)

4.68.2.22 #define DOS_ERRbadshare SMB_DOSERR(1, 32)

4.68.2.23 #define DOS_ERRlock SMB_DOSERR(1, 33)

4.68.2.24 #define DOS_ERRdentsupportipc SMB_DOSERR(1, 66)

4.68.2.25 #define DOS_ERRnoshare SMB_DOSERR(1, 67)

4.68.2.26 #define DOS_ERRfileexists SMB_DOSERR(1, 80)

4.68.2.27 #define DOS_ERRbaddirectory SMB_DOSERR(1, 87)

4.68.2.28 #define DOS_ERRinsufficientbuffer SMB_DOSERR(1, 122)

4.68.2.29 #define DOS_ERRinvalidname SMB_DOSERR(1, 123)

4.68.2.30 #define DOS_ERRdirnotempty SMB_DOSERR(1, 145)

4.68.2.31 #define DOS_ERRalreadyexists SMB_DOSERR(1, 183)

4.68.2.32 #define DOS_ERRbadpipe SMB_DOSERR(1, 230)

4.68.2.33 #define DOS_ERRpipebusy SMB_DOSERR(1, 231)

4.68.2.34 #define DOS_ERRpipeclosing SMB_DOSERR(1, 232)

4.68.2.35 **#define DOS_ERRnotconnected SMB_DOSERR(1, 233)**

4.68.2.36 **#define DOS_ERRmoredata SMB_DOSERR(1, 234)**

4.68.2.37 **#define IS_SRV_ERROR(err) ((err & CLASS_ERR_MODULE(2)) == CLASS_ERR_MODULE(2))**

4.68.2.38 **#define SRV_ERRerror SMB_DOSERR(2, 1)**

4.68.2.39 **#define SRV_ERRbadpw SMB_DOSERR(2, 2)**

4.68.2.40 **#define SRV_ERRaccess SMB_DOSERR(2, 4)**

4.68.2.41 **#define SRV_ERRinvtid SMB_DOSERR(2, 5)**

4.68.2.42 **#define SRV_ERRinvnetname SMB_DOSERR(2, 6)**

4.68.2.43 **#define SRV_ERRinvdevice SMB_DOSERR(2, 7)**

4.68.2.44 **#define SRV_ERRqfull SMB_DOSERR(2, 49)**

4.68.2.45 **#define SRV_ERRqtoobig SMB_DOSERR(2, 50)**

4.68.2.46 **#define SRV_ERRqeof SMB_DOSERR(2, 51)**

4.68.2.47 **#define SRV_ERRinvfid SMB_DOSERR(2, 52)**

4.68.2.48 **#define SRV_ERRsmbcmd SMB_DOSERR(2, 64)**

4.68.2.49 **#define SRV_ERRsrverror SMB_DOSERR(2, 65)**

4.68.2.50 **#define SRV_ERRfilespecs SMB_DOSERR(2, 67)**

4.68.2.51 **#define SRV_ERRbadpermits SMB_DOSERR(2, 69)**

4.68.2.52 **#define SRV_ERRsetattrmode SMB_DOSERR(2, 71)**

4.68.2.53 **#define SRV_ERRpaused SMB_DOSERR(2, 81)**

4.68.2.54 **#define SRV_ERRmsgoff SMB_DOSERR(2, 82)**

4.68.2.55 **#define SRV_ERRnroom SMB_DOSERR(2, 83)**

4.68.2.56 **#define SRV_ERRrmuns SMB_DOSERR(2, 87)**

4.68.2.57 **#define SRV_ERRtimeout SMB_DOSERR(2, 88)**

4.68.2.58 **#define SRV_ERRnoresource SMB_DOSERR(2, 89)**

4.68.2.59 **#define SRV_ERRtoomanyuids SMB_DOSERR(2, 90)**

4.68.2.60 **#define SRV_ERRinvuid SMB_DOSERR(2, 91)**

4.68.2.61 **#define SRV_ERRusempx SMB_DOSERR(2, 250)**

4.68.2.62 **#define SRV_ERRusestd SMB_DOSERR(2, 251)**

4.68.2.63 #define SRV_ERRcontmpx SMB_DOSERR(2, 252)

4.68.2.64 #define SRV_ERRnosupport SMB_DOSERR(2, 65535)

4.68.2.65 #define IS_HRD_ERROR(err) ((err & CLASS_ERR_MODULE(3)) == CLASS_ERR_MODULE(3))

4.68.2.66 #define HRD_ERRnowrite SMB_DOSERR(3, 19)

4.68.2.67 #define HRD_ERRbadunit SMB_DOSERR(3, 20)

4.68.2.68 #define HRD_ERRnotready SMB_DOSERR(3, 21)

4.68.2.69 #define HRD_ERRbadcmd SMB_DOSERR(3, 22)

4.68.2.70 #define HRD_ERRdata SMB_DOSERR(3, 23)

4.68.2.71 #define HRD_ERRbadreq SMB_DOSERR(3, 24)

4.68.2.72 #define HRD_ERRseek SMB_DOSERR(3, 25)

4.68.2.73 #define HRD_ERRbadmedia SMB_DOSERR(3, 26)

4.68.2.74 #define HRD_ERRbadsector SMB_DOSERR(3, 27)

4.68.2.75 #define HRD_ERRnopaper SMB_DOSERR(3, 28)

4.68.2.76 #define HRD_ERRwrite SMB_DOSERR(3, 29)

4.68.2.77 #define HRD_ERRread SMB_DOSERR(3, 30)

4.68.2.78 #define HRD_ERRgeneral SMB_DOSERR(3, 31)

4.68.2.79 #define HRD_ERRbadshare SMB_DOSERR(3, 32)

4.68.2.80 #define HRD_ERRlock SMB_DOSERR(3, 33)

4.68.2.81 #define HRD_ERRwrongdisk SMB_DOSERR(3, 34)

4.68.2.82 #define HRD_ERRFCBUnavail SMB_DOSERR(3, 35)

4.68.2.83 #define HRD_ERRsharebufexc SMB_DOSERR(3, 36)

4.68.2.84 #define HRD_ERRdiskfull SMB_DOSERR(3, 39)

4.69 SMB Error Codes

Macros

- #define `SMB_STATUS_OK` 0xc0000000
- #define `SMB_STATUS_INVALID` 0xffffffff
- #define `SMB_STATUS_SUCCESS` 0x00000000
- #define `SMB_STATUS_ENUMDIR` 0x0000010c
- #define `SMB_STATUS_PENDING` 0x00000103
- #define `SMB_STATUS_MORE_ENTRIES` 0x00000105
- #define `SMB_STATUS_SOME_UNMAPPED` 0x00000107
- #define `SMB_STATUS_NOTIFY_CLEANUP` 0x0000010b
- #define `SMB_STATUS_INSUFFICIENT_BUFFER` 0x0000007a
- #define `SMB_STATUS_UNKNOWN_PRINTER_DRIVER` 0x00000705
- #define `SMB_STATUS_INVALID_DATATYPE` 0x0000070c
- #define `SMB_STATUS_BUFFER_OVERFLOW` 0x80000005
- #define `SMB_STATUS_NO_MORE_FILES` 0x80000006
- #define `SMB_STATUS_DEVICE_OFF_LINE` 0x80000010
- #define `SMB_STATUS_NO_MORE_ENTRIES` 0x8000001a
- #define `SMB_STATUS_STOPPED_ON_SYMLINK` 0x8000002D
- #define `SMB_STATUS_UNSUCCESSFUL` 0xc0000001
- #define `SMB_STATUS_NOT_IMPLEMENTED` 0xc0000002
- #define `SMB_STATUS_INVALID_INFO_CLASS` 0xc0000003
- #define `SMB_STATUS_INFO_LENGTH_MISMATCH` 0xc0000004
- #define `SMB_STATUS_ACCESS_VIOLATION` 0xc0000005
- #define `SMB_STATUS_IN_PAGE_ERROR` 0xc0000006
- #define `SMB_STATUS_PAGEFILE_QUOTA` 0xc0000007
- #define `SMB_STATUS_INVALID_HANDLE` 0xc0000008
- #define `SMB_STATUS_BAD_INITIAL_STACK` 0xc0000009
- #define `SMB_STATUS_BAD_INITIAL_PC` 0xc000000a
- #define `SMB_STATUS_INVALID_CID` 0xc000000b
- #define `SMB_STATUS_TIMER_NOT_CANCELED` 0xc000000c
- #define `SMB_STATUS_INVALID_PARAMETER` 0xc000000d
- #define `SMB_STATUS_NO_SUCH_DEVICE` 0xc000000e
- #define `SMB_STATUS_NO_SUCH_FILE` 0xc000000f
- #define `SMB_STATUS_INVALID_DEVICE_REQUEST` 0xc0000010
- #define `SMB_STATUS_END_OF_FILE` 0xc0000011
- #define `SMB_STATUS_WRONG_VOLUME` 0xc0000012
- #define `SMB_STATUS_NO_MEDIA_IN_DEVICE` 0xc0000013
- #define `SMB_STATUS_UNRECOGNIZED_MEDIA` 0xc0000014
- #define `SMB_STATUS_NONEXISTENT_SECTOR` 0xc0000015
- #define `SMB_STATUS_MORE_PROCESSING_REQUIRED` 0xc0000016
- #define `SMB_STATUS_NO_MEMORY` 0xc0000017
- #define `SMB_STATUS_CONFLICTING_ADDRESSES` 0xc0000018
- #define `SMB_STATUS_NOT_MAPPED_VIEW` 0xc0000019
- #define `SMB_STATUS_UNABLE_TO_FREE_VM` 0xc000001a
- #define `SMB_STATUS_UNABLE_TO_DELETE_SECTION` 0xc000001b
- #define `SMB_STATUS_INVALID_SYSTEM_SERVICE` 0xc000001c
- #define `SMB_STATUS_ILLEGAL_INSTRUCTION` 0xc000001d
- #define `SMB_STATUS_INVALID_LOCK_SEQUENCE` 0xc000001e
- #define `SMB_STATUS_INVALID_VIEW_SIZE` 0xc000001f
- #define `SMB_STATUS_INVALID_FILE_FOR_SECTION` 0xc0000020
- #define `SMB_STATUS_ALREADY_COMMITTED` 0xc0000021
- #define `SMB_STATUS_ACCESS_DENIED` 0xc0000022

- #define SMB_STATUS_BUFFER_TOO_SMALL 0xc0000023
- #define SMB_STATUS_OBJECT_TYPE_MISMATCH 0xc0000024
- #define SMB_STATUS_NONCONTINUABLE_EXCEPTION 0xc0000025
- #define SMB_STATUS_INVALID_DISPOSITION 0xc0000026
- #define SMB_STATUS_UNWIND 0xc0000027
- #define SMB_STATUS_BAD_STACK 0xc0000028
- #define SMB_STATUS_INVALID_UNWIND_TARGET 0xc0000029
- #define SMB_STATUS_NOT_LOCKED 0xc000002a
- #define SMB_STATUS_PARITY_ERROR 0xc000002b
- #define SMB_STATUS_UNABLE_TO_DECOMMIT_VM 0xc000002c
- #define SMB_STATUS_NOT_COMMITTED 0xc000002d
- #define SMB_STATUS_INVALID_PORT_ATTRIBUTES 0xc000002e
- #define SMB_STATUS_PORT_MESSAGE_TOO_LONG 0xc000002f
- #define SMB_STATUS_INVALID_PARAMETER_MIX 0xc0000030
- #define SMB_STATUS_INVALID_QUOTA_LOWER 0xc0000031
- #define SMB_STATUS_DISK_CORRUPT_ERROR 0xc0000032
- #define SMB_STATUS_OBJECT_NAME_INVALID 0xc0000033
- #define SMB_STATUS_OBJECT_NAME_NOT_FOUND 0xc0000034
- #define SMB_STATUS_OBJECT_NAME_COLLISION 0xc0000035
- #define SMB_STATUS_HANDLE_NOT_WAITABLE 0xc0000036
- #define SMB_STATUS_PORT_DISCONNECTED 0xc0000037
- #define SMB_STATUS_DEVICE_ALREADY_ATTACHED 0xc0000038
- #define SMB_STATUS_OBJECT_PATH_INVALID 0xc0000039
- #define SMB_STATUS_OBJECT_PATH_NOT_FOUND 0xc000003a
- #define SMB_STATUS_OBJECT_PATH_SYNTAX_BAD 0xc000003b
- #define SMB_STATUS_DATA_OVERRUN 0xc000003c
- #define SMB_STATUS_DATA_LATE_ERROR 0xc000003d
- #define SMB_STATUS_DATA_ERROR 0xc000003e
- #define SMB_STATUS_CRC_ERROR 0xc000003f
- #define SMB_STATUS_SECTION_TOO_BIG 0xc0000040
- #define SMB_STATUS_PORT_CONNECTION_REFUSED 0xc0000041
- #define SMB_STATUS_INVALID_PORT_HANDLE 0xc0000042
- #define SMB_STATUS_SHARING_VIOLATION 0xc0000043
- #define SMB_STATUS_QUOTA_EXCEEDED 0xc0000044
- #define SMB_STATUS_INVALID_PAGE_PROTECTION 0xc0000045
- #define SMB_STATUS_MUTANT_NOT_OWNED 0xc0000046
- #define SMB_STATUS_SEMAPHORE_LIMIT_EXCEEDED 0xc0000047
- #define SMB_STATUS_PORT_ALREADY_SET 0xc0000048
- #define SMB_STATUS_SECTION_NOT_IMAGE 0xc0000049
- #define SMB_STATUS_SUSPEND_COUNT_EXCEEDED 0xc000004a
- #define SMB_STATUS_THREAD_IS_TERMINATING 0xc000004b
- #define SMB_STATUS_BAD_WORKING_SET_LIMIT 0xc000004c
- #define SMB_STATUS_INCOMPATIBLE_FILE_MAP 0xc000004d
- #define SMB_STATUS_SECTION_PROTECTION 0xc000004e
- #define SMB_STATUS_EAS_NOT_SUPPORTED 0xc000004f
- #define SMB_STATUS_EA_TOO_LARGE 0xc0000050
- #define SMB_STATUS_NONEXISTENT_EA_ENTRY 0xc0000051
- #define SMB_STATUS_NO_EAS_ON_FILE 0xc0000052
- #define SMB_STATUS_EA_CORRUPT_ERROR 0xc0000053
- #define SMB_STATUS_FILE_LOCK_CONFLICT 0xc0000054
- #define SMB_STATUS_LOCK_NOT_GRANTED 0xc0000055
- #define SMB_STATUS_DELETE_PENDING 0xc0000056
- #define SMB_STATUS_CTL_FILE_NOT_SUPPORTED 0xc0000057
- #define SMB_STATUS_UNKNOWN_REVISION 0xc0000058
- #define SMB_STATUS_REVISION_MISMATCH 0xc0000059

- #define SMB_STATUS_INVALID_OWNER 0xc000005a
- #define SMB_STATUS_INVALID_PRIMARY_GROUP 0xc000005b
- #define SMB_STATUS_NO_IMPERSONATION_TOKEN 0xc000005c
- #define SMB_STATUS_CANT_DISABLE_MANDATORY 0xc000005d
- #define SMB_STATUS_NO_LOGON_SERVERS 0xc000005e
- #define SMB_STATUS_NO_SUCH_LOGON_SESSION 0xc000005f
- #define SMB_STATUS_NO_SUCH_PRIVILEGE 0xc0000060
- #define SMB_STATUS_PRIVILEGE_NOT_HELD 0xc0000061
- #define SMB_STATUS_INVALID_ACCOUNT_NAME 0xc0000062
- #define SMB_STATUS_USER_EXISTS 0xc0000063
- #define SMB_STATUS_NO_SUCH_USER 0xc0000064
- #define SMB_STATUS_GROUP_EXISTS 0xc0000065
- #define SMB_STATUS_NO_SUCH_GROUP 0xc0000066
- #define SMB_STATUS_MEMBER_IN_GROUP 0xc0000067
- #define SMB_STATUS_MEMBER_NOT_IN_GROUP 0xc0000068
- #define SMB_STATUS_LAST_ADMIN 0xc0000069
- #define SMB_STATUS_WRONG_PASSWORD 0xc000006a
- #define SMB_STATUS_ILL_FORMED_PASSWORD 0xc000006b
- #define SMB_STATUS_PASSWORD_RESTRICTION 0xc000006c
- #define SMB_STATUS_LOGON_FAILURE 0xc000006d
- #define SMB_STATUS_ACCOUNT_RESTRICTION 0xc000006e
- #define SMB_STATUS_INVALID_LOGON_HOURS 0xc000006f
- #define SMB_STATUS_INVALID_WORKSTATION 0xc0000070
- #define SMB_STATUS_PASSWORD_EXPIRED 0xc0000071
- #define SMB_STATUS_ACCOUNT_DISABLED 0xc0000072
- #define SMB_STATUS_NONE_MAPPED 0xc0000073
- #define SMB_STATUS_TOO_MANY_LUIDS_REQUESTED 0xc0000074
- #define SMB_STATUS_LUIDS_EXHAUSTED 0xc0000075
- #define SMB_STATUS_INVALID_SUB_AUTHORITY 0xc0000076
- #define SMB_STATUS_INVALID_ACL 0xc0000077
- #define SMB_STATUS_INVALID_SID 0xc0000078
- #define SMB_STATUS_INVALID_SECURITY_DESCR 0xc0000079
- #define SMB_STATUS_PROCEDURE_NOT_FOUND 0xc000007a
- #define SMB_STATUS_INVALID_IMAGE_FORMAT 0xc000007b
- #define SMB_STATUS_NO_TOKEN 0xc000007c
- #define SMB_STATUS_BAD_INHERITANCE_ACL 0xc000007d
- #define SMB_STATUS_RANGE_NOT_LOCKED 0xc000007e
- #define SMB_STATUS_DISK_FULL 0xc000007f
- #define SMB_STATUS_SERVER_DISABLED 0xc0000080
- #define SMB_STATUS_SERVER_NOT_DISABLED 0xc0000081
- #define SMB_STATUS_TOO_MANY_GUIDS_REQUESTED 0xc0000082
- #define SMB_STATUS_GUIDS_EXHAUSTED 0xc0000083
- #define SMB_STATUS_INVALID_ID_AUTHORITY 0xc0000084
- #define SMB_STATUS_AGENTS_EXHAUSTED 0xc0000085
- #define SMB_STATUS_INVALID_VOLUME_LABEL 0xc0000086
- #define SMB_STATUS_SECTION_NOT_EXTENDED 0xc0000087
- #define SMB_STATUS_NOT_MAPPED_DATA 0xc0000088
- #define SMB_STATUS_RESOURCE_DATA_NOT_FOUND 0xc0000089
- #define SMB_STATUS_RESOURCE_TYPE_NOT_FOUND 0xc000008a
- #define SMB_STATUS_RESOURCE_NAME_NOT_FOUND 0xc000008b
- #define SMB_STATUS_ARRAY_BOUNDS_EXCEEDED 0xc000008c
- #define SMB_STATUS_FLOAT_DENORMAL_OPERAND 0xc000008d
- #define SMB_STATUS_FLOAT_DIVIDE_BY_ZERO 0xc000008e
- #define SMB_STATUS_FLOAT_INEXACT_RESULT 0xc000008f
- #define SMB_STATUS_FLOAT_INVALID_OPERATION 0xc0000090

- #define SMB_STATUS_FLOAT_OVERFLOW 0xc0000091
- #define SMB_STATUS_FLOAT_STACK_CHECK 0xc0000092
- #define SMB_STATUS_FLOAT_UNDERFLOW 0xc0000093
- #define SMB_STATUS_INTEGER_DIVIDE_BY_ZERO 0xc0000094
- #define SMB_STATUS_INTEGER_OVERFLOW 0xc0000095
- #define SMB_STATUS_PRIVILEGED_INSTRUCTION 0xc0000096
- #define SMB_STATUS_TOO_MANY_PAGING_FILES 0xc0000097
- #define SMB_STATUS_FILE_INVALID 0xc0000098
- #define SMB_STATUS_ALLOTTED_SPACE_EXCEEDED 0xc0000099
- #define SMB_STATUS_INSUFFICIENT_RESOURCES 0xc000009a
- #define SMB_STATUS_DFS_EXIT_PATH_FOUND 0xc000009b
- #define SMB_STATUS_DEVICE_DATA_ERROR 0xc000009c
- #define SMB_STATUS_DEVICE_NOT_CONNECTED 0xc000009d
- #define SMB_STATUS_DEVICE_POWER_FAILURE 0xc000009e
- #define SMB_STATUS_FREE_VM_NOT_AT_BASE 0xc000009f
- #define SMB_STATUS_MEMORY_NOT_ALLOCATED 0xc00000a0
- #define SMB_STATUS_WORKING_SET_QUOTA 0xc00000a1
- #define SMB_STATUS_MEDIA_WRITE_PROTECTED 0xc00000a2
- #define SMB_STATUS_DEVICE_NOT_READY 0xc00000a3
- #define SMB_STATUS_INVALID_GROUP_ATTRIBUTES 0xc00000a4
- #define SMB_STATUS_BAD_IMPERSONATION_LEVEL 0xc00000a5
- #define SMB_STATUS_CANT_OPEN_ANONYMOUS 0xc00000a6
- #define SMB_STATUS_BAD_VALIDATION_CLASS 0xc00000a7
- #define SMB_STATUS_BAD_TOKEN_TYPE 0xc00000a8
- #define SMB_STATUS_BAD_MASTER_BOOT_RECORD 0xc00000a9
- #define SMB_STATUS_INSTRUCTION_MISALIGNMENT 0xc00000aa
- #define SMB_STATUS_INSTANCE_NOT_AVAILABLE 0xc00000ab
- #define SMB_STATUS_PIPE_NOT_AVAILABLE 0xc00000ac
- #define SMB_STATUS_INVALID_PIPE_STATE 0xc00000ad
- #define SMB_STATUS_PIPE_BUSY 0xc00000ae
- #define SMB_STATUS_ILLEGAL_FUNCTION 0xc00000af
- #define SMB_STATUS_PIPE_DISCONNECTED 0xc00000b0
- #define SMB_STATUS_PIPE_CLOSING 0xc00000b1
- #define SMB_STATUS_PIPE_CONNECTED 0xc00000b2
- #define SMB_STATUS_PIPE_LISTENING 0xc00000b3
- #define SMB_STATUS_INVALID_READ_MODE 0xc00000b4
- #define SMB_STATUS_IO_TIMEOUT 0xc00000b5
- #define SMB_STATUS_FILE_FORCED_CLOSED 0xc00000b6
- #define SMB_STATUS_PROFILING_NOT_STARTED 0xc00000b7
- #define SMB_STATUS_PROFILING_NOT_STOPPED 0xc00000b8
- #define SMB_STATUS_COULD_NOT_INTERPRET 0xc00000b9
- #define SMB_STATUS_FILE_IS_A_DIRECTORY 0xc00000ba
- #define SMB_STATUS_NOT_SUPPORTED 0xc00000bb
- #define SMB_STATUS_REMOTE_NOT_LISTENING 0xc00000bc
- #define SMB_STATUS_DUPLICATE_NAME 0xc00000bd
- #define SMB_STATUS_BAD_NETWORK_PATH 0xc00000be
- #define SMB_STATUS_NETWORK_BUSY 0xc00000bf
- #define SMB_STATUS_DEVICE_DOES_NOT_EXIST 0xc00000c0
- #define SMB_STATUS_TOO_MANY_COMMANDS 0xc00000c1
- #define SMB_STATUS_ADAPTER_HARDWARE_ERROR 0xc00000c2
- #define SMB_STATUS_INVALID_NETWORK_RESPONSE 0xc00000c3
- #define SMB_STATUS_UNEXPECTED_NETWORK_ERROR 0xc00000c4
- #define SMB_STATUS_BAD_REMOTE_ADAPTER 0xc00000c5
- #define SMB_STATUS_PRINT_QUEUE_FULL 0xc00000c6
- #define SMB_STATUS_NO_SPOOL_SPACE 0xc00000c7

- #define SMB_STATUS_PRINT_CANCELLED 0xc00000c8
- #define SMB_STATUS_NETWORK_NAME_DELETED 0xc00000c9
- #define SMB_STATUS_NETWORK_ACCESS_DENIED 0xc00000ca
- #define SMB_STATUS_BAD_DEVICE_TYPE 0xc00000cb
- #define SMB_STATUS_BAD_NETWORK_NAME 0xc00000cc
- #define SMB_STATUS_TOO_MANY_NAMES 0xc00000cd
- #define SMB_STATUS_TOO_MANY_SESSIONS 0xc00000ce
- #define SMB_STATUS_SHARING_PAUSED 0xc00000cf
- #define SMB_STATUS_REQUEST_NOT_ACCEPTED 0xc00000d0
- #define SMB_STATUS_REDIRECTOR_PAUSED 0xc00000d1
- #define SMB_STATUS_NET_WRITE_FAULT 0xc00000d2
- #define SMB_STATUS_PROFILING_AT_LIMIT 0xc00000d3
- #define SMB_STATUS_NOT_SAME_DEVICE 0xc00000d4
- #define SMB_STATUS_FILE_RENAMED 0xc00000d5
- #define SMB_STATUS_VIRTUAL_CIRCUIT_CLOSED 0xc00000d6
- #define SMB_STATUS_NO_SECURITY_ON_OBJECT 0xc00000d7
- #define SMB_STATUS_CANT_WAIT 0xc00000d8
- #define SMB_STATUS_PIPE_EMPTY 0xc00000d9
- #define SMB_STATUS_CANT_ACCESS_DOMAIN_INFO 0xc00000da
- #define SMB_STATUS_CANT_TERMINATE_SELF 0xc00000db
- #define SMB_STATUS_INVALID_SERVER_STATE 0xc00000dc
- #define SMB_STATUS_INVALID_DOMAIN_STATE 0xc00000dd
- #define SMB_STATUS_INVALID_DOMAIN_ROLE 0xc00000de
- #define SMB_STATUS_NO_SUCH_DOMAIN 0xc00000df
- #define SMB_STATUS_DOMAIN_EXISTS 0xc00000e0
- #define SMB_STATUS_DOMAIN_LIMIT_EXCEEDED 0xc00000e1
- #define SMB_STATUS_OPLOCK_NOT_GRANTED 0xc00000e2
- #define SMB_STATUS_INVALID_OPLOCK_PROTOCOL 0xc00000e3
- #define SMB_STATUS_INTERNAL_DB_CORRUPTION 0xc00000e4
- #define SMB_STATUS_INTERNAL_ERROR 0xc00000e5
- #define SMB_STATUS_GENERIC_NOT_MAPPED 0xc00000e6
- #define SMB_STATUS_BAD_DESCRIPTOR_FORMAT 0xc00000e7
- #define SMB_STATUS_INVALID_USER_BUFFER 0xc00000e8
- #define SMB_STATUS_UNEXPECTED_IO_ERROR 0xc00000e9
- #define SMB_STATUS_UNEXPECTED_MM_CREATE_ERR 0xc00000ea
- #define SMB_STATUS_UNEXPECTED_MM_MAP_ERROR 0xc00000eb
- #define SMB_STATUS_UNEXPECTED_MM_EXTEND_ERR 0xc00000ec
- #define SMB_STATUS_NOT_LOGON_PROCESS 0xc00000ed
- #define SMB_STATUS_LOGON_SESSION_EXISTS 0xc00000ee
- #define SMB_STATUS_INVALID_PARAMETER_1 0xc00000ef
- #define SMB_STATUS_INVALID_PARAMETER_2 0xc00000f0
- #define SMB_STATUS_INVALID_PARAMETER_3 0xc00000f1
- #define SMB_STATUS_INVALID_PARAMETER_4 0xc00000f2
- #define SMB_STATUS_INVALID_PARAMETER_5 0xc00000f3
- #define SMB_STATUS_INVALID_PARAMETER_6 0xc00000f4
- #define SMB_STATUS_INVALID_PARAMETER_7 0xc00000f5
- #define SMB_STATUS_INVALID_PARAMETER_8 0xc00000f6
- #define SMB_STATUS_INVALID_PARAMETER_9 0xc00000f7
- #define SMB_STATUS_INVALID_PARAMETER_10 0xc00000f8
- #define SMB_STATUS_INVALID_PARAMETER_11 0xc00000f9
- #define SMB_STATUS_INVALID_PARAMETER_12 0xc00000fa
- #define SMB_STATUS_REDIRECTOR_NOT_STARTED 0xc00000fb
- #define SMB_STATUS_REDIRECTOR_STARTED 0xc00000fc
- #define SMB_STATUS_STACK_OVERFLOW 0xc00000fd
- #define SMB_STATUS_NO_SUCH_PACKAGE 0xc00000fe

- #define SMB_STATUS_BAD_FUNCTION_TABLE 0xc00000ff
- #define SMB_STATUS_DIRECTORY_NOT_EMPTY 0xc0000101
- #define SMB_STATUS_FILE_CORRUPT_ERROR 0xc0000102
- #define SMB_STATUS_NOT_A_DIRECTORY 0xc0000103
- #define SMB_STATUS_BAD_LOGON_SESSION_STATE 0xc0000104
- #define SMB_STATUS_LOGON_SESSION_COLLISION 0xc0000105
- #define SMB_STATUS_NAME_TOO_LONG 0xc0000106
- #define SMB_STATUS_FILES_OPEN 0xc0000107
- #define SMB_STATUS_CONNECTION_IN_USE 0xc0000108
- #define SMB_STATUS_MESSAGE_NOT_FOUND 0xc0000109
- #define SMB_STATUS_PROCESS_IS_TERMINATING 0xc000010a
- #define SMB_STATUS_INVALID_LOGON_TYPE 0xc000010b
- #define SMB_STATUS_NO_GUID_TRANSLATION 0xc000010c
- #define SMB_STATUS_CANNOT_IMPERSONATE 0xc000010d
- #define SMB_STATUS_IMAGE_ALREADY_LOADED 0xc000010e
- #define SMB_STATUS_ABIOS_NOT_PRESENT 0xc000010f
- #define SMB_STATUS_ABIOS_LID_NOT_EXIST 0xc0000110
- #define SMB_STATUS_ABIOS_LID_ALREADY_OWNED 0xc0000111
- #define SMB_STATUS_ABIOS_NOT_LID_OWNER 0xc0000112
- #define SMB_STATUS_ABIOS_INVALID_COMMAND 0xc0000113
- #define SMB_STATUS_ABIOS_INVALID_LID 0xc0000114
- #define SMB_STATUS_ABIOS_SELECTOR_NOT_AVAILABLE 0xc0000115
- #define SMB_STATUS_ABIOS_INVALID_SELECTOR 0xc0000116
- #define SMB_STATUS_NO_LDT 0xc0000117
- #define SMB_STATUS_INVALID_LDT_SIZE 0xc0000118
- #define SMB_STATUS_INVALID_LDT_OFFSET 0xc0000119
- #define SMB_STATUS_INVALID_LDT_DESCRIPTOR 0xc000011a
- #define SMB_STATUS_INVALID_IMAGE_NE_FORMAT 0xc000011b
- #define SMB_STATUS_RXACT_INVALID_STATE 0xc000011c
- #define SMB_STATUS_RXACT_COMMIT_FAILURE 0xc000011d
- #define SMB_STATUS_MAPPED_FILE_SIZE_ZERO 0xc000011e
- #define SMB_STATUS_TOO_MANY_OPENED_FILES 0xc000011f
- #define SMB_STATUS_CANCELLED 0xc0000120
- #define SMB_STATUS_CANNOT_DELETE 0xc0000121
- #define SMB_STATUS_INVALID_COMPUTER_NAME 0xc0000122
- #define SMB_STATUS_FILE_DELETED 0xc0000123
- #define SMB_STATUS_SPECIAL_ACCOUNT 0xc0000124
- #define SMB_STATUS_SPECIAL_GROUP 0xc0000125
- #define SMB_STATUS_SPECIAL_USER 0xc0000126
- #define SMB_STATUS_MEMBERS_PRIMARY_GROUP 0xc0000127
- #define SMB_STATUS_FILE_CLOSED 0xc0000128
- #define SMB_STATUS_TOO_MANY_THREADS 0xc0000129
- #define SMB_STATUS_THREAD_NOT_IN_PROCESS 0xc000012a
- #define SMB_STATUS_TOKEN_ALREADY_IN_USE 0xc000012b
- #define SMB_STATUS_PAGEFILE_QUOTA_EXCEEDED 0xc000012c
- #define SMB_STATUS_COMMITMENT_LIMIT 0xc000012d
- #define SMB_STATUS_INVALID_IMAGE_LE_FORMAT 0xc000012e
- #define SMB_STATUS_INVALID_IMAGE_NOT_MZ 0xc000012f
- #define SMB_STATUS_INVALID_IMAGE_PROTECT 0xc0000130
- #define SMB_STATUS_INVALID_IMAGE_WIN_16 0xc0000131
- #define SMB_STATUS_LOGON_SERVER_CONFLICT 0xc0000132
- #define SMB_STATUS_TIME_DIFFERENCE_AT_DC 0xc0000133
- #define SMB_STATUS_SYNCHRONIZATION_REQUIRED 0xc0000134
- #define SMB_STATUS_DLL_NOT_FOUND 0xc0000135
- #define SMB_STATUS_OPEN_FAILED 0xc0000136

- #define SMB_STATUS_IO_PRIVILEGE_FAILED 0xc0000137
- #define SMB_STATUS_ORDINAL_NOT_FOUND 0xc0000138
- #define SMB_STATUS_ENTRYPOINT_NOT_FOUND 0xc0000139
- #define SMB_STATUS_CONTROL_C_EXIT 0xc000013a
- #define SMB_STATUS_LOCAL_DISCONNECT 0xc000013b
- #define SMB_STATUS_REMOTE_DISCONNECT 0xc000013c
- #define SMB_STATUS_REMOTE_RESOURCES 0xc000013d
- #define SMB_STATUS_LINK_FAILED 0xc000013e
- #define SMB_STATUS_LINK_TIMEOUT 0xc000013f
- #define SMB_STATUS_INVALID_CONNECTION 0xc0000140
- #define SMB_STATUS_INVALID_ADDRESS 0xc0000141
- #define SMB_STATUS_DLL_INIT_FAILED 0xc0000142
- #define SMB_STATUS_MISSING_SYSTEMFILE 0xc0000143
- #define SMB_STATUS_UNHANDLED_EXCEPTION 0xc0000144
- #define SMB_STATUS_APP_INIT_FAILURE 0xc0000145
- #define SMB_STATUS_PAGEFILE_CREATE_FAILED 0xc0000146
- #define SMB_STATUS_NO_PAGEFILE 0xc0000147
- #define SMB_STATUS_INVALID_LEVEL 0xc0000148
- #define SMB_STATUS_WRONG_PASSWORD_CORE 0xc0000149
- #define SMB_STATUS_ILLEGAL_FLOAT_CONTEXT 0xc000014a
- #define SMB_STATUS_PIPE_BROKEN 0xc000014b
- #define SMB_STATUS_REGISTRY_CORRUPT 0xc000014c
- #define SMB_STATUS_REGISTRY_IO_FAILED 0xc000014d
- #define SMB_STATUS_NO_EVENT_PAIR 0xc000014e
- #define SMB_STATUS_UNRECOGNIZED_VOLUME 0xc000014f
- #define SMB_STATUS_SERIAL_NO_DEVICE_INITED 0xc0000150
- #define SMB_STATUS_NO_SUCH_ALIAS 0xc0000151
- #define SMB_STATUS_MEMBER_NOT_IN_ALIAS 0xc0000152
- #define SMB_STATUS_MEMBER_IN_ALIAS 0xc0000153
- #define SMB_STATUS_ALIAS_EXISTS 0xc0000154
- #define SMB_STATUS_LOGON_NOT_GRANTED 0xc0000155
- #define SMB_STATUS_TOO_MANY_SECRETS 0xc0000156
- #define SMB_STATUS_SECRET_TOO_LONG 0xc0000157
- #define SMB_STATUS_INTERNAL_DB_ERROR 0xc0000158
- #define SMB_STATUS_FULLSCREEN_MODE 0xc0000159
- #define SMB_STATUS_TOO_MANY_CONTEXT_IDS 0xc000015a
- #define SMB_STATUS_LOGON_TYPE_NOT_GRANTED 0xc000015b
- #define SMB_STATUS_NOT_REGISTRY_FILE 0xc000015c
- #define SMB_STATUS_NT_CROSS_ENCRYPTION_REQUIRED 0xc000015d
- #define SMB_STATUS_DOMAIN_CTRLR_CONFIG_ERROR 0xc000015e
- #define SMB_STATUS_FT_MISSING_MEMBER 0xc000015f
- #define SMB_STATUS_ILL_FORMED_SERVICE_ENTRY 0xc0000160
- #define SMB_STATUS_ILLEGAL_CHARACTER 0xc0000161
- #define SMB_STATUS_UNMAPPABLE_CHARACTER 0xc0000162
- #define SMB_STATUS_UNDEFINED_CHARACTER 0xc0000163
- #define SMB_STATUS_FLOPPY_VOLUME 0xc0000164
- #define SMB_STATUS_FLOPPY_ID_MARK_NOT_FOUND 0xc0000165
- #define SMB_STATUS_FLOPPY_WRONG_CYLINDER 0xc0000166
- #define SMB_STATUS_FLOPPY_UNKNOWN_ERROR 0xc0000167
- #define SMB_STATUS_FLOPPY_BAD_REGISTERS 0xc0000168
- #define SMB_STATUS_DISK_RECALIBRATE_FAILED 0xc0000169
- #define SMB_STATUS_DISK_OPERATION_FAILED 0xc000016a
- #define SMB_STATUS_DISK_RESET_FAILED 0xc000016b
- #define SMB_STATUS_SHARED_IRQ_BUSY 0xc000016c
- #define SMB_STATUS_FT_ORPHANING 0xc000016d

- #define SMB_STATUS_PARTITION_FAILURE 0xc0000172
- #define SMB_STATUS_INVALID_BLOCK_LENGTH 0xc0000173
- #define SMB_STATUS_DEVICE_NOT_PARTITIONED 0xc0000174
- #define SMB_STATUS_UNABLE_TO_LOCK_MEDIA 0xc0000175
- #define SMB_STATUS_UNABLE_TO_UNLOAD_MEDIA 0xc0000176
- #define SMB_STATUS_EOM_OVERFLOW 0xc0000177
- #define SMB_STATUS_NO_MEDIA 0xc0000178
- #define SMB_STATUS_NO_SUCH_MEMBER 0xc000017a
- #define SMB_STATUS_INVALID_MEMBER 0xc000017b
- #define SMB_STATUS_KEY_DELETED 0xc000017c
- #define SMB_STATUS_NO_LOG_SPACE 0xc000017d
- #define SMB_STATUS_TOO_MANY_SIDS 0xc000017e
- #define SMB_STATUS_LM_CROSS_ENCRYPTION_REQUIRED 0xc000017f
- #define SMB_STATUS_KEY_HAS_CHILDREN 0xc0000180
- #define SMB_STATUS_CHILD_MUST_BE_VOLATILE 0xc0000181
- #define SMB_STATUS_DEVICE_CONFIGURATION_ERROR 0xc0000182
- #define SMB_STATUS_DRIVER_INTERNAL_ERROR 0xc0000183
- #define SMB_STATUS_INVALID_DEVICE_STATE 0xc0000184
- #define SMB_STATUS_IO_DEVICE_ERROR 0xc0000185
- #define SMB_STATUS_DEVICE_PROTOCOL_ERROR 0xc0000186
- #define SMB_STATUS_BACKUP_CONTROLLER 0xc0000187
- #define SMB_STATUS_LOG_FILE_FULL 0xc0000188
- #define SMB_STATUS_TOO_LATE 0xc0000189
- #define SMB_STATUS_NO_TRUST_LSA_SECRET 0xc000018a
- #define SMB_STATUS_NO_TRUST_SAM_ACCOUNT 0xc000018b
- #define SMB_STATUS_TRUSTED_DOMAIN_FAILURE 0xc000018c
- #define SMB_STATUS_TRUSTED_RELATIONSHIP_FAILURE 0xc000018d
- #define SMB_STATUS_EVENTLOG_FILE_CORRUPT 0xc000018e
- #define SMB_STATUS_EVENTLOG_CANT_START 0xc000018f
- #define SMB_STATUS_TRUST_FAILURE 0xc0000190
- #define SMB_STATUS_MUTANT_LIMIT_EXCEEDED 0xc0000191
- #define SMB_STATUS_NETLOGON_NOT_STARTED 0xc0000192
- #define SMB_STATUS_ACCOUNT_EXPIRED 0xc0000193
- #define SMB_STATUS_POSSIBLE_DEADLOCK 0xc0000194
- #define SMB_STATUS_NETWORK_CREDENTIAL_CONFLICT 0xc0000195
- #define SMB_STATUS_REMOTE_SESSION_LIMIT 0xc0000196
- #define SMB_STATUS_EVENTLOG_FILE_CHANGED 0xc0000197
- #define SMB_STATUS_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT 0xc0000198
- #define SMB_STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT 0xc0000199
- #define SMB_STATUS_NOLOGON_SERVER_TRUST_ACCOUNT 0xc000019a
- #define SMB_STATUS_DOMAIN_TRUST_INCONSISTENT 0xc000019b
- #define SMB_STATUS_FS_DRIVER_REQUIRED 0xc000019c
- #define SMB_STATUS_INVALID_LOCK_RANGE 0xc00001a1
- #define SMB_STATUS_NO_USER_SESSION_KEY 0xc0000202
- #define SMB_STATUS_USER_SESSION_DELETED 0xc0000203
- #define SMB_STATUS_RESOURCE_LANG_NOT_FOUND 0xc0000204
- #define SMB_STATUS_INSUFF_SERVER_RESOURCES 0xc0000205
- #define SMB_STATUS_INVALID_BUFFER_SIZE 0xc0000206
- #define SMB_STATUS_INVALID_ADDRESS_COMPONENT 0xc0000207
- #define SMB_STATUS_INVALID_ADDRESS_WILDCARD 0xc0000208
- #define SMB_STATUS_TOO_MANY_ADDRESSES 0xc0000209
- #define SMB_STATUS_ADDRESS_ALREADY_EXISTS 0xc000020a
- #define SMB_STATUS_ADDRESS_CLOSED 0xc000020b
- #define SMB_STATUS_CONNECTION_DISCONNECTED 0xc000020c
- #define SMB_STATUS_CONNECTION_RESET 0xc000020d

- #define SMB_STATUS_TOO_MANY_NODES 0xc000020e
- #define SMB_STATUS_TRANSACTION_ABORTED 0xc000020f
- #define SMB_STATUS_TRANSACTION_TIMED_OUT 0xc0000210
- #define SMB_STATUS_TRANSACTION_NO_RELEASE 0xc0000211
- #define SMB_STATUS_TRANSACTION_NO_MATCH 0xc0000212
- #define SMB_STATUS_TRANSACTION_RESPONDED 0xc0000213
- #define SMB_STATUS_TRANSACTION_INVALID_ID 0xc0000214
- #define SMB_STATUS_TRANSACTION_INVALID_TYPE 0xc0000215
- #define SMB_STATUS_NOT_SERVER_SESSION 0xc0000216
- #define SMB_STATUS_NOT_CLIENT_SESSION 0xc0000217
- #define SMB_STATUS_CANNOT_LOAD_REGISTRY_FILE 0xc0000218
- #define SMB_STATUS_DEBUG_ATTACH_FAILED 0xc0000219
- #define SMB_STATUS_SYSTEM_PROCESS_TERMINATED 0xc000021a
- #define SMB_STATUS_DATA_NOT_ACCEPTED 0xc000021b
- #define SMB_STATUS_NO_BROWSER_SERVERS_FOUND 0xc000021c
- #define SMB_STATUS_VDM_HARD_ERROR 0xc000021d
- #define SMB_STATUS_DRIVER_CANCEL_TIMEOUT 0xc000021e
- #define SMB_STATUS_REPLY_MESSAGE_MISMATCH 0xc000021f
- #define SMB_STATUS_MAPPED_ALIGNMENT 0xc0000220
- #define SMB_STATUS_IMAGE_CHECKSUM_MISMATCH 0xc0000221
- #define SMB_STATUS_LOST_WRITEBEHIND_DATA 0xc0000222
- #define SMB_STATUS_CLIENT_SERVER_PARAMETERS_INVALID 0xc0000223
- #define SMB_STATUS_PASSWORD_MUST_CHANGE 0xc0000224
- #define SMB_STATUS_NOT_FOUND 0xc0000225
- #define SMB_STATUS_NOT_TINY_STREAM 0xc0000226
- #define SMB_STATUS_RECOVERY_FAILURE 0xc0000227
- #define SMB_STATUS_STACK_OVERFLOW_READ 0xc0000228
- #define SMB_STATUS_FAIL_CHECK 0xc0000229
- #define SMB_STATUS_DUPLICATE_OBJECTID 0xc000022a
- #define SMB_STATUS_OBJECTID_EXISTS 0xc000022b
- #define SMB_STATUS_CONVERT_TO_LARGE 0xc000022c
- #define SMB_STATUS_RETRY 0xc000022d
- #define SMB_STATUS_FOUND_OUT_OF_SCOPE 0xc000022e
- #define SMB_STATUS_ALLOCATE_BUCKET 0xc000022f
- #define SMB_STATUS_PROPSET_NOT_FOUND 0xc0000230
- #define SMB_STATUS_MARSHALL_OVERFLOW 0xc0000231
- #define SMB_STATUS_INVALID_VARIANT 0xc0000232
- #define SMB_STATUS_DOMAIN_CONTROLLER_NOT_FOUND 0xc0000233
- #define SMB_STATUS_ACCOUNT_LOCKED_OUT 0xc0000234
- #define SMB_STATUS_HANDLE_NOT_CLOSABLE 0xc0000235
- #define SMB_STATUS_CONNECTION_REFUSED 0xc0000236
- #define SMB_STATUS_GRACEFUL_DISCONNECT 0xc0000237
- #define SMB_STATUS_ADDRESS_ALREADY_ASSOCIATED 0xc0000238
- #define SMB_STATUS_ADDRESS_NOT_ASSOCIATED 0xc0000239
- #define SMB_STATUS_CONNECTION_INVALID 0xc000023a
- #define SMB_STATUS_CONNECTION_ACTIVE 0xc000023b
- #define SMB_STATUS_NETWORK_UNREACHABLE 0xc000023c
- #define SMB_STATUS_HOST_UNREACHABLE 0xc000023d
- #define SMB_STATUS_PROTOCOL_UNREACHABLE 0xc000023e
- #define SMB_STATUS_PORT_UNREACHABLE 0xc000023f
- #define SMB_STATUS_REQUEST_ABORTED 0xc0000240
- #define SMB_STATUS_CONNECTION_ABORTED 0xc0000241
- #define SMB_STATUS_BAD_COMPRESSION_BUFFER 0xc0000242
- #define SMB_STATUS_USER_MAPPED_FILE 0xc0000243
- #define SMB_STATUS_AUDIT_FAILED 0xc0000244

- #define SMB_STATUS_TIMER_RESOLUTION_NOT_SET 0xc0000245
- #define SMB_STATUS_CONNECTION_COUNT_LIMIT 0xc0000246
- #define SMB_STATUS_LOGIN_TIME_RESTRICTION 0xc0000247
- #define SMB_STATUS_LOGIN_WKSTA_RESTRICTION 0xc0000248
- #define SMB_STATUS_IMAGE_MP_UP_MISMATCH 0xc0000249
- #define SMB_STATUS_INSUFFICIENT_LOGON_INFO 0xc0000250
- #define SMB_STATUS_BAD_DLL_ENTRYPOINT 0xc0000251
- #define SMB_STATUS_BAD_SERVICE_ENTRYPOINT 0xc0000252
- #define SMB_STATUS_LPC_REPLY_LOST 0xc0000253
- #define SMB_STATUS_IP_ADDRESS_CONFLICT1 0xc0000254
- #define SMB_STATUS_IP_ADDRESS_CONFLICT2 0xc0000255
- #define SMB_STATUS_REGISTRY_QUOTA_LIMIT 0xc0000256
- #define SMB_STATUS_PATH_NOT_COVERED 0xc0000257
- #define SMB_STATUS_NO_CALLBACK_ACTIVE 0xc0000258
- #define SMB_STATUS_LICENSE_QUOTA_EXCEEDED 0xc0000259
- #define SMB_STATUS_PWD_TOO_SHORT 0xc000025a
- #define SMB_STATUS_PWD_TOO_RECENT 0xc000025b
- #define SMB_STATUS_PWD_HISTORY_CONFLICT 0xc000025c
- #define SMB_STATUS_PLUGPLAY_NO_DEVICE 0xc000025e
- #define SMB_STATUS_UNSUPPORTED_COMPRESSION 0xc000025f
- #define SMB_STATUS_INVALID_HW_PROFILE 0xc0000260
- #define SMB_STATUS_INVALID_PLUGPLAY_DEVICE_PATH 0xc0000261
- #define SMB_STATUS_DRIVER_ORDINAL_NOT_FOUND 0xc0000262
- #define SMB_STATUS_DRIVER_ENTRYPOINT_NOT_FOUND 0xc0000263
- #define SMB_STATUS_RESOURCE_NOT_OWNED 0xc0000264
- #define SMB_STATUS_TOO_MANY_LINKS 0xc0000265
- #define SMB_STATUS_QUOTA_LIST_INCONSISTENT 0xc0000266
- #define SMB_STATUS_FILE_IS_OFFLINE 0xc0000267
- #define SMB_STATUS_DFS_UNAVAILABLE 0xc000026d
- #define SMB_STATUS_VOLUME_DISMOUNTED 0xc000026e
- #define SMB_STATUS_NOT_A_REPARSE_POINT 0xc0000275
- #define SMB_STATUS_IO_REPARSE_TAG_NOT_HANDLED 0xc0000279
- #define SMB_STATUS_NETWORK_SESSION_EXPIRED 0xc000035C
- #define SMB_STATUS_INVALID_TOKEN 0xc0000465
- #define SMB_STATUS_SERVER_UNAVAILABLE 0xc0000466
- #define SMB_STATUS_FILE_NOT_AVAILABLE 0xc0000467
- #define SMB_STATUS_SHARE_UNAVAILABLE 0xc0000480
- #define SMB_STATUS_NO_SUCH_JOB 0xc0000ede
- #define SMB_STATUS_NO_PREAUTH_INTEGRITY_HASH_OVERLAP 0xC05D0000

4.69.1 Detailed Description

4.69.2 Macro Definition Documentation

4.69.2.1 #define SMB_STATUS_OK 0xc0000000

4.69.2.2 #define SMB_STATUS_INVALID 0xffffffff

4.69.2.3 #define SMB_STATUS_SUCCESS 0x00000000

4.69.2.4 #define SMB_STATUS_ENUMDIR 0x0000010c

4.69.2.5 #define SMB_STATUS_PENDING 0x00000103

4.69.2.6 #define SMB_STATUS_MORE_ENTRIES 0x00000105

4.69.2.7 #define SMB_STATUS_SOME_UNMAPPED 0x00000107

4.69.2.8 #define SMB_STATUS_NOTIFY_CLEANUP 0x0000010b

4.69.2.9 #define SMB_STATUS_INSUFFICIENT_BUFFER 0x0000007a

4.69.2.10 #define SMB_STATUS_UNKNOWN_PRINTER_DRIVER 0x00000705

4.69.2.11 #define SMB_STATUS_INVALID_DATATYPE 0x0000070c

4.69.2.12 #define SMB_STATUS_BUFFER_OVERFLOW 0x80000005

4.69.2.13 #define SMB_STATUS_NO_MORE_FILES 0x80000006

4.69.2.14 #define SMB_STATUS_DEVICE_OFF_LINE 0x80000010

4.69.2.15 #define SMB_STATUS_NO_MORE_ENTRIES 0x8000001a

4.69.2.16 #define SMB_STATUS_STOPPED_ON_SYMLINK 0x8000002D

4.69.2.17 #define SMB_STATUS_UNSUCCESSFUL 0xc0000001

4.69.2.18 #define SMB_STATUS_NOT_IMPLEMENTED 0xc0000002

4.69.2.19 #define SMB_STATUS_INVALID_INFO_CLASS 0xc0000003

4.69.2.20 #define SMB_STATUS_INFO_LENGTH_MISMATCH 0xc0000004

4.69.2.21 #define SMB_STATUS_ACCESS_VIOLATION 0xc0000005

4.69.2.22 #define SMB_STATUS_IN_PAGE_ERROR 0xc0000006

4.69.2.23 #define SMB_STATUS_PAGEFILE_QUOTA 0xc0000007

4.69.2.24 #define SMB_STATUS_INVALID_HANDLE 0xc0000008

4.69.2.25 #define SMB_STATUS_BAD_INITIAL_STACK 0xc0000009

4.69.2.26 #define SMB_STATUS_BAD_INITIAL_PC 0xc000000a

4.69.2.27 #define SMB_STATUS_INVALID_CID 0xc000000b

4.69.2.28 #define SMB_STATUS_TIMER_NOT_CANCELED 0xc000000c

4.69.2.29 #define SMB_STATUS_INVALID_PARAMETER 0xc000000d

4.69.2.30 #define SMB_STATUS_NO_SUCH_DEVICE 0xc000000e

4.69.2.31 #define SMB_STATUS_NO_SUCH_FILE 0xc000000f

4.69.2.32 #define SMB_STATUS_INVALID_DEVICE_REQUEST 0xc0000010

4.69.2.33 #define SMB_STATUS_END_OF_FILE 0xc0000011

4.69.2.34 #define SMB_STATUS_WRONG_VOLUME 0xc0000012

4.69.2.35 #define SMB_STATUS_NO_MEDIA_IN_DEVICE 0xc0000013

4.69.2.36 #define SMB_STATUS_UNRECOGNIZED_MEDIA 0xc0000014

4.69.2.37 #define SMB_STATUS_NONEXISTENT_SECTOR 0xc0000015

4.69.2.38 #define SMB_STATUS_MORE_PROCESSING_REQUIRED 0xc0000016

4.69.2.39 #define SMB_STATUS_NO_MEMORY 0xc0000017

4.69.2.40 #define SMB_STATUS_CONFLICTING_ADDRESSES 0xc0000018

4.69.2.41 #define SMB_STATUS_NOT_MAPPED_VIEW 0xc0000019

4.69.2.42 #define SMB_STATUS_UNABLE_TO_FREE_VM 0xc000001a

4.69.2.43 #define SMB_STATUS_UNABLE_TO_DELETE_SECTION 0xc000001b

4.69.2.44 #define SMB_STATUS_INVALID_SYSTEM_SERVICE 0xc000001c

4.69.2.45 #define SMB_STATUS_ILLEGAL_INSTRUCTION 0xc000001d

4.69.2.46 #define SMB_STATUS_INVALID_LOCK_SEQUENCE 0xc000001e

4.69.2.47 #define SMB_STATUS_INVALID_VIEW_SIZE 0xc000001f

4.69.2.48 #define SMB_STATUS_INVALID_FILE_FOR_SECTION 0xc0000020

4.69.2.49 #define SMB_STATUS_ALREADY_COMMITTED 0xc0000021

4.69.2.50 #define SMB_STATUS_ACCESS_DENIED 0xc0000022

4.69.2.51 #define SMB_STATUS_BUFFER_TOO_SMALL 0xc0000023

4.69.2.52 #define SMB_STATUS_OBJECT_TYPE_MISMATCH 0xc0000024

4.69.2.53 #define SMB_STATUS_NONCONTINUABLE_EXCEPTION 0xc0000025

4.69.2.54 #define SMB_STATUS_INVALID_DISPOSITION 0xc0000026

4.69.2.55 #define SMB_STATUS_UNWIND 0xc0000027

4.69.2.56 #define SMB_STATUS_BAD_STACK 0xc0000028

4.69.2.57 #define SMB_STATUS_INVALID_UNWIND_TARGET 0xc0000029

4.69.2.58 #define SMB_STATUS_NOT_LOCKED 0xc000002a

4.69.2.59 #define SMB_STATUS_PARITY_ERROR 0xc000002b

4.69.2.60 #define SMB_STATUS_UNABLE_TO_DECOMMIT_VM 0xc000002c

4.69.2.61 #define SMB_STATUS_NOT_COMMITTED 0xc000002d

4.69.2.62 #define SMB_STATUS_INVALID_PORT_ATTRIBUTES 0xc000002e

4.69.2.63 #define SMB_STATUS_PORT_MESSAGE_TOO_LONG 0xc000002f

4.69.2.64 #define SMB_STATUS_INVALID_PARAMETER_MIX 0xc0000030

4.69.2.65 #define SMB_STATUS_INVALID_QUOTA_LOWER 0xc0000031

4.69.2.66 #define SMB_STATUS_DISK_CORRUPT_ERROR 0xc0000032

4.69.2.67 #define SMB_STATUS_OBJECT_NAME_INVALID 0xc0000033

4.69.2.68 #define SMB_STATUS_OBJECT_NAME_NOT_FOUND 0xc0000034

4.69.2.69 #define SMB_STATUS_OBJECT_NAME_COLLISION 0xc0000035

4.69.2.70 #define SMB_STATUS_HANDLE_NOT_WAITABLE 0xc0000036

4.69.2.71 #define SMB_STATUS_PORT_DISCONNECTED 0xc0000037

4.69.2.72 #define SMB_STATUS_DEVICE_ALREADY_ATTACHED 0xc0000038

4.69.2.73 #define SMB_STATUS_OBJECT_PATH_INVALID 0xc0000039

4.69.2.74 #define SMB_STATUS_OBJECT_PATH_NOT_FOUND 0xc000003a

4.69.2.75 #define SMB_STATUS_OBJECT_PATH_SYNTAX_BAD 0xc000003b

4.69.2.76 #define SMB_STATUS_DATA_OVERRUN 0xc000003c

4.69.2.77 #define SMB_STATUS_DATA_LATE_ERROR 0xc000003d

4.69.2.78 #define SMB_STATUS_DATA_ERROR 0xc000003e

4.69.2.79 #define SMB_STATUS_CRC_ERROR 0xc000003f

4.69.2.80 #define SMB_STATUS_SECTION_TOO_BIG 0xc0000040

4.69.2.81 #define SMB_STATUS_PORT_CONNECTION_REFUSED 0xc0000041

4.69.2.82 #define SMB_STATUS_INVALID_PORT_HANDLE 0xc0000042

4.69.2.83 #define SMB_STATUS_SHARING_VIOLATION 0xc0000043

4.69.2.84 #define SMB_STATUS_QUOTA_EXCEEDED 0xc0000044

4.69.2.85 #define SMB_STATUS_INVALID_PAGE_PROTECTION 0xc0000045

4.69.2.86 #define SMB_STATUS_MUTANT_NOT_OWNED 0xc0000046

4.69.2.87 #define SMB_STATUS_SEMAPHORE_LIMIT_EXCEEDED 0xc0000047

4.69.2.88 #define SMB_STATUS_PORT_ALREADY_SET 0xc0000048

4.69.2.89 #define SMB_STATUS_SECTION_NOT_IMAGE 0xc0000049

4.69.2.90 #define SMB_STATUS_SUSPEND_COUNT_EXCEEDED 0xc000004a

4.69.2.91 #define SMB_STATUS_THREAD_IS_TERMINATING 0xc000004b

4.69.2.92 #define SMB_STATUS_BAD_WORKING_SET_LIMIT 0xc000004c

4.69.2.93 #define SMB_STATUS_INCOMPATIBLE_FILE_MAP 0xc000004d

4.69.2.94 #define SMB_STATUS_SECTION_PROTECTION 0xc000004e

4.69.2.95 #define SMB_STATUS_EAS_NOT_SUPPORTED 0xc000004f

4.69.2.96 #define SMB_STATUS_EA_TOO_LARGE 0xc0000050

4.69.2.97 #define SMB_STATUS_NONEXISTENT_EA_ENTRY 0xc0000051

4.69.2.98 #define SMB_STATUS_NO_EAS_ON_FILE 0xc0000052

4.69.2.99 #define SMB_STATUS_EA_CORRUPT_ERROR 0xc0000053

4.69.2.100 #define SMB_STATUS_FILE_LOCK_CONFLICT 0xc0000054

4.69.2.101 #define SMB_STATUS_LOCK_NOT_GRANTED 0xc0000055

4.69.2.102 #define SMB_STATUS_DELETE_PENDING 0xc0000056

4.69.2.103 #define SMB_STATUS_CTL_FILE_NOT_SUPPORTED 0xc0000057

4.69.2.104 #define SMB_STATUS_UNKNOWN_REVISION 0xc0000058

4.69.2.105 #define SMB_STATUS_REVISION_MISMATCH 0xc0000059

4.69.2.106 #define SMB_STATUS_INVALID_OWNER 0xc000005a

4.69.2.107 #define SMB_STATUS_INVALID_PRIMARY_GROUP 0xc000005b

4.69.2.108 #define SMB_STATUS_NO_IMPERSONATION_TOKEN 0xc000005c

4.69.2.109 #define SMB_STATUS_CANT_DISABLE_MANDATORY 0xc000005d

4.69.2.110 #define SMB_STATUS_NO_LOGON_SERVERS 0xc000005e

4.69.2.111 #define SMB_STATUS_NO_SUCH_LOGON_SESSION 0xc000005f

4.69.2.112 #define SMB_STATUS_NO_SUCH_PRIVILEGE 0xc0000060

4.69.2.113 #define SMB_STATUS_PRIVILEGE_NOT_HELD 0xc0000061

4.69.2.114 #define SMB_STATUS_INVALID_ACCOUNT_NAME 0xc0000062

4.69.2.115 #define SMB_STATUS_USER_EXISTS 0xc0000063

4.69.2.116 #define SMB_STATUS_NO_SUCH_USER 0xc0000064

4.69.2.117 #define SMB_STATUS_GROUP_EXISTS 0xc0000065

4.69.2.118 #define SMB_STATUS_NO_SUCH_GROUP 0xc0000066

4.69.2.119 #define SMB_STATUS_MEMBER_IN_GROUP 0xc0000067

4.69.2.120 #define SMB_STATUS_MEMBER_NOT_IN_GROUP 0xc0000068

4.69.2.121 #define SMB_STATUS_LAST_ADMIN 0xc0000069

4.69.2.122 #define SMB_STATUS_WRONG_PASSWORD 0xc000006a

4.69.2.123 #define SMB_STATUS_ILL_FORMED_PASSWORD 0xc000006b

4.69.2.124 #define SMB_STATUS_PASSWORD_RESTRICTION 0xc000006c

4.69.2.125 #define SMB_STATUS_LOGON_FAILURE 0xc000006d

4.69.2.126 #define SMB_STATUS_ACCOUNT_RESTRICTION 0xc000006e

4.69.2.127 #define SMB_STATUS_INVALID_LOGON_HOURS 0xc000006f

4.69.2.128 #define SMB_STATUS_INVALID_WORKSTATION 0xc0000070

4.69.2.129 #define SMB_STATUS_PASSWORD_EXPIRED 0xc0000071

4.69.2.130 #define SMB_STATUS_ACCOUNT_DISABLED 0xc0000072

4.69.2.131 #define SMB_STATUS_NONE_MAPPED 0xc0000073

4.69.2.132 #define SMB_STATUS_TOO_MANY_LUIDS_REQUESTED 0xc0000074

4.69.2.133 #define SMB_STATUS_LUIDS_EXHAUSTED 0xc0000075

4.69.2.134 #define SMB_STATUS_INVALID_SUB_AUTHORITY 0xc0000076

4.69.2.135 #define SMB_STATUS_INVALID_ACL 0xc0000077

4.69.2.136 #define SMB_STATUS_INVALID_SID 0xc0000078

4.69.2.137 #define SMB_STATUS_INVALID_SECURITY_DESCR 0xc0000079

4.69.2.138 #define SMB_STATUS_PROCEDURE_NOT_FOUND 0xc000007a

4.69.2.139 #define SMB_STATUS_INVALID_IMAGE_FORMAT 0xc000007b

4.69.2.140 #define SMB_STATUS_NO_TOKEN 0xc000007c

4.69.2.141 #define SMB_STATUS_BAD_INHERITANCE_ACL 0xc000007d

4.69.2.142 #define SMB_STATUS_RANGE_NOT_LOCKED 0xc000007e

4.69.2.143 #define SMB_STATUS_DISK_FULL 0xc000007f

4.69.2.144 #define SMB_STATUS_SERVER_DISABLED 0xc0000080

4.69.2.145 #define SMB_STATUS_SERVER_NOT_DISABLED 0xc0000081

4.69.2.146 #define SMB_STATUS_TOO_MANY_GUIDS_REQUESTED 0xc0000082

4.69.2.147 #define SMB_STATUS_GUIDS_EXHAUSTED 0xc0000083

4.69.2.148 #define SMB_STATUS_INVALID_ID_AUTHORITY 0xc0000084

4.69.2.149 #define SMB_STATUS_AGENTS_EXHAUSTED 0xc0000085

4.69.2.150 #define SMB_STATUS_INVALID_VOLUME_LABEL 0xc0000086

4.69.2.151 #define SMB_STATUS_SECTION_NOT_EXTENDED 0xc0000087

4.69.2.152 #define SMB_STATUS_NOT_MAPPED_DATA 0xc0000088

4.69.2.153 #define SMB_STATUS_RESOURCE_DATA_NOT_FOUND 0xc0000089

4.69.2.154 #define SMB_STATUS_RESOURCE_TYPE_NOT_FOUND 0xc000008a

4.69.2.155 #define SMB_STATUS_RESOURCE_NAME_NOT_FOUND 0xc000008b

4.69.2.156 #define SMB_STATUS_ARRAY_BOUNDS_EXCEEDED 0xc000008c

4.69.2.157 #define SMB_STATUS_FLOAT_DENORMAL_OPERAND 0xc000008d

4.69.2.158 #define SMB_STATUS_FLOAT_DIVIDE_BY_ZERO 0xc000008e

4.69.2.159 #define SMB_STATUS_FLOAT_INEXACT_RESULT 0xc000008f

4.69.2.160 #define SMB_STATUS_FLOAT_INVALID_OPERATION 0xc0000090

4.69.2.161 #define SMB_STATUS_FLOAT_OVERFLOW 0xc0000091

4.69.2.162 #define SMB_STATUS_FLOAT_STACK_CHECK 0xc0000092

4.69.2.163 #define SMB_STATUS_FLOAT_UNDERFLOW 0xc0000093

4.69.2.164 #define SMB_STATUS_INTEGER_DIVIDE_BY_ZERO 0xc0000094

4.69.2.165 #define SMB_STATUS_INTEGER_OVERFLOW 0xc0000095

4.69.2.166 #define SMB_STATUS_PRIVILEGED_INSTRUCTION 0xc0000096

4.69.2.167 #define SMB_STATUS_TOO_MANY_PAGING_FILES 0xc0000097

4.69.2.168 #define SMB_STATUS_FILE_INVALID 0xc0000098

4.69.2.169 #define SMB_STATUS_ALLOTTED_SPACE_EXCEEDED 0xc0000099

4.69.2.170 #define SMB_STATUS_INSUFFICIENT_RESOURCES 0xc000009a

4.69.2.171 #define SMB_STATUS_DFS_EXIT_PATH_FOUND 0xc000009b

4.69.2.172 #define SMB_STATUS_DEVICE_DATA_ERROR 0xc000009c

4.69.2.173 #define SMB_STATUS_DEVICE_NOT_CONNECTED 0xc000009d

4.69.2.174 #define SMB_STATUS_DEVICE_POWER_FAILURE 0xc000009e

4.69.2.175 #define SMB_STATUS_FREE_VM_NOT_AT_BASE 0xc000009f

4.69.2.176 #define SMB_STATUS_MEMORY_NOT_ALLOCATED 0xc00000a0

4.69.2.177 #define SMB_STATUS_WORKING_SET_QUOTA 0xc00000a1

4.69.2.178 #define SMB_STATUS_MEDIA_WRITE_PROTECTED 0xc00000a2

4.69.2.179 #define SMB_STATUS_DEVICE_NOT_READY 0xc00000a3

4.69.2.180 #define SMB_STATUS_INVALID_GROUP_ATTRIBUTES 0xc00000a4

4.69.2.181 #define SMB_STATUS_BAD_IMPERSONATION_LEVEL 0xc00000a5

4.69.2.182 #define SMB_STATUS_CANT_OPEN_ANONYMOUS 0xc00000a6

4.69.2.183 #define SMB_STATUS_BAD_VALIDATION_CLASS 0xc00000a7

4.69.2.184 #define SMB_STATUS_BAD_TOKEN_TYPE 0xc00000a8

4.69.2.185 #define SMB_STATUS_BAD_MASTER_BOOT_RECORD 0xc00000a9

4.69.2.186 #define SMB_STATUS_INSTRUCTION_MISALIGNMENT 0xc00000aa

4.69.2.187 #define SMB_STATUS_INSTANCE_NOT_AVAILABLE 0xc00000ab

4.69.2.188 #define SMB_STATUS_PIPE_NOT_AVAILABLE 0xc00000ac

4.69.2.189 #define SMB_STATUS_INVALID_PIPE_STATE 0xc00000ad

4.69.2.190 #define SMB_STATUS_PIPE_BUSY 0xc00000ae

4.69.2.191 #define SMB_STATUS_ILLEGAL_FUNCTION 0xc00000af

4.69.2.192 #define SMB_STATUS_PIPE_DISCONNECTED 0xc00000b0

4.69.2.193 #define SMB_STATUS_PIPE_CLOSING 0xc00000b1

4.69.2.194 #define SMB_STATUS_PIPE_CONNECTED 0xc00000b2

4.69.2.195 #define SMB_STATUS_PIPE_LISTENING 0xc00000b3

4.69.2.196 #define SMB_STATUS_INVALID_READ_MODE 0xc00000b4

4.69.2.197 #define SMB_STATUS_IO_TIMEOUT 0xc00000b5

4.69.2.198 #define SMB_STATUS_FILE_FORCED_CLOSED 0xc00000b6

4.69.2.199 #define SMB_STATUS_PROFILING_NOT_STARTED 0xc00000b7

4.69.2.200 #define SMB_STATUS_PROFILING_NOT_STOPPED 0xc00000b8

4.69.2.201 #define SMB_STATUS_COULD_NOT_INTERPRET 0xc00000b9

4.69.2.202 #define SMB_STATUS_FILE_IS_A_DIRECTORY 0xc00000ba

4.69.2.203 #define SMB_STATUS_NOT_SUPPORTED 0xc00000bb

4.69.2.204 #define SMB_STATUS_REMOTE_NOT_LISTENING 0xc00000bc

4.69.2.205 #define SMB_STATUS_DUPLICATE_NAME 0xc00000bd

4.69.2.206 #define SMB_STATUS_BAD_NETWORK_PATH 0xc00000be

4.69.2.207 #define SMB_STATUS_NETWORK_BUSY 0xc00000bf

4.69.2.208 #define SMB_STATUS_DEVICE_DOES_NOT_EXIST 0xc00000c0

4.69.2.209 #define SMB_STATUS_TOO_MANY_COMMANDS 0xc00000c1

4.69.2.210 #define SMB_STATUS_ADAPTER_HARDWARE_ERROR 0xc00000c2

4.69.2.211 #define SMB_STATUS_INVALID_NETWORK_RESPONSE 0xc00000c3

4.69.2.212 #define SMB_STATUS_UNEXPECTED_NETWORK_ERROR 0xc00000c4

4.69.2.213 #define SMB_STATUS_BAD_REMOTE_ADAPTER 0xc00000c5

4.69.2.214 #define SMB_STATUS_PRINT_QUEUE_FULL 0xc00000c6

4.69.2.215 #define SMB_STATUS_NO_SPOOL_SPACE 0xc00000c7

4.69.2.216 #define SMB_STATUS_PRINT_CANCELLED 0xc00000c8

4.69.2.217 #define SMB_STATUS_NETWORK_NAME_DELETED 0xc00000c9

4.69.2.218 #define SMB_STATUS_NETWORK_ACCESS_DENIED 0xc00000ca

4.69.2.219 #define SMB_STATUS_BAD_DEVICE_TYPE 0xc00000cb

4.69.2.220 #define SMB_STATUS_BAD_NETWORK_NAME 0xc00000cc

4.69.2.221 #define SMB_STATUS_TOO_MANY_NAMES 0xc00000cd

4.69.2.222 #define SMB_STATUS_TOO_MANY_SESSIONS 0xc00000ce

4.69.2.223 #define SMB_STATUS_SHARING_PAUSED 0xc00000cf

4.69.2.224 #define SMB_STATUS_REQUEST_NOT_ACCEPTED 0xc00000d0

4.69.2.225 #define SMB_STATUS_REDIRECTOR_PAUSED 0xc00000d1

4.69.2.226 #define SMB_STATUS_NET_WRITE_FAULT 0xc00000d2

4.69.2.227 #define SMB_STATUS_PROFILING_AT_LIMIT 0xc00000d3

4.69.2.228 #define SMB_STATUS_NOT_SAME_DEVICE 0xc00000d4

4.69.2.229 #define SMB_STATUS_FILE_RENAMED 0xc00000d5

4.69.2.230 #define SMB_STATUS_VIRTUAL_CIRCUIT_CLOSED 0xc00000d6

4.69.2.231 #define SMB_STATUS_NO_SECURITY_ON_OBJECT 0xc00000d7

4.69.2.232 #define SMB_STATUS_CANT_WAIT 0xc00000d8

4.69.2.233 #define SMB_STATUS_PIPE_EMPTY 0xc00000d9

4.69.2.234 #define SMB_STATUS_CANT_ACCESS_DOMAIN_INFO 0xc00000da

4.69.2.235 #define SMB_STATUS_CANT_TERMINATE_SELF 0xc00000db

4.69.2.236 #define SMB_STATUS_INVALID_SERVER_STATE 0xc00000dc

4.69.2.237 #define SMB_STATUS_INVALID_DOMAIN_STATE 0xc00000dd

4.69.2.238 #define SMB_STATUS_INVALID_DOMAIN_ROLE 0xc00000de

4.69.2.239 #define SMB_STATUS_NO_SUCH_DOMAIN 0xc00000df

4.69.2.240 #define SMB_STATUS_DOMAIN_EXISTS 0xc00000e0

4.69.2.241 #define SMB_STATUS_DOMAIN_LIMIT_EXCEEDED 0xc00000e1

4.69.2.242 #define SMB_STATUS_OPLOCK_NOT_GRANTED 0xc00000e2

4.69.2.243 #define SMB_STATUS_INVALID_OPLOCK_PROTOCOL 0xc00000e3

4.69.2.244 #define SMB_STATUS_INTERNAL_DB_CORRUPTION 0xc00000e4

4.69.2.245 #define SMB_STATUS_INTERNAL_ERROR 0xc00000e5

4.69.2.246 #define SMB_STATUS_GENERIC_NOT_MAPPED 0xc00000e6

4.69.2.247 #define SMB_STATUS_BAD_DESCRIPTOR_FORMAT 0xc00000e7

4.69.2.248 #define SMB_STATUS_INVALID_USER_BUFFER 0xc00000e8

4.69.2.249 #define SMB_STATUS_UNEXPECTED_IO_ERROR 0xc00000e9

4.69.2.250 #define SMB_STATUS_UNEXPECTED_MM_CREATE_ERR 0xc00000ea

4.69.2.251 #define SMB_STATUS_UNEXPECTED_MM_MAP_ERROR 0xc00000eb

4.69.2.252 #define SMB_STATUS_UNEXPECTED_MM_EXTEND_ERR 0xc00000ec

4.69.2.253 #define SMB_STATUS_NOT_LOGON_PROCESS 0xc00000ed

4.69.2.254 #define SMB_STATUS_LOGON_SESSION_EXISTS 0xc00000ee

4.69.2.255 #define SMB_STATUS_INVALID_PARAMETER_1 0xc00000ef

4.69.2.256 #define SMB_STATUS_INVALID_PARAMETER_2 0xc00000f0

4.69.2.257 #define SMB_STATUS_INVALID_PARAMETER_3 0xc00000f1

4.69.2.258 #define SMB_STATUS_INVALID_PARAMETER_4 0xc00000f2

4.69.2.259 #define SMB_STATUS_INVALID_PARAMETER_5 0xc00000f3

4.69.2.260 #define SMB_STATUS_INVALID_PARAMETER_6 0xc00000f4

4.69.2.261 #define SMB_STATUS_INVALID_PARAMETER_7 0xc00000f5

4.69.2.262 #define SMB_STATUS_INVALID_PARAMETER_8 0xc00000f6

4.69.2.263 #define SMB_STATUS_INVALID_PARAMETER_9 0xc00000f7

4.69.2.264 #define SMB_STATUS_INVALID_PARAMETER_10 0xc00000f8

4.69.2.265 #define SMB_STATUS_INVALID_PARAMETER_11 0xc00000f9

4.69.2.266 #define SMB_STATUS_INVALID_PARAMETER_12 0xc00000fa

4.69.2.267 #define SMB_STATUS_REDIRECTOR_NOT_STARTED 0xc00000fb

4.69.2.268 #define SMB_STATUS_REDIRECTOR_STARTED 0xc00000fc

4.69.2.269 #define SMB_STATUS_STACK_OVERFLOW 0xc00000fd

4.69.2.270 #define SMB_STATUS_NO_SUCH_PACKAGE 0xc00000fe

4.69.2.271 #define SMB_STATUS_BAD_FUNCTION_TABLE 0xc00000ff

4.69.2.272 #define SMB_STATUS_DIRECTORY_NOT_EMPTY 0xc0000101

4.69.2.273 #define SMB_STATUS_FILE_CORRUPT_ERROR 0xc0000102

4.69.2.274 #define SMB_STATUS_NOT_A_DIRECTORY 0xc0000103

4.69.2.275 #define SMB_STATUS_BAD_LOGON_SESSION_STATE 0xc0000104

4.69.2.276 #define SMB_STATUS_LOGON_SESSION_COLLISION 0xc0000105

4.69.2.277 #define SMB_STATUS_NAME_TOO_LONG 0xc0000106

4.69.2.278 #define SMB_STATUS_FILES_OPEN 0xc0000107

4.69.2.279 #define SMB_STATUS_CONNECTION_IN_USE 0xc0000108

4.69.2.280 #define SMB_STATUS_MESSAGE_NOT_FOUND 0xc0000109

4.69.2.281 #define SMB_STATUS_PROCESS_IS_TERMINATING 0xc000010a

4.69.2.282 #define SMB_STATUS_INVALID_LOGON_TYPE 0xc000010b

4.69.2.283 #define SMB_STATUS_NO_GUID_TRANSLATION 0xc000010c

4.69.2.284 #define SMB_STATUS_CANNOT_IMPERSONATE 0xc000010d

4.69.2.285 #define SMB_STATUS_IMAGE_ALREADY_LOADED 0xc000010e

4.69.2.286 #define SMB_STATUS_ABIOIS_NOT_PRESENT 0xc000010f

4.69.2.287 #define SMB_STATUS_ABIOIS_LID_NOT_EXIST 0xc0000110

4.69.2.288 #define SMB_STATUS_ABIOIS_LID_ALREADY_OWNED 0xc0000111

4.69.2.289 #define SMB_STATUS_ABIOIS_NOT_LID_OWNER 0xc0000112

4.69.2.290 #define SMB_STATUS_ABIOIS_INVALID_COMMAND 0xc0000113

4.69.2.291 #define SMB_STATUS_ABIOIS_INVALID_LID 0xc0000114

4.69.2.292 #define SMB_STATUS_ABIOIS_SELECTOR_NOT_AVAILABLE 0xc0000115

4.69.2.293 #define SMB_STATUS_ABIOIS_INVALID_SELECTOR 0xc0000116

4.69.2.294 #define SMB_STATUS_NO_LDT 0xc0000117

4.69.2.295 #define SMB_STATUS_INVALID_LDT_SIZE 0xc0000118

4.69.2.296 #define SMB_STATUS_INVALID_LDT_OFFSET 0xc0000119

4.69.2.297 #define SMB_STATUS_INVALID_LDT_DESCRIPTOR 0xc000011a

4.69.2.298 #define SMB_STATUS_INVALID_IMAGE_NE_FORMAT 0xc000011b

4.69.2.299 #define SMB_STATUS_RXACT_INVALID_STATE 0xc000011c

4.69.2.300 #define SMB_STATUS_RXACT_COMMIT_FAILURE 0xc000011d

4.69.2.301 #define SMB_STATUS_MAPPED_FILE_SIZE_ZERO 0xc000011e

4.69.2.302 #define SMB_STATUS_TOO_MANY_OPENED_FILES 0xc000011f

4.69.2.303 #define SMB_STATUS_CANCELLED 0xc0000120

4.69.2.304 #define SMB_STATUS_CANNOT_DELETE 0xc0000121

4.69.2.305 #define SMB_STATUS_INVALID_COMPUTER_NAME 0xc0000122

4.69.2.306 #define SMB_STATUS_FILE_DELETED 0xc0000123

4.69.2.307 #define SMB_STATUS_SPECIAL_ACCOUNT 0xc0000124

4.69.2.308 #define SMB_STATUS_SPECIAL_GROUP 0xc0000125

4.69.2.309 #define SMB_STATUS_SPECIAL_USER 0xc0000126

4.69.2.310 #define SMB_STATUS_MEMBERS_PRIMARY_GROUP 0xc0000127

4.69.2.311 #define SMB_STATUS_FILE_CLOSED 0xc0000128

4.69.2.312 #define SMB_STATUS_TOO_MANY_THREADS 0xc0000129

4.69.2.313 #define SMB_STATUS_THREAD_NOT_IN_PROCESS 0xc000012a

4.69.2.314 #define SMB_STATUS_TOKEN_ALREADY_IN_USE 0xc000012b

4.69.2.315 #define SMB_STATUS_PAGEFILE_QUOTA_EXCEEDED 0xc000012c

4.69.2.316 #define SMB_STATUS_COMMITMENT_LIMIT 0xc000012d

4.69.2.317 #define SMB_STATUS_INVALID_IMAGE_LE_FORMAT 0xc000012e

4.69.2.318 #define SMB_STATUS_INVALID_IMAGE_NOT_MZ 0xc000012f

4.69.2.319 #define SMB_STATUS_INVALID_IMAGE_PROTECT 0xc0000130

4.69.2.320 #define SMB_STATUS_INVALID_IMAGE_WIN_16 0xc0000131

4.69.2.321 #define SMB_STATUS_LOGON_SERVER_CONFLICT 0xc0000132

4.69.2.322 #define SMB_STATUS_TIME_DIFFERENCE_AT_DC 0xc0000133

4.69.2.323 #define SMB_STATUS_SYNCHRONIZATION_REQUIRED 0xc0000134

4.69.2.324 #define SMB_STATUS_DLL_NOT_FOUND 0xc0000135

4.69.2.325 #define SMB_STATUS_OPEN_FAILED 0xc0000136

4.69.2.326 #define SMB_STATUS_IO_PRIVILEGE_FAILED 0xc0000137

4.69.2.327 #define SMB_STATUS_ORDINAL_NOT_FOUND 0xc0000138

4.69.2.328 #define SMB_STATUS_ENTRYPOINT_NOT_FOUND 0xc0000139

4.69.2.329 #define SMB_STATUS_CONTROL_C_EXIT 0xc000013a

4.69.2.330 #define SMB_STATUS_LOCAL_DISCONNECT 0xc000013b

4.69.2.331 #define SMB_STATUS_REMOTE_DISCONNECT 0xc000013c

4.69.2.332 #define SMB_STATUS_REMOTE_RESOURCES 0xc000013d

4.69.2.333 #define SMB_STATUS_LINK_FAILED 0xc000013e

4.69.2.334 #define SMB_STATUS_LINK_TIMEOUT 0xc000013f

4.69.2.335 #define SMB_STATUS_INVALID_CONNECTION 0xc0000140

4.69.2.336 #define SMB_STATUS_INVALID_ADDRESS 0xc0000141

4.69.2.337 #define SMB_STATUS_DLL_INIT_FAILED 0xc0000142

4.69.2.338 #define SMB_STATUS_MISSING_SYSTEMFILE 0xc0000143

4.69.2.339 #define SMB_STATUS_UNHANDLED_EXCEPTION 0xc0000144

4.69.2.340 #define SMB_STATUS_APP_INIT_FAILURE 0xc0000145

4.69.2.341 #define SMB_STATUS_PAGEFILE_CREATE_FAILED 0xc0000146

4.69.2.342 #define SMB_STATUS_NO_PAGEFILE 0xc0000147

4.69.2.343 #define SMB_STATUS_INVALID_LEVEL 0xc0000148

4.69.2.344 #define SMB_STATUS_WRONG_PASSWORD_CORE 0xc0000149

4.69.2.345 #define SMB_STATUS_ILLEGAL_FLOAT_CONTEXT 0xc000014a

4.69.2.346 #define SMB_STATUS_PIPE_BROKEN 0xc000014b

4.69.2.347 #define SMB_STATUS_REGISTRY_CORRUPT 0xc000014c

4.69.2.348 #define SMB_STATUS_REGISTRY_IO_FAILED 0xc000014d

4.69.2.349 #define SMB_STATUS_NO_EVENT_PAIR 0xc000014e

4.69.2.350 #define SMB_STATUS_UNRECOGNIZED_VOLUME 0xc000014f

4.69.2.351 #define SMB_STATUS_SERIAL_NO_DEVICE_INITED 0xc0000150

4.69.2.352 #define SMB_STATUS_NO_SUCH_ALIAS 0xc0000151

4.69.2.353 #define SMB_STATUS_MEMBER_NOT_IN_ALIAS 0xc0000152

4.69.2.354 #define SMB_STATUS_MEMBER_IN_ALIAS 0xc0000153

4.69.2.355 #define SMB_STATUS_ALIAS_EXISTS 0xc0000154

4.69.2.356 #define SMB_STATUS_LOGON_NOT_GRANTED 0xc0000155

4.69.2.357 #define SMB_STATUS_TOO_MANY_SECRETS 0xc0000156

4.69.2.358 #define SMB_STATUS_SECRET_TOO_LONG 0xc0000157

4.69.2.359 #define SMB_STATUS_INTERNAL_DB_ERROR 0xc0000158

4.69.2.360 #define SMB_STATUS_FULLSCREEN_MODE 0xc0000159

4.69.2.361 #define SMB_STATUS_TOO_MANY_CONTEXT_IDS 0xc000015a

4.69.2.362 #define SMB_STATUS_LOGON_TYPE_NOT_GRANTED 0xc000015b

4.69.2.363 #define SMB_STATUS_NOT_REGISTRY_FILE 0xc000015c

4.69.2.364 #define SMB_STATUS_NT_CROSS_ENCRYPTION_REQUIRED 0xc000015d

4.69.2.365 #define SMB_STATUS_DOMAIN_CTRLR_CONFIG_ERROR 0xc000015e

4.69.2.366 #define SMB_STATUS_FT_MISSING_MEMBER 0xc000015f

4.69.2.367 #define SMB_STATUS_ILL_FORMED_SERVICE_ENTRY 0xc0000160

4.69.2.368 #define SMB_STATUS_ILLEGAL_CHARACTER 0xc0000161

4.69.2.369 #define SMB_STATUS_UNMAPPABLE_CHARACTER 0xc0000162

4.69.2.370 #define SMB_STATUS_UNDEFINED_CHARACTER 0xc0000163

4.69.2.371 #define SMB_STATUS_FLOPPY_VOLUME 0xc0000164

4.69.2.372 #define SMB_STATUS_FLOPPY_ID_MARK_NOT_FOUND 0xc0000165

4.69.2.373 #define SMB_STATUS_FLOPPY_WRONG_CYLINDER 0xc0000166

4.69.2.374 #define SMB_STATUS_FLOPPY_UNKNOWN_ERROR 0xc0000167

4.69.2.375 #define SMB_STATUS_FLOPPY_BAD_REGISTERS 0xc0000168

4.69.2.376 #define SMB_STATUS_DISK_RECALIBRATE_FAILED 0xc0000169

4.69.2.377 #define SMB_STATUS_DISK_OPERATION_FAILED 0xc000016a

4.69.2.378 #define SMB_STATUS_DISK_RESET_FAILED 0xc000016b

4.69.2.379 #define SMB_STATUS_SHARED_IRQ_BUSY 0xc000016c

4.69.2.380 #define SMB_STATUS_FT_ORPHANING 0xc000016d

4.69.2.381 #define SMB_STATUS_PARTITION_FAILURE 0xc0000172

4.69.2.382 #define SMB_STATUS_INVALID_BLOCK_LENGTH 0xc0000173

4.69.2.383 #define SMB_STATUS_DEVICE_NOT_PARTITIONED 0xc0000174

4.69.2.384 #define SMB_STATUS_UNABLE_TO_LOCK_MEDIA 0xc0000175

4.69.2.385 #define SMB_STATUS_UNABLE_TO_UNLOAD_MEDIA 0xc0000176

4.69.2.386 #define SMB_STATUS_EOM_OVERFLOW 0xc0000177

4.69.2.387 #define SMB_STATUS_NO_MEDIA 0xc0000178

4.69.2.388 #define SMB_STATUS_NO_SUCH_MEMBER 0xc000017a

4.69.2.389 #define SMB_STATUS_INVALID_MEMBER 0xc000017b

4.69.2.390 #define SMB_STATUS_KEY_DELETED 0xc000017c

4.69.2.391 #define SMB_STATUS_NO_LOG_SPACE 0xc000017d

4.69.2.392 #define SMB_STATUS_TOO_MANY_SIDS 0xc000017e

4.69.2.393 #define SMB_STATUS_LM_CROSS_ENCRYPTION_REQUIRED 0xc000017f

4.69.2.394 #define SMB_STATUS_KEY_HAS_CHILDREN 0xc0000180

4.69.2.395 #define SMB_STATUS_CHILD_MUST_BE_VOLATILE 0xc0000181

4.69.2.396 #define SMB_STATUS_DEVICE_CONFIGURATION_ERROR 0xc0000182

4.69.2.397 #define SMB_STATUS_DRIVER_INTERNAL_ERROR 0xc0000183

4.69.2.398 #define SMB_STATUS_INVALID_DEVICE_STATE 0xc0000184

4.69.2.399 #define SMB_STATUS_IO_DEVICE_ERROR 0xc0000185

4.69.2.400 #define SMB_STATUS_DEVICE_PROTOCOL_ERROR 0xc0000186

4.69.2.401 #define SMB_STATUS_BACKUP_CONTROLLER 0xc0000187

4.69.2.402 #define SMB_STATUS_LOG_FILE_FULL 0xc0000188

4.69.2.403 #define SMB_STATUS_TOO_LATE 0xc0000189

4.69.2.404 #define SMB_STATUS_NO_TRUST_LSA_SECRET 0xc000018a

4.69.2.405 #define SMB_STATUS_NO_TRUST_SAM_ACCOUNT 0xc000018b

4.69.2.406 #define SMB_STATUS_TRUSTED_DOMAIN_FAILURE 0xc000018c

4.69.2.407 #define SMB_STATUS_TRUSTED_RELATIONSHIP_FAILURE 0xc000018d

4.69.2.408 #define SMB_STATUS_EVENTLOG_FILE_CORRUPT 0xc000018e

4.69.2.409 #define SMB_STATUS_EVENTLOG_CANT_START 0xc000018f

4.69.2.410 #define SMB_STATUS_TRUST_FAILURE 0xc0000190

4.69.2.411 #define SMB_STATUS_MUTANT_LIMIT_EXCEEDED 0xc0000191

4.69.2.412 #define SMB_STATUS_NETLOGON_NOT_STARTED 0xc0000192

4.69.2.413 #define SMB_STATUS_ACCOUNT_EXPIRED 0xc0000193

4.69.2.414 #define SMB_STATUS_POSSIBLE_DEADLOCK 0xc0000194

4.69.2.415 #define SMB_STATUS_NETWORK_CREDENTIAL_CONFLICT 0xc0000195

4.69.2.416 #define SMB_STATUS_REMOTE_SESSION_LIMIT 0xc0000196

4.69.2.417 #define SMB_STATUS_EVENTLOG_FILE_CHANGED 0xc0000197

4.69.2.418 #define SMB_STATUS_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT 0xc0000198

4.69.2.419 #define SMB_STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT 0xc0000199

4.69.2.420 #define SMB_STATUS_NOLOGON_SERVER_TRUST_ACCOUNT 0xc000019a

4.69.2.421 #define SMB_STATUS_DOMAIN_TRUST_INCONSISTENT 0xc000019b

4.69.2.422 #define SMB_STATUS_FS_DRIVER_REQUIRED 0xc000019c

4.69.2.423 #define SMB_STATUS_INVALID_LOCK_RANGE 0xc00001a1

4.69.2.424 #define SMB_STATUS_NO_USER_SESSION_KEY 0xc0000202

4.69.2.425 #define SMB_STATUS_USER_SESSION_DELETED 0xc0000203

4.69.2.426 #define SMB_STATUS_RESOURCE_LANG_NOT_FOUND 0xc0000204

4.69.2.427 #define SMB_STATUS_INSUFF_SERVER_RESOURCES 0xc0000205

4.69.2.428 #define SMB_STATUS_INVALID_BUFFER_SIZE 0xc0000206

4.69.2.429 #define SMB_STATUS_INVALID_ADDRESS_COMPONENT 0xc0000207

4.69.2.430 #define SMB_STATUS_INVALID_ADDRESS_WILDCARD 0xc0000208

4.69.2.431 #define SMB_STATUS_TOO_MANY_ADDRESSES 0xc0000209

4.69.2.432 #define SMB_STATUS_ADDRESS_ALREADY_EXISTS 0xc000020a

4.69.2.433 #define SMB_STATUS_ADDRESS_CLOSED 0xc000020b

4.69.2.434 #define SMB_STATUS_CONNECTION_DISCONNECTED 0xc000020c

4.69.2.435 #define SMB_STATUS_CONNECTION_RESET 0xc000020d

4.69.2.436 #define SMB_STATUS_TOO_MANY_NODES 0xc000020e

4.69.2.437 #define SMB_STATUS_TRANSACTION_ABORTED 0xc000020f

4.69.2.438 #define SMB_STATUS_TRANSACTION_TIMED_OUT 0xc0000210

4.69.2.439 #define SMB_STATUS_TRANSACTION_NO_RELEASE 0xc0000211

4.69.2.440 #define SMB_STATUS_TRANSACTION_NO_MATCH 0xc0000212

4.69.2.441 #define SMB_STATUS_TRANSACTION_RESPONDED 0xc0000213

4.69.2.442 #define SMB_STATUS_TRANSACTION_INVALID_ID 0xc0000214

4.69.2.443 #define SMB_STATUS_TRANSACTION_INVALID_TYPE 0xc0000215

4.69.2.444 #define SMB_STATUS_NOT_SERVER_SESSION 0xc0000216

4.69.2.445 #define SMB_STATUS_NOT_CLIENT_SESSION 0xc0000217

4.69.2.446 #define SMB_STATUS_CANNOT_LOAD_REGISTRY_FILE 0xc0000218

4.69.2.447 #define SMB_STATUS_DEBUG_ATTACH_FAILED 0xc0000219

4.69.2.448 #define SMB_STATUS_SYSTEM_PROCESS_TERMINATED 0xc000021a

4.69.2.449 #define SMB_STATUS_DATA_NOT_ACCEPTED 0xc000021b

4.69.2.450 #define SMB_STATUS_NO_BROWSER_SERVERS_FOUND 0xc000021c

4.69.2.451 #define SMB_STATUS_VDM_HARD_ERROR 0xc000021d

4.69.2.452 #define SMB_STATUS_DRIVER_CANCEL_TIMEOUT 0xc000021e

4.69.2.453 #define SMB_STATUS_REPLY_MESSAGE_MISMATCH 0xc000021f

4.69.2.454 #define SMB_STATUS_MAPPED_ALIGNMENT 0xc0000220

4.69.2.455 #define SMB_STATUS_IMAGE_CHECKSUM_MISMATCH 0xc0000221

4.69.2.456 #define SMB_STATUS_LOST_WRITEBEHIND_DATA 0xc0000222

4.69.2.457 #define SMB_STATUS_CLIENT_SERVER_PARAMETERS_INVALID 0xc0000223

4.69.2.458 #define SMB_STATUS_PASSWORD_MUST_CHANGE 0xc0000224

4.69.2.459 #define SMB_STATUS_NOT_FOUND 0xc0000225

4.69.2.460 #define SMB_STATUS_NOT_TINY_STREAM 0xc0000226

4.69.2.461 #define SMB_STATUS_RECOVERY_FAILURE 0xc0000227

4.69.2.462 #define SMB_STATUS_STACK_OVERFLOW_READ 0xc0000228

4.69.2.463 #define SMB_STATUS_FAIL_CHECK 0xc0000229

4.69.2.464 #define SMB_STATUS_DUPLICATE_OBJECTID 0xc000022a

4.69.2.465 #define SMB_STATUS_OBJECTID_EXISTS 0xc000022b

4.69.2.466 #define SMB_STATUS_CONVERT_TO_LARGE 0xc000022c

4.69.2.467 #define SMB_STATUS_RETRY 0xc000022d

4.69.2.468 #define SMB_STATUS_FOUND_OUT_OF_SCOPE 0xc000022e

4.69.2.469 #define SMB_STATUS_ALLOCATE_BUCKET 0xc000022f

4.69.2.470 #define SMB_STATUS_PROPSET_NOT_FOUND 0xc0000230

4.69.2.471 #define SMB_STATUS_MARSHALL_OVERFLOW 0xc0000231

4.69.2.472 #define SMB_STATUS_INVALID_VARIANT 0xc0000232

4.69.2.473 #define SMB_STATUS_DOMAIN_CONTROLLER_NOT_FOUND 0xc0000233

4.69.2.474 #define SMB_STATUS_ACCOUNT_LOCKED_OUT 0xc0000234

4.69.2.475 #define SMB_STATUS_HANDLE_NOT_CLOSABLE 0xc0000235

4.69.2.476 #define SMB_STATUS_CONNECTION_REFUSED 0xc0000236

4.69.2.477 #define SMB_STATUS_GRACEFUL_DISCONNECT 0xc0000237

4.69.2.478 #define SMB_STATUS_ADDRESS_ALREADY_ASSOCIATED 0xc0000238

4.69.2.479 #define SMB_STATUS_ADDRESS_NOT_ASSOCIATED 0xc0000239

4.69.2.480 #define SMB_STATUS_CONNECTION_INVALID 0xc000023a

4.69.2.481 #define SMB_STATUS_CONNECTION_ACTIVE 0xc000023b

4.69.2.482 #define SMB_STATUS_NETWORK_UNREACHABLE 0xc000023c

4.69.2.483 #define SMB_STATUS_HOST_UNREACHABLE 0xc000023d

4.69.2.484 #define SMB_STATUS_PROTOCOL_UNREACHABLE 0xc000023e

4.69.2.485 #define SMB_STATUS_PORT_UNREACHABLE 0xc000023f

4.69.2.486 #define SMB_STATUS_REQUEST_ABORTED 0xc0000240

4.69.2.487 #define SMB_STATUS_CONNECTION_ABORTED 0xc0000241

4.69.2.488 #define SMB_STATUS_BAD_COMPRESSION_BUFFER 0xc0000242

4.69.2.489 #define SMB_STATUS_USER_MAPPED_FILE 0xc0000243

4.69.2.490 #define SMB_STATUS_AUDIT_FAILED 0xc0000244

4.69.2.491 #define SMB_STATUS_TIMER_RESOLUTION_NOT_SET 0xc0000245

4.69.2.492 #define SMB_STATUS_CONNECTION_COUNT_LIMIT 0xc0000246

4.69.2.493 #define SMB_STATUS_LOGIN_TIME_RESTRICTION 0xc0000247

4.69.2.494 #define SMB_STATUS_LOGIN_WKSTA_RESTRICTION 0xc0000248

4.69.2.495 #define SMB_STATUS_IMAGE_MP_UP_MISMATCH 0xc0000249

4.69.2.496 #define SMB_STATUS_INSUFFICIENT_LOGON_INFO 0xc0000250

4.69.2.497 #define SMB_STATUS_BAD_DLL_ENTRYPOINT 0xc0000251

4.69.2.498 #define SMB_STATUS_BAD_SERVICE_ENTRYPOINT 0xc0000252

4.69.2.499 #define SMB_STATUS_LPC_REPLY_LOST 0xc0000253

4.69.2.500 #define SMB_STATUS_IP_ADDRESS_CONFLICT1 0xc0000254

4.69.2.501 #define SMB_STATUS_IP_ADDRESS_CONFLICT2 0xc0000255

4.69.2.502 #define SMB_STATUS_REGISTRY_QUOTA_LIMIT 0xc0000256

4.69.2.503 #define SMB_STATUS_PATH_NOT_COVERED 0xc0000257

4.69.2.504 #define SMB_STATUS_NO_CALLBACK_ACTIVE 0xc0000258

4.69.2.505 #define SMB_STATUS_LICENSE_QUOTA_EXCEEDED 0xc0000259

4.69.2.506 #define SMB_STATUS_PWD_TOO_SHORT 0xc000025a

4.69.2.507 #define SMB_STATUS_PWD_TOO_RECENT 0xc000025b

4.69.2.508 #define SMB_STATUS_PWD_HISTORY_CONFLICT 0xc000025c

4.69.2.509 #define SMB_STATUS_PLUGPLAY_NO_DEVICE 0xc000025e

4.69.2.510 #define SMB_STATUS_UNSUPPORTED_COMPRESSION 0xc000025f

4.69.2.511 #define SMB_STATUS_INVALID_HW_PROFILE 0xc0000260

4.69.2.512 #define SMB_STATUS_INVALID_PLUGPLAY_DEVICE_PATH 0xc0000261

4.69.2.513 #define SMB_STATUS_DRIVER_ORDINAL_NOT_FOUND 0xc0000262

4.69.2.514 #define SMB_STATUS_DRIVER_ENTRYPOINT_NOT_FOUND 0xc0000263

4.69.2.515 #define SMB_STATUS_RESOURCE_NOT_OWNED 0xc0000264

4.69.2.516 #define SMB_STATUS_TOO_MANY_LINKS 0xc0000265

4.69.2.517 #define SMB_STATUS_QUOTA_LIST_INCONSISTENT 0xc0000266

4.69.2.518 #define SMB_STATUS_FILE_IS_OFFLINE 0xc0000267

4.69.2.519 #define SMB_STATUS_DFS_UNAVAILABLE 0xc000026d

4.69.2.520 #define SMB_STATUS_VOLUME_DISMOUNTED 0xc000026e

4.69.2.521 #define SMB_STATUS_NOT_A_REPARSE_POINT 0xc0000275

4.69.2.522 #define SMB_STATUS_IO_REPARSE_TAG_NOT_HANDLED 0xc0000279

4.69.2.523 #define SMB_STATUS_NETWORK_SESSION_EXPIRED 0xc000035C

4.69.2.524 #define SMB_STATUS_INVALID_TOKEN 0xc0000465

4.69.2.525 #define SMB_STATUS_SERVER_UNAVAILABLE 0xc0000466

4.69.2.526 #define SMB_STATUS_FILE_NOT_AVAILABLE 0xc0000467

4.69.2.527 #define SMB_STATUS_SHARE_UNAVAILABLE 0xc0000480

4.69.2.528 #define SMB_STATUS_NO_SUCH_JOB 0xc0000ede

4.69.2.529 #define SMB_STATUS_NO_PREAUTH_INTEGRITY_HASH_OVERLAP 0xc05D0000

4.70 Internal NT Statuses

Macros

- #define [SMB_STATUS_DISCONNECT](#) 0xe0000001
- #define [SMB_STATUS_INTERNAL_RELEASE_REQUEST_RESPONSE](#) 0xe0000002
- #define [SMB_STATUS_INTERNAL_BUFFER_TOO_SMALL](#) 0xe0000004
- #define [SMB_STATUS_DONOTRELEASERESPONSE](#) 0xe0000005

4.70.1 Detailed Description

4.70.2 Macro Definition Documentation

4.70.2.1 #define [SMB_STATUS_DISCONNECT](#) 0xe0000001

4.70.2.2 #define [SMB_STATUS_INTERNAL_RELEASE_REQUEST_RESPONSE](#) 0xe0000002

4.70.2.3 #define [SMB_STATUS_INTERNAL_BUFFER_TOO_SMALL](#) 0xe0000004

4.70.2.4 #define [SMB_STATUS_DONOTRELEASERESPONSE](#) 0xe0000005

4.71 NQ Error Codes

Macros

- `#define NQ_ERR_MODULE (NQ_INT)(255u << 24)`
- `#define IS_NQ_ERROR(err) ((err & NQ_ERR_MODULE) == NQ_ERR_MODULE)`
- `#define NQ_ERR_OK (0)`
- `#define NQ_ERR_BADPARAM (NQ_ERR_MODULE | 3)`
- `#define NQ_ERR_GETDATA (NQ_ERR_MODULE | 18)`
- `#define NQ_ERR_INVALIDMODE (NQ_ERR_MODULE | 19)`
- `#define NQ_ERR_NOSERVERMAC (NQ_ERR_MODULE | 20)`
- `#define NQ_ERR_SIGNATUREFAIL (NQ_ERR_MODULE | 21)`
- `#define NQ_ERR_OBJEXISTS (NQ_ERR_MODULE | 22)`
- `#define NQ_ERR_MOUNTERROR (NQ_ERR_MODULE | 23)`
- `#define NQ_ERR_UNABLETODISPOSE (NQ_ERR_MODULE | 24)`
- `#define NQ_ERR_INVALIDHANDLE (NQ_ERR_MODULE | 25)`
- `#define NQ_ERR_NEGOTIATEFAILED (NQ_ERR_MODULE | 26)`
- `#define NQ_ERR_PATHNOTCOVERED (NQ_ERR_MODULE | 27)`
- `#define NQ_ERR_DFSCACHEOVERFLOW (NQ_ERR_MODULE | 28)`
- `#define NQ_ERR_ACCOUNTLOCKEDOUT (NQ_ERR_MODULE | 29)`
- `#define NQ_ERR_USEREXISTS (NQ_ERR_MODULE | 30)`
- `#define NQ_ERR_USERNOTFOUND (NQ_ERR_MODULE | 31)`
- `#define NQ_ERR_NOTFOUND (NQ_ERR_MODULE | 32)`
- `#define NQ_ERR_LOGONFAILURE (NQ_ERR_MODULE | 33)`
- `#define NQ_ERR_VOLUMEDISMOUNTED (NQ_ERR_MODULE | 34)`
- `#define NQ_ERR_NETWORKERROR (NQ_ERR_MODULE | 35)`
- `#define NQ_ERR_BADCONNECTION (NQ_ERR_MODULE | 36)`
- `#define NQ_ERR_SHARINGPAUSED (NQ_ERR_MODULE | 37)`
- `#define NQ_ERR_BADDFS (NQ_ERR_MODULE | 38)`
- `#define NQ_ERR_IOTIMEOUT (NQ_ERR_MODULE | 39)`
- `#define NQ_ERR_TRYAGAIN (NQ_ERR_MODULE | 40)`
- `#define NQ_ERR_ACCOUNTEXPIRED (NQ_ERR_MODULE | 41)`
- `#define NQ_ERR_INVALIDLOGONHOURS (NQ_ERR_MODULE | 42)`
- `#define NQ_ERR_ACCOUNTDISABLED (NQ_ERR_MODULE | 43)`
- `#define NQ_ERR_SESSIONREAUTHREQUIRED (NQ_ERR_MODULE | 44)`
- `#define NQ_ERR_USERRELOGONFAILED (NQ_ERR_MODULE | 45)`
- `#define NQ_ERR_SERVERUNAVAILABLE (NQ_ERR_MODULE | 46)`
- `#define NQ_ERR_SHAREUNAVAILABLE (NQ_ERR_MODULE | 47)`
- `#define NQ_ERR_OUTOFMEMORY (NQ_ERR_MODULE | 901)`
- `#define NQ_ERR_BADFUNC (NQ_ERR_MODULE | 1001)`
- `#define NQ_ERR_BADFILE (NQ_ERR_MODULE | 1002)`
- `#define NQ_ERR_BADPATH (NQ_ERR_MODULE | 1003)`
- `#define NQ_ERR_NOFIDS (NQ_ERR_MODULE | 1004)`
- `#define NQ_ERR_NOACCESS (NQ_ERR_MODULE | 1005)`
- `#define NQ_ERR_BADFID (NQ_ERR_MODULE | 1006)`
- `#define NQ_ERR_BADMCB (NQ_ERR_MODULE | 1007)`
- `#define NQ_ERR_NOMEM (NQ_ERR_MODULE | 1008)`
- `#define NQ_ERR_BADMEM (NQ_ERR_MODULE | 1009)`
- `#define NQ_ERR_BADENV (NQ_ERR_MODULE | 1010)`
- `#define NQ_ERR_BADFORMAT (NQ_ERR_MODULE | 1011)`
- `#define NQ_ERR_BADACCESS (NQ_ERR_MODULE | 1012)`
- `#define NQ_ERR_BADDATA (NQ_ERR_MODULE | 1013)`
- `#define NQ_ERR_BADDRIVE (NQ_ERR_MODULE | 1015)`
- `#define NQ_ERR_REMCD (NQ_ERR_MODULE | 1016)`

- #define NQ_ERR_DIFFDEVICE (NQ_ERR_MODULE | 1017)
- #define NQ_ERR_NOFILES (NQ_ERR_MODULE | 1018)
- #define NQ_ERR_WRONGDIALECT (NQ_ERR_MODULE | 1019)
- #define NQ_ERR_BADSHARE (NQ_ERR_MODULE | 1032)
- #define NQ_ERR_LOCK (NQ_ERR_MODULE | 1033)
- #define NQ_ERR_DONTSUPPORTIPC (NQ_ERR_MODULE | 1066)
- #define NQ_ERR_SHARINGVIOLATION (NQ_ERR_MODULE | 1067)
- #define NQ_ERR_FILEEXISTS (NQ_ERR_MODULE | 1080)
- #define NQ_ERR_BADDIRECTORY (NQ_ERR_MODULE | 1087)
- #define NQ_ERR_INSUFFICIENTBUFFER (NQ_ERR_MODULE | 1122)
- #define NQ_ERR_INVALIDNAME (NQ_ERR_MODULE | 1123)
- #define NQ_ERR_DIRNOTEMPTY (NQ_ERR_MODULE | 1145)
- #define NQ_ERR_ALREADYEXISTS (NQ_ERR_MODULE | 1183)
- #define NQ_ERR_BADPIPE (NQ_ERR_MODULE | 1230)
- #define NQ_ERR_PIPEBUSY (NQ_ERR_MODULE | 1231)
- #define NQ_ERR_PIPECLOSING (NQ_ERR_MODULE | 1232)
- #define NQ_ERR_NOTCONNECTED (NQ_ERR_MODULE | 1233)
- #define NQ_ERR_MOREDATA (NQ_ERR_MODULE | 1234)
- #define NQ_ERR_INVALIDNAMELEN (NQ_ERR_MODULE | 1235)
- #define NQ_ERR_ERROR (NQ_ERR_MODULE | 2001)
- #define NQ_ERR_BADPW (NQ_ERR_MODULE | 2002)
- #define NQ_ERR_ACCESS (NQ_ERR_MODULE | 2004)
- #define NQ_ERR_INVTID (NQ_ERR_MODULE | 2005)
- #define NQ_ERR_INVNETNAME (NQ_ERR_MODULE | 2006)
- #define NQ_ERR_INVDEVICE (NQ_ERR_MODULE | 2007)
- #define NQ_ERR_QFULL (NQ_ERR_MODULE | 2049)
- #define NQ_ERR_QTOOBIG (NQ_ERR_MODULE | 2050)
- #define NQ_ERR_QEOF (NQ_ERR_MODULE | 2051)
- #define NQ_ERR_INVFID (NQ_ERR_MODULE | 2052)
- #define NQ_ERR_SMBCMD (NQ_ERR_MODULE | 2064)
- #define NQ_ERR_SRVERORROR (NQ_ERR_MODULE | 2065)
- #define NQ_ERR_FILESPECS (NQ_ERR_MODULE | 2067)
- #define NQ_ERR_BADPERMITS (NQ_ERR_MODULE | 2069)
- #define NQ_ERR_SETATTRMODE (NQ_ERR_MODULE | 2071)
- #define NQ_ERR_PAUSED (NQ_ERR_MODULE | 2081)
- #define NQ_ERR_MSGOFF (NQ_ERR_MODULE | 2082)
- #define NQ_ERR_NOROOM (NQ_ERR_MODULE | 2083)
- #define NQ_ERR_RMUNS (NQ_ERR_MODULE | 2087)
- #define NQ_ERR_TIMEOUT (NQ_ERR_MODULE | 2088)
- #define NQ_ERR_NORESOURCE (NQ_ERR_MODULE | 2089)
- #define NQ_ERR_TOOMANYUIDS (NQ_ERR_MODULE | 2090)
- #define NQ_ERR_INVUID (NQ_ERR_MODULE | 2091)
- #define NQ_ERR_USEMPX (NQ_ERR_MODULE | 2250)
- #define NQ_ERR_USESTD (NQ_ERR_MODULE | 2251)
- #define NQ_ERR_CONTMPX (NQ_ERR_MODULE | 2252)
- #define NQ_ERR_NOSUPPORT (NQ_ERR_MODULE | 2999)
- #define NQ_ERR_NOWRITE (NQ_ERR_MODULE | 3019)
- #define NQ_ERR_BADUNIT (NQ_ERR_MODULE | 3020)
- #define NQ_ERR_NOTREADY (NQ_ERR_MODULE | 3021)
- #define NQ_ERR_BADCMD (NQ_ERR_MODULE | 3022)
- #define NQ_ERR_DATA (NQ_ERR_MODULE | 3023)
- #define NQ_ERR_BADREQ (NQ_ERR_MODULE | 3024)
- #define NQ_ERR_SEEK (NQ_ERR_MODULE | 3025)
- #define NQ_ERR_BADMEDIA (NQ_ERR_MODULE | 3026)
- #define NQ_ERR_BADSECTOR (NQ_ERR_MODULE | 3027)

- `#define NQ_ERR_NOPAPER (NQ_ERR_MODULE | 3028)`
- `#define NQ_ERR_WRITE (NQ_ERR_MODULE | 3029)`
- `#define NQ_ERR_READ (NQ_ERR_MODULE | 3030)`
- `#define NQ_ERR_GENERAL (NQ_ERR_MODULE | 3031)`
- `#define NQ_ERR_WRONGDISK (NQ_ERR_MODULE | 3034)`
- `#define NQ_ERR_FCBUNAVAIL (NQ_ERR_MODULE | 3035)`
- `#define NQ_ERR_SHAREBUFEXC (NQ_ERR_MODULE | 3036)`
- `#define NQ_ERR_DISKFULL (NQ_ERR_MODULE | 3039)`

4.71.1 Detailed Description

4.71.2 Macro Definition Documentation

4.71.2.1 `#define NQ_ERR_MODULE (NQ_INT)(255u << 24)`

This module defines the NQ error subset

4.71.2.2 `#define IS_NQ_ERROR(err) ((err & NQ_ERR_MODULE) == NQ_ERR_MODULE)`

4.71.2.3 `#define NQ_ERR_OK (0)`

Success

4.71.2.4 `#define NQ_ERR_BADPARAM (NQ_ERR_MODULE | 3)`

Parameter error

4.71.2.5 `#define NQ_ERR_GETDATA (NQ_ERR_MODULE | 18)`

Error retrieving data

4.71.2.6 `#define NQ_ERR_INVALIDMODE (NQ_ERR_MODULE | 19)`

Invalid open mode

4.71.2.7 `#define NQ_ERR_NOSERVERMAC (NQ_ERR_MODULE | 20)`

Server doesn't support MAC signing

4.71.2.8 `#define NQ_ERR_SIGNATUREFAIL (NQ_ERR_MODULE | 21)`

MAC signature in incoming packet was broken

4.71.2.9 `#define NQ_ERR_OBJEXISTS (NQ_ERR_MODULE | 22)`

Database object already exists and cannot be created

4.71.2.10 `#define NQ_ERR_MOUNTERROR (NQ_ERR_MODULE | 23)`

Mount failed for a reason other than authentication

4.71.2.11 **#define NQ_ERR_UNABLETODISPOSE (NQ_ERR_MODULE | 24)**

Unable to dispose resources

4.71.2.12 **#define NQ_ERR_INVALIDHANDLE (NQ_ERR_MODULE | 25)**

Invalid handle passed

4.71.2.13 **#define NQ_ERR_NEGOTIATEFAILED (NQ_ERR_MODULE | 26)**

SMB negotiation failed

4.71.2.14 **#define NQ_ERR_PATHNOTCOVERED (NQ_ERR_MODULE | 27)**

Path should be resolved over DFS

4.71.2.15 **#define NQ_ERR_DFSCACHEOVERFLOW (NQ_ERR_MODULE | 28)**

DFS cache overflow

4.71.2.16 **#define NQ_ERR_ACCOUNTLOCKEDOUT (NQ_ERR_MODULE | 29)**

Account locked out

4.71.2.17 **#define NQ_ERR_USEREXISTS (NQ_ERR_MODULE | 30)**

Account already exists

4.71.2.18 **#define NQ_ERR_USERNOTFOUND (NQ_ERR_MODULE | 31)**

Account name not mapped

4.71.2.19 **#define NQ_ERR_NOTFOUND (NQ_ERR_MODULE | 32)**

Not found

4.71.2.20 **#define NQ_ERR_LOGONFAILURE (NQ_ERR_MODULE | 33)**

Logon failure

4.71.2.21 **#define NQ_ERR_VOLUMEDISMOUNTED (NQ_ERR_MODULE | 34)**

Volume not mounted

4.71.2.22 **#define NQ_ERR_NETWORKERROR (NQ_ERR_MODULE | 35)**

Network error: unreachable, busy, name deleted

4.71.2.23 #define NQ_ERR_BADCONNECTION (NQ_ERR_MODULE | 36)

Connection error: refused, disconnected, reset

4.71.2.24 #define NQ_ERR_SHARINGPAUSED (NQ_ERR_MODULE | 37)

File sharing error: has been temporarily paused

4.71.2.25 #define NQ_ERR_BADDFS (NQ_ERR_MODULE | 38)

DFS error: DFS not available on a server

4.71.2.26 #define NQ_ERR_IOTIMEOUT (NQ_ERR_MODULE | 39)

IO error: time-out period expired

4.71.2.27 #define NQ_ERR_TRYAGAIN (NQ_ERR_MODULE | 40)

Connection changed: possibly reconnect occurred - retry

4.71.2.28 #define NQ_ERR_ACCOUNTEXPIRED (NQ_ERR_MODULE | 41)

Account expired

4.71.2.29 #define NQ_ERR_INVALIDLOGONHOURS (NQ_ERR_MODULE | 42)

Invalid logon hours

4.71.2.30 #define NQ_ERR_ACCOUNTDISABLED (NQ_ERR_MODULE | 43)

Account is disabled

4.71.2.31 #define NQ_ERR_SESSIONREAUTHREQUIRED (NQ_ERR_MODULE | 44)

session reauthentication is required

4.71.2.32 #define NQ_ERR_USERRELOGONFAILED (NQ_ERR_MODULE | 45)

user relog failed

4.71.2.33 #define NQ_ERR_SERVERUNAVAILABLE (NQ_ERR_MODULE | 46)

the file server is temporarily unavailable

4.71.2.34 #define NQ_ERR_SHAREUNAVAILABLE (NQ_ERR_MODULE | 47)

the share is temporarily unavailable

4.71.2.35 #define NQ_ERR_OUTOFMEMORY (NQ_ERR_MODULE | 901)

Failed to allocate memory block

4.71.2.36 #define NQ_ERR_BADFUNC (NQ_ERR_MODULE | 1001)

Invalid function. The server did not recognize or could not perform a system call generated by the server, e.g. set the DIRECTORY attribute on a data file, invalid seek mode.

4.71.2.37 #define NQ_ERR_BADFILE (NQ_ERR_MODULE | 1002)

File not found. The last component of a file's pathname could not be found.

4.71.2.38 #define NQ_ERR_BADPATH (NQ_ERR_MODULE | 1003)

Directory invalid. A directory component in a pathname could not be found.

4.71.2.39 #define NQ_ERR_NOFIDS (NQ_ERR_MODULE | 1004)

Too many open files. The server has no file handles available.

4.71.2.40 #define NQ_ERR_NOACCESS (NQ_ERR_MODULE | 1005)

Access denied, the client's context does not permit the requested function. This includes the following conditions: invalid rename command, write to a file open for read only, read on a file open for write only, attempt to delete a non-empty directory.

4.71.2.41 #define NQ_ERR_BADFID (NQ_ERR_MODULE | 1006)

Invalid file handle. The file handle specified was not recognized by the server.

4.71.2.42 #define NQ_ERR_BADMCB (NQ_ERR_MODULE | 1007)

Memory control blocks destroyed

4.71.2.43 #define NQ_ERR_NOMEM (NQ_ERR_MODULE | 1008)

Insufficient server memory to perform the requested function

4.71.2.44 #define NQ_ERR_BADMEM (NQ_ERR_MODULE | 1009)

Invalid memory block address

4.71.2.45 #define NQ_ERR_BADENV (NQ_ERR_MODULE | 1010)

Invalid environment

4.71.2.46 #define NQ_ERR_BADFORMAT (NQ_ERR_MODULE | 1011)

Invalid format

4.71.2.47 #define NQ_ERR_BADACCESS (NQ_ERR_MODULE | 1012)

Invalid access

4.71.2.48 #define NQ_ERR_BADDATA (NQ_ERR_MODULE | 1013)

Invalid data (generated only by IOCTL calls within the server)

4.71.2.49 #define NQ_ERR_BADDRIVE (NQ_ERR_MODULE | 1015)

Invalid drive specified

4.71.2.50 #define NQ_ERR_REMCD (NQ_ERR_MODULE | 1016)

A Delete Directory request attempted to remove the server's current directory

4.71.2.51 #define NQ_ERR_DIFFDEVICE (NQ_ERR_MODULE | 1017)

Not the same device (e.g. a cross volume rename was attempted)

4.71.2.52 #define NQ_ERR_NOFILES (NQ_ERR_MODULE | 1018)

A File Search command can find no more files matching the specified criteria.

4.71.2.53 #define NQ_ERR_WRONGDIALECT (NQ_ERR_MODULE | 1019)

Dialect revision does not present in Client's active dialects list

4.71.2.54 #define NQ_ERR_BADSHARE (NQ_ERR_MODULE | 1032)

The network path cannot be located.

4.71.2.55 #define NQ_ERR_LOCK (NQ_ERR_MODULE | 1033)

A Lock request conflicted with an existing lock or specified an invalid mode, or an Unlock requested attempted to remove a lock held by another process.

4.71.2.56 #define NQ_ERR_DONTSUPPORTIPC (NQ_ERR_MODULE | 1066)

Server does not support IPC pseudo-filesystem, so that RPC operations are not available.

4.71.2.57 #define NQ_ERR_SHARINGVIOLATION (NQ_ERR_MODULE | 1067)

The sharing mode specified for Open conflicts with existing opens on the file.

4.71.2.58 #define NQ_ERR_FILEEXISTS (NQ_ERR_MODULE | 1080)

The file named in the request already exists.

4.71.2.59 #define NQ_ERR_BADDIRECTORY (NQ_ERR_MODULE | 1087)

The file specified is not a directory while directory expected.

4.71.2.60 #define NQ_ERR_INSUFFICIENTBUFFER (NQ_ERR_MODULE | 1122)

Server compose a response due to constraints applied on the request.

4.71.2.61 #define NQ_ERR_INVALIDNAME (NQ_ERR_MODULE | 1123)

File name or path contains invalid characters.

4.71.2.62 #define NQ_ERR_DIRNOTEMPTY (NQ_ERR_MODULE | 1145)

Directory not empty.

4.71.2.63 #define NQ_ERR_ALREADYEXISTS (NQ_ERR_MODULE | 1183)

Object already exists. Returned on an attempt to create file or directory which already exists on the server.

4.71.2.64 #define NQ_ERR_BADPIPE (NQ_ERR_MODULE | 1230)

RPC request specifies invalid handle or a handle that does correspond to an RPC pipe.

4.71.2.65 #define NQ_ERR_PIPEBUSY (NQ_ERR_MODULE | 1231)

Server cannot currently perform the required RPC operation. Client can try again later.

4.71.2.66 #define NQ_ERR_PIPECLOSING (NQ_ERR_MODULE | 1232)

RPC pipe is being closed.

4.71.2.67 #define NQ_ERR_NOTCONNECTED (NQ_ERR_MODULE | 1233)

Connection to server lost.

4.71.2.68 #define NQ_ERR_MOREDATA (NQ_ERR_MODULE | 1234)

The entire payload does not fit in the response.

4.71.2.69 #define NQ_ERR_INVALIDNAMELEN (NQ_ERR_MODULE | 1235)

Invalid name length.

4.71.2.70 #define NQ_ERR_ERROR (NQ_ERR_MODULE | 2001)

Non-specific error code, returned when none of specific error codes is applicable.

4.71.2.71 #define NQ_ERR_BADPW (NQ_ERR_MODULE | 2002)

Invalid password - name/password pair.

4.71.2.72 #define NQ_ERR_ACCESS (NQ_ERR_MODULE | 2004)

The client does not have the necessary access rights within the specified context for the requested function.

4.71.2.73 #define NQ_ERR_INVTID (NQ_ERR_MODULE | 2005)

The TID specified in a command was invalid.

4.71.2.74 #define NQ_ERR_INVNETNAME (NQ_ERR_MODULE | 2006)

Invalid network name (host name) specified.

4.71.2.75 #define NQ_ERR_INVDEVICE (NQ_ERR_MODULE | 2007)

Invalid device - printer request made to non-printer connection or non-printer request made to printer connection.

4.71.2.76 #define NQ_ERR_QFULL (NQ_ERR_MODULE | 2049)

Print queue full (files) – returned by open print file.

4.71.2.77 #define NQ_ERR_QTOOBIG (NQ_ERR_MODULE | 2050)

Print queue full – no space .

4.71.2.78 #define NQ_ERR_QEOF (NQ_ERR_MODULE | 2051)

End of file.

4.71.2.79 #define NQ_ERR_INVFID (NQ_ERR_MODULE | 2052)

Invalid file handle.

4.71.2.80 #define NQ_ERR_SMBCMD (NQ_ERR_MODULE | 2064)

The server did not recognize the command received.

4.71.2.81 #define NQ_ERR_SRVEERROR (NQ_ERR_MODULE | 2065)

The server encountered an internal error, e.g. system file unavailable.

4.71.2.82 #define NQ_ERR_FILESPCS (NQ_ERR_MODULE | 2067)

The file handle and pathname parameters contained an invalid combination of values.

4.71.2.83 #define NQ_ERR_BADPERMITS (NQ_ERR_MODULE | 2069)

The access permissions specified for a file or directory are not a valid combination. The server cannot set the requested attribute.

4.71.2.84 #define NQ_ERR_SETATTRMODE (NQ_ERR_MODULE | 2071)

The attribute mode in the Set File Attribute request is invalid.

4.71.2.85 #define NQ_ERR_PAUSED (NQ_ERR_MODULE | 2081)

Server paused (reserved for messaging).

4.71.2.86 #define NQ_ERR_MSGOFF (NQ_ERR_MODULE | 2082)

Not receiving messages (reserved for messaging).

4.71.2.87 #define NQ_ERR_NOROOM (NQ_ERR_MODULE | 2083)

No room to buffer message (reserved for messaging).

4.71.2.88 #define NQ_ERR_RMUNS (NQ_ERR_MODULE | 2087)

Too many remote user names (reserved for messaging).

4.71.2.89 #define NQ_ERR_TIMEOUT (NQ_ERR_MODULE | 2088)

Operation timed out.

4.71.2.90 #define NQ_ERR_NORESOURCE (NQ_ERR_MODULE | 2089)

No resources currently available for request.

4.71.2.91 #define NQ_ERR_TOOMANYUIDS (NQ_ERR_MODULE | 2090)

Too many users active on this session.

4.71.2.92 #define NQ_ERR_INVUID (NQ_ERR_MODULE | 2091)

The UID is not known as a valid user identifier on this session.

4.71.2.93 #define NQ_ERR_USEMPX (NQ_ERR_MODULE | 2250)

Temporarily unable to support Raw, use MPX mode.

4.71.2.94 #define NQ_ERR_USESTD (NQ_ERR_MODULE | 2251)

Temporarily unable to support Raw, use standard read/write.

4.71.2.95 **#define NQ_ERR_CONTMPX (NQ_ERR_MODULE | 2252)**

Continue in MPX mode.

4.71.2.96 **#define NQ_ERR_NOSUPPORT (NQ_ERR_MODULE | 2999)**

Function not supported.

4.71.2.97 **#define NQ_ERR_NOWRITE (NQ_ERR_MODULE | 3019)**

Attempt to write on write-protected media.

4.71.2.98 **#define NQ_ERR_BADUNIT (NQ_ERR_MODULE | 3020)**

Unknown unit.

4.71.2.99 **#define NQ_ERR_NOTREADY (NQ_ERR_MODULE | 3021)**

Drive not ready.

4.71.2.100 **#define NQ_ERR_BADCMD (NQ_ERR_MODULE | 3022)**

Unknown command.

4.71.2.101 **#define NQ_ERR_DATA (NQ_ERR_MODULE | 3023)**

Data error (CRC).

4.71.2.102 **#define NQ_ERR_BADREQ (NQ_ERR_MODULE | 3024)**

Bad request structure length.

4.71.2.103 **#define NQ_ERR_SEEK (NQ_ERR_MODULE | 3025)**

Seek error.

4.71.2.104 **#define NQ_ERR_BADMEDIA (NQ_ERR_MODULE | 3026)**

Unknown media type.

4.71.2.105 **#define NQ_ERR_BADSECTOR (NQ_ERR_MODULE | 3027)**

Sector not found.

4.71.2.106 **#define NQ_ERR_NOPAPER (NQ_ERR_MODULE | 3028)**

Printer out of paper.

4.71.2.107 **#define NQ_ERR_WRITE (NQ_ERR_MODULE | 3029)**

Write fault.

4.71.2.108 **#define NQ_ERR_READ (NQ_ERR_MODULE | 3030)**

Read fault.

4.71.2.109 **#define NQ_ERR_GENERAL (NQ_ERR_MODULE | 3031)**

General hardware failure.

4.71.2.110 **#define NQ_ERR_WRONGDISK (NQ_ERR_MODULE | 3034)**

The wrong disk was found in a drive.

4.71.2.111 **#define NQ_ERR_FCBUNAVAIL (NQ_ERR_MODULE | 3035)**

No FCBs are available to process request.

4.71.2.112 **#define NQ_ERR_SHAREBUFEXC (NQ_ERR_MODULE | 3036)**

A sharing buffer has been exceeded.

4.71.2.113 **#define NQ_ERR_DISKFULL (NQ_ERR_MODULE | 3039)**

The disk is full.

4.72 TCP and NetBIOS Layer Error Codes

Macros

- #define NQ_ERR_NBILLEGALSOCKETSLLOT (NQ_ERR_MODULE | 4000)
- #define NQ_ERR_NBNOTNETBIOSNAME (NQ_ERR_MODULE | 4001)
- #define NQ_ERR_NBTIMEOUT (NQ_ERR_MODULE | 4002)
- #define NQ_ERR_NBNEGATIVERESPONSE (NQ_ERR_MODULE | 4003)
- #define NQ_ERR_NBHOSTNAMENOTRESOLVED (NQ_ERR_MODULE | 4004)
- #define NQ_ERR_NBCANCELLISTENFAIL (NQ_ERR_MODULE | 4005)
- #define NQ_ERR_NBSOCKETOVERFLOW (NQ_ERR_MODULE | 4006)
- #define NQ_ERR_NBNOBINDBEFORELISTEN (NQ_ERR_MODULE | 4007)
- #define NQ_ERR_NBILLEGALDATAGRAMSOURCE (NQ_ERR_MODULE | 4008)
- #define NQ_ERR_NBILLEGALDATAGRAMDESTINATION (NQ_ERR_MODULE | 4009)
- #define NQ_ERR_NBINVALIDPARAMETER (NQ_ERR_MODULE | 4010)
- #define NQ_ERR_NBINTERNALERROR (NQ_ERR_MODULE | 4011)
- #define NQ_ERR_NBILLEGALDATAGRAMTYPE (NQ_ERR_MODULE | 4012)
- #define NQ_ERR_NBDDCOMMUNICATIONERROR (NQ_ERR_MODULE | 4013)
- #define NQ_ERR_NBBUFFEROVERFLOW (NQ_ERR_MODULE | 4014)
- #define NQ_ERR_NBRELEASENAMEFAIL (NQ_ERR_MODULE | 4015)
- #define NQ_ERR_NBLISTENFAIL (NQ_ERR_MODULE | 4016)
- #define NQ_ERR_SOCKETCREATE (NQ_ERR_MODULE | 5000)
- #define NQ_ERR_SOCKETBIND (NQ_ERR_MODULE | 5001)
- #define NQ_ERR_SOCKETNAME (NQ_ERR_MODULE | 5002)
- #define NQ_ERR_SOCKETSEND (NQ_ERR_MODULE | 5003)
- #define NQ_ERR_SOCKETSETOPTION (NQ_ERR_MODULE | 5004)
- #define NQ_ERR_SOCKETADDRESS (NQ_ERR_MODULE | 5005)
- #define NQ_ERR_CAPTUREZERO (NQ_ERR_MODULE | 6000)
- #define NQ_ERR_INVALIDUINT32SIZE (NQ_ERR_MODULE | 6001)
- #define NQ_ERR_RECONNECTREQUIRED (NQ_ERR_MODULE | 6002)
- #define NQ_ERR_SIZEERROR (NQ_ERR_MODULE | 6003)
- #define NQ_ERR_SEEKERROR (NQ_ERR_MODULE | 6004)
- #define NQ_ERR_ATTRERROR (NQ_ERR_MODULE | 6005)
- #define NQ_ERR_KERBEROSERROR (NQ_ERR_MODULE | 7000)
- #define NQ_ERR_RAPGENERALERROR (NQ_ERR_MODULE | 8000)
- #define NQ_ERR_RAPACCESSDENIED (NQ_ERR_MODULE | 8001)
- #define NQ_ERR_RAPMOREDATA (NQ_ERR_MODULE | 8002)
- #define NQ_ERR_RAPSERVERNOTSTARTED (NQ_ERR_MODULE | 8003)
- #define NQ_ERR_RAPBADTRANSACTCONFIG (NQ_ERR_MODULE | 8004)
- #define NQ_ERR_RAPSERVICENOTINSTALLED (NQ_ERR_MODULE | 8005)
- #define NQ_ERR_RAPDEVICENOTCONNECTED (NQ_ERR_MODULE | 8006)
- #define NQ_ERR_RAPNOTACTIVEINDOMAIN (NQ_ERR_MODULE | 8007)
- #define NQ_ERR_RAPNOBROWSERVERSFOUND (NQ_ERR_MODULE | 8008)
- #define NQ_ERR_ICONVERROR (NQ_ERR_MODULE | 9000)
- #define NQ_ERR_ICONVOPEN (NQ_ERR_MODULE | 9001)
- #define UD_NQ_MAXPWDLEN 257

4.72.1 Detailed Description

4.72.2 Macro Definition Documentation

4.72.2.1 #define NQ_ERR_NBILLEGALSOCKETSLLOT (NQ_ERR_MODULE | 4000)

Internal error

4.72.2.2 #define NQ_ERR_NBNOTNETBIOSNAME (NQ_ERR_MODULE | 4001)

NetBIOS name has illegal format

4.72.2.3 #define NQ_ERR_NBTIMEOUT (NQ_ERR_MODULE | 4002)

Select timeout on NBT/TCP layer

4.72.2.4 #define NQ_ERR_NBNEGATIVERESPONSE (NQ_ERR_MODULE | 4003)

Negative NetBIOS response received

4.72.2.5 #define NQ_ERR_NBHOSTNAMENOTRESOLVED (NQ_ERR_MODULE | 4004)

NQ failed to resolve host over NetBIOS

4.72.2.6 #define NQ_ERR_NBCANCELLISTENFAIL (NQ_ERR_MODULE | 4005)

Unable to cancel a listen operation

4.72.2.7 #define NQ_ERR_NBSOCKETOVERFLOW (NQ_ERR_MODULE | 4006)

Too many sockets in use

4.72.2.8 #define NQ_ERR_NBNOBINDBEFORELISTEN (NQ_ERR_MODULE | 4007)

Listen attempt on a socket that was not bound yet

4.72.2.9 #define NQ_ERR_NBILLEGALDATAGRAMSOURCE (NQ_ERR_MODULE | 4008)

Source name has illegal format

4.72.2.10 #define NQ_ERR_NBILLEGALDATAGRAMDESTINATION (NQ_ERR_MODULE | 4009)

Destination name has illegal format

4.72.2.11 #define NQ_ERR_NBINVALIDPARAMETER (NQ_ERR_MODULE | 4010)

Invalid parameter

4.72.2.12 #define NQ_ERR_NBINTERNALERROR (NQ_ERR_MODULE | 4011)

Internal error

4.72.2.13 #define NQ_ERR_NBILLEGALDATAGRAMTYPE (NQ_ERR_MODULE | 4012)

Unexpected datagram type

4.72.2.14 #define NQ_ERR_NBDDCOMMUNICATIONERROR (NQ_ERR_MODULE | 4013)

NQ failed to communicate with NetBIOS daemon

4.72.2.15 #define NQ_ERR_NBBUFFEROVERFLOW (NQ_ERR_MODULE | 4014)

Internal buffer size exceeded

4.72.2.16 #define NQ_ERR_NBRELEasenameFAIL (NQ_ERR_MODULE | 4015)

NQ failed to release a registered name

4.72.2.17 #define NQ_ERR_NBLISTENFAIL (NQ_ERR_MODULE | 4016)

NQ failed to listen

4.72.2.18 #define NQ_ERR_SOCKETCREATE (NQ_ERR_MODULE | 5000)

Socket failed to create

4.72.2.19 #define NQ_ERR_SOCKETBIND (NQ_ERR_MODULE | 5001)

Socket failed to bind

4.72.2.20 #define NQ_ERR_SOCKETNAME (NQ_ERR_MODULE | 5002)

Socket failed to get name

4.72.2.21 #define NQ_ERR_SOCKETSEND (NQ_ERR_MODULE | 5003)

Socket failed to send

4.72.2.22 #define NQ_ERR_SOCKETSETOPTION (NQ_ERR_MODULE | 5004)

Socket failed to set options

4.72.2.23 #define NQ_ERR_SOCKETADDRESS (NQ_ERR_MODULE | 5005)

Failed to get socket address

4.72.2.24 #define NQ_ERR_CAPTUREZERO (NQ_ERR_MODULE | 6000)

Error used when packet in internal capture should be zeroed

4.72.2.25 #define NQ_ERR_INVALIDUINT32SIZE (NQ_ERR_MODULE | 6001)

Invalid NQ_UINT32 size

4.72.2.26 #define NQ_ERR_RECONNECTREQUIRED (NQ_ERR_MODULE | 6002)

Error used when send fails and reconnect is required

4.72.2.27 #define NQ_ERR_SIZEERROR (NQ_ERR_MODULE | 6003)

Error requesting file size

4.72.2.28 #define NQ_ERR_SEEKERROR (NQ_ERR_MODULE | 6004)

Invalid seek result

4.72.2.29 #define NQ_ERR_ATTRERROR (NQ_ERR_MODULE | 6005)

Error requesting file attributes

4.72.2.30 #define NQ_ERR_KERBEROSERROR (NQ_ERR_MODULE | 7000)

Kerberos general error

4.72.2.31 #define NQ_ERR_RAPGENERALERROR (NQ_ERR_MODULE | 8000)

RAP general error

4.72.2.32 #define NQ_ERR_RAPACCESSDENIED (NQ_ERR_MODULE | 8001)

RAP user has insufficient privilege

4.72.2.33 #define NQ_ERR_RAPMOREDATA (NQ_ERR_MODULE | 8002)

RAP additional data is available

4.72.2.34 #define NQ_ERR_RAPSERVERNOTSTARTED (NQ_ERR_MODULE | 8003)

RAP the server service on the remote computer is not running

4.72.2.35 #define NQ_ERR_RAPBADTRANSACTCONFIG (NQ_ERR_MODULE | 8004)

RAP the server is not configured for transactions, IPC\$ is not shared

4.72.2.36 #define NQ_ERR_RAPSERVICENOTINSTALLED (NQ_ERR_MODULE | 8005)

RAP the service has not been started

4.72.2.37 #define NQ_ERR_RAPDEVICENOTCONNECTED (NQ_ERR_MODULE | 8006)

RAP the device is not connected

4.72.2.38 #define NQ_ERR_RAPNOTACTIVEINDOMAIN (NQ_ERR_MODULE | 8007)

RAP the computer is not active in this domain

4.72.2.39 #define NQ_ERR_RAPNOBROWSESERVERSFOUND (NQ_ERR_MODULE | 8008)

RAP the list of servers for this workgroup is not currently available

4.72.2.40 #define NQ_ERR_ICONVERROR (NQ_ERR_MODULE | 9000)

iconv failed to convert

4.72.2.41 #define NQ_ERR_ICONVOPEN (NQ_ERR_MODULE | 9001)

iconv_open failed to allocate a conversion descriptor

4.72.2.42 #define UD_NQ_MAXPWLEN 257

password buffer length

4.73 Authentication Statuses

Macros

- #define [NQ_CS_PWDFOUND](#) 0
- #define [NQ_CS_PWDNOAUTH](#) 1
- #define [NQ_CS_PWDNOUSER](#) 2
- #define [NQ_CS_PWDLMHASH](#) 3
- #define [NQ_CS_PWDANY](#) 4

4.73.1 Detailed Description

[udGetPassword\(\)](#) return codes

4.73.2 Macro Definition Documentation

4.73.2.1 #define NQ_CS_PWDFOUND 0

One of the possible values returned by the [udGetPassword\(\)](#) Function. User name was found and the password is returned as both LM and NTLM hash. password was found for the user, the same as 3

4.73.2.2 #define NQ_CS_PWDNOAUTH 1

One of the possible values returned by the [udGetPassword\(\)](#) Function. User authenticated without checking password. authentication is not required

4.73.2.3 #define NQ_CS_PWDNOUSER 2

One of the possible values returned by the [udGetPassword\(\)](#) Function. No such user. no such user

4.73.2.4 #define NQ_CS_PWDLMHASH 3

One of the possible values returned by the [udGetPassword\(\)](#) Function. User name was found and the password is LM hash. password found for the user and it is LM hash

4.73.2.5 #define NQ_CS_PWDANY 4

One of the possible values returned by the [udGetPassword\(\)](#) Function. User name was found and the password is returned as both LM and NTLM hash. password found for the user and its encryption is reported

4.74 Functions

Functions

- [NQ_STATUS](#) [udInit](#) (void)
- void [udStop](#) (void)
- void [udResetServerParams](#) (void)
- void [udCifsServerStarted](#) (void)
- void [udCifsServerClosed](#) (void)
- void [udNetBiosDaemonStarted](#) (void)
- void [udNetBiosDaemonClosed](#) (void)
- void [udGetScopeID](#) ([NQ_WCHAR](#) *buffer)
- void [udGetWins](#) ([NQ_WCHAR](#) *pServers)
- void [udGetDomain](#) ([NQ_WCHAR](#) *buffer, [NQ_BOOL](#) *isWorkgroup)
- void [udGetDnsParams](#) ([NQ_WCHAR](#) *domain, [NQ_WCHAR](#) *servers)
- [NQ_BOOL](#) [udGetCredentials](#) (const void *resource, [NQ_WCHAR](#) *userName, [NQ_WCHAR](#) *password, [NQ_WCHAR](#) *domain)
- [NQ_INT](#) [udGetPassword](#) (const [NQ_WCHAR](#) *userName, [NQ_CHAR](#) *password, [NQ_INT](#) *pwdIsHashed, [NQ_UINT32](#) *userNumber)
- [NQ_UINT32](#) [udGetSmbError](#) ([NQ_UINT32](#) sysErr)
- [NQ_UINT32](#) [udNqToSystemError](#) ([NQ_STATUS](#) nqErr)
- [NQ_INT](#) [udGetTransportPriority](#) ([NQ_UINT](#) transport)
- void [udGetServerComment](#) ([NQ_WCHAR](#) *buffer)
- void [udNetBiosDataIn](#) (void)
- void [udServerDataIn](#) (void)
- void [udServerShareConnect](#) (const [NQ_WCHAR](#) *share)
- void [udServerShareDisconnect](#) (const [NQ_WCHAR](#) *share)
- [NQ_INT](#) [udGetCodePage](#) (void)
- void [udSetCodePage](#) ([NQ_INT](#) codePage)
- [NQ_PORT](#) [udGetPort](#) ([NQ_PORT](#) port)
- void [udGetComputerId](#) ([NQ_BYTE](#) *buf)
- [NQ_COUNT](#) [udGetUserCount](#) (void)
- [NQ_BOOL](#) [udGetUserRidByName](#) (const [NQ_WCHAR](#) *name, [NQ_UINT32](#) *rid)
- [NQ_BOOL](#) [udGetUserNameByRid](#) ([NQ_UINT32](#) rid, [NQ_WCHAR](#) *nameBuffer, [NQ_WCHAR](#) *fullNameBuffer)
- [NQ_BOOL](#) [udGetUserInfo](#) ([NQ_UINT](#) index, [NQ_UINT32](#) *rid, [NQ_WCHAR](#) *name, [NQ_WCHAR](#) *fullName, [NQ_WCHAR](#) *description)
- [NQ_BOOL](#) [udSetUserAsAdministrator](#) ([NQ_UINT32](#) rid, [NQ_BOOL](#) isAdmin)
- [NQ_BOOL](#) [udSetUserInfo](#) ([NQ_UINT32](#) rid, const [NQ_WCHAR](#) *name, const [NQ_WCHAR](#) *fullName, const [NQ_WCHAR](#) *descr, const [NQ_WCHAR](#) *password)
- [NQ_BOOL](#) [udCreateUser](#) (const [NQ_WCHAR](#) *name, const [NQ_WCHAR](#) *fullName, const [NQ_WCHAR](#) *description)
- [NQ_BOOL](#) [udDeleteUserByRid](#) ([NQ_UINT32](#) rid)
- [NQ_BOOL](#) [udSaveShareInformation](#) (const [NQ_WCHAR](#) *name, const [NQ_WCHAR](#) *newName, const [NQ_WCHAR](#) *newMap, const [NQ_WCHAR](#) *newDescription)
- [NQ_BOOL](#) [udRemoveShare](#) (const [NQ_WCHAR](#) *name)
- void [udNetBiosError](#) ([NQ_STATUS](#) cause, const [NQ_CHAR](#) *name)
- [NQ_BOOL](#) [udGetComputerSecret](#) ([NQ_BYTE](#) **secret)
- [NQ_BOOL](#) [udGetComputerSecretByDomain](#) ([NQ_BYTE](#) *secret, const [NQ_WCHAR](#) *domainDNS, [NQ_WCHAR](#) *domainNB)
- void [udSetComputerSecret](#) ([NQ_BYTE](#) *secret)
- void [udSetComputerSecretByDomain](#) (const [NQ_BYTE](#) *secret, const [NQ_WCHAR](#) *domainDNS, const [NQ_WCHAR](#) *domainNB)
- void [udGetFileSystemName](#) (const [NQ_WCHAR](#) *shareName, const [NQ_WCHAR](#) *sharePath, [NQ_WCHAR](#) *fileName)

- void `udEventLog` (`NQ_UINT` module, `NQ_UINT` eventClass, `NQ_UINT` type, const `NQ_WCHAR` *userName, const `NQ_IPADDRESS` *plp, `NQ_UINT32` status, const `NQ_BYTE` *parameters)
- `NQ_BYTE` * `udAllocateBuffer` (`NQ_INT` idx, `NQ_COUNT` numBufs, `NQ_UINT` bufferSize)
- void `udReleaseBuffer` (`NQ_INT` idx, `NQ_COUNT` numBufs, `NQ_BYTE` *buffAddr, `NQ_UINT` bufferSize)
- `NQ_INT` `udGetSecurityDescriptor` (`NQ_INT` file, `NQ_UINT32` information, void *buffer)
- `NQ_STATUS` `udSetSecurityDescriptor` (`NQ_INT` file, `NQ_UINT32` information, const void *buffer, `NQ_UINT32` len)
- `NQ_COUNT` `udLoadShareSecurityDescriptor` (const `NQ_WCHAR` *shareName, `NQ_BYTE` *buffer, `NQ_COUNT` bufferLen)
- void `udSaveShareSecurityDescriptor` (const `NQ_WCHAR` *shareName, const `NQ_BYTE` *sd, `NQ_COUNT` sdLen)
- `NQ_BOOL` `udGetNextShareEx` (`NQ_WCHAR` *name, `NQ_WCHAR` *map, `NQ_BOOL` *isPrinter, `NQ_WCHAR` *description, `NQ_BOOL` *isEncrypted)
- `NQ_BOOL` `udGetNextMount` (`NQ_WCHAR` *name, `NQ_WCHAR` *map)
- `NQ_BOOL` `udGetMessageSigningPolicy` (`NQ_INT` *policy)
- `NQ_BOOL` `udGetGlobalEncryption` (void)
- `NQ_BOOL` `udGetInternalCapture` (void)
- void `udGetCaptureFileBaseFolder` (`NQ_CHAR` *buffer, `NQ_UINT` size)
- `NQ_BOOL` `udGetInternalTrace` (void)
- `NQ_INT` `udGetTaskPriorities` (void)
- void `udGetDriverName` (`NQ_CHAR` *buffer)
- void `udGetHostName` (`NQ_CHAR` *buffer, `NQ_UINT` length)
- void `udGetLogFileBaseFolder` (`NQ_CHAR` *buffer, `NQ_UINT` size)
- void `udGetMaxDiskSize` (`NQ_UINT32` *size)

4.74.1 Detailed Description

4.74.2 Function Documentation

4.74.2.1 `NQ_STATUS` `udInit` (void)

Initialize UD module.

The project level code must call this function on startup before calling `nqStart()`.

Returns

`NQ_SUCCESS` on success or `NQ_FAIL` on error.

4.74.2.2 void `udStop` (void)

Stop UD module.

The project level must call this function after calling `nqStop()`.

4.74.2.3 void `udResetServerParams` (void)

Reset server parameters

This function reset the value of server parameters that was loaded from server configuration file, it should be called during server restart.

4.74.2.4 void udCifsServerStarted (void)

NQ signals to the project level of the server startup complete. Now NQ Server is ready to accept incoming connections.

The default implementation of this function prints a message on console. Integrators may modify it to take measures on the server startup.

4.74.2.5 void udCifsServerClosed (void)

NQ signals to the project level of the server shutdown. Now NQ Server does not accept anymore incoming connections.

The default implementation of this function prints a message on console. Integrators may modify it to take measures on the server shutdown.

4.74.2.6 void udNetBiosDaemonStarted (void)

NQ signals to the project level of NetBIOS Daemon readiness.

The default implementation of this function is empty. Integrators may modify it to take measures on the daemon startup.

4.74.2.7 void udNetBiosDaemonClosed (void)

NQ signals to the project level of NetBIOS Daemon shutdown.

The default implementation of this function is empty. Integrators may modify it to take measures on the daemon shutdown.

4.74.2.8 void udGetScopeID (NQ_WCHAR * buffer)

Obtain the system's Scope ID for NetBIOS. Parameters

Parameters

<i>buffer</i>	Buffer for the result of at least 16 bytes.
---------------	---

4.74.2.9 void udGetWins (NQ_WCHAR * pServers)

Obtain wins addresses information.

This function allows to obtain the default WINS addresses. To add more WINS addresses, use [cmNetBiosSetWins\(\)](#) or [cmNetBiosSetWinsA\(\)](#).

Parameters

<i>pServers</i>	OUT A semicolon-delimited list of WINS servers.
-----------------	---

Note

Returns WINS servers as supplied by User Defined level (for example cm_cfg.txt).

4.74.2.10 void udGetDomain (NQ_WCHAR * buffer, NQ_BOOL * isWorkgroup)

Get the default domain or workgroup name.

NQ calls this function once on startup. The domain name may be modified later by calling the [cmDnsSetDomain\(\)](#) functions.

Parameters

<i>buffer</i>	Buffer for the result of at least 16 bytes.
<i>isWorkgroup,:</i>	<i>TRUE</i> for a workgroup name, <i>FALSE</i> for a domain name.

4.74.2.11 void udGetDnsParams (NQ_WCHAR * domain, NQ_WCHAR * servers)

Get DNS initialization parameters (DNS addresses and domain)

This function is called by NQ on startup. It must bring the default list of DNS servers. More DNS servers may be added later by calling the [cmDnsSetServers\(\)](#) function.

Parameters

<i>domain</i>	The default domain target belongs to.
<i>servers</i>	A semicolon-delimited list of DNS servers.

Note

Returns DNS servers as supplied by User Defined level (for example cm_cfg.txt)

4.74.2.12 NQ_BOOL udGetCredentials (const void * resource, NQ_WCHAR * userName, NQ_WCHAR * password, NQ_WCHAR * domain)

Get authentication parameters for connecting domain controllers.

NQ uses embedded SMB client for domain authentication, joining domain and domain logon. The credentials used during these transactions should be sufficient.

Parameters

<i>resource</i>	URI of the destination to connect to.
<i>userName</i>	Buffer for user name. This buffer is 256 wide characters at least.
<i>password</i>	Buffer for password. This buffer is 256 wide characters at least.
<i>domain</i>	Buffer for domain name. This buffer is 256 wide characters at least.

Returns

TRUE when credentials were obtained, *FALSE* on failure.

4.74.2.13 NQ_INT udGetPassword (const NQ_WCHAR * userName, NQ_CHAR * password, NQ_INT * pwdIsHashed, NQ_UINT32 * userNumber)

Check password for a specific user.

NQ Server calls this function on SMB logon (session setup) to authenticate a local user. This function is called in the following cases:

- No domain Controller was detected and, therefore, domain logon is not available.
- Domain logon failed for any reason. NQ calls this function to obtain the password for the user being authenticated. When this function returns with a password, NQ compares it with the values received from the client over SMB. On match user becomes authenticated with NQ. Since the password received over the wire is hashed, NQ requires hashed passwords.

Plain text passwords are deprecated.

The default implementation of this function reads passwords from a configuration file.

Parameters

<i>userName</i>	User name as a null-terminated string in UTF-16LE.
<i>password</i>	Buffer for user's password. This password must be hashed. Plain text passwords are deprecated.
<i>pwdIsHashed</i>	Buffer for password has flag as: <i>TRUE</i> when password is hashed, <i>FALSE</i> when password is plain text. Plain text passwords are deprecated.
<i>userNumber</i>	Unique user ID (RID).

Returns

One of the following:

- NQ_CS_PWDFOUND : User name was found and the password is returned as both LM and NTLM hash.
- NQ_CS_PWDNOAUTH : User was authenticated without checking password.
- NQ_CS_PWDNOUSER : No such user.
- NQ_CS_PWDLMHASH : User name was found and the password is LM hash.
- NQ_CS_PWDANY : User name was found and the password is returned as both LM and NTLM hash.

4.74.2.14 NQ_UINT32 udGetSmbError (NQ_UINT32 sysErr)

This function reads last system error and attempts to convert it into an SMB error.

This function may implement a customized error code conversion. If this conversion fails (a zero value is returned) NQ will use the default conversion. Its default implementation always returns zero.

Parameters

<i>sysErr</i>	system error code to convert
---------------	------------------------------

Returns

SMB error or zero to use the default conversion.

4.74.2.15 NQ_UINT32 udNqToSystemError (NQ_STATUS nqErr)

This function converts an internal NQ error into the system error code.

NQ calls this function at the very end of a client transaction processing to map internal error on system error.

Parameters

<i>nqErr</i>	NQ internal error code to convert. These error numbers are defined in udapi.h .
--------------	---

Returns

System error or 0 (then the system-dependent conversion will be taken).

4.74.2.16 NQ_INT udGetTransportPriority (NQ_UINT transport)

Get transport priority.

NQ Client uses this function to determine in which sequence it attempts to connect to the server. NQ Client can use three transports: NetBIOS over TCP port 139. TCP over IPv4 port 445. TCP over IPv6 port 445. The order in which NQ tries the above transports, depends on the priorities assigned by the function being described. NQ calls this function on each new connection and it does it once for each transport that was included in *udparams.h*.

Parameters

<i>transport</i>	The transport number as: <ul style="list-style-type: none"> • NS_TRANSPORT_NETBIOS - NetBIOS. • NS_TRANSPORT_IPV4 - IPv4. • NS_TRANSPORT_IPV6 - IPv6.
------------------	--

Returns

A priority assigned to the provided transport as:

- 0 - the transport isn't used
- 1..3 - the bigger number is highest priority If more then one transport has same priority built in order is used: IPv6, IPv4, NetBIOS.

4.74.2.17 void udGetServerComment (NQ_WCHAR * *buffer*)

Get server comment string.

NQ Server uses this string in several information responses.

Parameters

<i>buffer</i>	Buffer for the result of 256 characters at least.
---------------	---

4.74.2.18 void udNetBiosDataIn (void)

Project-level processing on incoming data to NetBios Daemon.

NQ calls this function when NetBIOS daemon receives a message. The default implementation of this function is empty.

4.74.2.19 void udServerDataIn (void)

Project-level processing on incoming data to NQ Server.

NQ calls this function when SMB engine receives a message. The default implementation of this function is empty.

4.74.2.20 void udServerShareConnect (const NQ_WCHAR * *share*)

Project-level processing on client connection to a share.

NQ calls this function when SMB engine receives a tree connect request (Tree Connect). The default implementation of this function is empty.

Parameters

<i>share</i>	Name of the share being connected.
--------------	------------------------------------

4.74.2.21 void udServerShareDisconnect (const NQ_WCHAR * *share*)

Project-level processing on client disconnect from a share.

NQ calls this function when SMB engine receives a tree disconnect request (Tree Disconnect). The default implementation of this function is empty.

Parameters

<i>share</i>	Name of the share being disconnected.
--------------	---------------------------------------

4.74.2.22 NQ_INT udGetCodePage (void)

Get current code page number.

NQ uses code pages when conversion from UTF-16LE to ANSI characters or vice versa happens. This may happen for host and domain names through NetBIOS and DNS.

Returns

Code page number according to IANA.

4.74.2.23 void udSetCodePage (NQ_INT codePage)

Set current code page number.

NQ uses code pages when conversion from UTF-16LE to ANSI characters or vice versa happens. This may happen for host and domain names through NetBIOS and DNS.

Parameters

<i>codePage</i>	Code page number.
-----------------	-------------------

4.74.2.24 NQ_PORT udGetPort (NQ_PORT port)

Redefine port number.

NQ uses the following ports:

- Port 137 - NetBIOS Naming Service listener
- Port 138 - NetBIOS Datagram Service listener
- Port 139 - NetBIOS Session Service listener
- Internal ports used for communication between NQ Server and NetBIOS daemon. For coexistence with other network components NQ allows to redefine any of those ports by means of this function. NQ calls it each time a port is about to use, providing it with a default value. NQ uses the return value as an actual port number. The default implementation simply returns the default value thus no redefinition is performed. For redefining a port, modify this function so that it will return an alternate port number for the default port of the interest. For other ports that should not be alternated this function should continue to return its only argument.

Parameters

<i>port</i>	The default port number in the host byte order.
-------------	---

Returns

Redefined port number in the host byte order. By default this is the same number that is provided on the call.

4.74.2.25 void udGetComputerId (NQ_BYTE * buf)

Get unique ID for the current machine. The returned 12-byte value should be:

- "Statistically unique" for the given machine.
- Persistently the same for each call. Recommended methods are:
- MAC address of the default adapter.
- Product serial number when available.

Parameters

<i>buf</i>	A 12 - byte buffer to be filled with a unique value.
------------	--

4.74.2.26 NQ_COUNT udGetUserCount (void)

Get number of local users.

Returns

Number of local user accounts.

4.74.2.27 NQ_BOOL udGetUserRidByName (const NQ_WCHAR * name, NQ_UINT32 * rid)

Get local user ID by name.

It is developer's responsibility to implement a particular user management mechanism. Persistent user management may keep a list of users in a file. User RID (relative ID) should be a number unique through over the local user database.

Parameters

<i>name</i>	User name as a UTF-16LE string.
<i>rid</i>	Buffer for user's unique ID (RID).

Returns

TRUE when user was found and *FALSE* for a non-existing user.

4.74.2.28 NQ_BOOL udGetUserNameByRid (NQ_UINT32 rid, NQ_WCHAR * nameBuffer, NQ_WCHAR * fullNameBuffer)

Get local user name by ID.

User RID (relative ID) should be a number unique through over the local user database. User name (i.e., - account name) should be also unique in the local account database. Full user name should not be necessarily unique.

Parameters

<i>rid</i>	A unique ID.
<i>nameBuffer</i>	Pointer to a buffer of 256 wide characters for user account name (short name). When the given RID corresponds to a local user the function fills this buffer with the account name.
<i>fullNameBuffer</i>	Pointer to a buffer of 256 wide characters for a full user name. When the given RID corresponds to a local user the function fills this buffer with a user-friendly name to be displayed on the client.

Returns

TRUE when the ID was identified as a local user ID and *FALSE* for a non-existing user.

4.74.2.29 **NQ_BOOL** udGetUserInfo (**NQ_UINT** *index*, **NQ_UINT32** * *rid*, **NQ_WCHAR** * *name*, **NQ_WCHAR** * *fullName*, **NQ_WCHAR** * *description*)

Enumerate local users.

The order of accounts is undefined, but all accounts should have indices from 0 to N where N is the number of users.

Parameters

<i>index</i>	Zero-based index of user accounts in the database.
<i>rid</i>	Buffer for RID for the index's user.
<i>name</i>	Pointer to a buffer of 256 wide characters for user account name (short name). When the given RID corresponds to a local user the function fills this buffer with the account name.
<i>fullName</i>	Pointer to a buffer of 256 wide characters for a full user name. When the given RID corresponds to a local user the function fills this buffer with the name to be displayed on the client.
<i>description</i>	Pointer to a buffer of 256 wide characters for account description. When the given RID corresponds to a local user the function fills this buffer with account description to be displayed on the client.

Returns

TRUE when a user with the given index is available or *FALSE* for the end of enumeration.

4.74.2.30 **NQ_BOOL** udSetUserAsAdministrator (**NQ_UINT32** *rid*, **NQ_BOOL** *isAdmin*)

Set user administrative rights.

NQ Server supports two local user groups: Users and Administrators. Any user is a member of the Users group. This function allows to make user a member in the Administrators group or to cancel user's membership in this group.

Parameters

<i>rid</i>	A unique account ID.
<i>isAdmin</i>	<i>TRUE</i> to set a local user as an administrator, <i>FALSE</i> to remove user from the Administrators group.

Returns

TRUE when operation has succeeded, *FALSE* on error.

4.74.2.31 **NQ_BOOL** udSetUserInfo (**NQ_UINT32** *rid*, const **NQ_WCHAR** * *name*, const **NQ_WCHAR** * *fullName*, const **NQ_WCHAR** * *descr*, const **NQ_WCHAR** * *password*)

Add or modify a local user account.

It is recommended that this function return *FALSE* when there is another user with the same account name. It is recommended to save password in an encrypted form. Use the following code fragment to encrypt a plain text Unicode password into an MD4 hash:

```
cmUnicodeToAnsi(asciiPassword, password);
cmHashPassword(asciiPassword, encryptedPassword);
cmMD4(encryptedPassword+16, (NQ_BYTE*)password);
cmWStrlen(password)*sizeof(NQ_WCHAR) );
```

Parameters

<i>rid</i>	A unique account ID.
<i>name</i>	New account name of up to 256 wide characters.
<i>fullName</i>	New full (user-friendly) user name of up to 256 wide characters.
<i>descr</i>	New account description of up to 256 wide characters.
<i>password</i>	New password. The password is plain text UTF-16LE.

Returns

TRUE when operation has succeeded, *FALSE* on error.

4.74.2.32 NQ_BOOL udCreateUser (const NQ_WCHAR * name, const NQ_WCHAR * fullName, const NQ_WCHAR * description)

Add a local user.

It is recommended that this function return *FALSE* when there is another user with the same account name. A new user may be created with an empty or a random password. An expected immediate call to [udSetUserInfo\(\)](#) (see above) will set new password.

Parameters

<i>name</i>	New account name of up to 256 wide characters.
<i>fullName</i>	New full (user-friendly) user name of up to 256 wide characters.
<i>description</i>	New account description of up to 256 wide characters.

Returns

TRUE when operation has succeeded, *FALSE* on error.

4.74.2.33 NQ_BOOL udDeleteUserByRid (NQ_UINT32 rid)

Description Remove a local user.

Parameters

<i>rid</i>	A unique account ID.
------------	----------------------

Returns

TRUE when operation has succeeded, *FALSE* on error.

4.74.2.34 NQ_BOOL udSaveShareInformation (const NQ_WCHAR * name, const NQ_WCHAR * newName, const NQ_WCHAR * newMap, const NQ_WCHAR * newDescription)

modify/create share information in a persistent store

Parameters

<i>name</i>	share to modify or NULL for a new share
<i>newName</i>	new share name

<i>newMap</i>	new share path
<i>newDescription</i>	new share description

Returns

TRUE on success, FALSE on failure

Note

user-level should return TRUE in the following cases:

- new share was persistently stored
- existing share was persistently modified
- new share was not persistently stored but it should be exposed until server shutdown
- share was not persistently modified but this modification should be exposed until server shutdown user-level should return FALSE when a new share should not be created or an existing share should not be modified

4.74.2.35 NQ_BOOL udRemoveShare (const NQ_WCHAR * name)

remove share from the persistent store

Parameters

<i>name</i>	share to remove
-------------	-----------------

Returns

TRUE on success, FALSE on failure

4.74.2.36 void udNetBiosError (NQ_STATUS cause, const NQ_CHAR * name)

Report to the project level about a NetBIOS error. Parameters

Parameters

<i>cause</i>	Error code as: <ul style="list-style-type: none"> • CM_NBERR_NOTNETBIOSNAME • CM_NBERR_TIMEOUT • CM_NBERR_NEGATIVERESPONSE • CM_NBERR_HOSTNAMENOTRESOLVED • CM_NBERR_CANCELLISTENFAIL • CN_NBERR_SOCKETOVERFLOW • CM_NBERR_NOBINDBEFORELISTEN • CM_NBERR_ILLEGALDATAGRAMSOURCE • CM_NBERR_ILLEGALDATAGRAMDESTINATION • CM_NBERR_INVALIDPARAMETER • CM_NBERR_INTERNALERROR • CM_NBERR_ILLEGALDATAGRAMTYPE • CM_NBERR_DDCOMMUNICATIONERROR • CM_NBERR_BUFFEROVERFLOW • CM_NBERR_RELEASENAMEFAIL Right now only CM_NBERR_NEGATIVERESPONSE is used while others are reserved for a future use.
<i>name</i>	The NetBIOS name that caused the error.

4.74.2.37 NQ_BOOL udGetComputerSecret (NQ_BYTE ** *secret*)

Read computer secret.

Computer secret is a binary array of 16 bytes that NQ receives from Domain Controller after joining a domain. NQ uses this array for subsequent logon into the domain.

Parameters

<i>secret</i>	Buffer for a secret pointer. If a value inside that buffer is non-NULL, this function should fill this buffer with a pointer to computer secret. If it is NULL, the function should check for computer secret without setting a pointer to it.
---------------	--

Returns

TRUE if the computer secret is available. *FALSE* means that the computer secret is not available for one of the following reasons:

- NQ did not join the domain yet.
- The computer secret was not properly saved.
- The computer secret is not accessible.

Note

NQ assumes that computer secret is persistent and was stored in a call to the [udSetComputerSecret\(\)](#) Function.

4.74.2.38 `NQ_BOOL udGetComputerSecretByDomain (NQ_BYTE * secret, const NQ_WCHAR * domainDNS, NQ_WCHAR * domainNB)`

Read computer secret by domain.

Computer secret is a binary array of 16 bytes that NQ receives from Domain Controller after joining a domain. NQ uses this array for subsequent logon into the domain. Parameters

Parameters

<i>secret</i>	Buffer for a secret pointer. If a value inside that buffer is non-NULL, this function should fill this buffer with a pointer to computer secret.
<i>domainDNS</i>	Domain name DNS
<i>domainNB</i>	Domain name NetBIOS. If it is NULL, the function checks for computer secret which refers only to domainDNS.

Returns

TRUE if the computer secret is available and (if the *secret* parameter was not NULL) was copied into the buffer. *FALSE* means that the computer secret is not available for one of the following reasons:

- NQ did not join the domain yet.
- The computer secret was not properly saved.
- The computer secret is not accessible. Buffer domainNB is populated with domain NetBIOS name.

Note

NQ assumes that computer secret is persistent and was stored in a call to the [udSetComputerSecretByDomain\(\)](#) Function.

4.74.2.39 `void udSetComputerSecret (NQ_BYTE * secret)`

Save computer secret.

Computer secret is a binary array of 16 bytes that NQ receives from Domain Controller after joining a domain. NQ uses this array for subsequent logon into the domain.

Parameters

<i>secret</i>	Computer secret pointer.
---------------	--------------------------

Note

NQ assumes that computer secret is persistent and after this call it will be available through a call to the `udGetComputerSecret` Function().

4.74.2.40 `void udSetComputerSecretByDomain (const NQ_BYTE * secret, const NQ_WCHAR * domainDNS, const NQ_WCHAR * domainNB)`

Save computer secret per domain.

Computer secret is a binary array of 16 bytes that NQ receives from Domain Controller after joining a domain. NQ uses this array for subsequent logon into the domain.

Parameters

<i>secret</i>	Computer secret pointer.
<i>domainDNS</i>	Domain name DNS
<i>domainNB</i>	Domain name NetBIOS. If it is NULL, the function saves computer secret which refers only to domainDNS.

Note

NQ assumes that computer secret is persistent and after this call it will be available through a call to the [udGetComputerSecretByDomain\(\)](#) Function.

4.74.2.41 void udGetFileSystemName (const NQ_WCHAR * *shareName*, const NQ_WCHAR * *sharePath*, NQ_WCHAR * *fileSystemName*)

Determine the file system name for the given share.

This function uses either share name or share path to determine the type of the target file system. Then it copies the name of this file system into the provided buffer. The name should not be longer than 100 characters.

Parameters

<i>shareName</i>	Pointer to the share name.
<i>sharePath</i>	Pointer to the share path.
<i>fileSystemName</i>	Pointer to buffer for the file system name.

4.74.2.42 void udEventLog (NQ_UINT *module*, NQ_UINT *eventClass*, NQ_UINT *type*, const NQ_WCHAR * *userName*, const NQ_IPADDRESS * *plp*, NQ_UINT32 *status*, const NQ_BYTE * *parameters*)

Event log function

Parameters

<i>module</i>	NQ module that originated this event
<i>eventClass</i>	event class
<i>type</i>	event type
<i>userName</i>	name of the user
<i>plp</i>	next side IP address
<i>status</i>	zero if the operation has succeeded or error code on failure for server event this code is the same that will be transmitted to the client for an NQ CIFS client event this value is the same that will be installed as system error
<i>parameters</i>	pointer to a structure that is filled with event data actual structure depends on event type

4.74.2.43 NQ_BYTE* udAllocateBuffer (NQ_INT *idx*, NQ_COUNT *numBufs*, NQ_UINT *bufferSize*)

allocate buffer in the user space

Parameters

<i>idx</i>	buffer index zero based
<i>numBufs</i>	total number of buffers to be allocated
<i>bufferSize</i>	buffer size in bytes

Returns

pointer to the buffer

4.74.2.44 void udReleaseBuffer (NQ_INT *idx*, NQ_COUNT *numBufs*, NQ_BYTE * *buffAddr*, NQ_UINT *bufferSize*)

release buffer in the user space

Parameters

<i>idx</i>	buffer index zero based
<i>numBufs</i>	total number of buffers to be released
<i>buffAddr</i>	buffer address
<i>bufferSize</i>	buffer size in bytes

4.74.2.45 NQ_INT udGetSecurityDescriptor (NQ_INT file, NQ_UINT32 information, void * buffer)

query user-defined security descriptor

Parameters

<i>file</i>	ID of an opened file
<i>information</i>	descriptor to get
<i>buffer</i>	output buffer

Returns

descriptor length

4.74.2.46 NQ_STATUS udSetSecurityDescriptor (NQ_INT file, NQ_UINT32 information, const void * buffer, NQ_UINT32 len)

write user-defined security descriptor

Parameters

<i>file</i>	ID of an opened file
<i>information</i>	descriptor to set
<i>buffer</i>	input buffer
<i>len</i>	descriptor length

Returns

NQ_SUCCESS or error code

4.74.2.47 NQ_COUNT udLoadShareSecurityDescriptor (const NQ_WCHAR * shareName, NQ_BYTE * buffer, NQ_COUNT bufferLen)

Get persistent security descriptor for share

Parameters

<i>shareName</i>	share name
<i>buffer</i>	buffer to read SD in
<i>bufferLen</i>	buffer length

Returns

SD length or zero on error

4.74.2.48 void udSaveShareSecurityDescriptor (const NQ_WCHAR * shareName, const NQ_BYTE * sd, NQ_COUNT sdLen)

Save persistent security descriptor for share

Parameters

<i>shareName</i>	share name
<i>sd</i>	pointer to SD
<i>sdLen</i>	SD length

4.74.2.49 NQ_BOOL udGetNextShareEx (NQ_WCHAR * name, NQ_WCHAR * map, NQ_BOOL * isPrinter, NQ_WCHAR * description, NQ_BOOL * isEncrypted)

Get next share in the list of shares for CS (extended).

Parameters

<i>name</i>	Pointer to the share name buffer.
<i>map</i>	Pointer to the share map buffer.
<i>isPrinter</i>	0 for file system, 1 for print queue.
<i>description</i>	buffer for the share description
<i>isEncrypted</i>	0 for regular share, 1 for encrypted share (SMB3)

Returns

TRUE - a share read *FALSE* - no more shares

4.74.2.50 NQ_BOOL udGetNextMount (NQ_WCHAR * name, NQ_WCHAR * map)

Get next mount in the list of mounted volumes for Client.

Parameters

<i>name</i>	Pointer to the share name buffer.
<i>map</i>	Pointer to the share map buffer.

Returns

TRUE - more volumes in the list *FALSE* - no more volumes

4.74.2.51 NQ_BOOL udGetMessageSigningPolicy (NQ_INT * policy)

Get message signing policy CS. Values: 0 - disabled, 1 - enabled, 2 - required.

Parameters

<i>policy</i>	Pointer to policy.
---------------	--------------------

Returns

TRUE when implemented *FALSE* not implemented (use CS default)

4.74.2.52 NQ_BOOL udGetGlobalEncryption (void)

Get server global encryption.

Returns

TRUE global encryption on *FALSE* global encryption off

4.74.2.53 NQ_BOOL udGetInternalCapture (void)

Determine whether to use internal capture.

Returns

TRUE or *FALSE*

4.74.2.54 void udGetCaptureFileBaseFolder (NQ_CHAR * buffer, NQ_UINT size)

Get base folder path of internal capture file

Parameters

<i>buffer</i>	buffer for the path
<i>size</i>	the size of that buffer

4.74.2.55 NQ_BOOL udGetInternalTrace (void)

Determine whether to use internal trace.

Returns

TRUE or *FALSE*

4.74.2.56 NQ_INT udGetTaskPriorities (void)

Get task priorities.

Returns

Task priority value.

4.74.2.57 void udGetDriverName (NQ_CHAR * buffer)

Get CIFS driver name.

Parameters

<i>buffer</i>	Buffer for name
---------------	-----------------

4.74.2.58 void udGetHostName (NQ_CHAR * buffer, NQ_UINT length)

Get host name.

Parameters

<i>buffer</i>	Buffer for name
<i>length</i>	Length to be returned (not null terminated)

4.74.2.59 void udGetLogFileBaseFolder (NQ_CHAR * buffer, NQ_UINT size)

Get base folder path of log file

Parameters

<i>buffer</i>	Buffer for the path
<i>size</i>	The size of that buffer

4.74.2.60 void udGetMaxDiskSize (NQ_UINT32 * *size*)

Get disk size limit for shares

Parameters

<i>size</i>	Pointer to size limit
-------------	-----------------------

4.75 Event Log Constants

Macros

- #define UD_LOG_MODULE_CS 1
- #define UD_LOG_MODULE_CC 2
- #define UD_LOG_CLASS_GEN 1
- #define UD_LOG_CLASS_FILE 2
- #define UD_LOG_CLASS_SHARE 3
- #define UD_LOG_CLASS_USER 4
- #define UD_LOG_CLASS_CONNECTION 5
- #define UD_LOG_GEN_START 1
- #define UD_LOG_GEN_STOP 2
- #define UD_LOG_GEN_NAMECONFLICT 3
- #define UD_LOG_GEN_NAMEREGFAIL 4
- #define UD_LOG_FILE_CREATE 1
- #define UD_LOG_FILE_OPEN 2
- #define UD_LOG_FILE_CLOSE 3
- #define UD_LOG_FILE_DELETE 4
- #define UD_LOG_FILE_RENAME 5
- #define UD_LOG_FILE_ATTRIBGET 6
- #define UD_LOG_FILE_ATTRIBSET 7
- #define UD_LOG_FILE_SIZESET 8
- #define UD_LOG_FILE_VOLUMEINFO 9
- #define UD_LOG_FILE_QUERYDIRECTORY 10
- #define UD_LOG_FILE_SEEK 11
- #define UD_LOG_FILE_LOCK 12
- #define UD_LOG_FILE_UNLOCK 13
- #define UD_LOG_SHARE_CONNECT 1
- #define UD_LOG_SHARE_DISCONNECT 2
- #define UD_LOG_USER_LOGON 1
- #define UD_LOG_USER_LOGOFF 2
- #define UD_LOG_CONNECTION_CONNECT 1
- #define UD_LOG_CONNECTION_DISCONNECT 2

4.75.1 Detailed Description

4.75.2 Macro Definition Documentation

4.75.2.1 #define UD_LOG_MODULE_CS 1

Designates NQ CIFS Server as the event origin

4.75.2.2 #define UD_LOG_MODULE_CC 2

Designates NQ CIFS Client as the event origin

4.75.2.3 #define UD_LOG_CLASS_GEN 1

Event class for start and stop events

4.75.2.4 #define UD_LOG_CLASS_FILE 2

Event class for file/directory access events

4.75.2.5 #define UD_LOG_CLASS_SHARE 3

Event class for share connect/disconnect events

4.75.2.6 #define UD_LOG_CLASS_USER 4

Event class for user

4.75.2.7 #define UD_LOG_CLASS_CONNECTION 5

Event class for connections

4.75.2.8 #define UD_LOG_GEN_START 1

Start event type

4.75.2.9 #define UD_LOG_GEN_STOP 2

Stop event type

4.75.2.10 #define UD_LOG_GEN_NAMECONFLICT 3

Name Conflict has occurred

4.75.2.11 #define UD_LOG_GEN_NAMEREGFAIL 4

Name Registration Failed

4.75.2.12 #define UD_LOG_FILE_CREATE 1

File/directory create event type

4.75.2.13 #define UD_LOG_FILE_OPEN 2

File/directory open event type

4.75.2.14 #define UD_LOG_FILE_CLOSE 3

File/directory close event type

4.75.2.15 #define UD_LOG_FILE_DELETE 4

File/directory delete event type

4.75.2.16 #define UD_LOG_FILE_RENAME 5

File/directory rename event type

4.75.2.17 #define UD_LOG_FILE_ATTRIBGET 6

File/directory query attributes event type

4.75.2.18 #define UD_LOG_FILE_ATTRIBSET 7

File/directory change attributes event type

4.75.2.19 #define UD_LOG_FILE_SIZESET 8

File truncate/extend event type

4.75.2.20 #define UD_LOG_FILE_VOLUMEINFO 9

Volume information query event type

4.75.2.21 #define UD_LOG_FILE_QUERYDIRECTORY 10

folder enumeration event type

4.75.2.22 #define UD_LOG_FILE_SEEK 11

File position change event type

4.75.2.23 #define UD_LOG_FILE_LOCK 12

File range lock event type

4.75.2.24 #define UD_LOG_FILE_UNLOCK 13

File range unlock event type

4.75.2.25 #define UD_LOG_SHARE_CONNECT 1

Share connection event type

4.75.2.26 #define UD_LOG_SHARE_DISCONNECT 2

Share disconnection event type

4.75.2.27 #define UD_LOG_USER_LOGON 1

User connection event type

4.75.2.28 #define UD_LOG_USER_LOGOFF 2

User disconnection event type

4.75.2.29 #define UD_LOG_CONNECTION_CONNECT 1

New connection connected [SYN]

4.75.2.30 #define UD_LOG_CONNECTION_DISCONNECT 2

disconnected connection [FIN]

Chapter 5

Data Structure Documentation

5.1 AMCredentials Struct Reference

```
#include <amapi.h>
```

Data Fields

- [AMDomain](#) domain
- [NQ_WCHAR](#) user [CM_BUFFERLENGTH([NQ_WCHAR](#), 257)]
- [NQ_WCHAR](#) password [CM_BUFFERLENGTH([NQ_WCHAR](#), 257)]
- [NQ_UINT](#) type

5.1.1 Detailed Description

Account credentials

5.1.2 Field Documentation

5.1.2.1 AMDomain AMCredentials::domain

Domain name

5.1.2.2 NQ_WCHAR AMCredentials::user[CM_BUFFERLENGTH(NQ_WCHAR, 257)]

User name

5.1.2.3 NQ_WCHAR AMCredentials::password[CM_BUFFERLENGTH(NQ_WCHAR, 257)]

Password buffer

5.1.2.4 NQ_UINT AMCredentials::type

Password type, see [Security Password Type](#)

The documentation for this struct was generated from the following file:

- [amapi.h](#)

5.2 AMCredentialsA Struct Reference

```
#include <amapi.h>
```

Data Fields

- [AMDDomainA domain](#)
- [NQ_CHAR user](#) [CM_BUFFERLENGTH([NQ_CHAR](#), 257)]
- [NQ_CHAR password](#) [CM_BUFFERLENGTH([NQ_CHAR](#), 257)]
- [NQ_UINT type](#)

5.2.1 Detailed Description

This structure carries user credentials as used for authentication against server in the following cases:

- SMB/SMB2 authentication
- Domain join
- Domain logon

5.2.2 Field Documentation

5.2.2.1 AMDomainA AMCredentialsA::domain

Domain name

5.2.2.2 NQ_CHAR AMCredentialsA::user[CM_BUFFERLENGTH(NQ_CHAR, 257)]

User name

5.2.2.3 NQ_CHAR AMCredentialsA::password[CM_BUFFERLENGTH(NQ_CHAR, 257)]

Password buffer

5.2.2.4 NQ_UINT AMCredentialsA::type

Password type, see [Security Password Type](#)

The documentation for this struct was generated from the following file:

- [amapi.h](#)

5.3 AMDomain Struct Reference

```
#include <amapi.h>
```

Data Fields

- [NQ_WCHAR name](#) [CM_BUFFERLENGTH([NQ_WCHAR](#), 256)]
- [NQ_WCHAR realm](#) [CM_BUFFERLENGTH([NQ_WCHAR](#), 256)]

5.3.1 Field Documentation

5.3.1.1 NQ_WCHAR AMDomain::name[CM_BUFFERLENGTH(NQ_WCHAR, 256)]

Domain name

5.3.1.2 NQ_WCHAR AMDomain::realm[CM_BUFFERLENGTH(NQ_WCHAR, 256)]

Domain realm name

The documentation for this struct was generated from the following file:

- [amapi.h](#)

5.4 AMDomainA Struct Reference

```
#include <amapi.h>
```

Data Fields

- [NQ_CHAR name](#) [CM_BUFFERLENGTH(NQ_CHAR, 256)]
- [NQ_CHAR realm](#) [CM_BUFFERLENGTH(NQ_CHAR, 256)]

5.4.1 Detailed Description

This structure is used for domain description. Two fields of this structure compose together a fully qualified domain name.

5.4.2 Field Documentation

5.4.2.1 NQ_CHAR AMDomainA::name[CM_BUFFERLENGTH(NQ_CHAR, 256)]

Domain name

5.4.2.2 NQ_CHAR AMDomainA::realm[CM_BUFFERLENGTH(NQ_CHAR, 256)]

Domain realm name

The documentation for this struct was generated from the following file:

- [amapi.h](#)

5.5 CCFileInfo Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_UINT64 creationTime](#)
- [NQ_UINT64 lastAccessTime](#)

- [NQ_UINT64 lastWriteTime](#)
- [NQ_UINT64 changeTime](#)
- [NQ_UINT64 endOfFile](#)
- [NQ_UINT64 allocationSize](#)
- [NQ_UINT32 attributes](#)
- [NQ_UINT32 numberOfLinks](#)
- [NQ_UINT64 fileIndex](#)
- [NQ_UINT64 volumId](#)

5.5.1 Detailed Description

This structure describes file information. It combines all possible file information fields that may be used in information calls. The file information class equivalent is *FileAllInformation*.

5.5.2 Field Documentation

5.5.2.1 NQ_UINT64 CCFileInfo::creationTime

File creation time.

5.5.2.2 NQ_UINT64 CCFileInfo::lastAccessTime

File last access time.

5.5.2.3 NQ_UINT64 CCFileInfo::lastWriteTime

File last write time.

5.5.2.4 NQ_UINT64 CCFileInfo::changeTime

File change time.

5.5.2.5 NQ_UINT64 CCFileInfo::endOfFile

File size.

5.5.2.6 NQ_UINT64 CCFileInfo::allocationSize

File allocation size.

5.5.2.7 NQ_UINT32 CCFileInfo::attributes

File attributes.

5.5.2.8 NQ_UINT32 CCFileInfo::numberOfLinks

Number of hard links to this file.

5.5.2.9 NQ_UINT64 CCFileInfo::fileIndex

File index.

5.5.2.10 NQ_UINT64 CCFileInfo::volumeld

Volume Id within which the file is opened.

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.6 CCFileInfo_t Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_UINT32 creationTimeLow](#)
- [NQ_UINT32 creationTimeHigh](#)
- [NQ_UINT32 lastAccessTimeLow](#)
- [NQ_UINT32 lastAccessTimeHigh](#)
- [NQ_UINT32 lastWriteTimeLow](#)
- [NQ_UINT32 lastWriteTimeHigh](#)
- [NQ_UINT32 attributes](#)
- [NQ_UINT32 volumeSerialNumber](#)
- [NQ_UINT32 allocationSizeLow](#)
- [NQ_UINT32 allocationSizeHigh](#)
- [NQ_UINT32 fileSizeLow](#)
- [NQ_UINT32 fileSizeHigh](#)
- [NQ_UINT32 numberOfLinks](#)
- [NQ_UINT32 fileIndexLow](#)
- [NQ_UINT32 fileIndexHigh](#)

5.6.1 Detailed Description

This structure is used when calling File Information related CIFS Client API calls, it is used by [ccGetFileInformation-ByNameA\(\)](#) for backward compatibility.

5.6.2 Field Documentation

5.6.2.1 NQ_UINT32 CCFileInfo_t::creationTimeLow

low 32 bits of file creation time

5.6.2.2 NQ_UINT32 CCFileInfo_t::creationTimeHigh

high 32 bits of file creation time

5.6.2.3 NQ_UINT32 CCFileInfo_t::lastAccessTimeLow

low 32 bits of file last access time

5.6.2.4 NQ_UINT32 CCFileInfo_t::lastAccessTimeHigh

high 32 bits of file last access time

5.6.2.5 NQ_UINT32 CCFileInfo_t::lastWriteTimeLow

low 32 bits of file last write time

5.6.2.6 NQ_UINT32 CCFileInfo_t::lastWriteTimeHigh

high 32 bits of file last write time

5.6.2.7 NQ_UINT32 CCFileInfo_t::attributes

file attributes as defined in [File attributes](#) module

5.6.2.8 NQ_UINT32 CCFileInfo_t::volumeSerialNumber

volume serial number

5.6.2.9 NQ_UINT32 CCFileInfo_t::allocationSizeLow

low 32 bits of file allocation size

5.6.2.10 NQ_UINT32 CCFileInfo_t::allocationSizeHigh

high 32 bits of file allocation size

5.6.2.11 NQ_UINT32 CCFileInfo_t::fileSizeLow

low 32 bits of file data size

5.6.2.12 NQ_UINT32 CCFileInfo_t::fileSizeHigh

high 32 bits of file data size

5.6.2.13 NQ_UINT32 CCFileInfo_t::numberOfLinks

number of hard links to this file

5.6.2.14 NQ_UINT32 CCFileInfo_t::fileIndexLow

low 32 bits of file ID (always zero)

5.6.2.15 NQ_UINT32 CCFileInfo_t::fileIndexHigh

high 32 bits of file ID (always zero)

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.7 CCFileTime Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_UINT32 timeLow](#)
- [NQ_UINT32 timeHigh](#)

5.7.1 Detailed Description

This structure is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC). The application can use the NQ system abstraction layer calls for converting time from OS native format to [CCFileTime](#).

5.7.2 Field Documentation

5.7.2.1 NQ_UINT32 CCFileTime::timeLow

high 32 bits of file time

5.7.2.2 NQ_UINT32 CCFileTime::timeHigh

low 32 bits of file time

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.8 CCFindFileData Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_UINT32 fileAttributes](#)
- [NQ_UINT32 creationTimeLow](#)
- [NQ_UINT32 creationTimeHigh](#)
- [NQ_UINT32 lastAccessTimeLow](#)
- [NQ_UINT32 lastAccessTimeHigh](#)
- [NQ_UINT32 lastWriteTimeLow](#)
- [NQ_UINT32 lastWriteTimeHigh](#)
- [NQ_UINT32 changeTimeLow](#)
- [NQ_UINT32 changeTimeHigh](#)
- [NQ_UINT32 fileSizeLow](#)
- [NQ_UINT32 fileSizeHigh](#)
- [NQ_UINT32 allocationSizeLow](#)
- [NQ_UINT32 allocationSizeHigh](#)
- [NQ_UINT32 fileNameLength](#)
- [NQ_WCHAR fileName](#) [CM_BUFFERLENGTH([NQ_WCHAR](#), UD_FS_FILENAMECOMPONENTLEN)]

5.8.1 Detailed Description

5.8.2 Field Documentation

5.8.2.1 NQ_UINT32 CCFindFileData::fileAttributes

file attributes as defined in [File attributes](#) module

5.8.2.2 NQ_UINT32 CCFindFileData::creationTimeLow

low 32 bits of file creation time

5.8.2.3 NQ_UINT32 CCFindFileData::creationTimeHigh

high 32 bits of file creation time

5.8.2.4 NQ_UINT32 CCFindFileData::lastAccessTimeLow

low 32 bits of file last access time

5.8.2.5 NQ_UINT32 CCFindFileData::lastAccessTimeHigh

high 32 bits of file last access time

5.8.2.6 NQ_UINT32 CCFindFileData::lastWriteTimeLow

low 32 bits of file last write time

5.8.2.7 NQ_UINT32 CCFindFileData::lastWriteTimeHigh

high 32 bits of file last write time

5.8.2.8 NQ_UINT32 CCFindFileData::changeTimeLow

low 32 bits of file change time

5.8.2.9 NQ_UINT32 CCFindFileData::changeTimeHigh

high 32 bits of file change time

5.8.2.10 NQ_UINT32 CCFindFileData::fileSizeLow

low 32 bits of file data size

5.8.2.11 NQ_UINT32 CCFindFileData::fileSizeHigh

high 32 bits of file data size

5.8.2.12 NQ_UINT32 CCFindFileData::allocationSizeLow

low 32 bits of file allocation size

5.8.2.13 NQ_UINT32 CCFindFileData::allocationSizeHigh

high 32 bits of file allocation size

5.8.2.14 NQ_UINT32 CCFindFileData::fileNameLength

file name length in characters

5.8.2.15 NQ_WCHAR CCFindFileData::fileName[CM_BUFFERLENGTH(NQ_WCHAR, UD_FS_FILENAMECOMPONENTLEN)]

file name buffer

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.9 CCFindFileDataA Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_UINT32 fileAttributes](#)
- [NQ_UINT32 creationTimeLow](#)
- [NQ_UINT32 creationTimeHigh](#)
- [NQ_UINT32 lastAccessTimeLow](#)
- [NQ_UINT32 lastAccessTimeHigh](#)
- [NQ_UINT32 lastWriteTimeLow](#)
- [NQ_UINT32 lastWriteTimeHigh](#)
- [NQ_UINT32 changeTimeLow](#)
- [NQ_UINT32 changeTimeHigh](#)
- [NQ_UINT32 fileSizeLow](#)
- [NQ_UINT32 fileSizeHigh](#)
- [NQ_UINT32 allocationSizeLow](#)
- [NQ_UINT32 allocationSizeHigh](#)
- [NQ_UINT32 fileNameLength](#)
- [NQ_CHAR fileName](#) [CM_BUFFERLENGTH([NQ_CHAR](#), UD_FS_FILENAMECOMPONENTLEN)]

5.9.1 Detailed Description

This structure is used when calling [ccFindFirstFile\(\)](#) and [ccFindNextFile\(\)](#) calls.

5.9.2 Field Documentation

5.9.2.1 NQ_UINT32 CCFindFileDataA::fileAttributes

file attributes as defined in [File attributes](#) module

5.9.2.2 NQ_UINT32 CCFindFileDataA::creationTimeLow

low 32 bits of file creation time

5.9.2.3 NQ_UINT32 CCFindFileDataA::creationTimeHigh

high 32 bits of file creation time

5.9.2.4 NQ_UINT32 CCFindFileDataA::lastAccessTimeLow

low 32 bits of file last access time

5.9.2.5 NQ_UINT32 CCFindFileDataA::lastAccessTimeHigh

high 32 bits of file last access time

5.9.2.6 NQ_UINT32 CCFindFileDataA::lastWriteTimeLow

low 32 bits of file last write time

5.9.2.7 NQ_UINT32 CCFindFileDataA::lastWriteTimeHigh

high 32 bits of file last write time

5.9.2.8 NQ_UINT32 CCFindFileDataA::changeTimeLow

low 32 bits of file change time

5.9.2.9 NQ_UINT32 CCFindFileDataA::changeTimeHigh

high 32 bits of file change time

5.9.2.10 NQ_UINT32 CCFindFileDataA::fileSizeLow

low 32 bits of file data size

5.9.2.11 NQ_UINT32 CCFindFileDataA::fileSizeHigh

high 32 bits of file data size

5.9.2.12 NQ_UINT32 CCFindFileDataA::allocationSizeLow

low 32 bits of file allocation size

5.9.2.13 NQ_UINT32 CCFindFileDataA::allocationSizeHigh

high 32 bits of file allocation size

5.9.2.14 NQ_UINT32 CCFindFileDataA::fileNameLength

file name length in characters

5.9.2.15 NQ_CHAR CCFindFileDataA::fileName[CM_BUFFERLENGTH(NQ_CHAR, UD_FS_FILENAMECOMPONENTLEN)]

file name buffer

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.10 CCMountInfo Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_TIME](#) creationTime
- [NQ_INT](#) userLogonType
- [NQ_WCHAR *](#) serverName
- [CCServerInfo](#) serverInfo
- [CCStatistics](#) statistics

5.10.1 Detailed Description

This structure describes mount information.

5.10.2 Field Documentation

5.10.2.1 NQ_TIME CCMountInfo::creationTime

mount point creation time.

5.10.2.2 NQ_INT CCMountInfo::userLogonType

user logon type, see: [CC_LOGON_TYPE_REGULAR](#), [CC_LOGON_TYPE_ANONYMOUS](#), [CC_LOGON_TYPE_GUEST](#)

5.10.2.3 NQ_WCHAR* CCMountInfo::serverName

server name

5.10.2.4 CCServerInfo CCMountInfo::serverInfo

server information

5.10.2.5 CCSStatistics CCMountInfo::statistics

statistics on data transferring from and to the share

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.11 CCMountInfoA Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_TIME](#) creationTime
- [NQ_INT](#) userLogonType
- [NQ_CHAR](#) * serverName
- [CCServerInfo](#) serverInfo
- [CCSStatistics](#) statistics

5.11.1 Detailed Description

This structure describes mount information. This is ASCII version of [CCMountInfo](#).

5.11.2 Field Documentation

5.11.2.1 NQ_TIME CCMountInfoA::creationTime

mount point creation time.

5.11.2.2 NQ_INT CCMountInfoA::userLogonType

user logon type, see: [CC_LOGON_TYPE_REGULAR](#), [CC_LOGON_TYPE_ANONYMOUS](#), [CC_LOGON_TYPE_GUEST](#)

5.11.2.3 NQ_CHAR* CCMountInfoA::serverName

server name

5.11.2.4 CCServerInfo CCMountInfoA::serverInfo

server information

5.11.2.5 CCSStatistics CCMountInfoA::statistics

statistics on data transferring from and to the share

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.12 CCNetDomainItem Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_WCHAR * name](#)
- [NQ_UINT32 type](#)
- [NQ_WCHAR * comment](#)
- [NQ_BYTE majorVersion](#)
- [NQ_BYTE minorVersion](#)

5.12.1 Detailed Description

5.12.2 Field Documentation

5.12.2.1 [NQ_WCHAR*](#) CCNetDomainItem::name

domain name

5.12.2.2 [NQ_UINT32](#) CCNetDomainItem::type

type

5.12.2.3 [NQ_WCHAR*](#) CCNetDomainItem::comment

domain comment

5.12.2.4 [NQ_BYTE](#) CCNetDomainItem::majorVersion

major version

5.12.2.5 [NQ_BYTE](#) CCNetDomainItem::minorVersion

minor version

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.13 CCNetServerItem Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_WCHAR * name](#)
- [NQ_UINT32 type](#)
- [NQ_WCHAR * comment](#)
- [NQ_BYTE majorVersion](#)
- [NQ_BYTE minorVersion](#)

5.13.1 Detailed Description

5.13.2 Field Documentation

5.13.2.1 NQ_WCHAR* CCNetServerItem::name

server name

5.13.2.2 NQ_UINT32 CCNetServerItem::type

type

5.13.2.3 NQ_WCHAR* CCNetServerItem::comment

server comment

5.13.2.4 NQ_BYTE CCNetServerItem::majorVersion

major version

5.13.2.5 NQ_BYTE CCNetServerItem::minorVersion

minor version

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.14 CCNetShareItem Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_WCHAR * name](#)
- [NQ_UINT32 type](#)
- [NQ_WCHAR * comment](#)

5.14.1 Detailed Description

5.14.2 Field Documentation

5.14.2.1 NQ_WCHAR* CCNetShareItem::name

share name

5.14.2.2 NQ_UINT32 CCNetShareItem::type

share type

5.14.2.3 NQ_WCHAR* CCNetShareItem::comment

share comment

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.15 CCNetShareItemA Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_CHAR * name](#)
- [NQ_UINT32 type](#)
- [NQ_CHAR * comment](#)

5.15.1 Detailed Description

This structure is used when calling [ccNetworkGetNextShareItem\(\)](#) [CCNetShareItem](#) - ASCII version

5.15.2 Field Documentation

5.15.2.1 NQ_CHAR* CCNetShareItemA::name

share name

5.15.2.2 NQ_UINT32 CCNetShareItemA::type

share type

5.15.2.3 NQ_CHAR* CCNetShareItemA::comment

share comment

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.16 CCServerInfo Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [CCSmbDialect dialect](#)
- [NQ_UINT32 serverCapabilities](#)
- [NQ_IPADDRESS * ips](#)
- [NQ_COUNT numIps](#)
- [NQ_INT serverSigningPolicy](#)

5.16.1 Detailed Description

This structure describes server information.

5.16.2 Field Documentation

5.16.2.1 CCSmbDialect CServerInfo::dialect

SMB dialect, see: [CCSmbDialect](#)

5.16.2.2 NQ_UINT32 CServerInfo::serverCapabilities

server capabilities, combination of the following bits: [CC_CAPABILITY_DFS](#) ,[CC_CAPABILITY_LEASING](#), [CC_CAPABILITY_LARGE_MTU](#), [CC_CAPABILITY_MULTI_CHANNEL](#), [CC_CAPABILITY_PERSISTENT_HANDLES](#), [CC_CAPABILITY_DIRECTORY_LEASING](#), [CC_CAPABILITY_ENCRYPTION](#)

5.16.2.3 NQ_IPADDRESS* CServerInfo::ips

pointer to server IP addresses.

5.16.2.4 NQ_COUNT CServerInfo::numIps

number of IP addresses in the array above.

5.16.2.5 NQ_INT CServerInfo::serverSigningPolicy

server signing policy, combination of the following bits: [CC_SIGNING_ENABLED](#) ,[CC_SIGNING_REQUIRED](#)

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.17 CCSStatistics Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_FLOAT readAmount](#)
- [NQ_FLOAT writeAmount](#)

5.17.1 Detailed Description

This structure contain statistics on data transferring

5.17.2 Field Documentation

5.17.2.1 NQ_FLOAT CCSStatistics::readAmount

amount of data which was read from share's files in GB

5.17.2.2 NQ_FLOAT CCSStatistics::writeAmount

amount of data which was written to share's files in GB

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.18 CCVolumeInfo Struct Reference

```
#include <ccapi.h>
```

Data Fields

- [NQ_UINT sectorsPerCluster](#)
- [NQ_UINT bytesPerSector](#)
- [NQ_UINT64 freeClusters](#)
- [NQ_UINT64 totalClusters](#)
- [NQ_UINT fsType](#)
- [NQ_UINT serialNumber](#)

5.18.1 Detailed Description

This structure describes volume information.

5.18.2 Field Documentation

5.18.2.1 NQ_UINT CCVolumeInfo::sectorsPerCluster

5.18.2.2 NQ_UINT CCVolumeInfo::bytesPerSector

5.18.2.3 NQ_UINT64 CCVolumeInfo::freeClusters

5.18.2.4 NQ_UINT64 CCVolumeInfo::totalClusters

5.18.2.5 NQ_UINT CCVolumeInfo::fsType

5.18.2.6 NQ_UINT CCVolumeInfo::serialNumber

The documentation for this struct was generated from the following file:

- [ccapi.h](#)

5.19 CMCodepage Struct Reference

```
#include <cmapi.h>
```

Data Fields

- [NQ_INT id](#)

- `NQ_INT(* toAnsi)(NQ_CHAR *outStr, NQ_UINT outLength, const NQ_WCHAR *inWStr, NQ_UINT inLength)`
- `NQ_INT(* toUnicode)(NQ_WCHAR *outWStr, NQ_UINT outLength, const NQ_CHAR *inStr, NQ_UINT inLength)`
- `NQ_INT(* toUpper)(NQ_CHAR *dst, const NQ_CHAR *src)`
- `void(* ansiToFs)(NQ_CHAR *str, NQ_INT size, const NQ_BYTE *firstIllegalChar, const NQ_BYTE *anyIllegalChar, NQ_INT numOfFirstIllChars, NQ_INT numOfAnyIllChars)`
- `void(* fsToAnsi)(NQ_CHAR *str, NQ_INT size, const NQ_BYTE *firstIllegalChar, const NQ_BYTE *anyIllegalChar, NQ_INT numOfFirstIllChars, NQ_INT numOfAnyIllChars)`
- `const NQ_WCHAR * a2uTab`

5.19.1 Detailed Description

This structure defines code page parameters.

5.19.2 Field Documentation

5.19.2.1 NQ_INT CMCodepage::id

Unique id value for this codepage. Code page is identified by this value. It is recommended to use respective Microsoft code page numbers (see notes).

Note

For Microsoft code page numbers see [http://msdn.microsoft.com/en-us/library/dd317756\(-VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd317756(-VS.85).aspx)

5.19.2.2 NQ_INT(* CMCodepage::toAnsi)(NQ_CHAR *outStr, NQ_UINT outLength, const NQ_WCHAR *inWStr, NQ_UINT inLength)

Codepage-specific function for Unicode to ANSI conversion

5.19.2.3 NQ_INT(* CMCodepage::toUnicode)(NQ_WCHAR *outWStr, NQ_UINT outLength, const NQ_CHAR *inStr, NQ_UINT inLength)

Codepage-specific function for ANSI to Unicode conversion

5.19.2.4 NQ_INT(* CMCodepage::toUpper)(NQ_CHAR *dst, const NQ_CHAR *src)

Codepage-specific function for capitalizing conversion Codepage-specific function for converting ANSI to the encoding used in the local FS

5.19.2.5 void(* CMCodepage::ansiToFs)(NQ_CHAR *str, NQ_INT size, const NQ_BYTE *firstIllegalChar, const NQ_BYTE *anyIllegalChar, NQ_INT numOfFirstIllChars, NQ_INT numOfAnyIllChars)

Codepage-specific function for converting the encoding used in the local FS into ANSI

5.19.2.6 void(* CMCodepage::fsToAnsi)(NQ_CHAR *str, NQ_INT size, const NQ_BYTE *firstIllegalChar, const NQ_BYTE *anyIllegalChar, NQ_INT numOfFirstIllChars, NQ_INT numOfAnyIllChars)

5.19.2.7 `const NQ_WCHAR* CMCodepage::a2uTab`

Fast ANSI to Unicode conversion table. Using this table avoids calling the functions defined in this structure. It is only used for single-byte encodings like Hebrew/Russian/Turkish/Greek/etc. This pointer may be NULL. It should be NULL for multiple-byte encodings.

The documentation for this struct was generated from the following file:

- [cmapi.h](#)

5.20 CMCrypterList Struct Reference

```
#include <cmapi.h>
```

Data Fields

- [CMAbstractHasher md4](#)
- [CMAbstractCipher md5](#)
- [CMAbstractCipher hmacmd5](#)
- [CMAbstractIOCipher sha256](#)
- [CMAbstractIOCipher aes128cmac](#)
- [CMAbstractHasher512 sha512](#)
- [CMAbstractCcmEncryption aes128ccmEncryption](#)
- [CMAbstractCcmDecryption aes128ccmDecryption](#)
- [CMAbstractGcmEncryption aes128gcmEncryption](#)
- [CMAbstractGcmDecryption aes128gcmDecryption](#)

5.20.1 Detailed Description

This structure designates a list of cryptographic algorithms currently used by NQ. It is used to replace existing cryptographic algorithms.

If a particular cryptographic algorithm is NULL, it is not replaced.

5.20.2 Field Documentation

5.20.2.1 `CMAbstractHasher CMCrypterList::md4`

MD4 hasher

5.20.2.2 `CMAbstractCipher CMCrypterList::md5`

MD5 hasher

5.20.2.3 `CMAbstractCipher CMCrypterList::hmacmd5`

HMACMD5 crypter

5.20.2.4 `CMAbstractIOCipher CMCrypterList::sha256`

SHA-256 crypter

5.20.2.5 **CMAbstractIOCipher** **CMCrypterList::aes128cmac**

AES-CMAC crypter

5.20.2.6 **CMAbstractHasher512** **CMCrypterList::sha512**

SHA-512 crypter

5.20.2.7 **CMAbstractCcmEncryption** **CMCrypterList::aes128ccmEncryption**

AES-CCM encryption algorithm

5.20.2.8 **CMAbstractCcmDecryption** **CMCrypterList::aes128ccmDecryption**

AES-CCM decryption algorithm

5.20.2.9 **CMAbstractGcmEncryption** **CMCrypterList::aes128gcmEncryption**

AES-GCM encryption algorithm

5.20.2.10 **CMAbstractGcmDecryption** **CMCrypterList::aes128gcmDecryption**

AES-GCM decryption algorithm

The documentation for this struct was generated from the following file:

- [cmapi.h](#)

5.21 **CMIOVecBlob** Struct Reference

```
#include <udapi.h>
```

Data Fields

- [NQ_IOBuf](#) data
- [NQ_COUNT](#) len

5.21.1 Detailed Description

Description A wrapper around IOVEC buffer to represent a message.

5.21.2 Field Documentation

5.21.2.1 **NQ_IOBuf** **CMIOVecBlob::data**

IOVec buffer or byte *

5.21.2.2 NQ_COUNT CMIOVecBlob::len

total packet length

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.22 CMProductInfo Struct Reference

```
#include <cmapi.h>
```

Data Fields

- [NQ_CHAR](#) productName [[CM_PRODUCT_STRING_LEN](#)]
- [NQ_CHAR](#) productType [[CM_PRODUCT_STRING_LEN](#)]
- [NQ_CHAR](#) version [[CM_PRODUCT_STRING_LEN](#)]
- [NQ_UINT32](#) checksum

5.22.1 Field Documentation

5.22.1.1 [NQ_CHAR](#) CMProductInfo::productName[[CM_PRODUCT_STRING_LEN](#)]

5.22.1.2 [NQ_CHAR](#) CMProductInfo::productType[[CM_PRODUCT_STRING_LEN](#)]

5.22.1.3 [NQ_CHAR](#) CMProductInfo::version[[CM_PRODUCT_STRING_LEN](#)]

5.22.1.4 [NQ_UINT32](#) CMProductInfo::checksum

The documentation for this struct was generated from the following file:

- [cmapi.h](#)

5.23 CMResolverRegisteredMethodDescription Struct Reference

```
#include <cmapi.h>
```

Data Fields

- [NQ_INT](#) activationPriority
- [NQ_UINT32](#) timeout
- [NQ_IPADDRESS](#) * serverIp
- [NQ_STATUS](#)(* [requestByName](#))(SYSocketHandle socket, const [NQ_WCHAR](#) *name, void *context, const [NQ_IPADDRESS](#) *serverIp, [NQ_COUNT](#) *numOfSentRequests)
- [NQ_STATUS](#)(* [responseByName](#))(SYSocketHandle socket, [NQ_IPADDRESS](#) **pAddressArray, [NQ_INT](#) *numIps, void **pContext)
- [NQ_STATUS](#)(* [requestByIp](#))(SYSocketHandle socket, const [NQ_IPADDRESS](#) *ip, void *context, const [NQ_IPADDRESS](#) *serverIp, [NQ_COUNT](#) *numOfSentRequests)
- [NQ_STATUS](#)(* [responseByIp](#))(SYSocketHandle socket, const [NQ_WCHAR](#) **pName, void **pContext)

5.23.1 Detailed Description

A descriptor of one resolver method. Use this structure to register a resolution method using the [cmResolver-RegisterExternalMethod\(\)](#)

5.23.2 Field Documentation

5.23.2.1 NQ_INT CMResolverRegisteredMethodDescription::activationPriority

This parameter may have one of the following values:

1. Activate registered method before all other methods.
2. Activate registered method after the existing NQ unicast methods and before existing NQ multicast methods. In this case registered method will be activated only if existing NQ unicast methods failed to resolve name or IP
3. Activate registered method only if all other methods failed.

5.23.2.2 NQ_UINT32 CMResolverRegisteredMethodDescription::timeout

Timeout in seconds to use with this method

5.23.2.3 NQ_IPADDRESS* CMResolverRegisteredMethodDescription::serverIP

IP of the server to perform resolution with. This value is only used by Unicast methods.

5.23.2.4 NQ_STATUS(* CMResolverRegisteredMethodDescription::requestByName)(SYSocketHandle socket, const NQ_WCHAR *name, void *context, const NQ_IPADDRESS *serverIp, NQ_COUNT *numOfSentRequests)

Routine for composing and sending a name resolution request

Parameters

<i>socket,:</i>	Socket handle to use for sending
<i>name,:</i>	Name to resolve
<i>context,:</i>	Pointer to a method-specific context. This value may be NULL on the first call.
<i>serverIp,:</i>	Pointer to the IP of the server to query or NULL for multicast
<i>numOfSent-Requests,:</i>	Return the number of requests that were sent.

Returns

NQ_SUCCESS request sent NQ_ERR_<*> error

5.23.2.5 NQ_STATUS(* CMResolverRegisteredMethodDescription::responseByName)(SYSocketHandle socket, NQ_IPADDRESS **pAddressArray, NQ_INT *numIps, void **pContext)

Routine for receiving and parsing a name resolution response

Parameters

<i>socket</i>	Socket handle to use for sending
<i>pAddressArray</i>	Address of the pointer which this call sets to an array of resolved IP addresses. It is caller's responsibility to release this array. On error, this pointer remains untouched.
<i>numIps</i>	Pointer to the number of resolved IPs.
<i>pContext</i>	Double pointer to a method-specific context. Method may dispose context and create a new one.

Returns

NQ_SUCCESS name successfully resolved NQ_ERR_MOREDATA more exchange expected NQ_ERR_NOACCESS more comprehensive method with the same code should be used NQ_ERR_<*> error

5.23.2.6 NQ_STATUS(* CMResolverRegisteredMethodDescription::requestByIp)(SYSocketHandle socket, const NQ_IPADDRESS *ip, void *context, const NQ_IPADDRESS *serverIp, NQ_COUNT *numOfSentRequests)

Routine for composing and sending an IP resolution request

Parameters

<i>socket</i>	Socket handle to use for sending
<i>ip</i>	Pointer to the IP address to resolve.
<i>context</i>	Pointer to a method-specific context. This value may be NULL on the first call.
<i>serverIp</i>	Pointer to the IP of the server to query or NULL for multicast
<i>numOfSent-Requests</i>	Return the number of requests that were sent.

Returns

NQ_SUCCESS request sent NQ_ERR_<*> error

5.23.2.7 NQ_STATUS(* CMResolverRegisteredMethodDescription::responseByIp)(SYSocketHandle socket, const NQ_WCHAR **pName, void **pContext)

Routine for receiving and parsing a name resolution response

Parameters

<i>socket</i>	Socket handle to use for sending
<i>pName</i>	Double pointer to the resolved name. On success, this variable will point to a newly allocated name. Its is caller's responsibility to release it later.
<i>pContext</i>	Double pointer to a method-specific context. Method may dispose context and create a new one.

Returns

NQ_SUCCESS name successfully resolved NQ_ERR_MOREDATA more exchange expected NQ_ERR_NOACCESS more comprehensive method with the same code should be used NQ_ERR_<*> error

The documentation for this struct was generated from the following file:

- [cmapi.h](#)

5.24 CSFileData Struct Reference

```
#include <csapi.h>
```

Data Fields

- [NQ_WCHAR fileNamePath](#) [CM_BUFFERLENGTH([NQ_WCHAR](#), UD_FS_FILENAMELEN)]
- [NQ_WCHAR userName](#) [CM_BUFFERLENGTH([NQ_WCHAR](#), CM_USERNAMELENGTH)]
- [NQ_WCHAR IP](#) [CM_BUFFERLENGTH([NQ_WCHAR](#), CM_IPADDR_MAXLEN)]
- [NQ_BOOL isDirectory](#)

5.24.1 Detailed Description

This structure holds open file or directory info. NQ holds one file entry per successful create request. This file info gives general info per entry. This structure is used when calling [csEnumerateOpenFiles\(\)](#)

5.24.2 Field Documentation

5.24.2.1 [NQ_WCHAR](#) CSFileData::fileNamePath[CM_BUFFERLENGTH([NQ_WCHAR](#), UD_FS_FILENAMELEN)]

File name buffer

5.24.2.2 [NQ_WCHAR](#) CSFileData::userName[CM_BUFFERLENGTH([NQ_WCHAR](#), CM_USERNAMELENGTH)]

File name buffer

5.24.2.3 [NQ_WCHAR](#) CSFileData::IP[CM_BUFFERLENGTH([NQ_WCHAR](#), CM_IPADDR_MAXLEN)]

IP buffer

5.24.2.4 [NQ_BOOL](#) CSFileData::isDirectory

Is this a directory or a file

The documentation for this struct was generated from the following file:

- [csapi.h](#)

5.25 EVCSOpenFileEntry Struct Reference

```
#include <csapi.h>
```

Data Fields

- [NQ_WCHAR userName](#) [256]
- [NQ_WCHAR fileName](#) [UD_FS_FILENAMELEN]
- [NQ_WCHAR shareName](#) [UD_FS_MAXSHARELEN]
- [NQ_IPADDRESS ip](#)
- [NQ_UINT16 access](#)

5.25.1 Detailed Description

This structure designates one file entry. A file entry is created by NQ Server upon a successful file open.

5.25.2 Field Documentation

5.25.2.1 NQ_WCHAR EVCSOpenFileEntry::userName[256]

User name buffer.

5.25.2.2 NQ_WCHAR EVCSOpenFileEntry::fileName[UD_FS_FILENAMELEN]

File name buffer. This buffer will be filled with full file path. should be 300 characters long at least.

5.25.2.3 NQ_WCHAR EVCSOpenFileEntry::shareName[UD_FS_MAXSHARELEN]

Share name buffer. his buffer should be 256 characters long at least.

5.25.2.4 NQ_IPADDRESS EVCSOpenFileEntry::ip

Next side (client) IP address.

5.25.2.5 NQ_UINT16 EVCSOpenFileEntry::access

Access bits as indicated by the open operation.

The documentation for this struct was generated from the following file:

- [csapi.h](#)

5.26 EVCSShareConnectionEntry Struct Reference

```
#include <csapi.h>
```

Data Fields

- [NQ_WCHAR userName](#) [256]
- [NQ_WCHAR shareName](#) [UD_FS_MAXSHARELEN]
- [NQ_IPADDRESS ip](#)
- [NQ_BOOL ipc](#)
- [NQ_BOOL printQueue](#)

5.26.1 Detailed Description

This structure designates one share connection entry. A share connection entry is created by NQ Server upon a successful TreeConnect request.

5.26.2 Field Documentation

5.26.2.1 NQ_WCHAR EVCSShareConnectionEntry::userName[256]

User name buffer. This buffer should be 256 characters long at least.

5.26.2.2 **NQ_WCHAR** EVCSShareConnectionEntry::shareName[UD_FS_MAXSHARELEN]

Share name buffer. This buffer should be 256 characters long at least.

5.26.2.3 **NQ_IPADDRESS** EVCSShareConnectionEntry::ip

Next side (client) IP address.

5.26.2.4 **NQ_BOOL** EVCSShareConnectionEntry::ipc

IPC flag. This flag is TRUE for the IPC\$ share and FALSE for a file share.

5.26.2.5 **NQ_BOOL** EVCSShareConnectionEntry::printQueue

Printer flag. When TRUE, this entry designates a printer share. For a file share this flag is FALSE.

The documentation for this struct was generated from the following file:

- [csapi.h](#)

5.27 FileDataA_t Struct Reference

```
#include <csapi.h>
```

Data Fields

- [NQ_CHAR](#) [fileNamePath](#) [CM_BUFFERLENGTH([NQ_CHAR](#), UD_FS_FILENAMELEN)]
- [NQ_CHAR](#) [userName](#) [CM_BUFFERLENGTH([NQ_CHAR](#), CM_USERNAMELENGTH)]
- [NQ_CHAR](#) [IP](#) [CM_BUFFERLENGTH([NQ_CHAR](#), CM_IPADDR_MAXLEN)]
- [NQ_BOOL](#) [isDirectory](#)

5.27.1 Detailed Description

This structure holds open file or directory info. NQ holds one file entry per successful create request. This file info gives general info per entry. This structure is used when calling [csEnumerateOpenFilesA\(\)](#)

5.27.2 Field Documentation

5.27.2.1 **NQ_CHAR** FileDataA_t::fileNamePath[CM_BUFFERLENGTH(NQ_CHAR, UD_FS_FILENAMELEN)]

File name and path buffer

5.27.2.2 **NQ_CHAR** FileDataA_t::userName[CM_BUFFERLENGTH(NQ_CHAR, CM_USERNAMELENGTH)]

User name buffer

5.27.2.3 **NQ_CHAR** FileDataA_t::IP[CM_BUFFERLENGTH(NQ_CHAR, CM_IPADDR_MAXLEN)]

IP buffer

5.27.2.4 NQ_BOOL FileDataA_t::isDirectory

Is this a directory or a file

The documentation for this struct was generated from the following file:

- [csapi.h](#)

5.28 NQ_INT64 Struct Reference

```
#include <udapi.h>
```

Data Fields

- [NQ_UINT32 low](#)
- [NQ_UINT32 high](#)
- [NQ_INT32 sign](#)

5.28.1 Detailed Description

[NQ_INT64](#) type

5.28.2 Field Documentation

5.28.2.1 NQ_UINT32 NQ_INT64::low

low part

5.28.2.2 NQ_UINT32 NQ_INT64::high

high part

5.28.2.3 NQ_INT32 NQ_INT64::sign

sign

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.29 NQ_IOVEC Struct Reference

```
#include <udapi.h>
```

Data Fields

- [NQ_BYTE * iov_base](#)
- [NQ_COUNT iov_len](#)

5.29.1 Field Documentation

5.29.1.1 NQ_BYTE* NQ_IOVEC::iov_base

Starting address

5.29.1.2 NQ_COUNT NQ_IOVEC::iov_len

Number of bytes

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.30 NQ_IOVECBUF Struct Reference

```
#include <udapi.h>
```

Data Fields

- [NQ_IOVEC * iovec](#)
- [NQ_COUNT iovSegmentCnt](#)

5.30.1 Field Documentation

5.30.1.1 NQ_IOVEC* NQ_IOVECBUF::iovec

IOVec array

5.30.1.2 NQ_COUNT NQ_IOVECBUF::iovSegmentCnt

Number of array elements

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.31 NQ_IOVecBuffer Struct Reference

```
#include <udapi.h>
```

Data Fields

- [NQ_IOVEC data \[3\]](#)
- [NQ_COUNT len](#)

5.31.1 Field Documentation

5.31.1.1 NQ_IOVEC NQ_IOVecBuffer::data[3]

The iovec buffer.

5.31.1.2 NQ_COUNT NQ_IOVecBuffer::len

Total data lenggh

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.32 NQ_IPADDRESS Struct Reference

```
#include <udapi.h>
```

Data Fields

- [NQ_BYTE version](#)
- union {
 - [NQ_IPADDRESS6 v6](#)
 - [NQ_IPADDRESS4 v4](#)
 - [NQ_IPADDRESSVR vr](#)
- } [addr](#)

5.32.1 Detailed Description

Structure describing [NQ_IPADDRESS](#) stores IP version and address

5.32.2 Field Documentation

5.32.2.1 NQ_BYTE NQ_IPADDRESS::version

IP version

5.32.2.2 NQ_IPADDRESS6 NQ_IPADDRESS::v6

IP address

5.32.2.3 NQ_IPADDRESS4 NQ_IPADDRESS::v4

IP address

5.32.2.4 NQ_IPADDRESSVR NQ_IPADDRESS::vr

IP address

5.32.2.5 union { ... } NQ_IPADDRESS::addr

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.33 NQ_IPADDRESSVR Struct Reference

```
#include <udapi.h>
```

Data Fields

- [NQ_INT](#) id
- [NQ_BYTE](#) version
- union {
 - [NQ_IPADDRESS6](#) v6
 - [NQ_IPADDRESS4](#) v4

5.33.1 Field Documentation

5.33.1.1 [NQ_INT](#) NQ_IPADDRESSVR::id

5.33.1.2 [NQ_BYTE](#) NQ_IPADDRESSVR::version

5.33.1.3 [NQ_IPADDRESS6](#) NQ_IPADDRESSVR::v6

5.33.1.4 [NQ_IPADDRESS4](#) NQ_IPADDRESSVR::v4

5.33.1.5 union { ... } NQ_IPADDRESSVR::addr

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.34 NQ_UINT64 Struct Reference

```
#include <udapi.h>
```

Data Fields

- [NQ_UINT32](#) low
- [NQ_UINT32](#) high

5.34.1 Detailed Description

[NQ_UINT64](#) type

5.34.2 Field Documentation

5.34.2.1 [NQ_UINT32](#) NQ_UINT64::low

low part

5.34.2.2 NQ_UINT32 NQ_UINT64::high

high part

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.35 SYDeviceMode Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_UINT16 specVersion](#)
- [NQ_UINT16 driverVersion](#)
- [NQ_UINT16 size](#)
- [NQ_UINT16 driverExtraLength](#)
- [const NQ_BYTE * driverExtraData](#)
- [NQ_UINT32 fields](#)
- [NQ_UINT16 orientation](#)
- [NQ_UINT16 paperSize](#)
- [NQ_UINT16 paperLength](#)
- [NQ_UINT16 paperWidth](#)
- [NQ_UINT16 scale](#)
- [NQ_UINT16 copies](#)
- [NQ_UINT16 defaultSource](#)
- [NQ_UINT16 printQuality](#)
- [NQ_UINT16 color](#)
- [NQ_UINT16 duplex](#)
- [NQ_UINT16 yResolution](#)
- [NQ_UINT16 ttOption](#)
- [NQ_UINT16 collate](#)
- [const NQ_WCHAR * formName](#)
- [NQ_UINT16 logPixels](#)
- [NQ_UINT32 bitsPerPel](#)
- [NQ_UINT32 pelsWidth](#)
- [NQ_UINT32 pelsHeight](#)
- [NQ_UINT32 displayFlags](#)
- [NQ_UINT32 displayFrequency](#)
- [NQ_UINT32 icmMethod](#)
- [NQ_UINT32 icmIntent](#)
- [NQ_UINT32 mediaType](#)
- [NQ_UINT32 ditherType](#)
- [NQ_UINT32 reserved1](#)
- [NQ_UINT32 reserved2](#)
- [NQ_UINT32 panningWidth](#)
- [NQ_UINT32 panningHeight](#)

5.35.1 Detailed Description

Printer information structures

5.35.2 Field Documentation

- 5.35.2.1 **NQ_UINT16** SYDeviceMode::specVersion
- 5.35.2.2 **NQ_UINT16** SYDeviceMode::driverVersion
- 5.35.2.3 **NQ_UINT16** SYDeviceMode::size
- 5.35.2.4 **NQ_UINT16** SYDeviceMode::driverExtraLength
- 5.35.2.5 **const NQ_BYTE*** SYDeviceMode::driverExtraData
- 5.35.2.6 **NQ_UINT32** SYDeviceMode::fields
- 5.35.2.7 **NQ_UINT16** SYDeviceMode::orientation
- 5.35.2.8 **NQ_UINT16** SYDeviceMode::paperSize
- 5.35.2.9 **NQ_UINT16** SYDeviceMode::paperLength
- 5.35.2.10 **NQ_UINT16** SYDeviceMode::paperWidth
- 5.35.2.11 **NQ_UINT16** SYDeviceMode::scale
- 5.35.2.12 **NQ_UINT16** SYDeviceMode::copies
- 5.35.2.13 **NQ_UINT16** SYDeviceMode::defaultSource
- 5.35.2.14 **NQ_UINT16** SYDeviceMode::printQuality
- 5.35.2.15 **NQ_UINT16** SYDeviceMode::color
- 5.35.2.16 **NQ_UINT16** SYDeviceMode::duplex
- 5.35.2.17 **NQ_UINT16** SYDeviceMode::yResolution
- 5.35.2.18 **NQ_UINT16** SYDeviceMode::ttOption
- 5.35.2.19 **NQ_UINT16** SYDeviceMode::collate
- 5.35.2.20 **const NQ_WCHAR*** SYDeviceMode::formName
- 5.35.2.21 **NQ_UINT16** SYDeviceMode::logPixels
- 5.35.2.22 **NQ_UINT32** SYDeviceMode::bitsPerPel
- 5.35.2.23 **NQ_UINT32** SYDeviceMode::pelsWidth
- 5.35.2.24 **NQ_UINT32** SYDeviceMode::pelsHeight
- 5.35.2.25 **NQ_UINT32** SYDeviceMode::displayFlags
- 5.35.2.26 **NQ_UINT32** SYDeviceMode::displayFrequency
- 5.35.2.27 **NQ_UINT32** SYDeviceMode::icmMethod

- 5.35.2.28 **NQ_UINT32** SYDeviceMode::icmIntent
- 5.35.2.29 **NQ_UINT32** SYDeviceMode::mediaType
- 5.35.2.30 **NQ_UINT32** SYDeviceMode::ditherType
- 5.35.2.31 **NQ_UINT32** SYDeviceMode::reserved1
- 5.35.2.32 **NQ_UINT32** SYDeviceMode::reserved2
- 5.35.2.33 **NQ_UINT32** SYDeviceMode::panningWidth
- 5.35.2.34 **NQ_UINT32** SYDeviceMode::panningHeight

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.36 SYFileInformation Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_TIME](#) creationTime
- [NQ_TIME](#) lastChangeTime
- [NQ_TIME](#) lastAccessTime
- [NQ_TIME](#) lastWriteTime
- [NQ_UINT32](#) sizeLow
- [NQ_UINT32](#) sizeHigh
- [NQ_UINT32](#) allocSizeLow
- [NQ_UINT32](#) allocSizeHigh
- [NQ_UINT32](#) attributes
- [NQ_UINT32](#) isDeleted
- [NQ_UINT32](#) numLinks
- [NQ_UINT32](#) fieldLow
- [NQ_UINT32](#) fieldHigh

5.36.1 Detailed Description

File information structure. It is used for querying information from the file system and modifying this information.

Note

This structure is designated here for reference only and it must not be modified.

5.36.2 Field Documentation

5.36.2.1 **NQ_TIME** SYFileInformation::creationTime

Seconds in UNIX format.

5.36.2.2 NQ_TIME SYFileInformation::lastChangeTime

Seconds in UNIX format.

5.36.2.3 NQ_TIME SYFileInformation::lastAccessTime

Seconds in UNIX format.

5.36.2.4 NQ_TIME SYFileInformation::lastWriteTime

Seconds in UNIX format.

5.36.2.5 NQ_UINT32 SYFileInformation::sizeLow

Low part of the file size.

5.36.2.6 NQ_UINT32 SYFileInformation::sizeHigh

High part of the file size.

5.36.2.7 NQ_UINT32 SYFileInformation::allocSizeLow

Low part of the file size.

5.36.2.8 NQ_UINT32 SYFileInformation::allocSizeHigh

High part of the file size.

5.36.2.9 NQ_UINT32 SYFileInformation::attributes

DOS FS format attributes.

5.36.2.10 NQ_UINT32 SYFileInformation::isDeleted

File is marked for deletion.

5.36.2.11 NQ_UINT32 SYFileInformation::numLinks

Number of hard links to the file.

5.36.2.12 NQ_UINT32 SYFileInformation::fileIdLow

Low part of file ID.

5.36.2.13 NQ_UINT32 SYFileInformation::fileIdHigh

High part of file ID.

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.37 SYPrinterDriver Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_UINT32](#) `osVersion`
- `const` [NQ_WCHAR](#) * `name`
- `const` [NQ_WCHAR](#) * `driverPath`
- `const` [NQ_WCHAR](#) * `dataFile`
- `const` [NQ_WCHAR](#) * `configFile`
- `const` [NQ_WCHAR](#) * `helpFile`
- `const` [NQ_WCHAR](#) ** `dependentFiles`
- `const` [NQ_WCHAR](#) * `monitorName`
- `const` [NQ_WCHAR](#) * `defaultDataType`
- `const` [NQ_WCHAR](#) ** `previousNames`
- [NQ_UINT32](#) `driverDate`
- [NQ_UINT32](#) `driverVersions` [2]
- `const` [NQ_WCHAR](#) * `manufacturer`
- `const` [NQ_WCHAR](#) * `manufacturerURL`
- `const` [NQ_WCHAR](#) * `hardwareID`
- `const` [NQ_WCHAR](#) * `provider`
- [NQ_UINT32](#) `attributes`
- [NQ_UINT32](#) `configVersion`
- [NQ_UINT32](#) `driverVersion`

5.37.1 Detailed Description

Printer driver information structure

5.37.2 Field Documentation

5.37.2.1 [NQ_UINT32](#) `SYPrinterDriver::osVersion`

5.37.2.2 `const` [NQ_WCHAR](#)* `SYPrinterDriver::name`

5.37.2.3 `const` [NQ_WCHAR](#)* `SYPrinterDriver::driverPath`

5.37.2.4 `const` [NQ_WCHAR](#)* `SYPrinterDriver::dataFile`

5.37.2.5 `const` [NQ_WCHAR](#)* `SYPrinterDriver::configFile`

5.37.2.6 `const` [NQ_WCHAR](#)* `SYPrinterDriver::helpFile`

5.37.2.7 `const` [NQ_WCHAR](#)** `SYPrinterDriver::dependentFiles`

5.37.2.8 `const` [NQ_WCHAR](#)* `SYPrinterDriver::monitorName`

5.37.2.9 `const` [NQ_WCHAR](#)* `SYPrinterDriver::defaultDataType`

5.37.2.10 `const` [NQ_WCHAR](#)** `SYPrinterDriver::previousNames`

5.37.2.11 [NQ_UINT32](#) `SYPrinterDriver::driverDate`

- 5.37.2.12 **NQ_UINT32** SYPrinterDriver::driverVersions[2]
- 5.37.2.13 **const NQ_WCHAR*** SYPrinterDriver::manufacturer
- 5.37.2.14 **const NQ_WCHAR*** SYPrinterDriver::manufacturerURL
- 5.37.2.15 **const NQ_WCHAR*** SYPrinterDriver::hardwareID
- 5.37.2.16 **const NQ_WCHAR*** SYPrinterDriver::provider
- 5.37.2.17 **NQ_UINT32** SYPrinterDriver::attributes
- 5.37.2.18 **NQ_UINT32** SYPrinterDriver::configVersion
- 5.37.2.19 **NQ_UINT32** SYPrinterDriver::driverVersion

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.38 SYPrinterInfo Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_UINT32](#) flags
- [NQ_WCHAR *](#) portName
- [NQ_WCHAR *](#) driverName
- [NQ_WCHAR *](#) comment
- [NQ_WCHAR *](#) location
- [SYDeviceMode](#) devMode
- [NQ_WCHAR *](#) sepFile
- [NQ_WCHAR *](#) printProcessor
- [NQ_WCHAR *](#) dataType
- [NQ_WCHAR *](#) parameters
- [NQ_UINT32](#) attributes
- [NQ_UINT32](#) priority
- [NQ_UINT32](#) defaultPriority
- [NQ_UINT32](#) startTime
- [NQ_UINT32](#) untilTime
- [NQ_UINT32](#) status
- [NQ_UINT32](#) cJobs
- [NQ_UINT32](#) totalJobs
- [NQ_UINT32](#) totalBytes
- [NQ_UINT32](#) globalCounter
- [NQ_UINT32](#) totalPages
- [NQ_UINT16](#) majorVersion
- [NQ_UINT16](#) buildVersion
- [NQ_UINT32](#) sessionCounter
- [NQ_UINT32](#) printerErrors
- [NQ_UINT32](#) cSetPrinter
- [NQ_UINT32](#) averagePpm

- [NQ_UINT32 deviceNotSelectedTimeout](#)
- [NQ_UINT32 transmissionRetryTimeout](#)
- [NQ_BYTE * securityDescriptor](#)
- [NQ_UINT32 securityDescriptorLength](#)

5.38.1 Detailed Description

United printer information structure

5.38.2 Field Documentation

5.38.2.1 NQ_UINT32 SYPrinterInfo::flags

Printer flags - see [Printer Flags](#)

5.38.2.2 NQ_WCHAR* SYPrinterInfo::portName

5.38.2.3 NQ_WCHAR* SYPrinterInfo::driverName

5.38.2.4 NQ_WCHAR* SYPrinterInfo::comment

5.38.2.5 NQ_WCHAR* SYPrinterInfo::location

5.38.2.6 SYDeviceMode SYPrinterInfo::devMode

5.38.2.7 NQ_WCHAR* SYPrinterInfo::sepFile

5.38.2.8 NQ_WCHAR* SYPrinterInfo::printProcessor

5.38.2.9 NQ_WCHAR* SYPrinterInfo::dataType

5.38.2.10 NQ_WCHAR* SYPrinterInfo::parameters

5.38.2.11 NQ_UINT32 SYPrinterInfo::attributes

5.38.2.12 NQ_UINT32 SYPrinterInfo::priority

5.38.2.13 NQ_UINT32 SYPrinterInfo::defaultPriority

5.38.2.14 NQ_UINT32 SYPrinterInfo::startTime

5.38.2.15 NQ_UINT32 SYPrinterInfo::untilTime

5.38.2.16 NQ_UINT32 SYPrinterInfo::status

5.38.2.17 NQ_UINT32 SYPrinterInfo::cJobs

5.38.2.18 NQ_UINT32 SYPrinterInfo::totalJobs

5.38.2.19 NQ_UINT32 SYPrinterInfo::totalBytes

5.38.2.20 NQ_UINT32 SYPrinterInfo::globalCounter

5.38.2.21 NQ_UINT32 SYPrinterInfo::totalPages

- 5.38.2.22 **NQ_UINT16** SYPrinterInfo::majorVersion
- 5.38.2.23 **NQ_UINT16** SYPrinterInfo::buildVersion
- 5.38.2.24 **NQ_UINT32** SYPrinterInfo::sessionCounter
- 5.38.2.25 **NQ_UINT32** SYPrinterInfo::printerErrors
- 5.38.2.26 **NQ_UINT32** SYPrinterInfo::cSetPrinter
- 5.38.2.27 **NQ_UINT32** SYPrinterInfo::averagePpm
- 5.38.2.28 **NQ_UINT32** SYPrinterInfo::deviceNotSelectedTimeout
- 5.38.2.29 **NQ_UINT32** SYPrinterInfo::transmissionRetryTimeout
- 5.38.2.30 **NQ_BYTE*** SYPrinterInfo::securityDescriptor
- 5.38.2.31 **NQ_UINT32** SYPrinterInfo::securityDescriptorLength

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.39 SYPrintFormInfo Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_UINT32](#) id
- const [NQ_WCHAR](#) * name
- [NQ_UINT32](#) flags
- [SYPrintSize](#) size
- [SYPrintRect](#) imageableArea

5.39.1 Detailed Description

Form information structure

5.39.2 Field Documentation

5.39.2.1 **NQ_UINT32** SYPrintFormInfo::id

Form ID

5.39.2.2 **const NQ_WCHAR*** SYPrintFormInfo::name

Form name

5.39.2.3 NQ_UINT32 SYPrintFormInfo::flags

See [Printer Form Flags](#)

5.39.2.4 SYPrintSize SYPrintFormInfo::size

Form size

5.39.2.5 SYPrintRect SYPrintFormInfo::imageableArea

Form shape

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.40 SYPrintJobInfo Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_UINT32 id](#)
- [const NQ_WCHAR * documentName](#)
- [const NQ_WCHAR * pStatus](#)
- [NQ_UINT32 status](#)
- [NQ_UINT32 priority](#)
- [NQ_UINT32 position](#)
- [NQ_UINT32 totalPages](#)
- [NQ_UINT32 pagesPrinted](#)
- [NQ_UINT32 submitTime](#)
- [NQ_UINT32 startTime](#)
- [NQ_UINT32 untilTime](#)
- [NQ_UINT32 time](#)
- [NQ_UINT32 size](#)
- [NQ_BYTE * securityDescriptor](#)
- [NQ_UINT32 securityDescriptorLength](#)
- [const void * user](#)

5.40.1 Detailed Description

United print job information structure

5.40.2 Field Documentation

5.40.2.1 NQ_UINT32 SYPrintJobInfo::id

Job ID

5.40.2.2 const NQ_WCHAR* SYPrintJobInfo::documentName

Name of the document being printed

5.40.2.3 const NQ_WCHAR* SYPrintJobInfo::pStatus

Optional text description of the status

5.40.2.4 NQ_UINT32 SYPrintJobInfo::status

Status code (when the above is null) - see [Printer Status Job](#)

5.40.2.5 NQ_UINT32 SYPrintJobInfo::priority

Job priority - see [Printer Priority Job](#)

5.40.2.6 NQ_UINT32 SYPrintJobInfo::position

Job position in the print queue (0 - current)

5.40.2.7 NQ_UINT32 SYPrintJobInfo::totalPages

Number of pages in the entire job

5.40.2.8 NQ_UINT32 SYPrintJobInfo::pagesPrinted

Number of pages printed so far

5.40.2.9 NQ_UINT32 SYPrintJobInfo::submitTime

Time this job was queued

5.40.2.10 NQ_UINT32 SYPrintJobInfo::startTime

The earliest time this job can be printed

5.40.2.11 NQ_UINT32 SYPrintJobInfo::untilTime

The latest time this job can be printed

5.40.2.12 NQ_UINT32 SYPrintJobInfo::time

Elapsed time since this job begun being printed

5.40.2.13 NQ_UINT32 SYPrintJobInfo::size

Job size in bytes

5.40.2.14 NQ_BYTE* SYPrintJobInfo::securityDescriptor

Pointer to security descriptor

5.40.2.15 NQ_UINT32 SYPrintJobInfo::securityDescriptorLength

Security desriptor length

5.40.2.16 const void* SYPrintJobInfo::user

User context

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.41 SYPrintRect Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_UINT32 left](#)
- [NQ_UINT32 top](#)
- [NQ_UINT32 right](#)
- [NQ_UINT32 bottom](#)

5.41.1 Detailed Description

structure defining a rectangle area

5.41.2 Field Documentation

5.41.2.1 NQ_UINT32 SYPrintRect::left

Horizontal shift to the left edge in thousands of millimeters

5.41.2.2 NQ_UINT32 SYPrintRect::top

Vertical shift to the top edge in thousands of millimeters

5.41.2.3 NQ_UINT32 SYPrintRect::right

Horizontal shift to the right edge in thousands of millimeters

5.41.2.4 NQ_UINT32 SYPrintRect::bottom

Vertical shift to the bottom edge in thousands of millimeters

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.42 SYPrintSize Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_UINT32 width](#)
- [NQ_UINT32 height](#)

5.42.1 Detailed Description

Size structure

5.42.2 Field Documentation

5.42.2.1 NQ_UINT32 SYPrintSize::width

Width in thousands of millimeters

5.42.2.2 NQ_UINT32 SYPrintSize::height

Height in thousands of millimeters

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.43 SYTimeFragments Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_UINT16 sec](#)
- [NQ_UINT16 min](#)
- [NQ_UINT16 hour](#)
- [NQ_UINT16 day](#)
- [NQ_UINT16 month](#)
- [NQ_UINT16 year](#)

5.43.1 Detailed Description

This structure is intended for composing/decomposing system time - contains a field for each time fraction.

Note

This structure is designated here for reference only and it must not be modified.

5.43.2 Field Documentation

5.43.2.1 NQ_UINT16 SYTimeFragments::sec

Seconds (0 - 59).

5.43.2.2 NQ_UINT16 SYTimeFragments::min

Minutes (0 - 59).

5.43.2.3 NQ_UINT16 SYTimeFragments::hour

Hours of day (0 - 23).

5.43.2.4 NQ_UINT16 SYTimeFragments::day

Day of the month (1-31).

5.43.2.5 NQ_UINT16 SYTimeFragments::month

Month (1 - 12).

5.43.2.6 NQ_UINT16 SYTimeFragments::year

Year in YYYY format.

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.44 SYVolumeInformation Struct Reference

```
#include <sycommon.h>
```

Data Fields

- [NQ_UINT32 serialNumberLow](#)
- [NQ_UINT32 serialNumberHigh](#)
- [NQ_UINT32 creationTimeLow](#)
- [NQ_UINT32 creationTimeHigh](#)
- [NQ_UINT32 fileIdLow](#)
- [NQ_UINT32 fileIdHigh](#)
- [NQ_UINT32 blockSizeLow](#)
- [NQ_UINT32 blockSizeHigh](#)
- [NQ_UINT32 blocksPerUnitLow](#)
- [NQ_UINT32 blocksPerUnitHigh](#)
- [NQ_UINT32 totalUnitsLow](#)
- [NQ_UINT32 totalUnitsHigh](#)
- [NQ_UINT32 freeUnitsLow](#)
- [NQ_UINT32 freeUnitsHigh](#)

5.44.1 Detailed Description

Volume information structure. It is used for querying volume information from the file system.

Note

This structure is designated here for reference only and it must not be modified.

5.44.2 Field Documentation

5.44.2.1 **NQ_UINT32 SYVolumeInformation::serialNumberLow**

VolumeSerial number (low part).

5.44.2.2 **NQ_UINT32 SYVolumeInformation::serialNumberHigh**

VolumeSerial number (high part).

5.44.2.3 **NQ_UINT32 SYVolumeInformation::creationTimeLow**

Time of the volume creation (low part).

5.44.2.4 **NQ_UINT32 SYVolumeInformation::creationTimeHigh**

Time of the volume creation (high part).

5.44.2.5 **NQ_UINT32 SYVolumeInformation::fileSystemIdLow**

ID of the file system (low part).

5.44.2.6 **NQ_UINT32 SYVolumeInformation::fileSystemIdHigh**

ID of the file system (high part).

5.44.2.7 **NQ_UINT32 SYVolumeInformation::blockSizeLow**

Block size in bytes (low part).

5.44.2.8 **NQ_UINT32 SYVolumeInformation::blockSizeHigh**

Block size in bytes (high part).

5.44.2.9 **NQ_UINT32 SYVolumeInformation::blocksPerUnitLow**

Number of blocks in an allocation unit (low part).

5.44.2.10 **NQ_UINT32 SYVolumeInformation::blocksPerUnitHigh**

Number of blocks in an allocation unit (high part).

5.44.2.11 NQ_UINT32 SYVolumeInformation::totalUnitsLow

Number of units in volume (low part).

5.44.2.12 NQ_UINT32 SYVolumeInformation::totalUnitsHigh

Number of units in volume (high part).

5.44.2.13 NQ_UINT32 SYVolumeInformation::freeUnitsLow

Number of free units in volume (low part).

5.44.2.14 NQ_UINT32 SYVolumeInformation::freeUnitsHigh

Number of free units in volume (high part).

The documentation for this struct was generated from the following file:

- [sycommon.h](#)

5.45 UDFileAccessEvent Struct Reference

```
#include <udapi.h>
```

Data Fields

- const NQ_WCHAR * [fileName](#)
- const NQ_WCHAR * [newName](#)
- NQ_UINT32 [access](#)
- NQ_BOOL [before](#)
- NQ_UINT32 [rid](#)
- NQ_UINT32 [tid](#)
- NQ_UINT32 [sizeLow](#)
- NQ_UINT32 [sizeHigh](#)
- NQ_UINT32 [offsetLow](#)
- NQ_UINT32 [offsetHigh](#)
- NQ_UINT32 [infoLevel](#)

5.45.1 Detailed Description

Data structure for an event from File/Directory Access class. Depending on the event type, some members of this structure may be undefined

5.45.2 Field Documentation

5.45.2.1 const NQ_WCHAR* UDFileAccessEvent::fileName

file to access

5.45.2.2 `const NQ_WCHAR* UDFileAccessEvent::newName`

new file name for rename operation

5.45.2.3 `NQ_UINT32 UDFileAccessEvent::access`

this value is has different meaning for different events: for UD_LOG_FILE_CREATE, UD_LOG_FILE_DELETE and UD_LOG_FILE_RENAME - no meaning. for UD_LOG_FILE_OPEN and UD_LOG_FILE_CLOSE: 0x0 - read 0x1 - write 0x2 - read/write 0x3 - execute 0x8000 - delete 0xF - all for UD_LOG_FILE_ATTRIB: 0x1 - readonly 0x2 - hidden 0x4 - system 0x8 - volume 0x10 - directory 0x20 - archive 0x80 - normal

5.45.2.4 `NQ_BOOL UDFileAccessEvent::before`

whether event taken before of after the operation

5.45.2.5 `NQ_UINT32 UDFileAccessEvent::rid`

Unique user ID

5.45.2.6 `NQ_UINT32 UDFileAccessEvent::tid`

5.45.2.7 `NQ_UINT32 UDFileAccessEvent::sizeLow`

<Unique tree ID First 32 bits of file size or range size

5.45.2.8 `NQ_UINT32 UDFileAccessEvent::sizeHigh`

Last 32 bits of file size or range size

5.45.2.9 `NQ_UINT32 UDFileAccessEvent::offsetLow`

First 32 bits of a position in file

5.45.2.10 `NQ_UINT32 UDFileAccessEvent::offsetHigh`

Last 32 bits of a position in file

5.45.2.11 `NQ_UINT32 UDFileAccessEvent::infoLevel`

Information lever for query and set info

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.46 UDShareAccessEvent Struct Reference

```
#include <udapi.h>
```

Data Fields

- const [NQ_WCHAR](#) * [shareName](#)
- [NQ_BOOL](#) [ipc](#)
- [NQ_BOOL](#) [printQueue](#)
- [NQ_UINT32](#) [rid](#)
- [NQ_UINT32](#) [tid](#)

5.46.1 Detailed Description

Data structure for an event from Share Access class. Depending on the event type, some members of this structure may be undefined

5.46.2 Field Documentation

5.46.2.1 const [NQ_WCHAR](#)* [UDShareAccessEvent::shareName](#)

name of the share being accessed

5.46.2.2 [NQ_BOOL](#) [UDShareAccessEvent::ipc](#)

TRUE for IPC share

5.46.2.3 [NQ_BOOL](#) [UDShareAccessEvent::printQueue](#)

TRUE for a printer queue share

5.46.2.4 [NQ_UINT32](#) [UDShareAccessEvent::rid](#)

Unique user ID

5.46.2.5 [NQ_UINT32](#) [UDShareAccessEvent::tid](#)

Unique tree ID

The documentation for this struct was generated from the following file:

- [udapi.h](#)

5.47 UDUserAccessEvent Struct Reference

```
#include <udapi.h>
```

Data Fields

- [NQ_UINT32](#) [rid](#)

5.47.1 Detailed Description

Data structure for an event from User Access class. Defined user ID

5.47.2 Field Documentation

5.47.2.1 NQ_UINT32 UDUserAccessEvent::rid

Unique user ID

The documentation for this struct was generated from the following file:

- [udapi.h](#)

Chapter 6

File Documentation

6.1 amapi.h File Reference

```
#include "cmapi.h"
```

Data Structures

- struct [AMDomainA](#)
- struct [AMDomain](#)
- struct [AMCredentialsA](#)
- struct [AMCredentials](#)

Macros

- #define [AM_MAXSECURITYLEVEL](#) 4
- #define [AM_CRYPTER_NONE](#) 0
- #define [AM_CRYPTER_LM](#) 1
- #define [AM_CRYPTER_NTLM](#) 2
- #define [AM_CRYPTER_LM2](#) 3
- #define [AM_CRYPTER_NTLM2](#) 4
- #define [AM_MECH_NTLSSSP](#) 1
- #define [AM_MECH_KERBEROS](#) 2
- #define [AM_CREDENTIALS_PASSWORD_PLAINTEXT](#) 0
- #define [AM_CREDENTIALS_PASSWORD_MD4](#) 1

Functions

- [NQ_BOOL](#) [amStart](#) (void)
- void [amShutdown](#) (void)
- void [amClientDefineLevel](#) ([NQ_UINT](#) level, [NQ_UINT](#) crypter1, [NQ_UINT](#) crypter2, [NQ_UINT32](#) mechanisms)
- void [amSetNonSecureAuthentication](#) ([NQ_BOOL](#) enableNonSecureAuthentication)

6.2 ccapi.h File Reference

```
#include "syapi.h"
#include "cmapi.h"
#include <amcredentials.h>
#include <ccparams.h>
```

Data Structures

- struct [CCFileTime](#)
- struct [CCFindFileDataA](#)
- struct [CCFindFileData](#)
- struct [CCFileInfo_t](#)
- struct [CCFileInfo](#)
- struct [CCVolumeInfo](#)
- struct [CCStatistics](#)
- struct [CCServerInfo](#)
- struct [CCMountInfoA](#)
- struct [CCMountInfo](#)
- struct [CCNetShareItemA](#)
- struct [CCNetShareItem](#)
- struct [CCNetServerItem](#)
- struct [CCNetDomainItem](#)

Macros

- #define [CM_USERNAMELENGTH](#) 256
- #define [PASSWORD_LENGTH](#) UD_NQ_MAXPWDLEN
- #define [DOMAIN_LENGTH](#) (CM_NQ_HOSTNAMESIZE)
- #define [NOTIFY_SRCH_HANDLE_ERROR](#) 1
- #define [NOTIFY_FILE_HANDLE_ERROR](#) 2
- #define [CIFS_ATTR_READONLY](#) 0x01
- #define [CIFS_ATTR_HIDDEN](#) 0x02
- #define [CIFS_ATTR_SYSTEM](#) 0x04
- #define [CIFS_ATTR_VOLUME](#) 0x08
- #define [CIFS_ATTR_DIR](#) 0x10
- #define [CIFS_ATTR_ARCHIVE](#) 0x20
- #define [CIFS_ATTR_ALL](#) ([CIFS_ATTR_HIDDEN](#) | [CIFS_ATTR_SYSTEM](#) | [CIFS_ATTR_DIR](#))
- #define [FILE_AM_READ](#) 0
- #define [FILE_AM_WRITE](#) 1
- #define [FILE_AM_READ_WRITE](#) 2
- #define [FILE_AM_SPECIAL_MASK](#) 0x8000000
- #define [FILE_SM_COMPAT](#) 0
- #define [FILE_SM_EXCLUSIVE](#) 1
- #define [FILE_SM_DENY_WRITE](#) 2
- #define [FILE_SM_DENY_READ](#) 3
- #define [FILE_SM_DENY_NONE](#) 4
- #define [FILE_LCL_UNKNOWN](#) 0
- #define [FILE_LCL_SEQUENTIAL](#) 1
- #define [FILE_LCL_RANDOM](#) 2
- #define [FILE_LCL_MIXED](#) 3
- #define [FILE_CA_FAIL](#) 0
- #define [FILE_CA_CREATE](#) 1

- #define FILE_OA_FAIL 0
- #define FILE_OA_OPEN 1
- #define FILE_OA_TRUNCATE 2
- #define SEEK_FILE_BEGIN 0
- #define SEEK_FILE_CURRENT 1
- #define SEEK_FILE_END 2
- #define NQ_CC_AUTH_LM_AND_NTLM 1
- #define NQ_CC_AUTH_NTLM 2
- #define NQ_CC_AUTH_NTLM_V2 3
- #define NQ_CC_AUTH_SPNEGO_KERBEROS 4
- #define CC_LOGON_TYPE_REGULAR 0
- #define CC_LOGON_TYPE_ANONYMOUS 1
- #define CC_LOGON_TYPE_GUEST 2
- #define CC_CAPABILITY_DFS 0x00000001
- #define CC_CAPABILITY_LEASING 0x00000002
- #define CC_CAPABILITY_LARGE_MTU 0x00000004
- #define CC_CAPABILITY_MULTI_CHANNEL 0x00000008
- #define CC_CAPABILITY_PERSISTENT_HANDLES 0x00000010
- #define CC_CAPABILITY_DIRECTORY_LEASING 0x00000020
- #define CC_CAPABILITY_ENCRYPTION 0x00000040
- #define CC_SIGNING_ENABLED 0x0001
- #define CC_SIGNING_REQUIRED 0x0002

Typedefs

- typedef void(* CCServerSideCopyCallback)(NQ_STATUS status, NQ_UINT32 totalBytesWritten, NQ_UINT64 totalFileSize, void *context)
- typedef struct CCFindFileDataA FindFileDataA_t
- typedef struct CCFileInfo_t FileInfo_t

Enumerations

- enum CCSmbDialect {
CC_SMB100, CC_SMB202, CC_SMB210, CC_SMB300,
CC_SMB302, CC_SMB311 }

Functions

- NQ_BOOL ccInit (void(*fsdNotify)(NQ_INT eventId, NQ_ULONG param))
- void ccShutdown (void)
- NQ_BOOL ccIsInitialized (void)
- NQ_INT ccAddMountA (const NQ_CHAR *mountPoint, const NQ_CHAR *remotePath, NQ_BOOL connect)
- NQ_INT ccAddMount (const NQ_WCHAR *mountPoint, const NQ_WCHAR *remotePath, NQ_BOOL connect)
- NQ_INT ccAddMountWithCredsA (const NQ_CHAR *mountPoint, const NQ_CHAR *remotePath, NQ_BOOL connect, const AMCredentialsA *userCredentials)
- NQ_INT ccAddMountWithCreds (const NQ_WCHAR *mountPoint, const NQ_WCHAR *remotePath, NQ_BOOL connect, const AMCredentials *userCredentials)
- NQ_INT ccRemoveMountA (const NQ_CHAR *mountPoint)
- NQ_INT ccRemoveMount (const NQ_WCHAR *mountPoint)
- NQ_GUID ccGetServerIdA (const NQ_CHAR *mountpoint)
- NQ_GUID ccGetServerId (const NQ_WCHAR *mountpoint)
- NQ_BOOL ccGetMountInformationA (const NQ_CHAR *mountPoint, CCMountInfoA *mountPointInfo)

- [NQ_BOOL ccGetMountInformation](#) (const [NQ_WCHAR](#) *mountPoint, [CCMountInfo](#) *mountPointInfo)
- [NQ_HANDLE ccFindFirstFileA](#) (const [NQ_CHAR](#) *srchPath, [FindFileDataA_t](#) *findFileData, [NQ_BOOL](#) extractFirst)
- [NQ_HANDLE ccFindFirstFile](#) (const [NQ_WCHAR](#) *srchPath, [CCFindFileData](#) *findFileData, [NQ_BOOL](#) extractFirst)
- [NQ_BOOL ccFindNextFileA](#) ([NQ_HANDLE](#) handle, [FindFileDataA_t](#) *findFileData)
- [NQ_BOOL ccFindNextFile](#) ([NQ_HANDLE](#) handle, [CCFindFileData](#) *findFileData)
- [NQ_BOOL ccFindClose](#) ([NQ_HANDLE](#) handle)
- [NQ_BOOL ccCreateDirectoryA](#) (const [NQ_CHAR](#) *pathName)
- [NQ_BOOL ccCreateDirectory](#) (const [NQ_WCHAR](#) *pathName)
- [NQ_BOOL ccRemoveDirectoryA](#) (const [NQ_CHAR](#) *pathName)
- [NQ_BOOL ccRemoveDirectory](#) (const [NQ_WCHAR](#) *pathName)
- [NQ_HANDLE ccCreateFileA](#) (const [NQ_CHAR](#) *fileName, [NQ_INT](#) access, [NQ_INT](#) shareMode, [NQ_INT](#) locality, [NQ_BOOL](#) writeThrough, [NQ_UINT16](#) attributes, [NQ_INT](#) createAction, [NQ_INT](#) openAction)
- [NQ_HANDLE ccCreateFile](#) (const [NQ_WCHAR](#) *fileName, [NQ_INT](#) access, [NQ_INT](#) shareMode, [NQ_INT](#) locality, [NQ_BOOL](#) writeThrough, [NQ_UINT16](#) attributes, [NQ_INT](#) createAction, [NQ_INT](#) openAction)
- [NQ_BOOL ccDeleteFileA](#) (const [NQ_CHAR](#) *fileName)
- [NQ_BOOL ccDeleteFile](#) (const [NQ_WCHAR](#) *fileName)
- [NQ_BOOL ccMoveFileA](#) (const [NQ_CHAR](#) *oldFilename, const [NQ_CHAR](#) *newFilename)
- [NQ_BOOL ccMoveFile](#) (const [NQ_WCHAR](#) *oldFilename, const [NQ_WCHAR](#) *newFilename)
- [NQ_BOOL ccReadFile](#) ([NQ_HANDLE](#) hndl, [NQ_IOBufPos](#) buffer, [NQ_UINT](#) count, [NQ_UINT](#) *readSize)
- [NQ_BOOL ccReadFileAsync](#) ([NQ_HANDLE](#) hndl, [NQ_IOBufPos](#) buffer, [NQ_UINT](#) count, void *context, void(*callback)([NQ_STATUS](#), [NQ_UINT](#), void *))
- [NQ_BOOL ccWriteFile](#) ([NQ_HANDLE](#) hndl, [NQ_IOBufPos](#) buffer, [NQ_UINT](#) count, [NQ_UINT](#) *writtenSize)
- [NQ_BOOL ccWriteFileAsync](#) ([NQ_HANDLE](#) hndl, [NQ_IOBufPos](#) buffer, [NQ_UINT](#) count, void *context, void(*callback)([NQ_STATUS](#), [NQ_UINT](#), void *))
- [NQ_BOOL ccFlushFile](#) ([NQ_HANDLE](#) handle)
- [NQ_BOOL ccCloseHandle](#) ([NQ_HANDLE](#) handle)
- [NQ_BOOL ccGetFilePointer](#) ([NQ_HANDLE](#) handle, [NQ_UINT64](#) *offset)
- [NQ_BOOL ccSetFilePointer](#) ([NQ_HANDLE](#) handle, [NQ_INT64](#) *offset, [NQ_INT](#) method)
- [NQ_BOOL ccServerSideDataCopyA](#) ([NQ_CHAR](#) *srcPath, [NQ_CHAR](#) *dstPath, void *context, [CCServerSideCopyCallback](#) *callBack)
- [NQ_BOOL ccServerSideDataCopy](#) ([NQ_WCHAR](#) *srcPath, [NQ_WCHAR](#) *dstPath, void *context, [CCServerSideCopyCallback](#) *callBack)
- [NQ_BOOL ccGetDiskFreeSpaceA](#) (const [NQ_CHAR](#) *pathName, [NQ_UINT](#) *sectorsPerCluster, [NQ_UINT](#) *bytesPerSector, [NQ_UINT](#) *freeClusters, [NQ_UINT](#) *totalClusters, [NQ_UINT](#) *fsType, [NQ_UINT](#) *serialNumber)
- [NQ_BOOL ccGetDiskFreeSpace](#) (const [NQ_WCHAR](#) *pathName, [NQ_UINT](#) *sectorsPerCluster, [NQ_UINT](#) *bytesPerSector, [NQ_UINT](#) *freeClusters, [NQ_UINT](#) *totalClusters, [NQ_UINT](#) *fsType, [NQ_UINT](#) *serialNumber)
- [NQ_BOOL ccGetDiskFreeSpaceExA](#) (const [NQ_CHAR](#) *pathName, [NQ_UINT](#) *sectorsPerCluster, [NQ_UINT](#) *bytesPerSector, [NQ_UINT64](#) *freeClusters, [NQ_UINT64](#) *totalClusters, [NQ_UINT](#) *fsType, [NQ_UINT](#) *serialNumber)
- [NQ_BOOL ccGetDiskFreeSpaceEx](#) (const [NQ_WCHAR](#) *pathName, [CCVolumeInfo](#) *volumeInfo)
- [NQ_UINT32 ccGetFileAttributesA](#) (const [NQ_CHAR](#) *fileName)
- [NQ_BOOL ccGetFileAttributes](#) (const [NQ_WCHAR](#) *fileName, [NQ_UINT32](#) *attributes)
- [NQ_BOOL ccSetFileAttributesA](#) (const [NQ_CHAR](#) *fileName, [NQ_UINT32](#) attributes)
- [NQ_BOOL ccSetFileAttributes](#) (const [NQ_WCHAR](#) *fileName, [NQ_UINT32](#) attributes)
- [NQ_BOOL ccGetFileInformationByNameA](#) (const [NQ_CHAR](#) *fileName, [FileInfo_t](#) *fileInfo)
- [NQ_BOOL ccGetFileInformationByName](#) (const [NQ_WCHAR](#) *fileName, [CCFileInfo](#) *fileInfo)
- [NQ_BOOL ccSetFileSizeByNameA](#) (const [NQ_CHAR](#) *fileName, [NQ_UINT32](#) sizeLow, [NQ_UINT32](#) sizeHigh)
- [NQ_BOOL ccSetFileSizeByName](#) (const [NQ_WCHAR](#) *fileName, [NQ_UINT64](#) size)
- [NQ_BOOL ccSetFileSizeByHandle](#) ([NQ_HANDLE](#) handle, [NQ_UINT64](#) size)
- [NQ_BOOL ccGetFileTime](#) ([NQ_HANDLE](#) handle, [CCFileTime](#) *pCreation, [CCFileTime](#) *pLastAccess, [CCFileTime](#) *pLastWrite)

- [NQ_BOOL ccSetFileTime](#) ([NQ_HANDLE](#) handle, [CCFileTime](#) *pCreation, [CCFileTime](#) *pLastAccess, [CCFileTime](#) *pLastWrite)
- [NQ_BOOL ccGetFileSize](#) ([NQ_HANDLE](#) handle, [NQ_UINT64](#) *fileSize)
- [NQ_BOOL ccGetFileInformationByHandle](#) ([NQ_HANDLE](#) handle, [CCFileInfo](#) *fileInfo)
- [void ccConfigSetTimeout](#) ([NQ_UINT32](#) secs)
- [NQ_UINT32 ccConfigGetTimeout](#) (void)
- [NQ_BOOL ccSetGetBackupListTimeout](#) ([NQ_UINT32](#) seconds)
- [NQ_BOOL ccNetworkSetDefaultDomainA](#) (const [NQ_CHAR](#) *domain)
- [NQ_BOOL ccNetworkSetDefaultDomain](#) (const [NQ_WCHAR](#) *domain)
- [void ccNetworkCacheSet](#) ([NQ_UINT32](#) timeout)
- [const NQ_CHAR * ccNetworkGetDefaultDomainA](#) ()
- [const NQ_WCHAR * ccNetworkGetDefaultDomain](#) ()
- [NQ_HANDLE ccNetworkEnumerateDomains](#) (void)
- [CCNetDomainItem * ccNetworkGetNextDomainItem](#) ([NQ_HANDLE](#) handle)
- [NQ_HANDLE ccNetworkEnumerateServersA](#) (const [NQ_CHAR](#) *domain)
- [NQ_HANDLE ccNetworkEnumerateServers](#) (const [NQ_WCHAR](#) *domain)
- [CCNetServerItem * ccNetworkGetNextServerItem](#) ([NQ_HANDLE](#) handle)
- [NQ_HANDLE ccNetworkEnumerateSharesA](#) (const [NQ_CHAR](#) *server)
- [NQ_HANDLE ccNetworkEnumerateShares](#) (const [NQ_WCHAR](#) *server)
- [const NQ_CHAR * ccNetworkGetNextItemNameA](#) ([NQ_HANDLE](#) handle)
- [const NQ_WCHAR * ccNetworkGetNextItemName](#) ([NQ_HANDLE](#) handle)
- [CCNetShareItem * ccNetworkGetNextShareItem](#) ([NQ_HANDLE](#) handle)
- [NQ_BOOL ccNetworkGetShareInfoA](#) (const [NQ_CHAR](#) *server, const [NQ_CHAR](#) *share, [NQ_UINT16](#) *type, [NQ_CHAR](#) *remarkBuffer, [NQ_INT](#) bufferSize)
- [NQ_BOOL ccNetworkGetShareInfo](#) (const [NQ_WCHAR](#) *server, const [NQ_WCHAR](#) *share, [NQ_UINT16](#) *type, [NQ_WCHAR](#) *remarkBuffer, [NQ_INT](#) bufferSize)
- [NQ_BOOL ccNetworkCloseHandle](#) ([NQ_HANDLE](#) handle)
- [NQ_BOOL ccNetworkResetHandle](#) ([NQ_HANDLE](#) handle)
- [NQ_BOOL ccDomainJoinA](#) (const [NQ_CHAR](#) *domain, const [NQ_CHAR](#) *computer, const [AMCredentialsA](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainJoin](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [AMCredentials](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainJoinPersistent](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [AMCredentials](#) *admin, const [NQ_WCHAR](#) *dns)
- [NQ_BOOL ccDomainUpdateSecretA](#) (const [NQ_CHAR](#) *domain, const [NQ_CHAR](#) *computer, const [AMCredentialsA](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainUpdateSecret](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [AMCredentials](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainLeaveA](#) (const [NQ_CHAR](#) *domain, const [NQ_CHAR](#) *computer, const [AMCredentialsA](#) *admin)
- [NQ_BOOL ccDomainLeave](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [AMCredentials](#) *admin)
- [NQ_BOOL ccDomainLogonA](#) (const [NQ_CHAR](#) *domain, const [NQ_CHAR](#) *computer, const [AMCredentialsA](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccDomainLogon](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *computer, const [AMCredentials](#) *admin, [NQ_BYTE](#) secret[16])
- [NQ_BOOL ccGetSigningPolicy](#) (void)
- [void ccSetSigningPolicy](#) ([NQ_BOOL](#) enable)
- [NQ_STATUS ccSetMinSecurityLevel](#) ([NQ_INT](#) level)
- [NQ_INT ccGetMinSecurityLevel](#) (void)
- [NQ_STATUS ccSetMaxSecurityLevel](#) ([NQ_INT](#) level)
- [NQ_INT ccGetMaxSecurityLevel](#) (void)
- [NQ_BOOL ccSetExclusiveAccessToFileA](#) ([NQ_CHAR](#) *fileName, [NQ_BOOL](#) makeExclusive)
- [NQ_BOOL ccSetExclusiveAccessToFile](#) ([NQ_WCHAR](#) *fileName, [NQ_BOOL](#) makeExclusive)
- [NQ_BOOL ccIsExclusiveAccessToFileA](#) ([NQ_CHAR](#) *fileName)

- [NQ_BOOL ccIsExclusiveAccessToFile](#) ([NQ_WCHAR](#) *fileName)
- [NQ_BOOL ccSetSmbDialect](#) ([CCSmbDialect](#) dialect, [NQ_BOOL](#) setActive)
- [NQ_BOOL ccCifsGetSMB1Support](#) (void)
- void [ccCifsSetSMB1Support](#) ([NQ_BOOL](#) isSupport)
- [NQ_BOOL ccResetCredentailsA](#) (const [NQ_CHAR](#) *mountpoint)
- [NQ_BOOL ccResetCredentails](#) (const [NQ_WCHAR](#) *mountpoint)
- [NQ_BOOL ccValidateUserA](#) ([NQ_CHAR](#) *server)
- [NQ_BOOL ccValidateUser](#) ([NQ_WCHAR](#) *server)
- [NQ_BOOL ccValidateUserByCredentialsA](#) ([NQ_CHAR](#) *server, const [AMCredentialsA](#) *userCredentials)
- [NQ_BOOL ccValidateUserByCredentials](#) ([NQ_WCHAR](#) *server, const [AMCredentials](#) *userCredentials)
- [NQ_UINT32 ccGetNTStatus](#) (void)
- [NQ_UINT32 ccErrorToSmbStatus](#) ([NQ_STATUS](#) nqError)
- [NQ_BOOL ccTransportRestartRecieveThread](#) ()
- void [ccDump](#) (void)
- void [ccCloseAllConnections](#) (void)
- void [ccCloseHiddenConnections](#) (void)
- void [ccThreadSubscribe](#) (void)
- void [ccThreadUnsubscribe](#) (void)

6.3 cmapi.h File Reference

```
#include "syapi.h"
#include <cmcommon.h>
#include <cmparams.h>
#include <cmutils.h>
#include <cmlist.h>
#include <cmmemory.h>
#include <cmselfip.h>
#include <cmresolver.h>
#include <cmthread.h>
#include <cmunicod.h>
#include <cmstrparse.h>
#include <cmstring.h>
#include <cmnbapi.h>
#include <cmfsapi.h>
#include <cmrpcdef.h>
#include <cmsdescr.h>
#include <cmvalida.h>
#include <cmtrace.h>
#include <cmbufman.h>
#include <udconfig.h>
```

Data Structures

- struct [CMProductInfo](#)
- struct [CMCodepage](#)
- struct [CMResolverRegisteredMethodDescription](#)
- struct [CMCrypterList](#)

Macros

- #define [NQ_CLIENT](#) 0x1
- #define [NQ_SERVER](#) 0x2
- #define [NQ_NDDAEMON](#) 0x4
- #define [NQ_RESOLVER_IPV4](#) 4
- #define [NQ_RESOLVER_IPV6](#) 6
- #define [NQ_RESOLVER_NONE](#) 0
- #define [NQ_RESOLVER_DNS](#) 1
- #define [NQ_RESOLVER_NETBIOS](#) 2
- #define [NQ_RESOLVER_EXTERNAL_METHOD](#) 5
- #define [NQ_RESOLVER_DNS_DC](#) 8
- #define [NQ_RESOLVER_NETBIOS_DC](#) 10
- #define [NQ_RESOLVER_WSD](#) 12
- #define [CM_PRODUCT_STRING_LEN](#) 20

Typedefs

- typedef void(* [CMAbstractHasher](#))(const [NQ_BYTE](#) *dataIn, [NQ_BYTE](#) *dataOut, [NQ_COUNT](#) length)
- typedef void(* [CMAbstractCipher](#))(const CMBlob *key, const CMBlob *key1, const CMIOBlob dataFragments[], [NQ_COUNT](#) numFragments, [NQ_BYTE](#) *buffer, [NQ_COUNT](#) bufferSize)
- typedef void(* [CMAbstractIOCipher](#))(const CMBlob *key, const CMBlob *key1, const CMIOBlob dataFragments[], [NQ_COUNT](#) numFragments, [NQ_BYTE](#) *buffer, [NQ_COUNT](#) bufferSize)
- typedef void(* [CMAbstractHasher512](#))(const CMBlob *key, const CMBlob *key1, const CMIOBlob dataFragments[], [NQ_COUNT](#) numFragments, [NQ_IOBufPos](#) buffer, [NQ_COUNT](#) bufferSize, [NQ_BYTE](#) *ctxBuff)
- typedef void(* [CMAbstractCcmEncryption](#))(const CMBlob *key, const CMBlob *key1, const CMIOBlob *prefix, CMIOBlob *message, [NQ_BYTE](#) *auth)
- typedef [NQ_BOOL](#)(* [CMAbstractCcmDecryption](#))(const CMBlob *key, const CMBlob *key1, const CMIOBlob *prefix, CMIOBlob *message, const [NQ_BYTE](#) *auth)
- typedef void(* [CMAbstractGcmEncryption](#))(const CMBlob *key, const CMBlob *key1, const CMIOBlob *prefix, CMIOBlob *message, [NQ_BYTE](#) *auth, [NQ_BYTE](#) *keyBuffer, [NQ_BYTE](#) *encMsgBuffer)
- typedef [NQ_BOOL](#)(* [CMAbstractGcmDecryption](#))(const CMBlob *key, const CMBlob *key1, const CMIOBlob *prefix, CMIOBlob *message, const [NQ_BYTE](#) *auth, [NQ_BYTE](#) *keyBuffer, [NQ_BYTE](#) *msgBuffer)
- typedef [NQ_INT](#)(* [CMResolverNameToIpA](#))(const [NQ_CHAR](#) *name, void *ip, [NQ_COUNT](#) index)
- typedef [NQ_INT](#)(* [CMResolverNameToIp](#))(const [NQ_WCHAR](#) *name, void *ip, [NQ_COUNT](#) index)
- typedef [NQ_BOOL](#)(* [CMResolverIpToNameA](#))(const [NQ_CHAR](#) *name, const void *ip, [NQ_INT](#) ipType)
- typedef [NQ_BOOL](#)(* [CMResolverIpToName](#))(const [NQ_WCHAR](#) *name, const void *ip, [NQ_INT](#) ipType)

Functions

- [NQ_STATUS](#) [cmInit](#) ([NQ_UINT32](#) component)
- void [cmExit](#) ([NQ_UINT32](#) component)
- [NQ_BOOL](#) [cmCodepageAdd](#) (const [CMCodepage](#) *codePage)
- [NQ_BOOL](#) [cmCodepageRemove](#) (const [CMCodepage](#) *codePage)
- void [cmWideCharToMultiByte](#) ([NQ_CHAR](#) *strMultiByte, const [NQ_WCHAR](#) *strWideChar)
- void [cmMultiByteToWideChar](#) ([NQ_WCHAR](#) *strWideChar, const [NQ_CHAR](#) *strMultiByte)
- void [cmSetExternalCrypters](#) (const [CMCrypterList](#) *crypters)
- void [cmResetExternalCrypters](#) (void)
- void [cmResolverSetExternalA](#) ([CMResolverNameToIpA](#) nameToIp, [CMResolverIpToNameA](#) ipToName)
- void [cmResolverSetExternal](#) ([CMResolverNameToIp](#) nameToIp, [CMResolverIpToName](#) ipToName)
- const [NQ_WCHAR](#) * [cmResolverGetHostName](#) (const [NQ_IPADDRESS](#) *ip)
- const [NQ_IPADDRESS](#) * [cmResolverGetHostIps](#) (const [NQ_WCHAR](#) *dnsList, const [NQ_WCHAR](#) *host, [NQ_INT](#) *numIps)

- const [NQ_WCHAR](#) * [cmResolverGetDCName](#) (const [NQ_WCHAR](#) *domain, const [NQ_WCHAR](#) *dnsList, [NQ_INT](#) *numDCs)
- void [cmResolverEnableMethod](#) ([NQ_INT](#) type, [NQ_BOOL](#) unicast, [NQ_BOOL](#) multicast)
- [NQ_BOOL](#) [cmResolverRegisterExternalMethod](#) (const [CMResolverRegisteredMethodDescription](#) *p-Method)
- [NQ_BOOL](#) [cmResolverUpdateExternalMethodsPriority](#) ([NQ_INT](#) requiredPriority)
- void [cmDnsSetServersA](#) (const [NQ_CHAR](#) *servers)
- void [cmDnsSetServers](#) (const [NQ_WCHAR](#) *servers)
- [NQ_IPADDRESS](#) * [cmDnsGetServer](#) ([NQ_COUNT](#) dnsID)
- [NQ_COUNT](#) [cmDnsGetNumDnsServers](#) (void)
- void [cmNetBiosSetWinsA](#) (const [NQ_CHAR](#) *servers)
- void [cmNetBiosSetWins](#) (const [NQ_WCHAR](#) *servers)
- [NQ_IPADDRESS4](#) [cmNetBiosGetWins](#) ([NQ_COUNT](#) winsID)
- [NQ_COUNT](#) [cmNetBiosGetNumWinsServers](#) (void)
- [NQ_STATUS](#) [cmDnsSetDomainA](#) (const [NQ_CHAR](#) *domainName)
- [NQ_STATUS](#) [cmDnsSetDomain](#) (const [NQ_WCHAR](#) *domainName)
- [NQ_STATUS](#) [cmNetBiosNameReload](#) (void)
- [CMProductInfo](#) * [cmGetCurrentProductInformation](#) (void)

6.4 csapi.h File Reference

```
#include "cmapi.h"
```

Data Structures

- struct [FileDataA_t](#)
- struct [CSFileData](#)
- struct [EVCSShareConnectionEntry](#)
- struct [EVCSOpenFileEntry](#)

Functions

- [NQ_STATUS](#) [csStart](#) (void)
- void [csStop](#) (void)
- [NQ_INT](#) [csAddShareA](#) (const [NQ_CHAR](#) *name, const [NQ_CHAR](#) *path, [NQ_BOOL](#) printQueue, const [NQ_CHAR](#) *comment, const [NQ_CHAR](#) *reserved)
- [NQ_INT](#) [csAddShare](#) (const [NQ_WCHAR](#) *name, const [NQ_WCHAR](#) *path, [NQ_BOOL](#) printQueue, const [NQ_WCHAR](#) *comment, const [NQ_CHAR](#) *reserved)
- [NQ_INT](#) [csRemoveShareA](#) (const [NQ_CHAR](#) *name)
- [NQ_INT](#) [csRemoveShare](#) (const [NQ_WCHAR](#) *name)
- [NQ_INT](#) [csSetShareEncryptionA](#) (const [NQ_CHAR](#) *name, [NQ_BOOL](#) isEncrypted)
- [NQ_INT](#) [csSetShareEncryption](#) (const [NQ_WCHAR](#) *name, [NQ_BOOL](#) isEncrypted)
- [NQ_BOOL](#) [csEnumerateOpenFilesA](#) ([NQ_UINT](#) index, [FileDataA_t](#) *fileData)
- [NQ_BOOL](#) [csEnumerateOpenFiles](#) ([NQ_UINT](#) index, [CSFileData](#) *fileData)
- [NQ_STATUS](#) [csCleanUserServerConnectionsA](#) (const [NQ_CHAR](#) *name, [NQ_BOOL](#) isDomainUser)
- [NQ_STATUS](#) [csCleanUserServerConnections](#) (const [NQ_WCHAR](#) *name, [NQ_BOOL](#) isDomainUser)
- void [csSetAnonymousAllowed](#) ([NQ_BOOL](#) on)
- [NQ_COUNT](#) [evcsEnumerateConnectedShares](#) ([NQ_COUNT](#) maxEntries, [EVCSShareConnectionEntry](#) *buffer)
- [NQ_COUNT](#) [evcsEnumerateOpenFiles](#) ([NQ_COUNT](#) maxEntries, [EVCSOpenFileEntry](#) *buffer)

6.5 ndapi.h File Reference

```
#include "cmapi.h"
#include <ndadaptr.h>
```

Functions

- [NQ_STATUS ndStart](#) (SYSemaphore *sem)
- void [ndStop](#) (void)
- void [ndNotifyConfigurationChange](#) (NDAdapterInfo *adapter)

6.6 nqapi.h File Reference

```
#include "cmapi.h"
```

Macros

- #define [CM_NB_GROUP](#) 0x80
- #define [CM_NB_UNIQUE](#) 0x40
- #define [CM_NB_POSTFIX_WORKSTATION](#) 0x0
- #define [CM_NB_POSTFIX_SERVER](#) 0x20
- #define [CM_NB_POSTFIX_DOMAINMASTERBROWSER](#) 0x1b
- #define [CM_NB_POSTFIX_DOMAINCONTROLLER](#) 0x1c
- #define [CM_NB_POSTFIX_MASTERBROWSER](#) 0x1d

Functions

- [NQ_INT nqStart](#) (void)
- [NQ_INT nqStop](#) (void)
- [NQ_STATUS nqRegisterNetBiosName](#) (const [NQ_CHAR](#) *name)
- [NQ_STATUS nqReleaseNetBiosName](#) (const [NQ_CHAR](#) *name)
- [NQ_STATUS nqRegisterNetBiosNameEx](#) (const [NQ_CHAR](#) *name, const [NQ_BYTE](#) flags)
- [NQ_STATUS nqReleaseNetBiosNameEx](#) (const [NQ_CHAR](#) *name, const [NQ_BYTE](#) flags)
- void [nqEnableTraceLog](#) ([NQ_BOOL](#) on)

6.7 syapi.h File Reference

```
#include "udapi.h"
#include "sycommon.h"
#include <syinclud.h>
#include <sypltfrm.h>
#include <syopsyst.h>
#include <sytrace.h>
```

Functions

- [NQ_BOOL syInit](#) (void)
- void [syStop](#) (void)

6.8 sycommon.h File Reference

Data Structures

- struct [SYFileInformation](#)
- struct [SYVolumeInformation](#)
- struct [SYTimeFragments](#)
- struct [SYDeviceMode](#)
- struct [SYPrinterDriver](#)
- struct [SYPrinterInfo](#)
- struct [SYPrintJobInfo](#)
- struct [SYPrintSize](#)
- struct [SYPrintRect](#)
- struct [SYPrintFormInfo](#)

Macros

- #define [SY_ATTR_READONLY](#) 0x001
- #define [SY_ATTR_HIDDEN](#) 0x002
- #define [SY_ATTR_SYSTEM](#) 0x004
- #define [SY_ATTR_VOLUME](#) 0x008
- #define [SY_ATTR_DIRECTORY](#) 0x010
- #define [SY_ATTR_ARCHIVE](#) 0x020
- #define [SY_ATTR_DEVICE](#) 0x040
- #define [SY_ATTR_NORMAL](#) 0x080
- #define [SY_ATTR_TEMPORARY](#) 0x100
- #define [SY_ATTR_SPARSE_FILE](#) 0x200
- #define [SY_ATTR_REPARSE_POINT](#) 0x400
- #define [SY_ATTR_COMPRESSED](#) 0x800
- #define [SY_ATTR_OFFLINE](#) 0x1000
- #define [SY_ATTR_NOT_CONTENT_INDEXED](#) 0x2000
- #define [SY_ATTR_ENCRYPTED](#) 0x4000
- #define [SY_PRINTEROSVERSION_WIN](#) 3
- #define [SY_DEVICEMODE_ORIENTATION](#) 0x00000001
- #define [SY_DEVICEMODE_PAPERSIZE](#) 0x00000002
- #define [SY_DEVICEMODE_PAPERLENGTH](#) 0x00000004
- #define [SY_DEVICEMODE_PAPERWIDTH](#) 0x00000008
- #define [SY_DEVICEMODE_SCALE](#) 0x00000010
- #define [SY_DEVICEMODE_POSITION](#) 0x00000020
- #define [SY_DEVICEMODE_NUP](#) 0x00000040
- #define [SY_DEVICEMODE_COPIES](#) 0x00000080
- #define [SY_DEVICEMODE_DEFAULTSOURCE](#) 0x00000100
- #define [SY_DEVICEMODE_PRINTQUALITY](#) 0x00000400
- #define [SY_DEVICEMODE_COLOR](#) 0x00000800
- #define [SY_DEVICEMODE_DUPLEX](#) 0x00001000
- #define [SY_DEVICEMODE_YRESOLUTION](#) 0x00002000
- #define [SY_DEVICEMODE_TTOPTION](#) 0x00004000
- #define [SY_DEVICEMODE_COLLATE](#) 0x00008000
- #define [SY_DEVICEMODE_FORMNAME](#) 0x00010000
- #define [SY_DEVICEMODE_LOGPIXELS](#) 0x00020000
- #define [SY_DEVICEMODE_BITSPERPEL](#) 0x00040000
- #define [SY_DEVICEMODE_PELWIDTH](#) 0x00080000
- #define [SY_DEVICEMODE_PELHEIGHT](#) 0x00100000
- #define [SY_DEVICEMODE_DISPLAYFLAGS](#) 0x00200000

- #define SY_DEVICEMODE_DISPLAYFREQUENCY 0x00400000
- #define SY_DEVICEMODE_ICMMETHOD 0x00800000
- #define SY_DEVICEMODE_ICMINTENT 0x01000000
- #define SY_DEVICEMODE_MEDIATYPE 0x02000000
- #define SY_DEVICEMODE_DITHERTYPE 0x04000000
- #define SY_DEVICEMODE_PANNINGWIDTH 0x08000000
- #define SY_DEVICEMODE_PANNINGHEIGHT 0x10000000
- #define SY_PRINTERSTATUS_PAUSED 0x00000001
- #define SY_PRINTERSTATUS_ERROR 0x00000002
- #define SY_PRINTERSTATUS_PENDINGDELETION 0x00000004
- #define SY_PRINTERSTATUS_PAPERJAM 0x00000008
- #define SY_PRINTERSTATUS_PAPEROUT 0x00000010
- #define SY_PRINTERSTATUS_MANUALFEED 0x00000020
- #define SY_PRINTERSTATUS_PAPERPROBLEM 0x00000040
- #define SY_PRINTERSTATUS_OFFLINE 0x00000080
- #define SY_PRINTERSTATUS_IOACTIVE 0x00000100
- #define SY_PRINTERSTATUS_BUSY 0x00000200
- #define SY_PRINTERSTATUS_PRINTING 0x00000400
- #define SY_PRINTERSTATUS_OUTPUTBINFULL 0x00000800
- #define SY_PRINTERSTATUS_NOTAVAILABLE 0x00001000
- #define SY_PRINTERSTATUS_WAITING 0x00002000
- #define SY_PRINTERSTATUS_PROCESSING 0x00004000
- #define SY_PRINTERSTATUS_INITIALIZING 0x00008000
- #define SY_PRINTERSTATUS_WARMINGUP 0x00010000
- #define SY_PRINTERSTATUS_TONERLOW 0x00020000
- #define SY_PRINTERSTATUS_NOTONER 0x00040000
- #define SY_PRINTERSTATUS_PAGEPUNT 0x00080000
- #define SY_PRINTERSTATUS_USERINTERVENTION 0x00100000
- #define SY_PRINTERSTATUS_OUTOFMEMORY 0x00200000
- #define SY_PRINTERSTATUS_DOOROPEN 0x00400000
- #define SY_PRINTERSTATUS_SERVERUNKNOWN 0x00800000
- #define SY_PRINTERSTATUS_POWERSAVE 0x01000000
- #define SY_PRINTERATTR_QUEUED 0x0001
- #define SY_PRINTERATTR_DIRECT 0x0002
- #define SY_PRINTERATTR_DEFAULT 0x0004
- #define SY_PRINTERATTR_SHARED 0x0008
- #define SY_PRINTERATTR_NETWORK 0x0010
- #define SY_PRINTERATTR_RESERVED 0x0020 /* */
- #define SY_PRINTERATTR_LOCAL 0x0040
- #define SY_PRINTERATTR_DEVQUERY 0x0080
- #define SY_PRINTERATTR_KEEPCJOBS 0x0100
- #define SY_PRINTERATTR_LIFO 0x0200
- #define SY_PRINTERATTR_OFFLINE 0x0400
- #define SY_PRINTERATTR_BIDI 0x0800
- #define SY_PRINTERATTR_RAWONLY 0x1000
- #define SY_PRINTERATTR_PUBLISHED 0x2000
- #define SY_PRINTERATTR_EXPAND 0x4000
- #define SY_PRINTERATTR_CONTAINER 0x8000
- #define SY_PRINTERATTR_ICON1 0x10000
- #define SY_PRINTERATTR_ICON2 0x20000
- #define SY_PRINTERATTR_ICON3 0x40000
- #define SY_PRINTERATTR_ICON4 0x80000
- #define SY_PRINTERATTR_ICON5 0x100000
- #define SY_PRINTERATTR_ICON6 0x200000
- #define SY_PRINTERATTR_ICON7 0x400000

- `#define SY_PRINTERATTR_ICON8 0x800000`
- `#define SY_PRINTJOBSTATUS_PAUSED 0x00000001`
- `#define SY_PRINTJOBSTATUS_ERROR 0x00000002`
- `#define SY_PRINTJOBSTATUS_DELETING 0x00000004`
- `#define SY_PRINTJOBSTATUS_SPOOLING 0x00000008`
- `#define SY_PRINTJOBSTATUS_PRINTING 0x00000010`
- `#define SY_PRINTJOBSTATUS_OFFLINE 0x00000020`
- `#define SY_PRINTJOBSTATUS_PAPERPROBLEM 0x00000040`
- `#define SY_PRINTJOBSTATUS_PRINTED 0x00000080`
- `#define SY_PRINTJOBSTATUS_DELETED 0x00000100`
- `#define SY_PRINTJOBSTATUS_BLOCKED 0x00000200`
- `#define SY_PRINTJOBSTATUS_USERINTERVENTION 0x00000400`
- `#define SY_PRINTJOBSTATUS_RESTART 0x00000800`
- `#define SY_PRINTJOB_MINPRIORITY 1`
- `#define SY_PRINTJOB_MAXPRIORITY 99`
- `#define SY_PRINTJOB_DEFPRIORITY 1`
- `#define SY_PRINTERCONTROL_PAUSE 1`
- `#define SY_PRINTERCONTROL_RESUME 2`
- `#define SY_PRINTERCONTROL_PURGE 3`
- `#define SY_PRINTERCONTROL_SETSTATUS 4`
- `#define SY_PRINTJOB_COM_PAUSE 1`
- `#define SY_PRINTJOB_COM_RESUME 2`
- `#define SY_PRINTJOB_COM_CANCEL 3`
- `#define SY_PRINTJOB_COM_RESTART 4`
- `#define SY_PRINTJOB_COM_DELETE 5`
- `#define SY_PRINTJOB_COM_SENDDTOPRINTER 6`
- `#define SY_PRINTJOB_COM_LASTPAGEEJECTED 7`
- `#define SY_PRINTFORMFLAG_USER 0x00000000`
- `#define SY_PRINTFORMFLAG_BUILTIN 0x00000001`
- `#define SY_PRINTFORMFLAG_PRINTER 0x00000002`

6.9 udapi.h File Reference

```
#include <udparams.h>
#include <sycompil.h>
```

Data Structures

- struct [NQ_IPADDRESSVR](#)
- struct [NQ_UINT64](#)
- struct [NQ_INT64](#)
- struct [NQ_IOVEC](#)
- struct [NQ_IOVECBUF](#)
- struct [NQ_IOVecBuffer](#)
- struct [CMIOVecBlob](#)
- struct [NQ_IPADDRESS](#)
- struct [UDFileAccessEvent](#)
- struct [UDShareAccessEvent](#)
- struct [UDUserAccessEvent](#)

Macros

- #define UD_DNS_SERVERSTRINGSIZE (16 * sizeof(NQ_WCHAR) * UD_NQ_MAXDNSSERVERS * UD_NS_MAXADAPTERS)
- #define NQ_IOBuf NQ_BYTE*
- #define NQ_IOBuffer NQ_IOVecBuffer *
- #define CMIOBufBlob CMIOVecBlob
- #define SMB_DOSERR(_class, _code) (((NQ_UINT32)_code << 16) | _class)
- #define CLASS_ERR_MODULE(_code) ((NQ_UINT32)_code << 16)
- #define IS_DOS_ERROR(err) ((err & CLASS_ERR_MODULE(1)) == CLASS_ERR_MODULE(1))
- #define SMB_ERRsuccess 0x00000000
- #define DOS_ERRbadfunc SMB_DOSERR(1, 1)
- #define DOS_ERRbadfile SMB_DOSERR(1, 2)
- #define DOS_ERRbadpath SMB_DOSERR(1, 3)
- #define DOS_ERRnofids SMB_DOSERR(1, 4)
- #define DOS_ERRnoaccess SMB_DOSERR(1, 5)
- #define DOS_ERRbadfid SMB_DOSERR(1, 6)
- #define DOS_ERRbadmcb SMB_DOSERR(1, 7)
- #define DOS_ERRnomem SMB_DOSERR(1, 8)
- #define DOS_ERRbadmem SMB_DOSERR(1, 9)
- #define DOS_ERRbadenv SMB_DOSERR(1, 10)
- #define DOS_ERRbadformat SMB_DOSERR(1, 11)
- #define DOS_ERRbadaccess SMB_DOSERR(1, 12)
- #define DOS_ERRbaddata SMB_DOSERR(1, 13)
- #define DOS_ERRbaddrive SMB_DOSERR(1, 15)
- #define DOS_ERRremcd SMB_DOSERR(1, 16)
- #define DOS_ERRdiffdevice SMB_DOSERR(1, 17)
- #define DOS_ERRnofiles SMB_DOSERR(1, 18)
- #define DOS_ERRbadshare SMB_DOSERR(1, 32)
- #define DOS_ERRlock SMB_DOSERR(1, 33)
- #define DOS_ERRdentsupportipc SMB_DOSERR(1, 66)
- #define DOS_ERRnoshare SMB_DOSERR(1, 67)
- #define DOS_ERRfileexists SMB_DOSERR(1, 80)
- #define DOS_ERRbaddirectory SMB_DOSERR(1, 87)
- #define DOS_ERRinsufficientbuffer SMB_DOSERR(1, 122)
- #define DOS_ERRinvalidname SMB_DOSERR(1, 123)
- #define DOS_ERRdirnotempty SMB_DOSERR(1, 145)
- #define DOS_ERRalreadyexists SMB_DOSERR(1, 183)
- #define DOS_ERRbadpipe SMB_DOSERR(1, 230)
- #define DOS_ERRpipebusy SMB_DOSERR(1, 231)
- #define DOS_ERRpipeclosing SMB_DOSERR(1, 232)
- #define DOS_ERRnotconnected SMB_DOSERR(1, 233)
- #define DOS_ERRmoredata SMB_DOSERR(1, 234)
- #define IS_SRV_ERROR(err) ((err & CLASS_ERR_MODULE(2)) == CLASS_ERR_MODULE(2))
- #define SRV_ERRerror SMB_DOSERR(2, 1)
- #define SRV_ERRbadpw SMB_DOSERR(2, 2)
- #define SRV_ERRaccess SMB_DOSERR(2, 4)
- #define SRV_ERRinvtid SMB_DOSERR(2, 5)
- #define SRV_ERRinvnetname SMB_DOSERR(2, 6)
- #define SRV_ERRinvdevice SMB_DOSERR(2, 7)
- #define SRV_ERRqfull SMB_DOSERR(2, 49)
- #define SRV_ERRqtoobig SMB_DOSERR(2, 50)
- #define SRV_ERRqeof SMB_DOSERR(2, 51)
- #define SRV_ERRinvfid SMB_DOSERR(2, 52)
- #define SRV_ERRsmbcmd SMB_DOSERR(2, 64)

- #define SRV_ERRsrverror SMB_DOSERR(2, 65)
- #define SRV_ERRfilespecs SMB_DOSERR(2, 67)
- #define SRV_ERRbadpermits SMB_DOSERR(2, 69)
- #define SRV_ERRsetattrmode SMB_DOSERR(2, 71)
- #define SRV_ERRpaused SMB_DOSERR(2, 81)
- #define SRV_ERRmsgoff SMB_DOSERR(2, 82)
- #define SRV_ERRnoroom SMB_DOSERR(2, 83)
- #define SRV_ERRrmuns SMB_DOSERR(2, 87)
- #define SRV_ERRtimeout SMB_DOSERR(2, 88)
- #define SRV_ERRnoresource SMB_DOSERR(2, 89)
- #define SRV_ERRtoomanyuids SMB_DOSERR(2, 90)
- #define SRV_ERRinvuid SMB_DOSERR(2, 91)
- #define SRV_ERRusempx SMB_DOSERR(2, 250)
- #define SRV_ERRusestd SMB_DOSERR(2, 251)
- #define SRV_ERRcontmpx SMB_DOSERR(2, 252)
- #define SRV_ERRnosupport SMB_DOSERR(2, 65535)
- #define IS_HRD_ERROR(err) ((err & CLASS_ERR_MODULE(3)) == CLASS_ERR_MODULE(3))
- #define HRD_ERRnowrite SMB_DOSERR(3, 19)
- #define HRD_ERRbadunit SMB_DOSERR(3, 20)
- #define HRD_ERRnotready SMB_DOSERR(3, 21)
- #define HRD_ERRbadcmd SMB_DOSERR(3, 22)
- #define HRD_ERRdata SMB_DOSERR(3, 23)
- #define HRD_ERRbadreq SMB_DOSERR(3, 24)
- #define HRD_ERRseek SMB_DOSERR(3, 25)
- #define HRD_ERRbadmedia SMB_DOSERR(3, 26)
- #define HRD_ERRbadsector SMB_DOSERR(3, 27)
- #define HRD_ERRnopaper SMB_DOSERR(3, 28)
- #define HRD_ERRwrite SMB_DOSERR(3, 29)
- #define HRD_ERRread SMB_DOSERR(3, 30)
- #define HRD_ERRgeneral SMB_DOSERR(3, 31)
- #define HRD_ERRbadshare SMB_DOSERR(3, 32)
- #define HRD_ERRlock SMB_DOSERR(3, 33)
- #define HRD_ERRwrongdisk SMB_DOSERR(3, 34)
- #define HRD_ERRFCBUnavail SMB_DOSERR(3, 35)
- #define HRD_ERRsharebufexc SMB_DOSERR(3, 36)
- #define HRD_ERRdiskfull SMB_DOSERR(3, 39)
- #define SMB_STATUS_OK 0xc0000000
- #define SMB_STATUS_INVALID 0xffffffff
- #define SMB_STATUS_SUCCESS 0x00000000
- #define SMB_STATUS_ENUMDIR 0x0000010c
- #define SMB_STATUS_PENDING 0x00000103
- #define SMB_STATUS_MORE_ENTRIES 0x00000105
- #define SMB_STATUS_SOME_UNMAPPED 0x00000107
- #define SMB_STATUS_NOTIFY_CLEANUP 0x0000010b
- #define SMB_STATUS_INSUFFICIENT_BUFFER 0x0000007a
- #define SMB_STATUS_UNKNOWN_PRINTER_DRIVER 0x00000705
- #define SMB_STATUS_INVALID_DATATYPE 0x0000070c
- #define SMB_STATUS_BUFFER_OVERFLOW 0x80000005
- #define SMB_STATUS_NO_MORE_FILES 0x80000006
- #define SMB_STATUS_DEVICE_OFF_LINE 0x80000010
- #define SMB_STATUS_NO_MORE_ENTRIES 0x8000001a
- #define SMB_STATUS_STOPPED_ON_SYMLINK 0x8000002D
- #define SMB_STATUS_UNSUCCESSFUL 0xc0000001
- #define SMB_STATUS_NOT_IMPLEMENTED 0xc0000002
- #define SMB_STATUS_INVALID_INFO_CLASS 0xc0000003

- #define SMB_STATUS_INFO_LENGTH_MISMATCH 0xc0000004
- #define SMB_STATUS_ACCESS_VIOLATION 0xc0000005
- #define SMB_STATUS_IN_PAGE_ERROR 0xc0000006
- #define SMB_STATUS_PAGEFILE_QUOTA 0xc0000007
- #define SMB_STATUS_INVALID_HANDLE 0xc0000008
- #define SMB_STATUS_BAD_INITIAL_STACK 0xc0000009
- #define SMB_STATUS_BAD_INITIAL_PC 0xc000000a
- #define SMB_STATUS_INVALID_CID 0xc000000b
- #define SMB_STATUS_TIMER_NOT_CANCELED 0xc000000c
- #define SMB_STATUS_INVALID_PARAMETER 0xc000000d
- #define SMB_STATUS_NO_SUCH_DEVICE 0xc000000e
- #define SMB_STATUS_NO_SUCH_FILE 0xc000000f
- #define SMB_STATUS_INVALID_DEVICE_REQUEST 0xc0000010
- #define SMB_STATUS_END_OF_FILE 0xc0000011
- #define SMB_STATUS_WRONG_VOLUME 0xc0000012
- #define SMB_STATUS_NO_MEDIA_IN_DEVICE 0xc0000013
- #define SMB_STATUS_UNRECOGNIZED_MEDIA 0xc0000014
- #define SMB_STATUS_NONEXISTENT_SECTOR 0xc0000015
- #define SMB_STATUS_MORE_PROCESSING_REQUIRED 0xc0000016
- #define SMB_STATUS_NO_MEMORY 0xc0000017
- #define SMB_STATUS_CONFLICTING_ADDRESSES 0xc0000018
- #define SMB_STATUS_NOT_MAPPED_VIEW 0xc0000019
- #define SMB_STATUS_UNABLE_TO_FREE_VM 0xc000001a
- #define SMB_STATUS_UNABLE_TO_DELETE_SECTION 0xc000001b
- #define SMB_STATUS_INVALID_SYSTEM_SERVICE 0xc000001c
- #define SMB_STATUS_ILLEGAL_INSTRUCTION 0xc000001d
- #define SMB_STATUS_INVALID_LOCK_SEQUENCE 0xc000001e
- #define SMB_STATUS_INVALID_VIEW_SIZE 0xc000001f
- #define SMB_STATUS_INVALID_FILE_FOR_SECTION 0xc0000020
- #define SMB_STATUS_ALREADY_COMMITTED 0xc0000021
- #define SMB_STATUS_ACCESS_DENIED 0xc0000022
- #define SMB_STATUS_BUFFER_TOO_SMALL 0xc0000023
- #define SMB_STATUS_OBJECT_TYPE_MISMATCH 0xc0000024
- #define SMB_STATUS_NONCONTINUABLE_EXCEPTION 0xc0000025
- #define SMB_STATUS_INVALID_DISPOSITION 0xc0000026
- #define SMB_STATUS_UNWIND 0xc0000027
- #define SMB_STATUS_BAD_STACK 0xc0000028
- #define SMB_STATUS_INVALID_UNWIND_TARGET 0xc0000029
- #define SMB_STATUS_NOT_LOCKED 0xc000002a
- #define SMB_STATUS_PARITY_ERROR 0xc000002b
- #define SMB_STATUS_UNABLE_TO_DECOMMIT_VM 0xc000002c
- #define SMB_STATUS_NOT_COMMITTED 0xc000002d
- #define SMB_STATUS_INVALID_PORT_ATTRIBUTES 0xc000002e
- #define SMB_STATUS_PORT_MESSAGE_TOO_LONG 0xc000002f
- #define SMB_STATUS_INVALID_PARAMETER_MIX 0xc0000030
- #define SMB_STATUS_INVALID_QUOTA_LOWER 0xc0000031
- #define SMB_STATUS_DISK_CORRUPT_ERROR 0xc0000032
- #define SMB_STATUS_OBJECT_NAME_INVALID 0xc0000033
- #define SMB_STATUS_OBJECT_NAME_NOT_FOUND 0xc0000034
- #define SMB_STATUS_OBJECT_NAME_COLLISION 0xc0000035
- #define SMB_STATUS_HANDLE_NOT_WAITABLE 0xc0000036
- #define SMB_STATUS_PORT_DISCONNECTED 0xc0000037
- #define SMB_STATUS_DEVICE_ALREADY_ATTACHED 0xc0000038
- #define SMB_STATUS_OBJECT_PATH_INVALID 0xc0000039
- #define SMB_STATUS_OBJECT_PATH_NOT_FOUND 0xc000003a

- #define SMB_STATUS_OBJECT_PATH_SYNTAX_BAD 0xc000003b
- #define SMB_STATUS_DATA_OVERRUN 0xc000003c
- #define SMB_STATUS_DATA_LATE_ERROR 0xc000003d
- #define SMB_STATUS_DATA_ERROR 0xc000003e
- #define SMB_STATUS_CRC_ERROR 0xc000003f
- #define SMB_STATUS_SECTION_TOO_BIG 0xc0000040
- #define SMB_STATUS_PORT_CONNECTION_REFUSED 0xc0000041
- #define SMB_STATUS_INVALID_PORT_HANDLE 0xc0000042
- #define SMB_STATUS_SHARING_VIOLATION 0xc0000043
- #define SMB_STATUS_QUOTA_EXCEEDED 0xc0000044
- #define SMB_STATUS_INVALID_PAGE_PROTECTION 0xc0000045
- #define SMB_STATUS_MUTANT_NOT_OWNED 0xc0000046
- #define SMB_STATUS_SEMAPHORE_LIMIT_EXCEEDED 0xc0000047
- #define SMB_STATUS_PORT_ALREADY_SET 0xc0000048
- #define SMB_STATUS_SECTION_NOT_IMAGE 0xc0000049
- #define SMB_STATUS_SUSPEND_COUNT_EXCEEDED 0xc000004a
- #define SMB_STATUS_THREAD_IS_TERMINATING 0xc000004b
- #define SMB_STATUS_BAD_WORKING_SET_LIMIT 0xc000004c
- #define SMB_STATUS_INCOMPATIBLE_FILE_MAP 0xc000004d
- #define SMB_STATUS_SECTION_PROTECTION 0xc000004e
- #define SMB_STATUS_EAS_NOT_SUPPORTED 0xc000004f
- #define SMB_STATUS_EA_TOO_LARGE 0xc0000050
- #define SMB_STATUS_NONEXISTENT_EA_ENTRY 0xc0000051
- #define SMB_STATUS_NO_EAS_ON_FILE 0xc0000052
- #define SMB_STATUS_EA_CORRUPT_ERROR 0xc0000053
- #define SMB_STATUS_FILE_LOCK_CONFLICT 0xc0000054
- #define SMB_STATUS_LOCK_NOT_GRANTED 0xc0000055
- #define SMB_STATUS_DELETE_PENDING 0xc0000056
- #define SMB_STATUS_CTL_FILE_NOT_SUPPORTED 0xc0000057
- #define SMB_STATUS_UNKNOWN_REVISION 0xc0000058
- #define SMB_STATUS_REVISION_MISMATCH 0xc0000059
- #define SMB_STATUS_INVALID_OWNER 0xc000005a
- #define SMB_STATUS_INVALID_PRIMARY_GROUP 0xc000005b
- #define SMB_STATUS_NO_IMPERSONATION_TOKEN 0xc000005c
- #define SMB_STATUS_CANT_DISABLE_MANDATORY 0xc000005d
- #define SMB_STATUS_NO_LOGON_SERVERS 0xc000005e
- #define SMB_STATUS_NO_SUCH_LOGON_SESSION 0xc000005f
- #define SMB_STATUS_NO_SUCH_PRIVILEGE 0xc0000060
- #define SMB_STATUS_PRIVILEGE_NOT_HELD 0xc0000061
- #define SMB_STATUS_INVALID_ACCOUNT_NAME 0xc0000062
- #define SMB_STATUS_USER_EXISTS 0xc0000063
- #define SMB_STATUS_NO_SUCH_USER 0xc0000064
- #define SMB_STATUS_GROUP_EXISTS 0xc0000065
- #define SMB_STATUS_NO_SUCH_GROUP 0xc0000066
- #define SMB_STATUS_MEMBER_IN_GROUP 0xc0000067
- #define SMB_STATUS_MEMBER_NOT_IN_GROUP 0xc0000068
- #define SMB_STATUS_LAST_ADMIN 0xc0000069
- #define SMB_STATUS_WRONG_PASSWORD 0xc000006a
- #define SMB_STATUS_ILL_FORMED_PASSWORD 0xc000006b
- #define SMB_STATUS_PASSWORD_RESTRICTION 0xc000006c
- #define SMB_STATUS_LOGON_FAILURE 0xc000006d
- #define SMB_STATUS_ACCOUNT_RESTRICTION 0xc000006e
- #define SMB_STATUS_INVALID_LOGON_HOURS 0xc000006f
- #define SMB_STATUS_INVALID_WORKSTATION 0xc0000070
- #define SMB_STATUS_PASSWORD_EXPIRED 0xc0000071

- #define SMB_STATUS_ACCOUNT_DISABLED 0xc0000072
- #define SMB_STATUS_NONE_MAPPED 0xc0000073
- #define SMB_STATUS_TOO_MANY_LUIDS_REQUESTED 0xc0000074
- #define SMB_STATUS_LUIDS_EXHAUSTED 0xc0000075
- #define SMB_STATUS_INVALID_SUB_AUTHORITY 0xc0000076
- #define SMB_STATUS_INVALID_ACL 0xc0000077
- #define SMB_STATUS_INVALID_SID 0xc0000078
- #define SMB_STATUS_INVALID_SECURITY_DESCR 0xc0000079
- #define SMB_STATUS_PROCEDURE_NOT_FOUND 0xc000007a
- #define SMB_STATUS_INVALID_IMAGE_FORMAT 0xc000007b
- #define SMB_STATUS_NO_TOKEN 0xc000007c
- #define SMB_STATUS_BAD_INHERITANCE_ACL 0xc000007d
- #define SMB_STATUS_RANGE_NOT_LOCKED 0xc000007e
- #define SMB_STATUS_DISK_FULL 0xc000007f
- #define SMB_STATUS_SERVER_DISABLED 0xc0000080
- #define SMB_STATUS_SERVER_NOT_DISABLED 0xc0000081
- #define SMB_STATUS_TOO_MANY_GUIDS_REQUESTED 0xc0000082
- #define SMB_STATUS_GUIDS_EXHAUSTED 0xc0000083
- #define SMB_STATUS_INVALID_ID_AUTHORITY 0xc0000084
- #define SMB_STATUS_AGENTS_EXHAUSTED 0xc0000085
- #define SMB_STATUS_INVALID_VOLUME_LABEL 0xc0000086
- #define SMB_STATUS_SECTION_NOT_EXTENDED 0xc0000087
- #define SMB_STATUS_NOT_MAPPED_DATA 0xc0000088
- #define SMB_STATUS_RESOURCE_DATA_NOT_FOUND 0xc0000089
- #define SMB_STATUS_RESOURCE_TYPE_NOT_FOUND 0xc000008a
- #define SMB_STATUS_RESOURCE_NAME_NOT_FOUND 0xc000008b
- #define SMB_STATUS_ARRAY_BOUNDS_EXCEEDED 0xc000008c
- #define SMB_STATUS_FLOAT_DENORMAL_OPERAND 0xc000008d
- #define SMB_STATUS_FLOAT_DIVIDE_BY_ZERO 0xc000008e
- #define SMB_STATUS_FLOAT_INEXACT_RESULT 0xc000008f
- #define SMB_STATUS_FLOAT_INVALID_OPERATION 0xc0000090
- #define SMB_STATUS_FLOAT_OVERFLOW 0xc0000091
- #define SMB_STATUS_FLOAT_STACK_CHECK 0xc0000092
- #define SMB_STATUS_FLOAT_UNDERFLOW 0xc0000093
- #define SMB_STATUS_INTEGER_DIVIDE_BY_ZERO 0xc0000094
- #define SMB_STATUS_INTEGER_OVERFLOW 0xc0000095
- #define SMB_STATUS_PRIVILEGED_INSTRUCTION 0xc0000096
- #define SMB_STATUS_TOO_MANY_PAGING_FILES 0xc0000097
- #define SMB_STATUS_FILE_INVALID 0xc0000098
- #define SMB_STATUS_ALLOTTED_SPACE_EXCEEDED 0xc0000099
- #define SMB_STATUS_INSUFFICIENT_RESOURCES 0xc000009a
- #define SMB_STATUS_DFS_EXIT_PATH_FOUND 0xc000009b
- #define SMB_STATUS_DEVICE_DATA_ERROR 0xc000009c
- #define SMB_STATUS_DEVICE_NOT_CONNECTED 0xc000009d
- #define SMB_STATUS_DEVICE_POWER_FAILURE 0xc000009e
- #define SMB_STATUS_FREE_VM_NOT_AT_BASE 0xc000009f
- #define SMB_STATUS_MEMORY_NOT_ALLOCATED 0xc00000a0
- #define SMB_STATUS_WORKING_SET_QUOTA 0xc00000a1
- #define SMB_STATUS_MEDIA_WRITE_PROTECTED 0xc00000a2
- #define SMB_STATUS_DEVICE_NOT_READY 0xc00000a3
- #define SMB_STATUS_INVALID_GROUP_ATTRIBUTES 0xc00000a4
- #define SMB_STATUS_BAD_IMPERSONATION_LEVEL 0xc00000a5
- #define SMB_STATUS_CANT_OPEN_ANONYMOUS 0xc00000a6
- #define SMB_STATUS_BAD_VALIDATION_CLASS 0xc00000a7
- #define SMB_STATUS_BAD_TOKEN_TYPE 0xc00000a8

- #define SMB_STATUS_BAD_MASTER_BOOT_RECORD 0xc00000a9
- #define SMB_STATUS_INSTRUCTION_MISALIGNMENT 0xc00000aa
- #define SMB_STATUS_INSTANCE_NOT_AVAILABLE 0xc00000ab
- #define SMB_STATUS_PIPE_NOT_AVAILABLE 0xc00000ac
- #define SMB_STATUS_INVALID_PIPE_STATE 0xc00000ad
- #define SMB_STATUS_PIPE_BUSY 0xc00000ae
- #define SMB_STATUS_ILLEGAL_FUNCTION 0xc00000af
- #define SMB_STATUS_PIPE_DISCONNECTED 0xc00000b0
- #define SMB_STATUS_PIPE_CLOSING 0xc00000b1
- #define SMB_STATUS_PIPE_CONNECTED 0xc00000b2
- #define SMB_STATUS_PIPE_LISTENING 0xc00000b3
- #define SMB_STATUS_INVALID_READ_MODE 0xc00000b4
- #define SMB_STATUS_IO_TIMEOUT 0xc00000b5
- #define SMB_STATUS_FILE_FORCED_CLOSED 0xc00000b6
- #define SMB_STATUS_PROFILING_NOT_STARTED 0xc00000b7
- #define SMB_STATUS_PROFILING_NOT_STOPPED 0xc00000b8
- #define SMB_STATUS_COULD_NOT_INTERPRET 0xc00000b9
- #define SMB_STATUS_FILE_IS_A_DIRECTORY 0xc00000ba
- #define SMB_STATUS_NOT_SUPPORTED 0xc00000bb
- #define SMB_STATUS_REMOTE_NOT_LISTENING 0xc00000bc
- #define SMB_STATUS_DUPLICATE_NAME 0xc00000bd
- #define SMB_STATUS_BAD_NETWORK_PATH 0xc00000be
- #define SMB_STATUS_NETWORK_BUSY 0xc00000bf
- #define SMB_STATUS_DEVICE_DOES_NOT_EXIST 0xc00000c0
- #define SMB_STATUS_TOO_MANY_COMMANDS 0xc00000c1
- #define SMB_STATUS_ADAPTER_HARDWARE_ERROR 0xc00000c2
- #define SMB_STATUS_INVALID_NETWORK_RESPONSE 0xc00000c3
- #define SMB_STATUS_UNEXPECTED_NETWORK_ERROR 0xc00000c4
- #define SMB_STATUS_BAD_REMOTE_ADAPTER 0xc00000c5
- #define SMB_STATUS_PRINT_QUEUE_FULL 0xc00000c6
- #define SMB_STATUS_NO_SPOOL_SPACE 0xc00000c7
- #define SMB_STATUS_PRINT_CANCELLED 0xc00000c8
- #define SMB_STATUS_NETWORK_NAME_DELETED 0xc00000c9
- #define SMB_STATUS_NETWORK_ACCESS_DENIED 0xc00000ca
- #define SMB_STATUS_BAD_DEVICE_TYPE 0xc00000cb
- #define SMB_STATUS_BAD_NETWORK_NAME 0xc00000cc
- #define SMB_STATUS_TOO_MANY_NAMES 0xc00000cd
- #define SMB_STATUS_TOO_MANY_SESSIONS 0xc00000ce
- #define SMB_STATUS_SHARING_PAUSED 0xc00000cf
- #define SMB_STATUS_REQUEST_NOT_ACCEPTED 0xc00000d0
- #define SMB_STATUS_REDIRECTOR_PAUSED 0xc00000d1
- #define SMB_STATUS_NET_WRITE_FAULT 0xc00000d2
- #define SMB_STATUS_PROFILING_AT_LIMIT 0xc00000d3
- #define SMB_STATUS_NOT_SAME_DEVICE 0xc00000d4
- #define SMB_STATUS_FILE_RENAMED 0xc00000d5
- #define SMB_STATUS_VIRTUAL_CIRCUIT_CLOSED 0xc00000d6
- #define SMB_STATUS_NO_SECURITY_ON_OBJECT 0xc00000d7
- #define SMB_STATUS_CANT_WAIT 0xc00000d8
- #define SMB_STATUS_PIPE_EMPTY 0xc00000d9
- #define SMB_STATUS_CANT_ACCESS_DOMAIN_INFO 0xc00000da
- #define SMB_STATUS_CANT_TERMINATE_SELF 0xc00000db
- #define SMB_STATUS_INVALID_SERVER_STATE 0xc00000dc
- #define SMB_STATUS_INVALID_DOMAIN_STATE 0xc00000dd
- #define SMB_STATUS_INVALID_DOMAIN_ROLE 0xc00000de
- #define SMB_STATUS_NO_SUCH_DOMAIN 0xc00000df

- #define SMB_STATUS_DOMAIN_EXISTS 0xc00000e0
- #define SMB_STATUS_DOMAIN_LIMIT_EXCEEDED 0xc00000e1
- #define SMB_STATUS_OPLOCK_NOT_GRANTED 0xc00000e2
- #define SMB_STATUS_INVALID_OPLOCK_PROTOCOL 0xc00000e3
- #define SMB_STATUS_INTERNAL_DB_CORRUPTION 0xc00000e4
- #define SMB_STATUS_INTERNAL_ERROR 0xc00000e5
- #define SMB_STATUS_GENERIC_NOT_MAPPED 0xc00000e6
- #define SMB_STATUS_BAD_DESCRIPTOR_FORMAT 0xc00000e7
- #define SMB_STATUS_INVALID_USER_BUFFER 0xc00000e8
- #define SMB_STATUS_UNEXPECTED_IO_ERROR 0xc00000e9
- #define SMB_STATUS_UNEXPECTED_MM_CREATE_ERR 0xc00000ea
- #define SMB_STATUS_UNEXPECTED_MM_MAP_ERROR 0xc00000eb
- #define SMB_STATUS_UNEXPECTED_MM_EXTEND_ERR 0xc00000ec
- #define SMB_STATUS_NOT_LOGON_PROCESS 0xc00000ed
- #define SMB_STATUS_LOGON_SESSION_EXISTS 0xc00000ee
- #define SMB_STATUS_INVALID_PARAMETER_1 0xc00000ef
- #define SMB_STATUS_INVALID_PARAMETER_2 0xc00000f0
- #define SMB_STATUS_INVALID_PARAMETER_3 0xc00000f1
- #define SMB_STATUS_INVALID_PARAMETER_4 0xc00000f2
- #define SMB_STATUS_INVALID_PARAMETER_5 0xc00000f3
- #define SMB_STATUS_INVALID_PARAMETER_6 0xc00000f4
- #define SMB_STATUS_INVALID_PARAMETER_7 0xc00000f5
- #define SMB_STATUS_INVALID_PARAMETER_8 0xc00000f6
- #define SMB_STATUS_INVALID_PARAMETER_9 0xc00000f7
- #define SMB_STATUS_INVALID_PARAMETER_10 0xc00000f8
- #define SMB_STATUS_INVALID_PARAMETER_11 0xc00000f9
- #define SMB_STATUS_INVALID_PARAMETER_12 0xc00000fa
- #define SMB_STATUS_REDIRECTOR_NOT_STARTED 0xc00000fb
- #define SMB_STATUS_REDIRECTOR_STARTED 0xc00000fc
- #define SMB_STATUS_STACK_OVERFLOW 0xc00000fd
- #define SMB_STATUS_NO_SUCH_PACKAGE 0xc00000fe
- #define SMB_STATUS_BAD_FUNCTION_TABLE 0xc00000ff
- #define SMB_STATUS_DIRECTORY_NOT_EMPTY 0xc0000101
- #define SMB_STATUS_FILE_CORRUPT_ERROR 0xc0000102
- #define SMB_STATUS_NOT_A_DIRECTORY 0xc0000103
- #define SMB_STATUS_BAD_LOGON_SESSION_STATE 0xc0000104
- #define SMB_STATUS_LOGON_SESSION_COLLISION 0xc0000105
- #define SMB_STATUS_NAME_TOO_LONG 0xc0000106
- #define SMB_STATUS_FILES_OPEN 0xc0000107
- #define SMB_STATUS_CONNECTION_IN_USE 0xc0000108
- #define SMB_STATUS_MESSAGE_NOT_FOUND 0xc0000109
- #define SMB_STATUS_PROCESS_IS_TERMINATING 0xc000010a
- #define SMB_STATUS_INVALID_LOGON_TYPE 0xc000010b
- #define SMB_STATUS_NO_GUID_TRANSLATION 0xc000010c
- #define SMB_STATUS_CANNOT_IMPERSONATE 0xc000010d
- #define SMB_STATUS_IMAGE_ALREADY_LOADED 0xc000010e
- #define SMB_STATUS_ABIOS_NOT_PRESENT 0xc000010f
- #define SMB_STATUS_ABIOS_LID_NOT_EXIST 0xc0000110
- #define SMB_STATUS_ABIOS_LID_ALREADY_OWNED 0xc0000111
- #define SMB_STATUS_ABIOS_NOT_LID_OWNER 0xc0000112
- #define SMB_STATUS_ABIOS_INVALID_COMMAND 0xc0000113
- #define SMB_STATUS_ABIOS_INVALID_LID 0xc0000114
- #define SMB_STATUS_ABIOS_SELECTOR_NOT_AVAILABLE 0xc0000115
- #define SMB_STATUS_ABIOS_INVALID_SELECTOR 0xc0000116
- #define SMB_STATUS_NO_LDT 0xc0000117

- #define SMB_STATUS_INVALID_LDT_SIZE 0xc0000118
- #define SMB_STATUS_INVALID_LDT_OFFSET 0xc0000119
- #define SMB_STATUS_INVALID_LDT_DESCRIPTOR 0xc000011a
- #define SMB_STATUS_INVALID_IMAGE_NE_FORMAT 0xc000011b
- #define SMB_STATUS_RXACT_INVALID_STATE 0xc000011c
- #define SMB_STATUS_RXACT_COMMIT_FAILURE 0xc000011d
- #define SMB_STATUS_MAPPED_FILE_SIZE_ZERO 0xc000011e
- #define SMB_STATUS_TOO_MANY_OPENED_FILES 0xc000011f
- #define SMB_STATUS_CANCELLED 0xc0000120
- #define SMB_STATUS_CANNOT_DELETE 0xc0000121
- #define SMB_STATUS_INVALID_COMPUTER_NAME 0xc0000122
- #define SMB_STATUS_FILE_DELETED 0xc0000123
- #define SMB_STATUS_SPECIAL_ACCOUNT 0xc0000124
- #define SMB_STATUS_SPECIAL_GROUP 0xc0000125
- #define SMB_STATUS_SPECIAL_USER 0xc0000126
- #define SMB_STATUS_MEMBERS_PRIMARY_GROUP 0xc0000127
- #define SMB_STATUS_FILE_CLOSED 0xc0000128
- #define SMB_STATUS_TOO_MANY_THREADS 0xc0000129
- #define SMB_STATUS_THREAD_NOT_IN_PROCESS 0xc000012a
- #define SMB_STATUS_TOKEN_ALREADY_IN_USE 0xc000012b
- #define SMB_STATUS_PAGEFILE_QUOTA_EXCEEDED 0xc000012c
- #define SMB_STATUS_COMMITMENT_LIMIT 0xc000012d
- #define SMB_STATUS_INVALID_IMAGE_LE_FORMAT 0xc000012e
- #define SMB_STATUS_INVALID_IMAGE_NOT_MZ 0xc000012f
- #define SMB_STATUS_INVALID_IMAGE_PROTECT 0xc0000130
- #define SMB_STATUS_INVALID_IMAGE_WIN_16 0xc0000131
- #define SMB_STATUS_LOGON_SERVER_CONFLICT 0xc0000132
- #define SMB_STATUS_TIME_DIFFERENCE_AT_DC 0xc0000133
- #define SMB_STATUS_SYNCHRONIZATION_REQUIRED 0xc0000134
- #define SMB_STATUS_DLL_NOT_FOUND 0xc0000135
- #define SMB_STATUS_OPEN_FAILED 0xc0000136
- #define SMB_STATUS_IO_PRIVILEGE_FAILED 0xc0000137
- #define SMB_STATUS_ORDINAL_NOT_FOUND 0xc0000138
- #define SMB_STATUS_ENTRYPOINT_NOT_FOUND 0xc0000139
- #define SMB_STATUS_CONTROL_C_EXIT 0xc000013a
- #define SMB_STATUS_LOCAL_DISCONNECT 0xc000013b
- #define SMB_STATUS_REMOTE_DISCONNECT 0xc000013c
- #define SMB_STATUS_REMOTE_RESOURCES 0xc000013d
- #define SMB_STATUS_LINK_FAILED 0xc000013e
- #define SMB_STATUS_LINK_TIMEOUT 0xc000013f
- #define SMB_STATUS_INVALID_CONNECTION 0xc0000140
- #define SMB_STATUS_INVALID_ADDRESS 0xc0000141
- #define SMB_STATUS_DLL_INIT_FAILED 0xc0000142
- #define SMB_STATUS_MISSING_SYSTEMFILE 0xc0000143
- #define SMB_STATUS_UNHANDLED_EXCEPTION 0xc0000144
- #define SMB_STATUS_APP_INIT_FAILURE 0xc0000145
- #define SMB_STATUS_PAGEFILE_CREATE_FAILED 0xc0000146
- #define SMB_STATUS_NO_PAGEFILE 0xc0000147
- #define SMB_STATUS_INVALID_LEVEL 0xc0000148
- #define SMB_STATUS_WRONG_PASSWORD_CORE 0xc0000149
- #define SMB_STATUS_ILLEGAL_FLOAT_CONTEXT 0xc000014a
- #define SMB_STATUS_PIPE_BROKEN 0xc000014b
- #define SMB_STATUS_REGISTRY_CORRUPT 0xc000014c
- #define SMB_STATUS_REGISTRY_IO_FAILED 0xc000014d
- #define SMB_STATUS_NO_EVENT_PAIR 0xc000014e

- #define SMB_STATUS_UNRECOGNIZED_VOLUME 0xc000014f
- #define SMB_STATUS_SERIAL_NO_DEVICE_INITED 0xc0000150
- #define SMB_STATUS_NO_SUCH_ALIAS 0xc0000151
- #define SMB_STATUS_MEMBER_NOT_IN_ALIAS 0xc0000152
- #define SMB_STATUS_MEMBER_IN_ALIAS 0xc0000153
- #define SMB_STATUS_ALIAS_EXISTS 0xc0000154
- #define SMB_STATUS_LOGON_NOT_GRANTED 0xc0000155
- #define SMB_STATUS_TOO_MANY_SECRETS 0xc0000156
- #define SMB_STATUS_SECRET_TOO_LONG 0xc0000157
- #define SMB_STATUS_INTERNAL_DB_ERROR 0xc0000158
- #define SMB_STATUS_FULLSCREEN_MODE 0xc0000159
- #define SMB_STATUS_TOO_MANY_CONTEXT_IDS 0xc000015a
- #define SMB_STATUS_LOGON_TYPE_NOT_GRANTED 0xc000015b
- #define SMB_STATUS_NOT_REGISTRY_FILE 0xc000015c
- #define SMB_STATUS_NT_CROSS_ENCRYPTION_REQUIRED 0xc000015d
- #define SMB_STATUS_DOMAIN_CTRLR_CONFIG_ERROR 0xc000015e
- #define SMB_STATUS_FT_MISSING_MEMBER 0xc000015f
- #define SMB_STATUS_ILL_FORMED_SERVICE_ENTRY 0xc0000160
- #define SMB_STATUS_ILLEGAL_CHARACTER 0xc0000161
- #define SMB_STATUS_UNMAPPABLE_CHARACTER 0xc0000162
- #define SMB_STATUS_UNDEFINED_CHARACTER 0xc0000163
- #define SMB_STATUS_FLOPPY_VOLUME 0xc0000164
- #define SMB_STATUS_FLOPPY_ID_MARK_NOT_FOUND 0xc0000165
- #define SMB_STATUS_FLOPPY_WRONG_CYLINDER 0xc0000166
- #define SMB_STATUS_FLOPPY_UNKNOWN_ERROR 0xc0000167
- #define SMB_STATUS_FLOPPY_BAD_REGISTERS 0xc0000168
- #define SMB_STATUS_DISK_RECALIBRATE_FAILED 0xc0000169
- #define SMB_STATUS_DISK_OPERATION_FAILED 0xc000016a
- #define SMB_STATUS_DISK_RESET_FAILED 0xc000016b
- #define SMB_STATUS_SHARED_IRQ_BUSY 0xc000016c
- #define SMB_STATUS_FT_ORPHANING 0xc000016d
- #define SMB_STATUS_PARTITION_FAILURE 0xc0000172
- #define SMB_STATUS_INVALID_BLOCK_LENGTH 0xc0000173
- #define SMB_STATUS_DEVICE_NOT_PARTITIONED 0xc0000174
- #define SMB_STATUS_UNABLE_TO_LOCK_MEDIA 0xc0000175
- #define SMB_STATUS_UNABLE_TO_UNLOAD_MEDIA 0xc0000176
- #define SMB_STATUS_EOM_OVERFLOW 0xc0000177
- #define SMB_STATUS_NO_MEDIA 0xc0000178
- #define SMB_STATUS_NO_SUCH_MEMBER 0xc000017a
- #define SMB_STATUS_INVALID_MEMBER 0xc000017b
- #define SMB_STATUS_KEY_DELETED 0xc000017c
- #define SMB_STATUS_NO_LOG_SPACE 0xc000017d
- #define SMB_STATUS_TOO_MANY_SIDS 0xc000017e
- #define SMB_STATUS_LM_CROSS_ENCRYPTION_REQUIRED 0xc000017f
- #define SMB_STATUS_KEY_HAS_CHILDREN 0xc0000180
- #define SMB_STATUS_CHILD_MUST_BE_VOLATILE 0xc0000181
- #define SMB_STATUS_DEVICE_CONFIGURATION_ERROR 0xc0000182
- #define SMB_STATUS_DRIVER_INTERNAL_ERROR 0xc0000183
- #define SMB_STATUS_INVALID_DEVICE_STATE 0xc0000184
- #define SMB_STATUS_IO_DEVICE_ERROR 0xc0000185
- #define SMB_STATUS_DEVICE_PROTOCOL_ERROR 0xc0000186
- #define SMB_STATUS_BACKUP_CONTROLLER 0xc0000187
- #define SMB_STATUS_LOG_FILE_FULL 0xc0000188
- #define SMB_STATUS_TOO_LATE 0xc0000189
- #define SMB_STATUS_NO_TRUST_LSA_SECRET 0xc000018a

- #define SMB_STATUS_NO_TRUST_SAM_ACCOUNT 0xc000018b
- #define SMB_STATUS_TRUSTED_DOMAIN_FAILURE 0xc000018c
- #define SMB_STATUS_TRUSTED_RELATIONSHIP_FAILURE 0xc000018d
- #define SMB_STATUS_EVENTLOG_FILE_CORRUPT 0xc000018e
- #define SMB_STATUS_EVENTLOG_CANT_START 0xc000018f
- #define SMB_STATUS_TRUST_FAILURE 0xc0000190
- #define SMB_STATUS_MUTANT_LIMIT_EXCEEDED 0xc0000191
- #define SMB_STATUS_NETLOGON_NOT_STARTED 0xc0000192
- #define SMB_STATUS_ACCOUNT_EXPIRED 0xc0000193
- #define SMB_STATUS_POSSIBLE_DEADLOCK 0xc0000194
- #define SMB_STATUS_NETWORK_CREDENTIAL_CONFLICT 0xc0000195
- #define SMB_STATUS_REMOTE_SESSION_LIMIT 0xc0000196
- #define SMB_STATUS_EVENTLOG_FILE_CHANGED 0xc0000197
- #define SMB_STATUS_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT 0xc0000198
- #define SMB_STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT 0xc0000199
- #define SMB_STATUS_NOLOGON_SERVER_TRUST_ACCOUNT 0xc000019a
- #define SMB_STATUS_DOMAIN_TRUST_INCONSISTENT 0xc000019b
- #define SMB_STATUS_FS_DRIVER_REQUIRED 0xc000019c
- #define SMB_STATUS_INVALID_LOCK_RANGE 0xc00001a1
- #define SMB_STATUS_NO_USER_SESSION_KEY 0xc0000202
- #define SMB_STATUS_USER_SESSION_DELETED 0xc0000203
- #define SMB_STATUS_RESOURCE_LANG_NOT_FOUND 0xc0000204
- #define SMB_STATUS_INSUFF_SERVER_RESOURCES 0xc0000205
- #define SMB_STATUS_INVALID_BUFFER_SIZE 0xc0000206
- #define SMB_STATUS_INVALID_ADDRESS_COMPONENT 0xc0000207
- #define SMB_STATUS_INVALID_ADDRESS_WILDCARD 0xc0000208
- #define SMB_STATUS_TOO_MANY_ADDRESSES 0xc0000209
- #define SMB_STATUS_ADDRESS_ALREADY_EXISTS 0xc000020a
- #define SMB_STATUS_ADDRESS_CLOSED 0xc000020b
- #define SMB_STATUS_CONNECTION_DISCONNECTED 0xc000020c
- #define SMB_STATUS_CONNECTION_RESET 0xc000020d
- #define SMB_STATUS_TOO_MANY_NODES 0xc000020e
- #define SMB_STATUS_TRANSACTION_ABORTED 0xc000020f
- #define SMB_STATUS_TRANSACTION_TIMED_OUT 0xc0000210
- #define SMB_STATUS_TRANSACTION_NO_RELEASE 0xc0000211
- #define SMB_STATUS_TRANSACTION_NO_MATCH 0xc0000212
- #define SMB_STATUS_TRANSACTION_RESPONDED 0xc0000213
- #define SMB_STATUS_TRANSACTION_INVALID_ID 0xc0000214
- #define SMB_STATUS_TRANSACTION_INVALID_TYPE 0xc0000215
- #define SMB_STATUS_NOT_SERVER_SESSION 0xc0000216
- #define SMB_STATUS_NOT_CLIENT_SESSION 0xc0000217
- #define SMB_STATUS_CANNOT_LOAD_REGISTRY_FILE 0xc0000218
- #define SMB_STATUS_DEBUG_ATTACH_FAILED 0xc0000219
- #define SMB_STATUS_SYSTEM_PROCESS_TERMINATED 0xc000021a
- #define SMB_STATUS_DATA_NOT_ACCEPTED 0xc000021b
- #define SMB_STATUS_NO_BROWSER_SERVERS_FOUND 0xc000021c
- #define SMB_STATUS_VDM_HARD_ERROR 0xc000021d
- #define SMB_STATUS_DRIVER_CANCEL_TIMEOUT 0xc000021e
- #define SMB_STATUS_REPLY_MESSAGE_MISMATCH 0xc000021f
- #define SMB_STATUS_MAPPED_ALIGNMENT 0xc0000220
- #define SMB_STATUS_IMAGE_CHECKSUM_MISMATCH 0xc0000221
- #define SMB_STATUS_LOST_WRITEBEHIND_DATA 0xc0000222
- #define SMB_STATUS_CLIENT_SERVER_PARAMETERS_INVALID 0xc0000223
- #define SMB_STATUS_PASSWORD_MUST_CHANGE 0xc0000224
- #define SMB_STATUS_NOT_FOUND 0xc0000225

- #define SMB_STATUS_NOT_TINY_STREAM 0xc0000226
- #define SMB_STATUS_RECOVERY_FAILURE 0xc0000227
- #define SMB_STATUS_STACK_OVERFLOW_READ 0xc0000228
- #define SMB_STATUS_FAIL_CHECK 0xc0000229
- #define SMB_STATUS_DUPLICATE_OBJECTID 0xc000022a
- #define SMB_STATUS_OBJECTID_EXISTS 0xc000022b
- #define SMB_STATUS_CONVERT_TO_LARGE 0xc000022c
- #define SMB_STATUS_RETRY 0xc000022d
- #define SMB_STATUS_FOUND_OUT_OF_SCOPE 0xc000022e
- #define SMB_STATUS_ALLOCATE_BUCKET 0xc000022f
- #define SMB_STATUS_PROPSET_NOT_FOUND 0xc0000230
- #define SMB_STATUS_MARSHALL_OVERFLOW 0xc0000231
- #define SMB_STATUS_INVALID_VARIANT 0xc0000232
- #define SMB_STATUS_DOMAIN_CONTROLLER_NOT_FOUND 0xc0000233
- #define SMB_STATUS_ACCOUNT_LOCKED_OUT 0xc0000234
- #define SMB_STATUS_HANDLE_NOT_CLOSABLE 0xc0000235
- #define SMB_STATUS_CONNECTION_REFUSED 0xc0000236
- #define SMB_STATUS_GRACEFUL_DISCONNECT 0xc0000237
- #define SMB_STATUS_ADDRESS_ALREADY_ASSOCIATED 0xc0000238
- #define SMB_STATUS_ADDRESS_NOT_ASSOCIATED 0xc0000239
- #define SMB_STATUS_CONNECTION_INVALID 0xc000023a
- #define SMB_STATUS_CONNECTION_ACTIVE 0xc000023b
- #define SMB_STATUS_NETWORK_UNREACHABLE 0xc000023c
- #define SMB_STATUS_HOST_UNREACHABLE 0xc000023d
- #define SMB_STATUS_PROTOCOL_UNREACHABLE 0xc000023e
- #define SMB_STATUS_PORT_UNREACHABLE 0xc000023f
- #define SMB_STATUS_REQUEST_ABORTED 0xc0000240
- #define SMB_STATUS_CONNECTION_ABORTED 0xc0000241
- #define SMB_STATUS_BAD_COMPRESSION_BUFFER 0xc0000242
- #define SMB_STATUS_USER_MAPPED_FILE 0xc0000243
- #define SMB_STATUS_AUDIT_FAILED 0xc0000244
- #define SMB_STATUS_TIMER_RESOLUTION_NOT_SET 0xc0000245
- #define SMB_STATUS_CONNECTION_COUNT_LIMIT 0xc0000246
- #define SMB_STATUS_LOGIN_TIME_RESTRICTION 0xc0000247
- #define SMB_STATUS_LOGIN_WKSTA_RESTRICTION 0xc0000248
- #define SMB_STATUS_IMAGE_MP_UP_MISMATCH 0xc0000249
- #define SMB_STATUS_INSUFFICIENT_LOGON_INFO 0xc0000250
- #define SMB_STATUS_BAD_DLL_ENTRYPOINT 0xc0000251
- #define SMB_STATUS_BAD_SERVICE_ENTRYPOINT 0xc0000252
- #define SMB_STATUS_LPC_REPLY_LOST 0xc0000253
- #define SMB_STATUS_IP_ADDRESS_CONFLICT1 0xc0000254
- #define SMB_STATUS_IP_ADDRESS_CONFLICT2 0xc0000255
- #define SMB_STATUS_REGISTRY_QUOTA_LIMIT 0xc0000256
- #define SMB_STATUS_PATH_NOT_COVERED 0xc0000257
- #define SMB_STATUS_NO_CALLBACK_ACTIVE 0xc0000258
- #define SMB_STATUS_LICENSE_QUOTA_EXCEEDED 0xc0000259
- #define SMB_STATUS_PWD_TOO_SHORT 0xc000025a
- #define SMB_STATUS_PWD_TOO_RECENT 0xc000025b
- #define SMB_STATUS_PWD_HISTORY_CONFLICT 0xc000025c
- #define SMB_STATUS_PLUGPLAY_NO_DEVICE 0xc000025e
- #define SMB_STATUS_UNSUPPORTED_COMPRESSION 0xc000025f
- #define SMB_STATUS_INVALID_HW_PROFILE 0xc0000260
- #define SMB_STATUS_INVALID_PLUGPLAY_DEVICE_PATH 0xc0000261
- #define SMB_STATUS_DRIVER_ORDINAL_NOT_FOUND 0xc0000262
- #define SMB_STATUS_DRIVER_ENTRYPOINT_NOT_FOUND 0xc0000263

```

• #define SMB_STATUS_RESOURCE_NOT_OWNED 0xc0000264
• #define SMB_STATUS_TOO_MANY_LINKS 0xc0000265
• #define SMB_STATUS_QUOTA_LIST_INCONSISTENT 0xc0000266
• #define SMB_STATUS_FILE_IS_OFFLINE 0xc0000267
• #define SMB_STATUS_DFS_UNAVAILABLE 0xc000026d
• #define SMB_STATUS_VOLUME_DISMOUNTED 0xc000026e
• #define SMB_STATUS_NOT_A_REPARSE_POINT 0xc0000275
• #define SMB_STATUS_IO_REPARSE_TAG_NOT_HANDLED 0xc0000279
• #define SMB_STATUS_NETWORK_SESSION_EXPIRED 0xc000035C
• #define SMB_STATUS_INVALID_TOKEN 0xc0000465
• #define SMB_STATUS_SERVER_UNAVAILABLE 0xc0000466
• #define SMB_STATUS_FILE_NOT_AVAILABLE 0xc0000467
• #define SMB_STATUS_SHARE_UNAVAILABLE 0xc0000480
• #define SMB_STATUS_NO_SUCH_JOB 0xc0000ede
• #define SMB_STATUS_NO_PREAUTH_INTEGRITY_HASH_OVERLAP 0xC05D0000
• #define SMB_STATUS_DISCONNECT 0xe0000001
• #define SMB_STATUS_INTERNAL_RELEASE_REQUEST_RESPONSE 0xe0000002
• #define SMB_STATUS_INTERNAL_BUFFER_TOO_SMALL 0xe0000004
• #define SMB_STATUS_DONOTRELEASERESPONSE 0xe0000005
• #define NQ_ERR_MODULE (NQ_INT)(255u << 24)
• #define IS_NQ_ERROR(err) ((err & NQ_ERR_MODULE) == NQ_ERR_MODULE)
• #define NQ_ERR_OK (0)
• #define NQ_ERR_BADPARAM (NQ_ERR_MODULE | 3)
• #define NQ_ERR_GETDATA (NQ_ERR_MODULE | 18)
• #define NQ_ERR_INVALIDMODE (NQ_ERR_MODULE | 19)
• #define NQ_ERR_NOSERVERMAC (NQ_ERR_MODULE | 20)
• #define NQ_ERR_SIGNATUREFAIL (NQ_ERR_MODULE | 21)
• #define NQ_ERR_OBJEXISTS (NQ_ERR_MODULE | 22)
• #define NQ_ERR_MOUNTERROR (NQ_ERR_MODULE | 23)
• #define NQ_ERR_UNABLETODISPOSE (NQ_ERR_MODULE | 24)
• #define NQ_ERR_INVALIDHANDLE (NQ_ERR_MODULE | 25)
• #define NQ_ERR_NEGOTIATEFAILED (NQ_ERR_MODULE | 26)
• #define NQ_ERR_PATHNOTCOVERED (NQ_ERR_MODULE | 27)
• #define NQ_ERR_DFSCACHEOVERFLOW (NQ_ERR_MODULE | 28)
• #define NQ_ERR_ACCOUNTLOCKEDOUT (NQ_ERR_MODULE | 29)
• #define NQ_ERR_USEREXISTS (NQ_ERR_MODULE | 30)
• #define NQ_ERR_USERNOTFOUND (NQ_ERR_MODULE | 31)
• #define NQ_ERR_NOTFOUND (NQ_ERR_MODULE | 32)
• #define NQ_ERR_LOGONFAILURE (NQ_ERR_MODULE | 33)
• #define NQ_ERR_VOLUMEDISMOUNTED (NQ_ERR_MODULE | 34)
• #define NQ_ERR_NETWORKERROR (NQ_ERR_MODULE | 35)
• #define NQ_ERR_BADCONNECTION (NQ_ERR_MODULE | 36)
• #define NQ_ERR_SHARINGPAUSED (NQ_ERR_MODULE | 37)
• #define NQ_ERR_BADDFS (NQ_ERR_MODULE | 38)
• #define NQ_ERR_IOTIMEOUT (NQ_ERR_MODULE | 39)
• #define NQ_ERR_TRYAGAIN (NQ_ERR_MODULE | 40)
• #define NQ_ERR_ACCOUNTEXPIRED (NQ_ERR_MODULE | 41)
• #define NQ_ERR_INVALIDLOGONHOURS (NQ_ERR_MODULE | 42)
• #define NQ_ERR_ACCOUNTDISABLED (NQ_ERR_MODULE | 43)
• #define NQ_ERR_SESSIONREAUTHREQUIRED (NQ_ERR_MODULE | 44)
• #define NQ_ERR_USERRELOGONFAILED (NQ_ERR_MODULE | 45)
• #define NQ_ERR_SERVERUNAVAILABLE (NQ_ERR_MODULE | 46)
• #define NQ_ERR_SHAREUNAVAILABLE (NQ_ERR_MODULE | 47)
• #define NQ_ERR_OUTOFMEMORY (NQ_ERR_MODULE | 901)
• #define NQ_ERR_BADFUNC (NQ_ERR_MODULE | 1001)

```


- #define NQ_ERR_BADFILE (NQ_ERR_MODULE | 1002)
- #define NQ_ERR_BADPATH (NQ_ERR_MODULE | 1003)
- #define NQ_ERR_NOFIDS (NQ_ERR_MODULE | 1004)
- #define NQ_ERR_NOACCESS (NQ_ERR_MODULE | 1005)
- #define NQ_ERR_BADFID (NQ_ERR_MODULE | 1006)
- #define NQ_ERR_BADMCB (NQ_ERR_MODULE | 1007)
- #define NQ_ERR_NOMEM (NQ_ERR_MODULE | 1008)
- #define NQ_ERR_BADMEM (NQ_ERR_MODULE | 1009)
- #define NQ_ERR_BADENV (NQ_ERR_MODULE | 1010)
- #define NQ_ERR_BADFORMAT (NQ_ERR_MODULE | 1011)
- #define NQ_ERR_BADACCESS (NQ_ERR_MODULE | 1012)
- #define NQ_ERR_BADDATA (NQ_ERR_MODULE | 1013)
- #define NQ_ERR_BADDRIVE (NQ_ERR_MODULE | 1015)
- #define NQ_ERR_REMCD (NQ_ERR_MODULE | 1016)
- #define NQ_ERR_DIFFDEVICE (NQ_ERR_MODULE | 1017)
- #define NQ_ERR_NOFILES (NQ_ERR_MODULE | 1018)
- #define NQ_ERR_WRONGDIALECT (NQ_ERR_MODULE | 1019)
- #define NQ_ERR_BADSHARE (NQ_ERR_MODULE | 1032)
- #define NQ_ERR_LOCK (NQ_ERR_MODULE | 1033)
- #define NQ_ERR_DONTSUPPORTIPC (NQ_ERR_MODULE | 1066)
- #define NQ_ERR_SHARINGVIOLATION (NQ_ERR_MODULE | 1067)
- #define NQ_ERR_FILEXISTS (NQ_ERR_MODULE | 1080)
- #define NQ_ERR_BADDIRECTORY (NQ_ERR_MODULE | 1087)
- #define NQ_ERR_INSUFFICIENTBUFFER (NQ_ERR_MODULE | 1122)
- #define NQ_ERR_INVALIDNAME (NQ_ERR_MODULE | 1123)
- #define NQ_ERR_DIRNOTEMPTY (NQ_ERR_MODULE | 1145)
- #define NQ_ERR_ALREADYEXISTS (NQ_ERR_MODULE | 1183)
- #define NQ_ERR_BADPIPE (NQ_ERR_MODULE | 1230)
- #define NQ_ERR_PIPEBUSY (NQ_ERR_MODULE | 1231)
- #define NQ_ERR_PIPECLOSING (NQ_ERR_MODULE | 1232)
- #define NQ_ERR_NOTCONNECTED (NQ_ERR_MODULE | 1233)
- #define NQ_ERR_MOREDATA (NQ_ERR_MODULE | 1234)
- #define NQ_ERR_INVALIDNAMELEN (NQ_ERR_MODULE | 1235)
- #define NQ_ERR_ERROR (NQ_ERR_MODULE | 2001)
- #define NQ_ERR_BADPW (NQ_ERR_MODULE | 2002)
- #define NQ_ERR_ACCESS (NQ_ERR_MODULE | 2004)
- #define NQ_ERR_INVTID (NQ_ERR_MODULE | 2005)
- #define NQ_ERR_INVNETNAME (NQ_ERR_MODULE | 2006)
- #define NQ_ERR_INVDEVICE (NQ_ERR_MODULE | 2007)
- #define NQ_ERR_QFULL (NQ_ERR_MODULE | 2049)
- #define NQ_ERR_QTOOBIG (NQ_ERR_MODULE | 2050)
- #define NQ_ERR_QEOF (NQ_ERR_MODULE | 2051)
- #define NQ_ERR_INVFID (NQ_ERR_MODULE | 2052)
- #define NQ_ERR_SMBCMD (NQ_ERR_MODULE | 2064)
- #define NQ_ERR_SRVEERROR (NQ_ERR_MODULE | 2065)
- #define NQ_ERR_FILESPECS (NQ_ERR_MODULE | 2067)
- #define NQ_ERR_BADPERMITS (NQ_ERR_MODULE | 2069)
- #define NQ_ERR_SETATTRMODE (NQ_ERR_MODULE | 2071)
- #define NQ_ERR_PAUSED (NQ_ERR_MODULE | 2081)
- #define NQ_ERR_MSGOFF (NQ_ERR_MODULE | 2082)
- #define NQ_ERR_NOROOM (NQ_ERR_MODULE | 2083)
- #define NQ_ERR_RMUNS (NQ_ERR_MODULE | 2087)
- #define NQ_ERR_TIMEOUT (NQ_ERR_MODULE | 2088)
- #define NQ_ERR_NORESOURCE (NQ_ERR_MODULE | 2089)
- #define NQ_ERR_TOOMANYUIDS (NQ_ERR_MODULE | 2090)

- #define NQ_ERR_INVUID (NQ_ERR_MODULE | 2091)
- #define NQ_ERR_USEMPX (NQ_ERR_MODULE | 2250)
- #define NQ_ERR_USESTD (NQ_ERR_MODULE | 2251)
- #define NQ_ERR_CONTMPX (NQ_ERR_MODULE | 2252)
- #define NQ_ERR_NOSUPPORT (NQ_ERR_MODULE | 2999)
- #define NQ_ERR_NOWRITE (NQ_ERR_MODULE | 3019)
- #define NQ_ERR_BADUNIT (NQ_ERR_MODULE | 3020)
- #define NQ_ERR_NOTREADY (NQ_ERR_MODULE | 3021)
- #define NQ_ERR_BADCMD (NQ_ERR_MODULE | 3022)
- #define NQ_ERR_DATA (NQ_ERR_MODULE | 3023)
- #define NQ_ERR_BADREQ (NQ_ERR_MODULE | 3024)
- #define NQ_ERR_SEEK (NQ_ERR_MODULE | 3025)
- #define NQ_ERR_BADMEDIA (NQ_ERR_MODULE | 3026)
- #define NQ_ERR_BADSECTOR (NQ_ERR_MODULE | 3027)
- #define NQ_ERR_NOPAPER (NQ_ERR_MODULE | 3028)
- #define NQ_ERR_WRITE (NQ_ERR_MODULE | 3029)
- #define NQ_ERR_READ (NQ_ERR_MODULE | 3030)
- #define NQ_ERR_GENERAL (NQ_ERR_MODULE | 3031)
- #define NQ_ERR_WRONGDISK (NQ_ERR_MODULE | 3034)
- #define NQ_ERR_FCBUNAVAIL (NQ_ERR_MODULE | 3035)
- #define NQ_ERR_SHAREBUFEXC (NQ_ERR_MODULE | 3036)
- #define NQ_ERR_DISKFULL (NQ_ERR_MODULE | 3039)
- #define NQ_ERR_NBILLEGALSOCKETSLLOT (NQ_ERR_MODULE | 4000)
- #define NQ_ERR_NBNOTNETBIOSNAME (NQ_ERR_MODULE | 4001)
- #define NQ_ERR_NBTIMEOUT (NQ_ERR_MODULE | 4002)
- #define NQ_ERR_NBNEGATIVERESPONSE (NQ_ERR_MODULE | 4003)
- #define NQ_ERR_NBHOSTNAMENOTRESOLVED (NQ_ERR_MODULE | 4004)
- #define NQ_ERR_NBCANCELLISTENFAIL (NQ_ERR_MODULE | 4005)
- #define NQ_ERR_NBCKETOVERFLOW (NQ_ERR_MODULE | 4006)
- #define NQ_ERR_NBNOBINDBEFORELISTEN (NQ_ERR_MODULE | 4007)
- #define NQ_ERR_NBILLEGALDATAGRAMSOURCE (NQ_ERR_MODULE | 4008)
- #define NQ_ERR_NBILLEGALDATAGRAMDESTINATION (NQ_ERR_MODULE | 4009)
- #define NQ_ERR_NBINVALIDPARAMETER (NQ_ERR_MODULE | 4010)
- #define NQ_ERR_NBINTERNALERROR (NQ_ERR_MODULE | 4011)
- #define NQ_ERR_NBILLEGALDATAGRAMTYPE (NQ_ERR_MODULE | 4012)
- #define NQ_ERR_NBDDCOMMUNICATIONERROR (NQ_ERR_MODULE | 4013)
- #define NQ_ERR_NBBUFFEROVERFLOW (NQ_ERR_MODULE | 4014)
- #define NQ_ERR_NBRELEASENAMEFAIL (NQ_ERR_MODULE | 4015)
- #define NQ_ERR_NBLISTENFAIL (NQ_ERR_MODULE | 4016)
- #define NQ_ERR_SOCKETCREATE (NQ_ERR_MODULE | 5000)
- #define NQ_ERR_SOCKETBIND (NQ_ERR_MODULE | 5001)
- #define NQ_ERR_SOCKETNAME (NQ_ERR_MODULE | 5002)
- #define NQ_ERR_SOCKETSEND (NQ_ERR_MODULE | 5003)
- #define NQ_ERR_SOCKETSETOPTION (NQ_ERR_MODULE | 5004)
- #define NQ_ERR_SOCKETADDRESS (NQ_ERR_MODULE | 5005)
- #define NQ_ERR_CAPTUREZERO (NQ_ERR_MODULE | 6000)
- #define NQ_ERR_INVALIDUINT32SIZE (NQ_ERR_MODULE | 6001)
- #define NQ_ERR_RECONNECTREQUIRED (NQ_ERR_MODULE | 6002)
- #define NQ_ERR_SIZEERROR (NQ_ERR_MODULE | 6003)
- #define NQ_ERR_SEEKERROR (NQ_ERR_MODULE | 6004)
- #define NQ_ERR_ATTRERROR (NQ_ERR_MODULE | 6005)
- #define NQ_ERR_KERBEROSERROR (NQ_ERR_MODULE | 7000)
- #define NQ_ERR_RAPGENERALERROR (NQ_ERR_MODULE | 8000)
- #define NQ_ERR_RAPACCESSDENIED (NQ_ERR_MODULE | 8001)
- #define NQ_ERR_RAPMOREDATA (NQ_ERR_MODULE | 8002)

- #define [NQ_ERR_RAPSERVERNOTSTARTED](#) ([NQ_ERR_MODULE](#) | 8003)
- #define [NQ_ERR_RAPBADTRANSACTCONFIG](#) ([NQ_ERR_MODULE](#) | 8004)
- #define [NQ_ERR_RAPSERVICENOTINSTALLED](#) ([NQ_ERR_MODULE](#) | 8005)
- #define [NQ_ERR_RAPDEVICENOTCONNECTED](#) ([NQ_ERR_MODULE](#) | 8006)
- #define [NQ_ERR_RAPNOTACTIVEINDOMAIN](#) ([NQ_ERR_MODULE](#) | 8007)
- #define [NQ_ERR_RAPNOBROWSERSERVERSFFOUND](#) ([NQ_ERR_MODULE](#) | 8008)
- #define [NQ_ERR_ICONVERROR](#) ([NQ_ERR_MODULE](#) | 9000)
- #define [NQ_ERR_ICONVOPEN](#) ([NQ_ERR_MODULE](#) | 9001)
- #define [UD_NQ_MAXPWDLEN](#) 257
- #define [NQ_CS_PWDFOUND](#) 0
- #define [NQ_CS_PWDNOAUTH](#) 1
- #define [NQ_CS_PWDNOUSER](#) 2
- #define [NQ_CS_PWDLMHASH](#) 3
- #define [NQ_CS_PWDANY](#) 4
- #define [UD_LOG_MODULE_CS](#) 1
- #define [UD_LOG_MODULE_CC](#) 2
- #define [UD_LOG_CLASS_GEN](#) 1
- #define [UD_LOG_CLASS_FILE](#) 2
- #define [UD_LOG_CLASS_SHARE](#) 3
- #define [UD_LOG_CLASS_USER](#) 4
- #define [UD_LOG_CLASS_CONNECTION](#) 5
- #define [UD_LOG_GEN_START](#) 1
- #define [UD_LOG_GEN_STOP](#) 2
- #define [UD_LOG_GEN_NAMECONFLICT](#) 3
- #define [UD_LOG_GEN_NAMEREGFAIL](#) 4
- #define [UD_LOG_FILE_CREATE](#) 1
- #define [UD_LOG_FILE_OPEN](#) 2
- #define [UD_LOG_FILE_CLOSE](#) 3
- #define [UD_LOG_FILE_DELETE](#) 4
- #define [UD_LOG_FILE_RENAME](#) 5
- #define [UD_LOG_FILE_ATTRIBGET](#) 6
- #define [UD_LOG_FILE_ATTRIBSET](#) 7
- #define [UD_LOG_FILE_SIZESET](#) 8
- #define [UD_LOG_FILE_VOLUMEINFO](#) 9
- #define [UD_LOG_FILE_QUERYDIRECTORY](#) 10
- #define [UD_LOG_FILE_SEEK](#) 11
- #define [UD_LOG_FILE_LOCK](#) 12
- #define [UD_LOG_FILE_UNLOCK](#) 13
- #define [UD_LOG_SHARE_CONNECT](#) 1
- #define [UD_LOG_SHARE_DISCONNECT](#) 2
- #define [UD_LOG_USER_LOGON](#) 1
- #define [UD_LOG_USER_LOGOFF](#) 2
- #define [UD_LOG_CONNECTION_CONNECT](#) 1
- #define [UD_LOG_CONNECTION_DISCONNECT](#) 2

Typedefs

- typedef char [NQ_CHAR](#)
- typedef unsigned char [NQ_BYTE](#)
- typedef int [NQ_INT](#)
- typedef unsigned int [NQ_UINT](#)
- typedef int [NQ_BOOL](#)
- typedef [NQ_UINT](#) [NQ_INDEX](#)
- typedef [NQ_UINT](#) [NQ_COUNT](#)

- typedef short [NQ_INT16](#)
- typedef long [NQ_INT32](#)
- typedef unsigned short [NQ_UINT16](#)
- typedef unsigned long [NQ_UINT32](#)
- typedef [NQ_UINT16](#) [NQ_WCHAR](#)
- typedef void * [NQ_HANDLE](#)
- typedef [NQ_INT](#) [NQ_STATUS](#)
- typedef [NQ_UINT16](#) [NQ_PORT](#)
- typedef [NQ_UINT32](#) [NQ_IPADDRESS4](#)
- typedef [NQ_UINT16](#) [NQ_IPADDRESS6](#) [8]
- typedef unsigned long [NQ_ULONG](#)
- typedef float [NQ_FLOAT](#)
- typedef [NQ_UINT64](#) [NQ_TIME](#)

Functions

- [NQ_STATUS](#) [udInit](#) (void)
- void [udStop](#) (void)
- void [udResetServerParams](#) (void)
- void [udCifsServerStarted](#) (void)
- void [udCifsServerClosed](#) (void)
- void [udNetBiosDaemonStarted](#) (void)
- void [udNetBiosDaemonClosed](#) (void)
- void [udGetScopeId](#) ([NQ_WCHAR](#) *buffer)
- void [udGetWins](#) ([NQ_WCHAR](#) *pServers)
- void [udGetDomain](#) ([NQ_WCHAR](#) *buffer, [NQ_BOOL](#) *isWorkgroup)
- void [udGetDnsParams](#) ([NQ_WCHAR](#) *domain, [NQ_WCHAR](#) *servers)
- [NQ_BOOL](#) [udGetCredentials](#) (const void *resource, [NQ_WCHAR](#) *userName, [NQ_WCHAR](#) *password, [NQ_WCHAR](#) *domain)
- [NQ_INT](#) [udGetPassword](#) (const [NQ_WCHAR](#) *userName, [NQ_CHAR](#) *password, [NQ_INT](#) *pwdIsHashed, [NQ_UINT32](#) *userNumber)
- [NQ_UINT32](#) [udGetSmbError](#) ([NQ_UINT32](#) sysErr)
- [NQ_UINT32](#) [udNqToSystemError](#) ([NQ_STATUS](#) nqErr)
- [NQ_INT](#) [udGetTransportPriority](#) ([NQ_UINT](#) transport)
- void [udGetServerComment](#) ([NQ_WCHAR](#) *buffer)
- void [udNetBiosDataIn](#) (void)
- void [udServerDataIn](#) (void)
- void [udServerShareConnect](#) (const [NQ_WCHAR](#) *share)
- void [udServerShareDisconnect](#) (const [NQ_WCHAR](#) *share)
- [NQ_INT](#) [udGetCodePage](#) (void)
- void [udSetCodePage](#) ([NQ_INT](#) codePage)
- [NQ_PORT](#) [udGetPort](#) ([NQ_PORT](#) port)
- void [udGetComputerId](#) ([NQ_BYTE](#) *buf)
- [NQ_COUNT](#) [udGetUserCount](#) (void)
- [NQ_BOOL](#) [udGetUserRidByName](#) (const [NQ_WCHAR](#) *name, [NQ_UINT32](#) *rid)
- [NQ_BOOL](#) [udGetUserNameByRid](#) ([NQ_UINT32](#) rid, [NQ_WCHAR](#) *nameBuffer, [NQ_WCHAR](#) *fullNameBuffer)
- [NQ_BOOL](#) [udGetUserInfo](#) ([NQ_UINT](#) index, [NQ_UINT32](#) *rid, [NQ_WCHAR](#) *name, [NQ_WCHAR](#) *fullName, [NQ_WCHAR](#) *description)
- [NQ_BOOL](#) [udSetUserAsAdministrator](#) ([NQ_UINT32](#) rid, [NQ_BOOL](#) isAdmin)
- [NQ_BOOL](#) [udSetUserInfo](#) ([NQ_UINT32](#) rid, const [NQ_WCHAR](#) *name, const [NQ_WCHAR](#) *fullName, const [NQ_WCHAR](#) *descr, const [NQ_WCHAR](#) *password)
- [NQ_BOOL](#) [udCreateUser](#) (const [NQ_WCHAR](#) *name, const [NQ_WCHAR](#) *fullName, const [NQ_WCHAR](#) *description)

- `NQ_BOOL udDeleteUserByRid (NQ_UINT32 rid)`
- `NQ_BOOL udSaveShareInformation (const NQ_WCHAR *name, const NQ_WCHAR *newName, const NQ_WCHAR *newMap, const NQ_WCHAR *newDescription)`
- `NQ_BOOL udRemoveShare (const NQ_WCHAR *name)`
- `void udNetBiosError (NQ_STATUS cause, const NQ_CHAR *name)`
- `NQ_BOOL udGetComputerSecret (NQ_BYTE **secret)`
- `NQ_BOOL udGetComputerSecretByDomain (NQ_BYTE *secret, const NQ_WCHAR *domainDNS, NQ_WCHAR *domainNB)`
- `void udSetComputerSecret (NQ_BYTE *secret)`
- `void udSetComputerSecretByDomain (const NQ_BYTE *secret, const NQ_WCHAR *domainDNS, const NQ_WCHAR *domainNB)`
- `void udGetFileSystemName (const NQ_WCHAR *shareName, const NQ_WCHAR *sharePath, NQ_WCHAR *fileName)`
- `void udEventLog (NQ_UINT module, NQ_UINT eventClass, NQ_UINT type, const NQ_WCHAR *userName, const NQ_IPADDRESS *plp, NQ_UINT32 status, const NQ_BYTE *parameters)`
- `NQ_BYTE * udAllocateBuffer (NQ_INT idx, NQ_COUNT numBufs, NQ_UINT bufferSize)`
- `void udReleaseBuffer (NQ_INT idx, NQ_COUNT numBufs, NQ_BYTE *buffAddr, NQ_UINT bufferSize)`
- `NQ_INT udGetSecurityDescriptor (NQ_INT file, NQ_UINT32 information, void *buffer)`
- `NQ_STATUS udSetSecurityDescriptor (NQ_INT file, NQ_UINT32 information, const void *buffer, NQ_UINT32 len)`
- `NQ_COUNT udLoadShareSecurityDescriptor (const NQ_WCHAR *shareName, NQ_BYTE *buffer, NQ_COUNT bufferSize)`
- `void udSaveShareSecurityDescriptor (const NQ_WCHAR *shareName, const NQ_BYTE *sd, NQ_COUNT sdLen)`
- `NQ_BOOL udGetNextShareEx (NQ_WCHAR *name, NQ_WCHAR *map, NQ_BOOL *isPrinter, NQ_WCHAR *description, NQ_BOOL *isEncrypted)`
- `NQ_BOOL udGetNextMount (NQ_WCHAR *name, NQ_WCHAR *map)`
- `NQ_BOOL udGetMessageSigningPolicy (NQ_INT *policy)`
- `NQ_BOOL udGetGlobalEncryption (void)`
- `NQ_BOOL udGetInternalCapture (void)`
- `void udGetCaptureFileBaseFolder (NQ_CHAR *buffer, NQ_UINT size)`
- `NQ_BOOL udGetInternalTrace (void)`
- `NQ_INT udGetTaskPriorities (void)`
- `void udGetDriverName (NQ_CHAR *buffer)`
- `void udGetHostName (NQ_CHAR *buffer, NQ_UINT length)`
- `void udGetLogFileBaseFolder (NQ_CHAR *buffer, NQ_UINT size)`
- `void udGetMaxDiskSize (NQ_UINT32 *size)`

6.9.1 Macro Definition Documentation

6.9.1.1 `#define UD_DNS_SERVERSTRINGSIZE (16 * sizeof(NQ_WCHAR) * UD_NQ_MAXDNSSERVERS * UD_NS_MAXADAPTERS)`

6.9.1.2 `#define NQ_IOBuf NQ_BYTE*`

6.9.1.3 `#define NQ_IOBuffer NQ_IOVecBuffer *`

Abstract data buffer According to compile-time parameters, this type may evaluate to one of the following: Plain buffer. Data is placed as a contiguous array of bytes. IOVEC scatter-gather structure. NQ manipulates this type the same manner regardless of its implementation.

6.9.1.4 `#define CMIOBufBlob CMIOVecBlob`

6.9.2 Typedef Documentation

6.9.2.1 `typedef char NQ_CHAR`

6.9.2.2 `typedef unsigned char NQ_BYTE`

6.9.2.3 `typedef int NQ_INT`

6.9.2.4 `typedef unsigned int NQ_UINT`

6.9.2.5 `typedef int NQ_BOOL`

6.9.2.6 `typedef NQ_UINT NQ_INDEX`

6.9.2.7 `typedef NQ_UINT NQ_COUNT`

6.9.2.8 `typedef short NQ_INT16`

6.9.2.9 `typedef long NQ_INT32`

6.9.2.10 `typedef unsigned short NQ_UINT16`

6.9.2.11 `typedef unsigned long NQ_UINT32`

6.9.2.12 `typedef NQ_UINT16 NQ_WCHAR`

6.9.2.13 `typedef void* NQ_HANDLE`

6.9.2.14 `typedef NQ_INT NQ_STATUS`

6.9.2.15 `typedef NQ_UINT16 NQ_PORT`

6.9.2.16 `typedef NQ_UINT32 NQ_IPADDRESS4`

6.9.2.17 `typedef NQ_UINT16 NQ_IPADDRESS6[8]`

6.9.2.18 `typedef unsigned long NQ_ULONG`

6.9.2.19 `typedef float NQ_FLOAT`

6.9.2.20 `typedef NQ_UINT64 NQ_TIME`

Index

- a2uTab
 - CMCodepage, [238](#)
- AM_CREDENTIALS_PASSWORD_MD4
 - Security Password Type, [11](#)
- AM_CREDENTIALS_PASSWORD_PLAINTEXT
 - Security Password Type, [11](#)
- AM_CRYPTER_LM
 - Encryption Algorithms, [9](#)
- AM_CRYPTER_LM2
 - Encryption Algorithms, [9](#)
- AM_CRYPTER_NONE
 - Encryption Algorithms, [9](#)
- AM_CRYPTER_NTLM
 - Encryption Algorithms, [9](#)
- AM_CRYPTER_NTLM2
 - Encryption Algorithms, [9](#)
- AM_MAXSECURITYLEVEL
 - Authentication, [6](#)
- AM_MECH_KERBEROS
 - Security Mechanisms, [10](#)
- AM_MECH_NTLMSSP
 - Security Mechanisms, [10](#)
- AMCredentials, [221](#)
 - domain, [221](#)
 - password, [221](#)
 - type, [221](#)
 - user, [221](#)
- AMCredentialsA, [222](#)
 - domain, [222](#)
 - password, [222](#)
 - type, [222](#)
 - user, [222](#)
- AMDomain, [222](#)
 - name, [223](#)
 - realm, [223](#)
- AMDomainA, [223](#)
 - name, [223](#)
 - realm, [223](#)
- access
 - EVCSOpenFileEntry, [245](#)
 - UDFileAccessEvent, [266](#)
- activationPriority
 - CMResolverRegisteredMethodDescription, [242](#)
- addr
 - NQ_IPADDRESS, [249](#)
 - NQ_IPADDRESSVR, [250](#)
- aes128ccmDecryption
 - CMCrypterList, [240](#)
- aes128ccmEncryption
 - CMCrypterList, [240](#)
- aes128cmac
 - CMCrypterList, [239](#)
- aes128gcmDecryption
 - CMCrypterList, [240](#)
- aes128gcmEncryption
 - CMCrypterList, [240](#)
- allocSizeHigh
 - SYFileInformation, [254](#)
- allocSizeLow
 - SYFileInformation, [254](#)
- allocationSize
 - CCFileInfo, [224](#)
- allocationSizeHigh
 - CCFileInfo_t, [226](#)
 - CCFindFileData, [229](#)
 - CCFindFileDataA, [230](#)
- allocationSizeLow
 - CCFileInfo_t, [226](#)
 - CCFindFileData, [228](#)
 - CCFindFileDataA, [230](#)
- amClientDefineLevel
 - Authentication, [7](#)
- amSetNonSecureAuthentication
 - Authentication, [8](#)
- amShutdown
 - Authentication, [7](#)
- amStart
 - Authentication, [7](#)
- amapi.h, [269](#)
- Anonymous, [115](#)
 - csSetAnonymousAllowed, [115](#)
- ansiToFfs
 - CMCodepage, [238](#)
- attributes
 - CCFileInfo, [224](#)
 - CCFileInfo_t, [226](#)
 - SYFileInformation, [254](#)
 - SYPrinterDriver, [256](#)
 - SYPrinterInfo, [257](#)
- Authentication, [6](#)
 - AM_MAXSECURITYLEVEL, [6](#)
 - amClientDefineLevel, [7](#)
 - amSetNonSecureAuthentication, [8](#)
 - amShutdown, [7](#)
 - amStart, [7](#)
- Authentication Level, [22](#)
 - NQ_CC_AUTH_LM_AND_NTLM, [22](#)
 - NQ_CC_AUTH_NTLM, [22](#)

NQ_CC_AUTH_NTLM_V2, 22
 NQ_CC_AUTH_SPNEGO_KERBEROS, 22
 Authentication Statuses, 196
 NQ_CS_PWDANY, 196
 NQ_CS_PWDFOUND, 196
 NQ_CS_PWDLMDHASH, 196
 NQ_CS_PWDNOAUTH, 196
 NQ_CS_PWDNOUSER, 196
 Auxiliary, 105
 cmNetBiosNameReload, 105
 Auxiliary methods, 80
 ccCloseAllConnections, 80
 ccCloseHiddenConnections, 80
 ccDump, 80
 ccThreadSubscribe, 81
 ccThreadUnsubscribe, 81
 ccTransportRestartRecieveThread, 80
 averagePpm
 SYPrinterInfo, 258
 before
 UDFileAccessEvent, 266
 bitsPerPel
 SYDeviceMode, 252
 blockSizeHigh
 SYVolumeInformation, 264
 blockSizeLow
 SYVolumeInformation, 264
 blocksPerUnitHigh
 SYVolumeInformation, 264
 blocksPerUnitLow
 SYVolumeInformation, 264
 bottom
 SYPrintRect, 261
 buildVersion
 SYPrinterInfo, 258
 bytesPerSector
 CCVolumeInfo, 237
 CC_CAPABILITY_DFS
 Server capabilities, 24
 CC_CAPABILITY_DIRECTORY_LEASING
 Server capabilities, 24
 CC_CAPABILITY_ENCRYPTION
 Server capabilities, 24
 CC_CAPABILITY_LARGE_MTU
 Server capabilities, 24
 CC_CAPABILITY_LEASING
 Server capabilities, 24
 CC_CAPABILITY_MULTI_CHANNEL
 Server capabilities, 24
 CC_CAPABILITY_PERSISTENT_HANDLES
 Server capabilities, 24
 CC_LOGON_TYPE_ANONYMOUS
 User logon type, 23
 CC_LOGON_TYPE_GUEST
 User logon type, 23
 CC_LOGON_TYPE_REGULAR
 User logon type, 23
 CC_SIGNING_ENABLED
 signing policy, 25
 CC_SIGNING_REQUIRED
 signing policy, 25
 CC_SMB100
 Dialects Operations, 74
 CC_SMB202
 Dialects Operations, 74
 CC_SMB210
 Dialects Operations, 74
 CC_SMB300
 Dialects Operations, 74
 CC_SMB302
 Dialects Operations, 74
 CC_SMB311
 Dialects Operations, 74
 CCFileInfo, 223
 allocationSize, 224
 attributes, 224
 changeTime, 224
 creationTime, 224
 endOfFile, 224
 fileIndex, 224
 lastAccessTime, 224
 lastWriteTime, 224
 numberOfLinks, 224
 volumeld, 225
 CCFileInfo_t, 225
 allocationSizeHigh, 226
 allocationSizeLow, 226
 attributes, 226
 creationTimeHigh, 225
 creationTimeLow, 225
 fileIndexHigh, 226
 fileIndexLow, 226
 fileSizeHigh, 226
 fileSizeLow, 226
 lastAccessTimeHigh, 225
 lastAccessTimeLow, 225
 lastWriteTimeHigh, 226
 lastWriteTimeLow, 226
 numberOfLinks, 226
 volumeSerialNumber, 226
 CCFileTime, 227
 timeHigh, 227
 timeLow, 227
 CCFindFileData, 227
 allocationSizeHigh, 229
 allocationSizeLow, 228
 changeTimeHigh, 228
 changeTimeLow, 228
 creationTimeHigh, 228
 creationTimeLow, 228
 fileAttributes, 228
 fileName, 229
 fileNameLength, 229
 fileSizeHigh, 228
 fileSizeLow, 228

- lastAccessTimeHigh, [228](#)
- lastAccessTimeLow, [228](#)
- lastWriteTimeHigh, [228](#)
- lastWriteTimeLow, [228](#)
- CCFindFileDataA, [229](#)
 - allocationSizeHigh, [230](#)
 - allocationSizeLow, [230](#)
 - changeTimeHigh, [230](#)
 - changeTimeLow, [230](#)
 - creationTimeHigh, [230](#)
 - creationTimeLow, [229](#)
 - fileAttributes, [229](#)
 - fileName, [231](#)
 - fileNameLength, [230](#)
 - fileSizeHigh, [230](#)
 - fileSizeLow, [230](#)
 - lastAccessTimeHigh, [230](#)
 - lastAccessTimeLow, [230](#)
 - lastWriteTimeHigh, [230](#)
 - lastWriteTimeLow, [230](#)
- CCMountInfo, [231](#)
 - creationTime, [231](#)
 - serverInfo, [231](#)
 - serverName, [231](#)
 - statistics, [231](#)
 - userLogonType, [231](#)
- CCMountInfoA, [232](#)
 - creationTime, [232](#)
 - serverInfo, [232](#)
 - serverName, [232](#)
 - statistics, [232](#)
 - userLogonType, [232](#)
- CCNetDomainItem, [233](#)
 - comment, [233](#)
 - majorVersion, [233](#)
 - minorVersion, [233](#)
 - name, [233](#)
 - type, [233](#)
- CCNetServerItem, [233](#)
 - comment, [234](#)
 - majorVersion, [234](#)
 - minorVersion, [234](#)
 - name, [234](#)
 - type, [234](#)
- CCNetShareItem, [234](#)
 - comment, [234](#)
 - name, [234](#)
 - type, [234](#)
- CCNetShareItemA, [235](#)
 - comment, [235](#)
 - name, [235](#)
 - type, [235](#)
- CCSStatistics, [236](#)
 - readAmount, [236](#)
 - writeAmount, [236](#)
- CCServerInfo, [235](#)
 - dialect, [236](#)
 - ips, [236](#)
 - numIps, [236](#)
 - serverCapabilities, [236](#)
 - serverSigningPolicy, [236](#)
- CCServerSideCopyCallback
 - File Operations, [38](#)
- CCSmbDialect
 - Dialects Operations, [74](#)
- CCVolumeInfo, [237](#)
 - bytesPerSector, [237](#)
 - freeClusters, [237](#)
 - fsType, [237](#)
 - sectorsPerCluster, [237](#)
 - serialNumber, [237](#)
 - totalClusters, [237](#)
- CIFS_ATTR_ALL
 - File attributes, [15](#)
- CIFS_ATTR_ARCHIVE
 - File attributes, [15](#)
- CIFS_ATTR_DIR
 - File attributes, [15](#)
- CIFS_ATTR_HIDDEN
 - File attributes, [15](#)
- CIFS_ATTR_READONLY
 - File attributes, [15](#)
- CIFS_ATTR_SYSTEM
 - File attributes, [15](#)
- CIFS_ATTR_VOLUME
 - File attributes, [15](#)
- cJobs
 - SYPrinterInfo, [257](#)
- CLASS_ERR_MODULE
 - SMB Errors, [145](#)
- CM_NB_GROUP
 - NetBIOS Name Registration, [120](#)
- CM_NB_POSTFIX_DOMAINCONTROLLER
 - NetBIOS Name Registration, [120](#)
- CM_NB_POSTFIX_DOMAINMASTERBROWSER
 - NetBIOS Name Registration, [120](#)
- CM_NB_POSTFIX_MASTERBROWSER
 - NetBIOS Name Registration, [120](#)
- CM_NB_POSTFIX_SERVER
 - NetBIOS Name Registration, [120](#)
- CM_NB_POSTFIX_WORKSTATION
 - NetBIOS Name Registration, [120](#)
- CM_NB_UNIQUE
 - NetBIOS Name Registration, [120](#)
- CM_PRODUCT_STRING_LEN
 - Product Information, [92](#)
- CM_USERNAMELENGTH
 - Credential fields length, [13](#)
- CMAbstractCcmDecryption
 - Cryptographic, [98](#)
- CMAbstractCcmEncryption
 - Cryptographic, [98](#)
- CMAbstractCipher
 - Cryptographic, [96](#)
- CMAbstractGcmDecryption
 - Cryptographic, [99](#)

CMAbstractGcmEncryption
Cryptographic, [98](#)

CMAbstractHasher
Cryptographic, [96](#)

CMAbstractHasher512
Cryptographic, [97](#)

CMAbstractIOCipher
Cryptographic, [97](#)

CMCodepage, [237](#)
a2uTab, [238](#)
ansiToFs, [238](#)
fsToAnsi, [238](#)
id, [238](#)
toAnsi, [238](#)
toUnicode, [238](#)
toUpper, [238](#)

CMCrypterList, [239](#)
aes128ccmDecryption, [240](#)
aes128ccmEncryption, [240](#)
aes128cmac, [239](#)
aes128gcmDecryption, [240](#)
aes128gcmEncryption, [240](#)
hmacmd5, [239](#)
md4, [239](#)
md5, [239](#)
sha256, [239](#)
sha512, [240](#)

CMIOBufBlob
udapi.h, [297](#)

CMIOVecBlob, [240](#)
data, [240](#)
len, [240](#)

CMProductInfo, [241](#)
checksum, [241](#)
productName, [241](#)
productType, [241](#)
version, [241](#)

CMResolverIpToName
Resolver, [87](#)

CMResolverIpToNameA
Resolver, [87](#)

CMResolverNameToIp
Resolver, [86](#)

CMResolverNameToIpA
Resolver, [86](#)

CMResolverRegisteredMethodDescription, [241](#)
activationPriority, [242](#)
requestByIp, [243](#)
requestByName, [242](#)
responseByIp, [243](#)
responseByName, [242](#)
serverIP, [242](#)
timeout, [242](#)

CSFileData, [243](#)
fileNamePath, [244](#)
IP, [244](#)
isDirectory, [244](#)
userName, [244](#)

cSetPrinter
SYPrinterInfo, [258](#)

ccAddMount
Mount and Server, [27](#)

ccAddMountA
Mount and Server, [27](#)

ccAddMountWithCreds
Mount and Server, [28](#)

ccAddMountWithCredsA
Mount and Server, [28](#)

ccCifsGetSMB1Support
Dialects Operations, [74](#)

ccCifsSetSMB1Support
Dialects Operations, [75](#)

ccCloseAllConnections
Auxiliary methods, [80](#)

ccCloseHandle
File Operations, [44](#)

ccCloseHiddenConnections
Auxiliary methods, [80](#)

ccConfigGetTimeout
Timeout Operations, [55](#)

ccConfigSetTimeout
Timeout Operations, [55](#)

ccCreateDirectory
Directory Operations, [36](#)

ccCreateDirectoryA
Directory Operations, [36](#)

ccCreateFile
File Operations, [39](#)

ccCreateFileA
File Operations, [39](#)

ccDeleteFile
File Operations, [40](#)

ccDeleteFileA
File Operations, [40](#)

ccDomainJoin
Domain Operations, [65](#)

ccDomainJoinA
Domain Operations, [65](#)

ccDomainJoinPersistent
Domain Operations, [66](#)

ccDomainLeave
Domain Operations, [68](#)

ccDomainLeaveA
Domain Operations, [67](#)

ccDomainLogon
Domain Operations, [69](#)

ccDomainLogonA
Domain Operations, [68](#)

ccDomainUpdateSecret
Domain Operations, [67](#)

ccDomainUpdateSecretA
Domain Operations, [66](#)

ccDump
Auxiliary methods, [80](#)

ccErrorToSmbStatus
Errors, [79](#)

ccFindClose	Search Operations, 35
ccFindFirstFile	Search Operations, 34
ccFindFirstFileA	Search Operations, 33
ccFindNextFile	Search Operations, 35
ccFindNextFileA	Search Operations, 34
ccFlushFile	File Operations, 44
ccGetDiskFreeSpace	File Info Operations, 48
ccGetDiskFreeSpaceA	File Info Operations, 47
ccGetDiskFreeSpaceEx	File Info Operations, 49
ccGetDiskFreeSpaceExA	File Info Operations, 48
ccGetFileAttributes	File Info Operations, 49
ccGetFileAttributesA	File Info Operations, 49
ccGetFileInformationByHandle	File Info Operations, 53
ccGetFileInformationByName	File Info Operations, 50
ccGetFileInformationByNameA	File Info Operations, 50
ccGetFilePointer	File Info Operations, 46
ccGetFileSize	File Info Operations, 53
ccGetFileTime	File Info Operations, 51
ccGetMaxSecurityLevel	Special Operations, 71
ccGetMinSecurityLevel	Special Operations, 71
ccGetMountInformation	Mount information, 31
ccGetMountInformationA	Mount information, 31
ccGetNTStatus	Errors, 79
ccGetServerId	Mount and Server, 30
ccGetServerIdA	Mount and Server, 30
ccGetSigningPolicy	Special Operations, 70
ccInit	Initialization and Shutdown, 26
ccIsExclusiveAccessToFile	Special Operations, 73
ccIsExclusiveAccessToFileA	Special Operations, 72
ccIsInitialized	Initialization and Shutdown, 26
ccMoveFile	File Operations, 40
ccMoveFileA	File Operations, 40
ccNetworkCacheSet	Network Browsing, 58
ccNetworkCloseHandle	Network Browsing, 63
ccNetworkEnumerateDomains	Network Browsing, 58
ccNetworkEnumerateServers	Network Browsing, 60
ccNetworkEnumerateServersA	Network Browsing, 59
ccNetworkEnumerateShares	Network Browsing, 61
ccNetworkEnumerateSharesA	Network Browsing, 60
ccNetworkGetDefaultDomain	Network Browsing, 58
ccNetworkGetDefaultDomainA	Network Browsing, 58
ccNetworkGetNextDomainItem	Network Browsing, 59
ccNetworkGetNextItemName	Network Browsing, 61
ccNetworkGetNextItemNameA	Network Browsing, 61
ccNetworkGetNextServerItem	Network Browsing, 60
ccNetworkGetNextShareItem	Network Browsing, 62
ccNetworkGetShareInfo	Network Browsing, 63
ccNetworkGetShareInfoA	Network Browsing, 62
ccNetworkResetHandle	Network Browsing, 63
ccNetworkSetDefaultDomain	Network Browsing, 56
ccNetworkSetDefaultDomainA	Network Browsing, 56
ccReadFile	File Operations, 42
ccReadFileAsync	File Operations, 42
ccRemoveDirectory	Directory Operations, 37
ccRemoveDirectoryA	Directory Operations, 36
ccRemoveMount	Mount and Server, 29
ccRemoveMountA	Mount and Server, 29
ccResetCredentails	User Credentials, 76

ccResetCredentailsA
 User Credentials, 76
ccServerSideDataCopy
 File Info Operations, 47
ccServerSideDataCopyA
 File Info Operations, 46
ccSetExclusiveAccessToFile
 Special Operations, 72
ccSetExclusiveAccessToFileA
 Special Operations, 71
ccSetFileAttributes
 File Info Operations, 50
ccSetFileAttributesA
 File Info Operations, 49
ccSetFilePointer
 File Info Operations, 46
ccSetFileSizeByHandle
 File Info Operations, 51
ccSetFileSizeByName
 File Info Operations, 51
ccSetFileSizeByNameA
 File Info Operations, 51
ccSetFileTime
 File Info Operations, 53
ccSetGetBackupListTimeout
 Timeout Operations, 55
ccSetMaxSecurityLevel
 Special Operations, 71
ccSetMinSecurityLevel
 Special Operations, 70
ccSetSigningPolicy
 Special Operations, 70
ccSetSmbDialect
 Dialects Operations, 74
ccShutdown
 Initialization and Shutdown, 26
ccThreadSubscribe
 Auxiliary methods, 81
ccThreadUnsubscribe
 Auxiliary methods, 81
ccTransportRestartRecieveThread
 Auxiliary methods, 80
ccValidateUser
 User Credentials, 77
ccValidateUserA
 User Credentials, 77
ccValidateUserByCredentials
 User Credentials, 78
ccValidateUserByCredentialsA
 User Credentials, 77
ccWriteFile
 File Operations, 43
ccWriteFileAsync
 File Operations, 43
ccapi.h, 270
changeTime
 CCFileInfo, 224
changeTimeHigh
 CCFindFileData, 228
 CCFindFileDataA, 230
changeTimeLow
 CCFindFileData, 228
 CCFindFileDataA, 230
checksum
 CMPProductInfo, 241
Cleanup, 114
 csCleanUserServerConnections, 114
 csCleanUserServerConnectionsA, 114
Client, 12
cmCodepageAdd
 Codepage, 93
cmCodepageRemove
 Codepage, 93
cmDnsGetNumDnsServers
 DNS and WINS, 103
cmDnsGetServer
 DNS and WINS, 102
cmDnsSetDomain
 DNS and WINS, 104
cmDnsSetDomainA
 DNS and WINS, 104
cmDnsSetServers
 DNS and WINS, 102
cmDnsSetServersA
 DNS and WINS, 102
cmExit
 Initialization and Shutdown, 83
cmGetCurrentProductInformation
 Product Information, 92
cmInit
 Initialization and Shutdown, 83
cmMultiByteToWideChar
 Codepage, 95
cmNetBiosGetNumWinsServers
 DNS and WINS, 104
cmNetBiosGetWins
 DNS and WINS, 103
cmNetBiosNameReload
 Auxiliary, 105
cmNetBiosSetWins
 DNS and WINS, 103
cmNetBiosSetWinsA
 DNS and WINS, 103
cmResetExternalCrypters
 Cryptographic, 101
cmResolverEnableMethod
 Resolver, 90
cmResolverGetDCName
 Resolver, 90
cmResolverGetHostIps
 Resolver, 89
cmResolverGetHostName
 Resolver, 89
cmResolverRegisterExternalMethod
 Resolver, 91
cmResolverSetExternal

- CMAbstractHasher512, [97](#)
- CMAbstractIOCipher, [97](#)
- cmResetExternalCrypters, [101](#)
- cmSetExternalCrypters, [100](#)
- csAddShare
 - Shares Management, [108](#)
- csAddShareA
 - Shares Management, [108](#)
- csCleanUserServerConnections
 - Cleanup, [114](#)
- csCleanUserServerConnectionsA
 - Cleanup, [114](#)
- csEnumerateOpenFiles
 - Open Files Enumeration, [112](#)
- csEnumerateOpenFilesA
 - Open Files Enumeration, [112](#)
- csRemoveShare
 - Shares Management, [109](#)
- csRemoveShareA
 - Shares Management, [109](#)
- csSetAnonymousAllowed
 - Anonymous, [115](#)
- csSetShareEncryption
 - Shares Management, [110](#)
- csSetShareEncryptionA
 - Shares Management, [110](#)
- csStart
 - Initialization and Shutdown, [107](#)
- csStop
 - Initialization and Shutdown, [107](#)
- csapi.h, [276](#)
- DNS and WINS, [102](#)
 - cmDnsGetNumDnsServers, [103](#)
 - cmDnsGetServer, [102](#)
 - cmDnsSetDomain, [104](#)
 - cmDnsSetDomainA, [104](#)
 - cmDnsSetServers, [102](#)
 - cmDnsSetServersA, [102](#)
 - cmNetBiosGetNumWinsServers, [104](#)
 - cmNetBiosGetWins, [103](#)
 - cmNetBiosSetWins, [103](#)
 - cmNetBiosSetWinsA, [103](#)
- DOMAIN_LENGTH
 - Credential fields length, [13](#)
- DOS_ERRAlreadyexists
 - SMB Errors, [146](#)
- DOS_ERRBadaccess
 - SMB Errors, [146](#)
- DOS_ERRBaddata
 - SMB Errors, [146](#)
- DOS_ERRBaddirectory
 - SMB Errors, [146](#)
- DOS_ERRBaddrive
 - SMB Errors, [146](#)
- DOS_ERRBadenv
 - SMB Errors, [146](#)
- DOS_ERRBadfid
 - SMB Errors, [146](#)

DOS_ERRbadfile
 SMB Errors, [145](#)
 DOS_ERRbadformat
 SMB Errors, [146](#)
 DOS_ERRbadfunc
 SMB Errors, [145](#)
 DOS_ERRbadmcb
 SMB Errors, [146](#)
 DOS_ERRbadmem
 SMB Errors, [146](#)
 DOS_ERRbadpath
 SMB Errors, [145](#)
 DOS_ERRbadpipe
 SMB Errors, [146](#)
 DOS_ERRbadshare
 SMB Errors, [146](#)
 DOS_ERRdiffdevice
 SMB Errors, [146](#)
 DOS_ERRdirnotempty
 SMB Errors, [146](#)
 DOS_ERRdentsupportipc
 SMB Errors, [146](#)
 DOS_ERRfileexists
 SMB Errors, [146](#)
 DOS_ERRinsufficientbuffer
 SMB Errors, [146](#)
 DOS_ERRinvalidname
 SMB Errors, [146](#)
 DOS_ERRlock
 SMB Errors, [146](#)
 DOS_ERRmoredata
 SMB Errors, [147](#)
 DOS_ERRnoaccess
 SMB Errors, [146](#)
 DOS_ERRnofids
 SMB Errors, [146](#)
 DOS_ERRnofiles
 SMB Errors, [146](#)
 DOS_ERRnomem
 SMB Errors, [146](#)
 DOS_ERRnoshare
 SMB Errors, [146](#)
 DOS_ERRnotconnected
 SMB Errors, [146](#)
 DOS_ERRpipebusy
 SMB Errors, [146](#)
 DOS_ERRpipeclosing
 SMB Errors, [146](#)
 DOS_ERRremcd
 SMB Errors, [146](#)
 data
 CMIOVecBlob, [240](#)
 NQ_IOVecBuffer, [248](#)
 dataFile
 SYPrinterDriver, [255](#)
 dataType
 SYPrinterInfo, [257](#)
 day
 SYTimeFragments, [263](#)
 defaultDataType
 SYPrinterDriver, [255](#)
 defaultPriority
 SYPrinterInfo, [257](#)
 defaultSource
 SYDeviceMode, [252](#)
 dependentFiles
 SYPrinterDriver, [255](#)
 devMode
 SYPrinterInfo, [257](#)
 deviceNotSelectedTimeout
 SYPrinterInfo, [258](#)
 dialect
 CCServerInfo, [236](#)
 Dialects Operations, [74](#)
 CC_SMB100, [74](#)
 CC_SMB202, [74](#)
 CC_SMB210, [74](#)
 CC_SMB300, [74](#)
 CC_SMB302, [74](#)
 CC_SMB311, [74](#)
 CCSmbDialect, [74](#)
 ccCifsGetSMB1Support, [74](#)
 ccCifsSetSMB1Support, [75](#)
 ccSetSmbDialect, [74](#)
 Directory Operations, [36](#)
 ccCreateDirectory, [36](#)
 ccCreateDirectoryA, [36](#)
 ccRemoveDirectory, [37](#)
 ccRemoveDirectoryA, [36](#)
 displayFlags
 SYDeviceMode, [252](#)
 displayFrequency
 SYDeviceMode, [252](#)
 ditherType
 SYDeviceMode, [253](#)
 documentName
 SYPrintJobInfo, [259](#)
 domain
 AMCredentials, [221](#)
 AMCredentialsA, [222](#)
 Domain Operations, [65](#)
 ccDomainJoin, [65](#)
 ccDomainJoinA, [65](#)
 ccDomainJoinPersistent, [66](#)
 ccDomainLeave, [68](#)
 ccDomainLeaveA, [67](#)
 ccDomainLogon, [69](#)
 ccDomainLogonA, [68](#)
 ccDomainUpdateSecret, [67](#)
 ccDomainUpdateSecretA, [66](#)
 Driver Notification Events, [14](#)
 NOTIFY_FILE_HANDLE_ERROR, [14](#)
 NOTIFY_SRCH_HANDLE_ERROR, [14](#)
 driverDate
 SYPrinterDriver, [255](#)
 driverExtraData

SYDeviceMode, [252](#)
 driverExtraLength
 SYDeviceMode, [252](#)
 driverName
 SYPrinterInfo, [257](#)
 driverPath
 SYPrinterDriver, [255](#)
 driverVersion
 SYDeviceMode, [252](#)
 SYPrinterDriver, [256](#)
 driverVersions
 SYPrinterDriver, [255](#)
 duplex
 SYDeviceMode, [252](#)

 EVCSOpenFileEntry, [244](#)
 access, [245](#)
 fileName, [245](#)
 ip, [245](#)
 shareName, [245](#)
 userName, [245](#)
 EVCSShareConnectionEntry, [245](#)
 ip, [246](#)
 ipc, [246](#)
 printQueue, [246](#)
 shareName, [245](#)
 userName, [245](#)
 Encryption Algorithms, [9](#)
 AM_CRYPTER_LM, [9](#)
 AM_CRYPTER_LM2, [9](#)
 AM_CRYPTER_NONE, [9](#)
 AM_CRYPTER_NTLM, [9](#)
 AM_CRYPTER_NTLM2, [9](#)
 endOfFile
 CCFileInfo, [224](#)
 Errors, [79](#)
 ccErrorToSmbStatus, [79](#)
 ccGetNTStatus, [79](#)
 evcsEnumerateConnectedShares
 Event Log, [116](#)
 evcsEnumerateOpenFiles
 Event Log, [116](#)
 Event Log, [116](#)
 evcsEnumerateConnectedShares, [116](#)
 evcsEnumerateOpenFiles, [116](#)
 Event Log Constants, [217](#)
 UD_LOG_CLASS_CONNECTION, [218](#)
 UD_LOG_CLASS_FILE, [217](#)
 UD_LOG_CLASS_GEN, [217](#)
 UD_LOG_CLASS_SHARE, [218](#)
 UD_LOG_CLASS_USER, [218](#)
 UD_LOG_CONNECTION_CONNECT, [220](#)
 UD_LOG_CONNECTION_DISCONNECT, [220](#)
 UD_LOG_FILE_ATTRIBGET, [219](#)
 UD_LOG_FILE_ATTRIBSET, [219](#)
 UD_LOG_FILE_CLOSE, [218](#)
 UD_LOG_FILE_CREATE, [218](#)
 UD_LOG_FILE_DELETE, [218](#)
 UD_LOG_FILE_LOCK, [219](#)
 UD_LOG_FILE_OPEN, [218](#)
 UD_LOG_FILE_QUERYDIRECTORY, [219](#)
 UD_LOG_FILE_RENAME, [218](#)
 UD_LOG_FILE_SEEK, [219](#)
 UD_LOG_FILE_SIZESET, [219](#)
 UD_LOG_FILE_UNLOCK, [219](#)
 UD_LOG_FILE_VOLUMEINFO, [219](#)
 UD_LOG_GEN_NAMECONFLICT, [218](#)
 UD_LOG_GEN_NAMEREGFAIL, [218](#)
 UD_LOG_GEN_START, [218](#)
 UD_LOG_GEN_STOP, [218](#)
 UD_LOG_MODULE_CC, [217](#)
 UD_LOG_MODULE_CS, [217](#)
 UD_LOG_SHARE_CONNECT, [219](#)
 UD_LOG_SHARE_DISCONNECT, [219](#)
 UD_LOG_USER_LOGOFF, [219](#)
 UD_LOG_USER_LOGON, [219](#)

 FILE_AM_READ
 File access mode, [16](#)
 FILE_AM_READ_WRITE
 File access mode, [16](#)
 FILE_AM_SPECIAL_MASK
 File access mode, [16](#)
 FILE_AM_WRITE
 File access mode, [16](#)
 FILE_CA_CREATE
 File create action, [19](#)
 FILE_CA_FAIL
 File create action, [19](#)
 FILE_LCL_MIXED
 File locality, [18](#)
 FILE_LCL_RANDOM
 File locality, [18](#)
 FILE_LCL_SEQUENTIAL
 File locality, [18](#)
 FILE_LCL_UNKNOWN
 File locality, [18](#)
 FILE_OA_FAIL
 File open action, [20](#)
 FILE_OA_OPEN
 File open action, [20](#)
 FILE_OA_TRUNCATE
 File open action, [20](#)
 FILE_SM_COMPAT
 File share mode, [17](#)
 FILE_SM_DENY_NONE
 File share mode, [17](#)
 FILE_SM_DENY_READ
 File share mode, [17](#)
 FILE_SM_DENY_WRITE
 File share mode, [17](#)
 FILE_SM_EXCLUSIVE
 File share mode, [17](#)
 fields
 SYDeviceMode, [252](#)
 File access mode, [16](#)
 FILE_AM_READ, [16](#)
 FILE_AM_READ_WRITE, [16](#)

- FILE_AM_SPECIAL_MASK, 16
- FILE_AM_WRITE, 16
- File Attributes, 127
 - SY_ATTR_ARCHIVE, 127
 - SY_ATTR_COMPRESSED, 128
 - SY_ATTR_DEVICE, 127
 - SY_ATTR_DIRECTORY, 127
 - SY_ATTR_ENCRYPTED, 128
 - SY_ATTR_HIDDEN, 127
 - SY_ATTR_NORMAL, 127
 - SY_ATTR_NOT_CONTENT_INDEXED, 128
 - SY_ATTR_OFFLINE, 128
 - SY_ATTR_READONLY, 127
 - SY_ATTR_REPARSE_POINT, 128
 - SY_ATTR_SPARSE_FILE, 128
 - SY_ATTR_SYSTEM, 127
 - SY_ATTR_TEMPORARY, 128
 - SY_ATTR_VOLUME, 127
- File attributes, 15
 - CIFS_ATTR_ALL, 15
 - CIFS_ATTR_ARCHIVE, 15
 - CIFS_ATTR_DIR, 15
 - CIFS_ATTR_HIDDEN, 15
 - CIFS_ATTR_READONLY, 15
 - CIFS_ATTR_SYSTEM, 15
 - CIFS_ATTR_VOLUME, 15
- File create action, 19
 - FILE_CA_CREATE, 19
 - FILE_CA_FAIL, 19
- File Info Operations, 45
 - ccGetDiskFreeSpace, 48
 - ccGetDiskFreeSpaceA, 47
 - ccGetDiskFreeSpaceEx, 49
 - ccGetDiskFreeSpaceExA, 48
 - ccGetFileAttributes, 49
 - ccGetFileAttributesA, 49
 - ccGetFileInformationByHandle, 53
 - ccGetFileInformationByName, 50
 - ccGetFileInformationByNameA, 50
 - ccGetFilePointer, 46
 - ccGetFileSize, 53
 - ccGetFileTime, 51
 - ccServerSideDataCopy, 47
 - ccServerSideDataCopyA, 46
 - ccSetFileAttributes, 50
 - ccSetFileAttributesA, 49
 - ccSetFilePointer, 46
 - ccSetFileSizeByHandle, 51
 - ccSetFileSizeByName, 51
 - ccSetFileSizeByNameA, 51
 - ccSetFileTime, 53
- FileInfo_t, 46
- File Information, 126
- File locality, 18
 - FILE_LCL_MIXED, 18
 - FILE_LCL_RANDOM, 18
 - FILE_LCL_SEQUENTIAL, 18
 - FILE_LCL_UNKNOWN, 18
- File open action, 20
 - FILE_OA_FAIL, 20
 - FILE_OA_OPEN, 20
 - FILE_OA_TRUNCATE, 20
- File Operations, 38
 - CCServerSideCopyCallback, 38
 - ccCloseHandle, 44
 - ccCreateFile, 39
 - ccCreateFileA, 39
 - ccDeleteFile, 40
 - ccDeleteFileA, 40
 - ccFlushFile, 44
 - ccMoveFile, 40
 - ccMoveFileA, 40
 - ccReadFile, 42
 - ccReadFileAsync, 42
 - ccWriteFile, 43
 - ccWriteFileAsync, 43
- File seek constants, 21
 - SEEK_FILE_BEGIN, 21
 - SEEK_FILE_CURRENT, 21
 - SEEK_FILE_END, 21
- File share mode, 17
 - FILE_SM_COMPAT, 17
 - FILE_SM_DENY_NONE, 17
 - FILE_SM_DENY_READ, 17
 - FILE_SM_DENY_WRITE, 17
 - FILE_SM_EXCLUSIVE, 17
- fileAttributes
 - CCFindFileData, 228
 - CCFindFileDataA, 229
- FileDataA_t, 246
 - fileNamePath, 246
 - IP, 246
 - isDirectory, 246
 - userName, 246
- fileIdHigh
 - SYFileInformation, 254
- fileIdLow
 - SYFileInformation, 254
- fileIndex
 - CCFileInfo, 224
- fileIndexHigh
 - CCFileInfo_t, 226
- fileIndexLow
 - CCFileInfo_t, 226
- FileInfo_t
 - File Info Operations, 46
- fileName
 - CCFindFileData, 229
 - CCFindFileDataA, 231
 - EVCSOpenFileEntry, 245
 - UDFileAccessEvent, 265
- fileNameLength
 - CCFindFileData, 229
 - CCFindFileDataA, 230
- fileNamePath
 - CSFileData, 244

FileDataA_t, [246](#)
 fileSizeHigh
 CCFileInfo_t, [226](#)
 CCFindFileData, [228](#)
 CCFindFileDataA, [230](#)
 fileSizeLow
 CCFileInfo_t, [226](#)
 CCFindFileData, [228](#)
 CCFindFileDataA, [230](#)
 fileSizeLow
 CCFileInfo_t, [226](#)
 CCFindFileData, [228](#)
 CCFindFileDataA, [230](#)
 fileSystemIdHigh
 SYVolumeInformation, [264](#)
 fileSystemIdLow
 SYVolumeInformation, [264](#)
 FindFileDataA_t
 Search Operations, [33](#)
 flags
 SYPrinterInfo, [257](#)
 SYPrintFormInfo, [258](#)
 formName
 SYDeviceMode, [252](#)
 freeClusters
 CCVolumeInfo, [237](#)
 freeUnitsHigh
 SYVolumeInformation, [265](#)
 freeUnitsLow
 SYVolumeInformation, [265](#)
 fsToAnsi
 CMCodepage, [238](#)
 fsType
 CCVolumeInfo, [237](#)
 Functions, [197](#)
 udAllocateBuffer, [211](#)
 udCifsServerClosed, [199](#)
 udCifsServerStarted, [198](#)
 udCreateUser, [207](#)
 udDeleteUserByRid, [207](#)
 udEventLog, [211](#)
 udGetCaptureFileBaseFolder, [215](#)
 udGetCodePage, [204](#)
 udGetComputerId, [204](#)
 udGetComputerSecret, [209](#)
 udGetComputerSecretByDomain, [209](#)
 udGetCredentials, [201](#)
 udGetDnsParams, [201](#)
 udGetDomain, [199](#)
 udGetDriverName, [215](#)
 udGetFileSystemName, [211](#)
 udGetGlobalEncryption, [214](#)
 udGetHostName, [215](#)
 udGetInternalCapture, [214](#)
 udGetInternalTrace, [215](#)
 udGetLogFileBaseFolder, [215](#)
 udGetMaxDiskSize, [216](#)
 udGetMessageSigningPolicy, [214](#)
 udGetNextMount, [214](#)
 udGetNextShareEx, [214](#)
 udGetPassword, [201](#)
 udGetPort, [204](#)
 udGetScopeId, [199](#)
 udGetSecurityDescriptor, [213](#)
 udGetServerComment, [203](#)
 udGetSmbError, [202](#)
 udGetTaskPriorities, [215](#)
 udGetTransportPriority, [202](#)
 udGetUserCount, [205](#)
 udGetUserInfo, [205](#)
 udGetUserNameByRid, [205](#)
 udGetUserRidByName, [205](#)
 udGetWins, [199](#)
 udInit, [198](#)
 udLoadShareSecurityDescriptor, [213](#)
 udNetBiosDaemonClosed, [199](#)
 udNetBiosDaemonStarted, [199](#)
 udNetBiosDataIn, [203](#)
 udNetBiosError, [208](#)
 udNqToSystemError, [202](#)
 udReleaseBuffer, [211](#)
 udRemoveShare, [208](#)
 udResetServerParams, [198](#)
 udSaveShareInformation, [207](#)
 udSaveShareSecurityDescriptor, [213](#)
 udServerDataIn, [203](#)
 udServerShareConnect, [203](#)
 udServerShareDisconnect, [203](#)
 udSetCodePage, [204](#)
 udSetComputerSecret, [210](#)
 udSetComputerSecretByDomain, [210](#)
 udSetSecurityDescriptor, [213](#)
 udSetUserAsAdministrator, [206](#)
 udSetUserInfo, [206](#)
 udStop, [198](#)
 globalCounter
 SYPrinterInfo, [257](#)
 HRD_ERRFCBUnavail
 SMB Errors, [148](#)
 HRD_ERRbadcmd
 SMB Errors, [148](#)
 HRD_ERRbadmedia
 SMB Errors, [148](#)
 HRD_ERRbadreq
 SMB Errors, [148](#)
 HRD_ERRbadsector
 SMB Errors, [148](#)
 HRD_ERRbadshare
 SMB Errors, [148](#)
 HRD_ERRbadunit
 SMB Errors, [148](#)
 HRD_ERRdata
 SMB Errors, [148](#)
 HRD_ERRdiskfull
 SMB Errors, [148](#)
 HRD_ERRgeneral
 SMB Errors, [148](#)
 HRD_ERRlock
 SMB Errors, [148](#)

HRD_ERRnopaper
 SMB Errors, [148](#)
 HRD_ERRnotready
 SMB Errors, [148](#)
 HRD_ERRnowrite
 SMB Errors, [148](#)
 HRD_ERRread
 SMB Errors, [148](#)
 HRD_ERRseek
 SMB Errors, [148](#)
 HRD_ERRsharebufexc
 SMB Errors, [148](#)
 HRD_ERRwrite
 SMB Errors, [148](#)
 HRD_ERRwrongdisk
 SMB Errors, [148](#)
 hardwareID
 SYPrinterDriver, [256](#)
 height
 SYPrintSize, [262](#)
 helpFile
 SYPrinterDriver, [255](#)
 high
 NQ_INT64, [247](#)
 NQ_UINT64, [250](#)
 hmacmd5
 CMCrypterList, [239](#)
 hour
 SYTimeFragments, [263](#)
 IP
 CSFileData, [244](#)
 FileDataA_t, [246](#)
 IS_DOS_ERROR
 SMB Errors, [145](#)
 IS_HRD_ERROR
 SMB Errors, [148](#)
 IS_NQ_ERROR
 NQ Error Codes, [181](#)
 IS_SRV_ERROR
 SMB Errors, [147](#)
 icmIntent
 SYDeviceMode, [252](#)
 icmMethod
 SYDeviceMode, [252](#)
 id
 CMCodepage, [238](#)
 NQ_IPADDRESSVR, [250](#)
 SYPrintFormInfo, [258](#)
 SYPrintJobInfo, [259](#)
 imageableArea
 SYPrintFormInfo, [259](#)
 infoLevel
 UDFileAccessEvent, [266](#)
 Initialization and Shutdown, [26](#), [83](#), [107](#), [119](#)
 cclnit, [26](#)
 ccIsInitialized, [26](#)
 ccShutdown, [26](#)
 cmExit, [83](#)
 cmInit, [83](#)
 csStart, [107](#)
 csStop, [107](#)
 NQ_CLIENT, [83](#)
 NQ_NDDAEMON, [83](#)
 NQ_SERVER, [83](#)
 nqStart, [119](#)
 nqStop, [119](#)
 Internal NT Statues, [178](#)
 SMB_STATUS_DISCONNECT, [178](#)
 SMB_STATUS_DONOTRELEASERESPONSE,
 [178](#)
 SMB_STATUS_INTERNAL_BUFFER_TOO_SMA-
 LL, [178](#)
 SMB_STATUS_INTERNAL_RELEASE_REQUES-
 T_RESPONSE, [178](#)
 iov_base
 NQ_IOVEC, [248](#)
 iov_len
 NQ_IOVEC, [248](#)
 iovSegmentCnt
 NQ_IOVECBUF, [248](#)
 iovec
 NQ_IOVECBUF, [248](#)
 ip
 EVCSOpenFileEntry, [245](#)
 EVCSShareConnectionEntry, [246](#)
 ipc
 EVCSShareConnectionEntry, [246](#)
 UDShareAccessEvent, [267](#)
 ips
 CCServerInfo, [236](#)
 isDeleted
 SYFileInformation, [254](#)
 isDirectory
 CSFileData, [244](#)
 FileDataA_t, [246](#)
 lastAccessTime
 CCFileInfo, [224](#)
 SYFileInformation, [254](#)
 lastAccessTimeHigh
 CCFileInfo_t, [225](#)
 CCFindFileData, [228](#)
 CCFindFileDataA, [230](#)
 lastAccessTimeLow
 CCFileInfo_t, [225](#)
 CCFindFileData, [228](#)
 CCFindFileDataA, [230](#)
 lastChangeTime
 SYFileInformation, [253](#)
 lastWriteTime
 CCFileInfo, [224](#)
 SYFileInformation, [254](#)
 lastWriteTimeHigh
 CCFileInfo_t, [226](#)
 CCFindFileData, [228](#)
 CCFindFileDataA, [230](#)
 lastWriteTimeLow

CCFileInfo_t, [226](#)
CCFindFileData, [228](#)
CCFindFileDataA, [230](#)
left
 SYPrintRect, [261](#)
len
 CMIOVecBlob, [240](#)
 NQ_IOVecBuffer, [248](#)
location
 SYPrinterInfo, [257](#)
logPixels
 SYDeviceMode, [252](#)
low
 NQ_INT64, [247](#)
 NQ_UINT64, [250](#)
majorVersion
 CCNetDomainItem, [233](#)
 CCNetServerItem, [234](#)
 SYPrinterInfo, [257](#)
manufacturer
 SYPrinterDriver, [256](#)
manufacturerURL
 SYPrinterDriver, [256](#)
md4
 CMCrypterList, [239](#)
md5
 CMCrypterList, [239](#)
mediaType
 SYDeviceMode, [253](#)
min
 SYTimeFragments, [263](#)
minorVersion
 CCNetDomainItem, [233](#)
 CCNetServerItem, [234](#)
monitorName
 SYPrinterDriver, [255](#)
month
 SYTimeFragments, [263](#)
Mount and Server, [27](#)
 ccAddMount, [27](#)
 ccAddMountA, [27](#)
 ccAddMountWithCreds, [28](#)
 ccAddMountWithCredsA, [28](#)
 ccGetServerId, [30](#)
 ccGetServerIdA, [30](#)
 ccRemoveMount, [29](#)
 ccRemoveMountA, [29](#)
Mount information, [31](#)
 ccGetMountInformation, [31](#)
 ccGetMountInformationA, [31](#)
NOTIFY_FILE_HANDLE_ERROR
 Driver Notification Events, [14](#)
NOTIFY_SRCH_HANDLE_ERROR
 Driver Notification Events, [14](#)
NQ Common, [118](#)
NQ Error Codes, [179](#)
 IS_NQ_ERROR, [181](#)
 NQ_ERR_ACCESS, [187](#)
 NQ_ERR_ACCOUNTDISABLED, [183](#)
 NQ_ERR_ACCOUNTEXPIRED, [183](#)
 NQ_ERR_ACCOUNTLOCKEDOUT, [182](#)
 NQ_ERR_ALREADYEXISTS, [186](#)
 NQ_ERR_BADACCESS, [184](#)
 NQ_ERR_BADCMD, [189](#)
 NQ_ERR_BADCONNECTION, [182](#)
 NQ_ERR_BADDATA, [185](#)
 NQ_ERR_BADDFS, [183](#)
 NQ_ERR_BADDIRECTORY, [185](#)
 NQ_ERR_BADDRIVE, [185](#)
 NQ_ERR_BADENV, [184](#)
 NQ_ERR_BADFID, [184](#)
 NQ_ERR_BADFILE, [184](#)
 NQ_ERR_BADFORMAT, [184](#)
 NQ_ERR_BADFUNC, [184](#)
 NQ_ERR_BADMCB, [184](#)
 NQ_ERR_BADMEDIA, [189](#)
 NQ_ERR_BADMEM, [184](#)
 NQ_ERR_BADPARAM, [181](#)
 NQ_ERR_BADPATH, [184](#)
 NQ_ERR_BADPERMITS, [187](#)
 NQ_ERR_BADPIPE, [186](#)
 NQ_ERR_BADPW, [186](#)
 NQ_ERR_BADREQ, [189](#)
 NQ_ERR_BADSECTOR, [189](#)
 NQ_ERR_BADSHARE, [185](#)
 NQ_ERR_BADUNIT, [189](#)
 NQ_ERR_CONTMPX, [188](#)
 NQ_ERR_DATA, [189](#)
 NQ_ERR_DFSCACHEOVERFLOW, [182](#)
 NQ_ERR_DIFFDEVICE, [185](#)
 NQ_ERR_DIRNOTEMPTY, [186](#)
 NQ_ERR_DISKFULL, [190](#)
 NQ_ERR_DONTSUPPORTIPC, [185](#)
 NQ_ERR_ERROR, [186](#)
 NQ_ERR_FCBUNAVAIL, [190](#)
 NQ_ERR_FILESPECS, [187](#)
 NQ_ERR_FILEXISTS, [185](#)
 NQ_ERR_GENERAL, [190](#)
 NQ_ERR_GETDATA, [181](#)
 NQ_ERR_INSUFFICIENTBUFFER, [186](#)
 NQ_ERR_INVALIDHANDLE, [182](#)
 NQ_ERR_INVALIDLOGONHOURS, [183](#)
 NQ_ERR_INVALIDMODE, [181](#)
 NQ_ERR_INVALIDNAME, [186](#)
 NQ_ERR_INVALIDNAMELEN, [186](#)
 NQ_ERR_INVDEVICE, [187](#)
 NQ_ERR_INVFID, [187](#)
 NQ_ERR_INVNETNAME, [187](#)
 NQ_ERR_INVTID, [187](#)
 NQ_ERR_INVUID, [188](#)
 NQ_ERR_IOTIMEOUT, [183](#)
 NQ_ERR_LOCK, [185](#)
 NQ_ERR_LOGONFAILURE, [182](#)
 NQ_ERR_MODULE, [181](#)
 NQ_ERR_MOREDATA, [186](#)

NQ_ERR_MOUNTERROR, 181
 NQ_ERR_MSGOFF, 188
 NQ_ERR_NEGOTIATEFAILED, 182
 NQ_ERR_NETWORKERROR, 182
 NQ_ERR_NOACCESS, 184
 NQ_ERR_NOFIDS, 184
 NQ_ERR_NOFILES, 185
 NQ_ERR_NOMEM, 184
 NQ_ERR_NOPAPER, 189
 NQ_ERR_NORESOURCE, 188
 NQ_ERR_NOROOM, 188
 NQ_ERR_NOSERVERMAC, 181
 NQ_ERR_NOSUPPORT, 189
 NQ_ERR_NOTCONNECTED, 186
 NQ_ERR_NOTFOUND, 182
 NQ_ERR_NOTREADY, 189
 NQ_ERR_NOWRITE, 189
 NQ_ERR_OBJEXISTS, 181
 NQ_ERR_OK, 181
 NQ_ERR_OUTOFMEMORY, 183
 NQ_ERR_PATHNOTCOVERED, 182
 NQ_ERR_PAUSED, 188
 NQ_ERR_PIPEBUSY, 186
 NQ_ERR_PIPECLOSING, 186
 NQ_ERR_QEOF, 187
 NQ_ERR_QFULL, 187
 NQ_ERR_QTOOBIG, 187
 NQ_ERR_READ, 190
 NQ_ERR_REMCD, 185
 NQ_ERR_RMUNS, 188
 NQ_ERR_SEEK, 189
 NQ_ERR_SERVERUNAVAILABLE, 183
 NQ_ERR_SESSIONREAUTHREQUIRED, 183
 NQ_ERR_SETATTRMODE, 188
 NQ_ERR_SHAREBUFEXC, 190
 NQ_ERR_SHAREUNAVAILABLE, 183
 NQ_ERR_SHARINGPAUSED, 183
 NQ_ERR_SHARINGVIOLATION, 185
 NQ_ERR_SIGNATUREFAIL, 181
 NQ_ERR_SMBCMD, 187
 NQ_ERR_SRVERERROR, 187
 NQ_ERR_TIMEOUT, 188
 NQ_ERR_TOOMANYUIDS, 188
 NQ_ERR_TRYAGAIN, 183
 NQ_ERR_UNABLETODISPOSE, 181
 NQ_ERR_USEMPX, 188
 NQ_ERR_USEREXISTS, 182
 NQ_ERR_USERNOTFOUND, 182
 NQ_ERR_USERRELOGONFAILED, 183
 NQ_ERR_USESTD, 188
 NQ_ERR_VOLUMEDISMOUNTED, 182
 NQ_ERR_WRITE, 189
 NQ_ERR_WRONGDIALECT, 185
 NQ_ERR_WRONGDISK, 190
 NQ_BOOL
 udapi.h, 298
 NQ_BYTE
 udapi.h, 298
 NQ_CC_AUTH_LM_AND_NTLM
 Authentication Level, 22
 NQ_CC_AUTH_NTLM
 Authentication Level, 22
 NQ_CC_AUTH_NTLM_V2
 Authentication Level, 22
 NQ_CC_AUTH_SPNEGO_KERBEROS
 Authentication Level, 22
 NQ_CHAR
 udapi.h, 298
 NQ_CLIENT
 Initialization and Shutdown, 83
 NQ_COUNT
 udapi.h, 298
 NQ_CS_PWDANY
 Authentication Statuses, 196
 NQ_CS_PWDFOUND
 Authentication Statuses, 196
 NQ_CS_PWDLMHASH
 Authentication Statuses, 196
 NQ_CS_PWDNOAUTH
 Authentication Statuses, 196
 NQ_CS_PWDNOUSER
 Authentication Statuses, 196
 NQ_ERR_ACCESS
 NQ Error Codes, 187
 NQ_ERR_ACCOUNTDISABLED
 NQ Error Codes, 183
 NQ_ERR_ACCOUNTEXPIRED
 NQ Error Codes, 183
 NQ_ERR_ACCOUNTLOCKEDOUT
 NQ Error Codes, 182
 NQ_ERR_ALREADYEXISTS
 NQ Error Codes, 186
 NQ_ERR_ATTRERROR
 TCP and NetBIOS Layer Error Codes, 194
 NQ_ERR_BADACCESS
 NQ Error Codes, 184
 NQ_ERR_BADCMD
 NQ Error Codes, 189
 NQ_ERR_BADCONNECTION
 NQ Error Codes, 182
 NQ_ERR_BADDATA
 NQ Error Codes, 185
 NQ_ERR_BADDFS
 NQ Error Codes, 183
 NQ_ERR_BADDIRECTORY
 NQ Error Codes, 185
 NQ_ERR_BADDRIVE
 NQ Error Codes, 185
 NQ_ERR_BADENV
 NQ Error Codes, 184
 NQ_ERR_BADFID
 NQ Error Codes, 184
 NQ_ERR_BADFILE
 NQ Error Codes, 184
 NQ_ERR_BADFORMAT
 NQ Error Codes, 184

NQ_ERR_BADFUNC	NQ_ERR_INSUFFICIENTBUFFER
NQ Error Codes, 184	NQ Error Codes, 186
NQ_ERR_BADMCB	NQ_ERR_INVALIDHANDLE
NQ Error Codes, 184	NQ Error Codes, 182
NQ_ERR_BADMEDIA	NQ_ERR_INVALIDLOGONHOURS
NQ Error Codes, 189	NQ Error Codes, 183
NQ_ERR_BADMEM	NQ_ERR_INVALIDMODE
NQ Error Codes, 184	NQ Error Codes, 181
NQ_ERR_BADPARAM	NQ_ERR_INVALIDNAME
NQ Error Codes, 181	NQ Error Codes, 186
NQ_ERR_BADPATH	NQ_ERR_INVALIDNAMELEN
NQ Error Codes, 184	NQ Error Codes, 186
NQ_ERR_BADPERMITS	NQ_ERR_INVALIDUINT32SIZE
NQ Error Codes, 187	TCP and NetBIOS Layer Error Codes, 193
NQ_ERR_BADPIPE	NQ_ERR_INVDEVICE
NQ Error Codes, 186	NQ Error Codes, 187
NQ_ERR_BADPW	NQ_ERR_INVFID
NQ Error Codes, 186	NQ Error Codes, 187
NQ_ERR_BADREQ	NQ_ERR_INVNETNAME
NQ Error Codes, 189	NQ Error Codes, 187
NQ_ERR_BADSECTOR	NQ_ERR_INVTID
NQ Error Codes, 189	NQ Error Codes, 187
NQ_ERR_BADSHARE	NQ_ERR_INVUID
NQ Error Codes, 185	NQ Error Codes, 188
NQ_ERR_BADUNIT	NQ_ERR_IOTIMEOUT
NQ Error Codes, 189	NQ Error Codes, 183
NQ_ERR_CAPTUREZERO	NQ_ERR_KERBEROSERROR
TCP and NetBIOS Layer Error Codes, 193	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_CONTMPX	NQ_ERR_LOCK
NQ Error Codes, 188	NQ Error Codes, 185
NQ_ERR_DATA	NQ_ERR_LOGONFAILURE
NQ Error Codes, 189	NQ Error Codes, 182
NQ_ERR_DFSCACHEOVERFLOW	NQ_ERR_MODULE
NQ Error Codes, 182	NQ Error Codes, 181
NQ_ERR_DIFFDEVICE	NQ_ERR_MOREDATA
NQ Error Codes, 185	NQ Error Codes, 186
NQ_ERR_DIRNOTEMPTY	NQ_ERR_MOUNTERROR
NQ Error Codes, 186	NQ Error Codes, 181
NQ_ERR_DISKFULL	NQ_ERR_MSGOFF
NQ Error Codes, 190	NQ Error Codes, 188
NQ_ERR_DONTSUPPORTIPC	NQ_ERR_NBBUFFEROVERFLOW
NQ Error Codes, 185	TCP and NetBIOS Layer Error Codes, 193
NQ_ERR_ERROR	NQ_ERR_NBCANCELLISTENFAIL
NQ Error Codes, 186	TCP and NetBIOS Layer Error Codes, 192
NQ_ERR_FCBUNAVAIL	NQ_ERR_NBDDCOMMUNICATIONERROR
NQ Error Codes, 190	TCP and NetBIOS Layer Error Codes, 192
NQ_ERR_FILESPCS	NQ_ERR_NBHOSTNAMENOTRESOLVED
NQ Error Codes, 187	TCP and NetBIOS Layer Error Codes, 192
NQ_ERR_FILEXISTS	NQ_ERR_NBILLEGALDATAGRAMDESTINATION
NQ Error Codes, 185	TCP and NetBIOS Layer Error Codes, 192
NQ_ERR_GENERAL	NQ_ERR_NBILLEGALDATAGRAMSOURCE
NQ Error Codes, 190	TCP and NetBIOS Layer Error Codes, 192
NQ_ERR_GETDATA	NQ_ERR_NBILLEGALDATAGRAMTYPE
NQ Error Codes, 181	TCP and NetBIOS Layer Error Codes, 192
NQ_ERR_ICONVERROR	NQ_ERR_NBILLEGALSOCKETSLLOT
TCP and NetBIOS Layer Error Codes, 195	TCP and NetBIOS Layer Error Codes, 191
NQ_ERR_ICONVOPEN	NQ_ERR_NBINTERNALERROR
TCP and NetBIOS Layer Error Codes, 195	TCP and NetBIOS Layer Error Codes, 192

NQ_ERR_NBINVALIDPARAMETER	NQ_ERR_PIPECLOSING
TCP and NetBIOS Layer Error Codes, 192	NQ Error Codes, 186
NQ_ERR_NBLISTENFAIL	NQ_ERR_QEOF
TCP and NetBIOS Layer Error Codes, 193	NQ Error Codes, 187
NQ_ERR_NBNEGATIVERESPONSE	NQ_ERR_QFULL
TCP and NetBIOS Layer Error Codes, 192	NQ Error Codes, 187
NQ_ERR_NBNOBINDBEFORELISTEN	NQ_ERR_QTOOBIG
TCP and NetBIOS Layer Error Codes, 192	NQ Error Codes, 187
NQ_ERR_NBNOTNETBIOSNAME	NQ_ERR_RAPACCESSDENIED
TCP and NetBIOS Layer Error Codes, 191	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_NBRELEASENAMEFAIL	NQ_ERR_RAPBADTRANSACTCONFIG
TCP and NetBIOS Layer Error Codes, 193	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_NBSOCKETOVERFLOW	NQ_ERR_RAPDEVICENOTCONNECTED
TCP and NetBIOS Layer Error Codes, 192	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_NBTIMEOUT	NQ_ERR_RAPGENERALERROR
TCP and NetBIOS Layer Error Codes, 192	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_NEGOTIATEFAILED	NQ_ERR_RAPMOREDATA
NQ Error Codes, 182	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_NETWORKERROR	NQ_ERR_RAPNOBROWSESERVERSFOUND
NQ Error Codes, 182	TCP and NetBIOS Layer Error Codes, 195
NQ_ERR_NOACCESS	NQ_ERR_RAPNOTACTIVEINDOMAIN
NQ Error Codes, 184	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_NOFIDS	NQ_ERR_RAPSERVERNOTSTARTED
NQ Error Codes, 184	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_NOFILES	NQ_ERR_RAPSERVICENOTINSTALLED
NQ Error Codes, 185	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_NOMEM	NQ_ERR_READ
NQ Error Codes, 184	NQ Error Codes, 190
NQ_ERR_NOPAPER	NQ_ERR_RECONNECTREQUIRED
NQ Error Codes, 189	TCP and NetBIOS Layer Error Codes, 193
NQ_ERR_NORESOURCE	NQ_ERR_REMCD
NQ Error Codes, 188	NQ Error Codes, 185
NQ_ERR_NOROOM	NQ_ERR_RMUNS
NQ Error Codes, 188	NQ Error Codes, 188
NQ_ERR_NOSERVERMAC	NQ_ERR_SEEK
NQ Error Codes, 181	NQ Error Codes, 189
NQ_ERR_NOSUPPORT	NQ_ERR_SEEKERROR
NQ Error Codes, 189	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_NOTCONNECTED	NQ_ERR_SERVERUNAVAILABLE
NQ Error Codes, 186	NQ Error Codes, 183
NQ_ERR_NOTFOUND	NQ_ERR_SESSIONREAUTHREQUIRED
NQ Error Codes, 182	NQ Error Codes, 183
NQ_ERR_NOTREADY	NQ_ERR_SETATTRMODE
NQ Error Codes, 189	NQ Error Codes, 188
NQ_ERR_NOWRITE	NQ_ERR_SHAREBUFEXC
NQ Error Codes, 189	NQ Error Codes, 190
NQ_ERR_OBJEXISTS	NQ_ERR_SHAREUNAVAILABLE
NQ Error Codes, 181	NQ Error Codes, 183
NQ_ERR_OK	NQ_ERR_SHARINGPAUSED
NQ Error Codes, 181	NQ Error Codes, 183
NQ_ERR_OUTOFMEMORY	NQ_ERR_SHARINGVIOLATION
NQ Error Codes, 183	NQ Error Codes, 185
NQ_ERR_PATHNOTCOVERED	NQ_ERR_SIGNATUREFAIL
NQ Error Codes, 182	NQ Error Codes, 181
NQ_ERR_PAUSED	NQ_ERR_SIZEERROR
NQ Error Codes, 188	TCP and NetBIOS Layer Error Codes, 194
NQ_ERR_PIPEBUSY	NQ_ERR_SMBCMD
NQ Error Codes, 186	NQ Error Codes, 187

NQ_ERR_SOCKETADDRESS	NQ_IOBuffer
TCP and NetBIOS Layer Error Codes, 193	udapi.h, 297
NQ_ERR_SOCKETBIND	NQ_IOVEC, 247
TCP and NetBIOS Layer Error Codes, 193	iov_base, 248
NQ_ERR_SOCKETCREATE	iov_len, 248
TCP and NetBIOS Layer Error Codes, 193	NQ_IOVECBUF, 248
NQ_ERR_SOCKETNAME	iovSegmentCnt, 248
TCP and NetBIOS Layer Error Codes, 193	iovec, 248
NQ_ERR_SOCKETSEND	NQ_IOVecBuffer, 248
TCP and NetBIOS Layer Error Codes, 193	data, 248
NQ_ERR_SOCKETSETOPTION	len, 248
TCP and NetBIOS Layer Error Codes, 193	NQ_IPADDRESS, 249
NQ_ERR_SRVERERROR	addr, 249
NQ Error Codes, 187	v4, 249
NQ_ERR_TIMEOUT	v6, 249
NQ Error Codes, 188	version, 249
NQ_ERR_TOOMANYUIDS	vr, 249
NQ Error Codes, 188	NQ_IPADDRESS4
NQ_ERR_TRYAGAIN	udapi.h, 298
NQ Error Codes, 183	NQ_IPADDRESS6
NQ_ERR_UNABLETODISPOSE	udapi.h, 298
NQ Error Codes, 181	NQ_IPADDRESSVR, 250
NQ_ERR_USEMPX	addr, 250
NQ Error Codes, 188	id, 250
NQ_ERR_USEREXISTS	v4, 250
NQ Error Codes, 182	v6, 250
NQ_ERR_USERNOTFOUND	version, 250
NQ Error Codes, 182	NQ_NDDAEMON
NQ_ERR_USERRELOGONFAILED	Initialization and Shutdown, 83
NQ Error Codes, 183	NQ_PORT
NQ_ERR_USESTD	udapi.h, 298
NQ Error Codes, 188	NQ_RESOLVER_DNS
NQ_ERR_VOLUMEDISMOUNTED	Resolver, 86
NQ Error Codes, 182	NQ_RESOLVER_DNS_DC
NQ_ERR_WRITE	Resolver, 86
NQ Error Codes, 189	NQ_RESOLVER_EXTERNAL_METHOD
NQ_ERR_WRONGDIALECT	Resolver, 86
NQ Error Codes, 185	NQ_RESOLVER_IPV4
NQ_ERR_WRONGDISK	Resolver, 85
NQ Error Codes, 190	NQ_RESOLVER_IPV6
NQ_FLOAT	Resolver, 85
udapi.h, 298	NQ_RESOLVER_NETBIOS
NQ_HANDLE	Resolver, 86
udapi.h, 298	NQ_RESOLVER_NETBIOS_DC
NQ_INDEX	Resolver, 86
udapi.h, 298	NQ_RESOLVER_NONE
NQ_INT	Resolver, 85
udapi.h, 298	NQ_RESOLVER_WSD
NQ_INT16	Resolver, 86
udapi.h, 298	NQ_SERVER
NQ_INT32	Initialization and Shutdown, 83
udapi.h, 298	NQ_STATUS
NQ_INT64, 247	udapi.h, 298
high, 247	NQ_TIME
low, 247	udapi.h, 298
sign, 247	NQ_UINT
NQ_IOBuf	udapi.h, 298
udapi.h, 297	NQ_UINT16

- udapi.h, [298](#)
- NQ_UINT32
 - udapi.h, [298](#)
- NQ_UINT64, [250](#)
 - high, [250](#)
 - low, [250](#)
- NQ_ULONG
 - udapi.h, [298](#)
- NQ_WCHAR
 - udapi.h, [298](#)
- name
 - AMDDomain, [223](#)
 - AMDDomainA, [223](#)
 - CCNetDomainItem, [233](#)
 - CCNetServerItem, [234](#)
 - CCNetShareItem, [234](#)
 - CCNetShareItemA, [235](#)
 - SYPrinterDriver, [255](#)
 - SYPrintFormInfo, [258](#)
- ndNotifyConfigurationChange
 - NetBIOS Daemon, [117](#)
- ndStart
 - NetBIOS Daemon, [117](#)
- ndStop
 - NetBIOS Daemon, [117](#)
- ndapi.h, [277](#)
- NetBIOS Daemon, [117](#)
 - ndNotifyConfigurationChange, [117](#)
 - ndStart, [117](#)
 - ndStop, [117](#)
- NetBIOS Name Registration, [120](#)
 - CM_NB_GROUP, [120](#)
 - CM_NB_POSTFIX_DOMAINCONTROLLER, [120](#)
 - CM_NB_POSTFIX_DOMAINMASTERBROWSER, [120](#)
 - CM_NB_POSTFIX_MASTERBROWSER, [120](#)
 - CM_NB_POSTFIX_SERVER, [120](#)
 - CM_NB_POSTFIX_WORKSTATION, [120](#)
 - CM_NB_UNIQUE, [120](#)
 - nqRegisterNetBiosName, [120](#)
 - nqRegisterNetBiosNameEx, [122](#)
 - nqReleaseNetBiosName, [120](#)
 - nqReleaseNetBiosNameEx, [122](#)
- Network Browsing, [56](#)
 - ccNetworkCacheSet, [58](#)
 - ccNetworkCloseHandle, [63](#)
 - ccNetworkEnumerateDomains, [58](#)
 - ccNetworkEnumerateServers, [60](#)
 - ccNetworkEnumerateServersA, [59](#)
 - ccNetworkEnumerateShares, [61](#)
 - ccNetworkEnumerateSharesA, [60](#)
 - ccNetworkGetDefaultDomain, [58](#)
 - ccNetworkGetDefaultDomainA, [58](#)
 - ccNetworkGetNextDomainItem, [59](#)
 - ccNetworkGetNextItemName, [61](#)
 - ccNetworkGetNextItemNameA, [61](#)
 - ccNetworkGetNextServerItem, [60](#)
 - ccNetworkGetNextShareItem, [62](#)
 - ccNetworkGetShareInfo, [63](#)
 - ccNetworkGetShareInfoA, [62](#)
 - ccNetworkResetHandle, [63](#)
 - ccNetworkSetDefaultDomain, [56](#)
 - ccNetworkSetDefaultDomainA, [56](#)
- newName
 - UDFileAccessEvent, [265](#)
- nqEnableTraceLog
 - Utils, [123](#)
- nqRegisterNetBiosName
 - NetBIOS Name Registration, [120](#)
- nqRegisterNetBiosNameEx
 - NetBIOS Name Registration, [122](#)
- nqReleaseNetBiosName
 - NetBIOS Name Registration, [120](#)
- nqReleaseNetBiosNameEx
 - NetBIOS Name Registration, [122](#)
- nqStart
 - Initialization and Shutdown, [119](#)
- nqStop
 - Initialization and Shutdown, [119](#)
- nqapi.h, [277](#)
- numlps
 - CCServerInfo, [236](#)
- numLinks
 - SYFileInformation, [254](#)
- numberOfLinks
 - CCFileInfo, [224](#)
 - CCFileInfo_t, [226](#)
- offsetHigh
 - UDFileAccessEvent, [266](#)
- offsetLow
 - UDFileAccessEvent, [266](#)
- Open Files Enumeration, [112](#)
 - csEnumerateOpenFiles, [112](#)
 - csEnumerateOpenFilesA, [112](#)
- orientation
 - SYDeviceMode, [252](#)
- osVersion
 - SYPrinterDriver, [255](#)
- PASSWORD_LENGTH
 - Credential fields length, [13](#)
- pStatus
 - SYPrintJobInfo, [259](#)
- pagesPrinted
 - SYPrintJobInfo, [260](#)
- panningHeight
 - SYDeviceMode, [253](#)
- panningWidth
 - SYDeviceMode, [253](#)
- paperLength
 - SYDeviceMode, [252](#)
- paperSize
 - SYDeviceMode, [252](#)
- paperWidth
 - SYDeviceMode, [252](#)
- parameters

- SYPrinterInfo, [257](#)
- password
 - AMCredentials, [221](#)
 - AMCredentialsA, [222](#)
- pelsHeight
 - SYDeviceMode, [252](#)
- pelsWidth
 - SYDeviceMode, [252](#)
- portName
 - SYPrinterInfo, [257](#)
- position
 - SYPrintJobInfo, [260](#)
- previousNames
 - SYPrinterDriver, [255](#)
- printProcessor
 - SYPrinterInfo, [257](#)
- printQuality
 - SYDeviceMode, [252](#)
- printQueue
 - EVCSShareConnectionEntry, [246](#)
 - UDShareAccessEvent, [267](#)
- Printer, [129](#)
 - SY_PRINTEROSVERSION_WIN, [129](#)
- Printer Attribute Flags, [134](#)
 - SY_PRINTERATTR_BIDI, [135](#)
 - SY_PRINTERATTR_CONTAINER, [135](#)
 - SY_PRINTERATTR_DEFAULT, [134](#)
 - SY_PRINTERATTR_DEVQUERY, [135](#)
 - SY_PRINTERATTR_DIRECT, [134](#)
 - SY_PRINTERATTR_EXPAND, [135](#)
 - SY_PRINTERATTR_ICON1, [135](#)
 - SY_PRINTERATTR_ICON2, [135](#)
 - SY_PRINTERATTR_ICON3, [135](#)
 - SY_PRINTERATTR_ICON4, [135](#)
 - SY_PRINTERATTR_ICON5, [135](#)
 - SY_PRINTERATTR_ICON6, [136](#)
 - SY_PRINTERATTR_ICON7, [136](#)
 - SY_PRINTERATTR_ICON8, [136](#)
 - SY_PRINTERATTR_KEEPCONNECTIONS, [135](#)
 - SY_PRINTERATTR_LIFO, [135](#)
 - SY_PRINTERATTR_LOCAL, [135](#)
 - SY_PRINTERATTR_NETWORK, [134](#)
 - SY_PRINTERATTR_OFFLINE, [135](#)
 - SY_PRINTERATTR_PUBLISHED, [135](#)
 - SY_PRINTERATTR_QUEUED, [134](#)
 - SY_PRINTERATTR_RAWONLY, [135](#)
 - SY_PRINTERATTR_RESERVED, [135](#)
 - SY_PRINTERATTR_SHARED, [134](#)
- Printer Control Commands, [140](#)
 - SY_PRINTERCONTROL_PAUSE, [140](#)
 - SY_PRINTERCONTROL_PURGE, [140](#)
 - SY_PRINTERCONTROL_RESUME, [140](#)
 - SY_PRINTERCONTROL_SETSTATUS, [140](#)
- Printer Flags, [130](#)
 - SY_DEVICEMODE_BITSPERPEL, [131](#)
 - SY_DEVICEMODE_COLLATE, [131](#)
 - SY_DEVICEMODE_COLOR, [131](#)
 - SY_DEVICEMODE_COPIES, [130](#)
 - SY_DEVICEMODE_DEFAULTSOURCE, [131](#)
 - SY_DEVICEMODE_DISPLAYFLAGS, [131](#)
 - SY_DEVICEMODE_DISPLAYFREQUENCY, [131](#)
 - SY_DEVICEMODE_DITHERTYPE, [131](#)
 - SY_DEVICEMODE_DUPLEX, [131](#)
 - SY_DEVICEMODE_FORMNAME, [131](#)
 - SY_DEVICEMODE_ICMINTENT, [131](#)
 - SY_DEVICEMODE_ICMMETHOD, [131](#)
 - SY_DEVICEMODE_LOGPIXELS, [131](#)
 - SY_DEVICEMODE_MEDIATYPE, [131](#)
 - SY_DEVICEMODE_NUP, [130](#)
 - SY_DEVICEMODE_ORIENTATION, [130](#)
 - SY_DEVICEMODE_PANNINGHEIGHT, [131](#)
 - SY_DEVICEMODE_PANNINGWIDTH, [131](#)
 - SY_DEVICEMODE_PAPERLENGTH, [130](#)
 - SY_DEVICEMODE_PAPERSIZE, [130](#)
 - SY_DEVICEMODE_PAPERWIDTH, [130](#)
 - SY_DEVICEMODE_PELHEIGHT, [131](#)
 - SY_DEVICEMODE_PELWIDTH, [131](#)
 - SY_DEVICEMODE_POSITION, [130](#)
 - SY_DEVICEMODE_PRINTQUALITY, [131](#)
 - SY_DEVICEMODE_SCALE, [130](#)
 - SY_DEVICEMODE_TTOPTION, [131](#)
 - SY_DEVICEMODE_YRESOLUTION, [131](#)
- Printer Form Flags, [142](#)
 - SY_PRINTFORMFLAG_BUILTIN, [142](#)
 - SY_PRINTFORMFLAG_PRINTER, [142](#)
 - SY_PRINTFORMFLAG_USER, [142](#)
- Printer Job Control Commands, [141](#)
 - SY_PRINTJOB_COM_CANCEL, [141](#)
 - SY_PRINTJOB_COM_DELETE, [141](#)
 - SY_PRINTJOB_COM_LASTPAGEEJECTED, [141](#)
 - SY_PRINTJOB_COM_PAUSE, [141](#)
 - SY_PRINTJOB_COM_RESTART, [141](#)
 - SY_PRINTJOB_COM_RESUME, [141](#)
 - SY_PRINTJOB_COM_SENDDTOPRINTER, [141](#)
- Printer Priority Job, [139](#)
 - SY_PRINTJOB_DEFPPRIORITY, [139](#)
 - SY_PRINTJOB_MAXPPRIORITY, [139](#)
 - SY_PRINTJOB_MINPPRIORITY, [139](#)
- Printer Status Flags, [132](#)
 - SY_PRINTERSTATUS_BUSY, [132](#)
 - SY_PRINTERSTATUS_DOOROPEN, [133](#)
 - SY_PRINTERSTATUS_ERROR, [132](#)
 - SY_PRINTERSTATUS_INITIALIZING, [133](#)
 - SY_PRINTERSTATUS_IOACTIVE, [132](#)
 - SY_PRINTERSTATUS_MANUALFEED, [132](#)
 - SY_PRINTERSTATUS_NOTAVAILABLE, [133](#)
 - SY_PRINTERSTATUS_NOTONER, [133](#)
 - SY_PRINTERSTATUS_OFFLINE, [132](#)
 - SY_PRINTERSTATUS_OUTOFMEMORY, [133](#)
 - SY_PRINTERSTATUS_OUTPUTBINFULL, [133](#)
 - SY_PRINTERSTATUS_PAGEPUNT, [133](#)
 - SY_PRINTERSTATUS_PAPERJAM, [132](#)
 - SY_PRINTERSTATUS_PAPEROUT, [132](#)
 - SY_PRINTERSTATUS_PAPERPROBLEM, [132](#)
 - SY_PRINTERSTATUS_PAUSED, [132](#)
 - SY_PRINTERSTATUS_PENDINGDELETION, [132](#)

- SY_PRINTERSTATUS_POWERSAVE, [133](#)
- SY_PRINTERSTATUS_PRINTING, [133](#)
- SY_PRINTERSTATUS_PROCESSING, [133](#)
- SY_PRINTERSTATUS_SERVERUNKNOWN, [133](#)
- SY_PRINTERSTATUS_TONERLOW, [133](#)
- SY_PRINTERSTATUS_USERINTERVENTION, [133](#)
- SY_PRINTERSTATUS_WAITING, [133](#)
- SY_PRINTERSTATUS_WARMINGUP, [133](#)
- Printer Status Job, [137](#)
 - SY_PRINTJOBSTATUS_BLOCKED, [138](#)
 - SY_PRINTJOBSTATUS_DELETED, [138](#)
 - SY_PRINTJOBSTATUS_DELETING, [137](#)
 - SY_PRINTJOBSTATUS_ERROR, [137](#)
 - SY_PRINTJOBSTATUS_OFFLINE, [137](#)
 - SY_PRINTJOBSTATUS_PAPERPROBLEM, [137](#)
 - SY_PRINTJOBSTATUS_PAUSED, [137](#)
 - SY_PRINTJOBSTATUS_PRINTED, [137](#)
 - SY_PRINTJOBSTATUS_PRINTING, [137](#)
 - SY_PRINTJOBSTATUS_RESTART, [138](#)
 - SY_PRINTJOBSTATUS_SPOOLING, [137](#)
 - SY_PRINTJOBSTATUS_USERINTERVENTION, [138](#)
- printerErrors
 - SYPrinterInfo, [258](#)
- priority
 - SYPrinterInfo, [257](#)
 - SYPrintJobInfo, [260](#)
- Product Information, [92](#)
 - CM_PRODUCT_STRING_LEN, [92](#)
 - cmGetCurrentProductInformation, [92](#)
- productName
 - CMProductInfo, [241](#)
- productType
 - CMProductInfo, [241](#)
- provider
 - SYPrinterDriver, [256](#)
- readAmount
 - CCSStatistics, [236](#)
- realm
 - AMDDomain, [223](#)
 - AMDDomainA, [223](#)
- requestByIp
 - CMResolverRegisteredMethodDescription, [243](#)
- requestByName
 - CMResolverRegisteredMethodDescription, [242](#)
- reserved1
 - SYDeviceMode, [253](#)
- reserved2
 - SYDeviceMode, [253](#)
- Resolver, [85](#)
 - CMResolverIpToName, [87](#)
 - CMResolverIpToNameA, [87](#)
 - CMResolverNameToIp, [86](#)
 - CMResolverNameToIpA, [86](#)
 - cmResolverEnableMethod, [90](#)
 - cmResolverGetDCName, [90](#)
 - cmResolverGetHostIps, [89](#)
 - cmResolverGetHostName, [89](#)
 - cmResolverRegisterExternalMethod, [91](#)
 - cmResolverSetExternal, [88](#)
 - cmResolverSetExternalA, [88](#)
 - cmResolverUpdateExternalMethodsPriority, [91](#)
 - NQ_RESOLVER_DNS, [86](#)
 - NQ_RESOLVER_DNS_DC, [86](#)
 - NQ_RESOLVER_EXTERNAL_METHOD, [86](#)
 - NQ_RESOLVER_IPV4, [85](#)
 - NQ_RESOLVER_IPV6, [85](#)
 - NQ_RESOLVER_NETBIOS, [86](#)
 - NQ_RESOLVER_NETBIOS_DC, [86](#)
 - NQ_RESOLVER_NONE, [85](#)
 - NQ_RESOLVER_WSD, [86](#)
- responseByIp
 - CMResolverRegisteredMethodDescription, [243](#)
- responseByName
 - CMResolverRegisteredMethodDescription, [242](#)
- rid
 - UDFileAccessEvent, [266](#)
 - UDShareAccessEvent, [267](#)
 - UDUserAccessEvent, [268](#)
- right
 - SYPrintRect, [261](#)
- SEEK_FILE_BEGIN
 - File seek constants, [21](#)
- SEEK_FILE_CURRENT
 - File seek constants, [21](#)
- SEEK_FILE_END
 - File seek constants, [21](#)
- SMB Error Codes, [149](#)
 - SMB_STATUS_ABIOS_INVALID_COMMAND, [169](#)
 - SMB_STATUS_ABIOS_INVALID_LID, [169](#)
 - SMB_STATUS_ABIOS_INVALID_SELECTOR, [169](#)
 - SMB_STATUS_ABIOS_LID_ALREADY_OWNED, [169](#)
 - SMB_STATUS_ABIOS_LID_NOT_EXIST, [169](#)
 - SMB_STATUS_ABIOS_NOT_LID_OWNER, [169](#)
 - SMB_STATUS_ABIOS_NOT_PRESENT, [168](#)
 - SMB_STATUS_ABIOS_SELECTOR_NOT_AVAILABLE, [169](#)
 - SMB_STATUS_ACCESS_DENIED, [160](#)
 - SMB_STATUS_ACCESS_VIOLATION, [159](#)
 - SMB_STATUS_ACCOUNT_DISABLED, [163](#)
 - SMB_STATUS_ACCOUNT_EXPIRED, [173](#)
 - SMB_STATUS_ACCOUNT_LOCKED_OUT, [175](#)
 - SMB_STATUS_ACCOUNT_RESTRICTION, [163](#)
 - SMB_STATUS_ADAPTER_HARDWARE_ERROR, [166](#)
 - SMB_STATUS_ADDRESS_ALREADY_ASSOCIATED, [175](#)
 - SMB_STATUS_ADDRESS_ALREADY_EXISTS, [174](#)
 - SMB_STATUS_ADDRESS_CLOSED, [174](#)
 - SMB_STATUS_ADDRESS_NOT_ASSOCIATED, [175](#)

SMB_STATUS_AGENTS_EXHAUSTED, 164
SMB_STATUS_ALIAS_EXISTS, 171
SMB_STATUS_ALLOCATE_BUCKET, 175
SMB_STATUS_ALLOTTED_SPACE_EXCEEDED, 164
SMB_STATUS_ALREADY_COMMITTED, 160
SMB_STATUS_APP_INIT_FAILURE, 170
SMB_STATUS_ARRAY_BOUNDS_EXCEEDED, 164
SMB_STATUS_AUDIT_FAILED, 176
SMB_STATUS_BACKUP_CONTROLLER, 173
SMB_STATUS_BAD_COMPRESSION_BUFFER, 176
SMB_STATUS_BAD_DESCRIPTOR_FORMAT, 167
SMB_STATUS_BAD_DEVICE_TYPE, 166
SMB_STATUS_BAD_DLL_ENTRYPOINT, 176
SMB_STATUS_BAD_FUNCTION_TABLE, 168
SMB_STATUS_BAD_IMPERSONATION_LEVEL, 165
SMB_STATUS_BAD_INHERITANCE_ACL, 163
SMB_STATUS_BAD_INITIAL_PC, 159
SMB_STATUS_BAD_INITIAL_STACK, 159
SMB_STATUS_BAD_LOGON_SESSION_STATE, 168
SMB_STATUS_BAD_MASTER_BOOT_RECORD, 165
SMB_STATUS_BAD_NETWORK_NAME, 166
SMB_STATUS_BAD_NETWORK_PATH, 166
SMB_STATUS_BAD_REMOTE_ADAPTER, 166
SMB_STATUS_BAD_SERVICE_ENTRYPOINT, 176
SMB_STATUS_BAD_STACK, 160
SMB_STATUS_BAD_TOKEN_TYPE, 165
SMB_STATUS_BAD_VALIDATION_CLASS, 165
SMB_STATUS_BAD_WORKING_SET_LIMIT, 162
SMB_STATUS_BUFFER_OVERFLOW, 159
SMB_STATUS_BUFFER_TOO_SMALL, 160
SMB_STATUS_CANCELLED, 169
SMB_STATUS_CANNOT_DELETE, 169
SMB_STATUS_CANNOT_IMPERSONATE, 168
SMB_STATUS_CANNOT_LOAD_REGISTRY_FILE, 174
SMB_STATUS_CANT_ACCESS_DOMAIN_INFO, 167
SMB_STATUS_CANT_DISABLE_MANDATORY, 162
SMB_STATUS_CANT_OPEN_ANONYMOUS, 165
SMB_STATUS_CANT_TERMINATE_SELF, 167
SMB_STATUS_CANT_WAIT, 167
SMB_STATUS_CHILD_MUST_BE_VOLATILE, 172
SMB_STATUS_CLIENT_SERVER_PARAMETERS_INVALID, 175
SMB_STATUS_COMMITMENT_LIMIT, 170
SMB_STATUS_CONFLICTING_ADDRESSES, 160
SMB_STATUS_CONNECTION_ABORTED, 176
SMB_STATUS_CONNECTION_ACTIVE, 175
SMB_STATUS_CONNECTION_COUNT_LIMIT, 176
SMB_STATUS_CONNECTION_DISCONNECTED, 174
SMB_STATUS_CONNECTION_IN_USE, 168
SMB_STATUS_CONNECTION_INVALID, 175
SMB_STATUS_CONNECTION_REFUSED, 175
SMB_STATUS_CONNECTION_RESET, 174
SMB_STATUS_CONTROL_C_EXIT, 170
SMB_STATUS_CONVERT_TO_LARGE, 175
SMB_STATUS_COULD_NOT_INTERPRET, 165
SMB_STATUS_CRC_ERROR, 161
SMB_STATUS_CTL_FILE_NOT_SUPPORTED, 162
SMB_STATUS_DATA_ERROR, 161
SMB_STATUS_DATA_LATE_ERROR, 161
SMB_STATUS_DATA_NOT_ACCEPTED, 174
SMB_STATUS_DATA_OVERRUN, 161
SMB_STATUS_DEBUG_ATTACH_FAILED, 174
SMB_STATUS_DELETE_PENDING, 162
SMB_STATUS_DEVICE_ALREADY_ATTACHED, 161
SMB_STATUS_DEVICE_CONFIGURATION_ERROR, 172
SMB_STATUS_DEVICE_DATA_ERROR, 164
SMB_STATUS_DEVICE_DOES_NOT_EXIST, 166
SMB_STATUS_DEVICE_NOT_CONNECTED, 164
SMB_STATUS_DEVICE_NOT_PARTITIONED, 172
SMB_STATUS_DEVICE_NOT_READY, 165
SMB_STATUS_DEVICE_OFF_LINE, 159
SMB_STATUS_DEVICE_POWER_FAILURE, 164
SMB_STATUS_DEVICE_PROTOCOL_ERROR, 173
SMB_STATUS_DFS_EXIT_PATH_FOUND, 164
SMB_STATUS_DFS_UNAVAILABLE, 177
SMB_STATUS_DIRECTORY_NOT_EMPTY, 168
SMB_STATUS_DISK_CORRUPT_ERROR, 161
SMB_STATUS_DISK_FULL, 163
SMB_STATUS_DISK_OPERATION_FAILED, 172
SMB_STATUS_DISK_RECALIBRATE_FAILED, 172
SMB_STATUS_DISK_RESET_FAILED, 172
SMB_STATUS_DLL_INIT_FAILED, 170
SMB_STATUS_DLL_NOT_FOUND, 170
SMB_STATUS_DOMAIN_CONTROLLER_NOT_FOUND, 175
SMB_STATUS_DOMAIN_CTRLR_CONFIG_ERROR, 171
SMB_STATUS_DOMAIN_EXISTS, 167
SMB_STATUS_DOMAIN_LIMIT_EXCEEDED, 167
SMB_STATUS_DOMAIN_TRUST_INCONSISTENT, 173

SMB_STATUS_DRIVER_CANCEL_TIMEOUT, 174
SMB_STATUS_DRIVER_ENTRYPOINT_NOT_FOUND, 177
SMB_STATUS_DRIVER_INTERNAL_ERROR, 172
SMB_STATUS_DRIVER_ORDINAL_NOT_FOUND, 177
SMB_STATUS_DUPLICATE_NAME, 166
SMB_STATUS_DUPLICATE_OBJECTID, 175
SMB_STATUS_EA_CORRUPT_ERROR, 162
SMB_STATUS_EA_TOO_LARGE, 162
SMB_STATUS_EAS_NOT_SUPPORTED, 162
SMB_STATUS_END_OF_FILE, 159
SMB_STATUS_ENTRYPOINT_NOT_FOUND, 170
SMB_STATUS_ENUMDIR, 158
SMB_STATUS_EOM_OVERFLOW, 172
SMB_STATUS_EVENTLOG_CANT_START, 173
SMB_STATUS_EVENTLOG_FILE_CHANGED, 173
SMB_STATUS_EVENTLOG_FILE_CORRUPT, 173
SMB_STATUS_FAIL_CHECK, 175
SMB_STATUS_FILE_CLOSED, 169
SMB_STATUS_FILE_CORRUPT_ERROR, 168
SMB_STATUS_FILE_DELETED, 169
SMB_STATUS_FILE_FORCED_CLOSED, 165
SMB_STATUS_FILE_INVALID, 164
SMB_STATUS_FILE_IS_A_DIRECTORY, 165
SMB_STATUS_FILE_IS_OFFLINE, 177
SMB_STATUS_FILE_LOCK_CONFLICT, 162
SMB_STATUS_FILE_NOT_AVAILABLE, 177
SMB_STATUS_FILE_RENAMED, 166
SMB_STATUS_FILES_OPEN, 168
SMB_STATUS_FLOAT_DENORMAL_OPERAND, 164
SMB_STATUS_FLOAT_DIVIDE_BY_ZERO, 164
SMB_STATUS_FLOAT_INEXACT_RESULT, 164
SMB_STATUS_FLOAT_INVALID_OPERATION, 164
SMB_STATUS_FLOAT_OVERFLOW, 164
SMB_STATUS_FLOAT_STACK_CHECK, 164
SMB_STATUS_FLOAT_UNDERFLOW, 164
SMB_STATUS_FLOPPY_BAD_REGISTERS, 172
SMB_STATUS_FLOPPY_ID_MARK_NOT_FOUND, 172
SMB_STATUS_FLOPPY_UNKNOWN_ERROR, 172
SMB_STATUS_FLOPPY_VOLUME, 172
SMB_STATUS_FLOPPY_WRONG_CYLINDER, 172
SMB_STATUS_FOUND_OUT_OF_SCOPE, 175
SMB_STATUS_FREE_VM_NOT_AT_BASE, 165
SMB_STATUS_FS_DRIVER_REQUIRED, 173
SMB_STATUS_FT_MISSING_MEMBER, 171
SMB_STATUS_FT_ORPHANING, 172
SMB_STATUS_FULLSCREEN_MODE, 171
SMB_STATUS_GENERIC_NOT_MAPPED, 167
SMB_STATUS_GRACEFUL_DISCONNECT, 175
SMB_STATUS_GROUP_EXISTS, 162
SMB_STATUS_GUIDS_EXHAUSTED, 164
SMB_STATUS_HANDLE_NOT_CLOSABLE, 175
SMB_STATUS_HANDLE_NOT_WAITABLE, 161
SMB_STATUS_HOST_UNREACHABLE, 176
SMB_STATUS_ILL_FORMED_PASSWORD, 163
SMB_STATUS_ILL_FORMED_SERVICE_ENTRY, 171
SMB_STATUS_ILLEGAL_CHARACTER, 171
SMB_STATUS_ILLEGAL_FLOAT_CONTEXT, 171
SMB_STATUS_ILLEGAL_FUNCTION, 165
SMB_STATUS_ILLEGAL_INSTRUCTION, 160
SMB_STATUS_IMAGE_ALREADY_LOADED, 168
SMB_STATUS_IMAGE_CHECKSUM_MISMATCH, 175
SMB_STATUS_IMAGE_MP_UP_MISMATCH, 176
SMB_STATUS_IN_PAGE_ERROR, 159
SMB_STATUS_INCOMPATIBLE_FILE_MAP, 162
SMB_STATUS_INFO_LENGTH_MISMATCH, 159
SMB_STATUS_INSTANCE_NOT_AVAILABLE, 165
SMB_STATUS_INSTRUCTION_MISALIGNMENT, 165
SMB_STATUS_INSUFF_SERVER_RESOURCES, 174
SMB_STATUS_INSUFFICIENT_BUFFER, 159
SMB_STATUS_INSUFFICIENT_LOGON_INFO, 176
SMB_STATUS_INSUFFICIENT_RESOURCES, 164
SMB_STATUS_INTEGER_DIVIDE_BY_ZERO, 164
SMB_STATUS_INTEGER_OVERFLOW, 164
SMB_STATUS_INTERNAL_DB_CORRUPTION, 167
SMB_STATUS_INTERNAL_DB_ERROR, 171
SMB_STATUS_INTERNAL_ERROR, 167
SMB_STATUS_INVALID, 158
SMB_STATUS_INVALID_ACCOUNT_NAME, 162
SMB_STATUS_INVALID_ACL, 163
SMB_STATUS_INVALID_ADDRESS, 170
SMB_STATUS_INVALID_ADDRESS_COMPONENT, 174
SMB_STATUS_INVALID_ADDRESS_WILDCARD, 174
SMB_STATUS_INVALID_BLOCK_LENGTH, 172
SMB_STATUS_INVALID_BUFFER_SIZE, 174
SMB_STATUS_INVALID_CID, 159
SMB_STATUS_INVALID_COMPUTER_NAME, 169
SMB_STATUS_INVALID_CONNECTION, 170
SMB_STATUS_INVALID_DATATYPE, 159
SMB_STATUS_INVALID_DEVICE_REQUEST, 159
SMB_STATUS_INVALID_DEVICE_STATE, 172

SMB_STATUS_INVALID_DISPOSITION, 160
SMB_STATUS_INVALID_DOMAIN_ROLE, 167
SMB_STATUS_INVALID_DOMAIN_STATE, 167
SMB_STATUS_INVALID_FILE_FOR_SECTION, 160
SMB_STATUS_INVALID_GROUP_ATTRIBUTES, 165
SMB_STATUS_INVALID_HANDLE, 159
SMB_STATUS_INVALID_HW_PROFILE, 177
SMB_STATUS_INVALID_ID_AUTHORITY, 164
SMB_STATUS_INVALID_IMAGE_FORMAT, 163
SMB_STATUS_INVALID_IMAGE_LE_FORMAT, 170
SMB_STATUS_INVALID_IMAGE_NE_FORMAT, 169
SMB_STATUS_INVALID_IMAGE_NOT_MZ, 170
SMB_STATUS_INVALID_IMAGE_PROTECT, 170
SMB_STATUS_INVALID_IMAGE_WIN_16, 170
SMB_STATUS_INVALID_INFO_CLASS, 159
SMB_STATUS_INVALID_LDT_DESCRIPTOR, 169
SMB_STATUS_INVALID_LDT_OFFSET, 169
SMB_STATUS_INVALID_LDT_SIZE, 169
SMB_STATUS_INVALID_LEVEL, 171
SMB_STATUS_INVALID_LOCK_RANGE, 173
SMB_STATUS_INVALID_LOCK_SEQUENCE, 160
SMB_STATUS_INVALID_LOGON_HOURS, 163
SMB_STATUS_INVALID_LOGON_TYPE, 168
SMB_STATUS_INVALID_MEMBER, 172
SMB_STATUS_INVALID_NETWORK_RESPONSE, 166
SMB_STATUS_INVALID_OPLOCK_PROTOCOL, 167
SMB_STATUS_INVALID_OWNER, 162
SMB_STATUS_INVALID_PAGE_PROTECTION, 161
SMB_STATUS_INVALID_PARAMETER, 159
SMB_STATUS_INVALID_PARAMETER_1, 167
SMB_STATUS_INVALID_PARAMETER_10, 168
SMB_STATUS_INVALID_PARAMETER_11, 168
SMB_STATUS_INVALID_PARAMETER_12, 168
SMB_STATUS_INVALID_PARAMETER_2, 167
SMB_STATUS_INVALID_PARAMETER_3, 167
SMB_STATUS_INVALID_PARAMETER_4, 167
SMB_STATUS_INVALID_PARAMETER_5, 168
SMB_STATUS_INVALID_PARAMETER_6, 168
SMB_STATUS_INVALID_PARAMETER_7, 168
SMB_STATUS_INVALID_PARAMETER_8, 168
SMB_STATUS_INVALID_PARAMETER_9, 168
SMB_STATUS_INVALID_PARAMETER_MIX, 161
SMB_STATUS_INVALID_PIPE_STATE, 165
SMB_STATUS_INVALID_PLUGPLAY_DEVICE_PATH, 177
SMB_STATUS_INVALID_PORT_ATTRIBUTES, 160
SMB_STATUS_INVALID_PORT_HANDLE, 161
SMB_STATUS_INVALID_PRIMARY_GROUP, 162
SMB_STATUS_INVALID_QUOTA_LOWER, 161
SMB_STATUS_INVALID_READ_MODE, 165
SMB_STATUS_INVALID_SECURITY_DESCR, 163
SMB_STATUS_INVALID_SERVER_STATE, 167
SMB_STATUS_INVALID_SID, 163
SMB_STATUS_INVALID_SUB_AUTHORITY, 163
SMB_STATUS_INVALID_SYSTEM_SERVICE, 160
SMB_STATUS_INVALID_TOKEN, 177
SMB_STATUS_INVALID_UNWIND_TARGET, 160
SMB_STATUS_INVALID_USER_BUFFER, 167
SMB_STATUS_INVALID_VARIANT, 175
SMB_STATUS_INVALID_VIEW_SIZE, 160
SMB_STATUS_INVALID_VOLUME_LABEL, 164
SMB_STATUS_INVALID_WORKSTATION, 163
SMB_STATUS_IO_DEVICE_ERROR, 173
SMB_STATUS_IO_PRIVILEGE_FAILED, 170
SMB_STATUS_IO_REPARSE_TAG_NOT_HANDLED, 177
SMB_STATUS_IO_TIMEOUT, 165
SMB_STATUS_IP_ADDRESS_CONFLICT1, 176
SMB_STATUS_IP_ADDRESS_CONFLICT2, 176
SMB_STATUS_KEY_DELETED, 172
SMB_STATUS_KEY_HAS_CHILDREN, 172
SMB_STATUS_LAST_ADMIN, 163
SMB_STATUS_LICENSE_QUOTA_EXCEEDED, 176
SMB_STATUS_LINK_FAILED, 170
SMB_STATUS_LINK_TIMEOUT, 170
SMB_STATUS_LM_CROSS_ENCRYPTION_REQUIRED, 172
SMB_STATUS_LOCAL_DISCONNECT, 170
SMB_STATUS_LOCK_NOT_GRANTED, 162
SMB_STATUS_LOG_FILE_FULL, 173
SMB_STATUS_LOGIN_TIME_RESTRICTION, 176
SMB_STATUS_LOGIN_WKSTA_RESTRICTION, 176
SMB_STATUS_LOGON_FAILURE, 163
SMB_STATUS_LOGON_NOT_GRANTED, 171
SMB_STATUS_LOGON_SERVER_CONFLICT, 170
SMB_STATUS_LOGON_SESSION_COLLISION, 168
SMB_STATUS_LOGON_SESSION_EXISTS, 167
SMB_STATUS_LOGON_TYPE_NOT_GRANTED, 171
SMB_STATUS_LOST_WRITEBEHIND_DATA, 175
SMB_STATUS_LPC_REPLY_LOST, 176
SMB_STATUS_LUIDS_EXHAUSTED, 163
SMB_STATUS_MAPPED_ALIGNMENT, 174
SMB_STATUS_MAPPED_FILE_SIZE_ZERO, 169
SMB_STATUS_MARSHALL_OVERFLOW, 175

SMB_STATUS_MEDIA_WRITE_PROTECTED, 165
SMB_STATUS_MEMBER_IN_ALIAS, 171
SMB_STATUS_MEMBER_IN_GROUP, 163
SMB_STATUS_MEMBER_NOT_IN_ALIAS, 171
SMB_STATUS_MEMBER_NOT_IN_GROUP, 163
SMB_STATUS_MEMBERS_PRIMARY_GROUP, 169
SMB_STATUS_MEMORY_NOT_ALLOCATED, 165
SMB_STATUS_MESSAGE_NOT_FOUND, 168
SMB_STATUS_MISSING_SYSTEMFILE, 170
SMB_STATUS_MORE_ENTRIES, 158
SMB_STATUS_MORE_PROCESSING_REQUIRED, 160
SMB_STATUS_MUTANT_LIMIT_EXCEEDED, 173
SMB_STATUS_MUTANT_NOT_OWNED, 161
SMB_STATUS_NAME_TOO_LONG, 168
SMB_STATUS_NET_WRITE_FAULT, 166
SMB_STATUS_NETLOGON_NOT_STARTED, 173
SMB_STATUS_NETWORK_ACCESS_DENIED, 166
SMB_STATUS_NETWORK_BUSY, 166
SMB_STATUS_NETWORK_CREDENTIAL_CONFLICT, 173
SMB_STATUS_NETWORK_NAME_DELETED, 166
SMB_STATUS_NETWORK_SESSION_EXPIRED, 177
SMB_STATUS_NETWORK_UNREACHABLE, 175
SMB_STATUS_NO_BROWSER_SERVERS_FOUND, 174
SMB_STATUS_NO_CALLBACK_ACTIVE, 176
SMB_STATUS_NO_EAS_ON_FILE, 162
SMB_STATUS_NO_EVENT_PAIR, 171
SMB_STATUS_NO_GUID_TRANSLATION, 168
SMB_STATUS_NO_IMPERSONATION_TOKEN, 162
SMB_STATUS_NO_LDT, 169
SMB_STATUS_NO_LOG_SPACE, 172
SMB_STATUS_NO_LOGON_SERVERS, 162
SMB_STATUS_NO_MEDIA, 172
SMB_STATUS_NO_MEDIA_IN_DEVICE, 160
SMB_STATUS_NO_MEMORY, 160
SMB_STATUS_NO_MORE_ENTRIES, 159
SMB_STATUS_NO_MORE_FILES, 159
SMB_STATUS_NO_PAGEFILE, 170
SMB_STATUS_NO_PREAUTH_INTEGRITY_HASH_OVERLAP, 177
SMB_STATUS_NO_SECURITY_ON_OBJECT, 167
SMB_STATUS_NO_SPOOL_SPACE, 166
SMB_STATUS_NO_SUCH_ALIAS, 171
SMB_STATUS_NO_SUCH_DEVICE, 159
SMB_STATUS_NO_SUCH_DOMAIN, 167
SMB_STATUS_NO_SUCH_FILE, 159
SMB_STATUS_NO_SUCH_GROUP, 162
SMB_STATUS_NO_SUCH_JOB, 177
SMB_STATUS_NO_SUCH_LOGON_SESSION, 162
SMB_STATUS_NO_SUCH_MEMBER, 172
SMB_STATUS_NO_SUCH_PACKAGE, 168
SMB_STATUS_NO_SUCH_PRIVILEGE, 162
SMB_STATUS_NO_SUCH_USER, 162
SMB_STATUS_NO_TOKEN, 163
SMB_STATUS_NO_TRUST_LSA_SECRET, 173
SMB_STATUS_NO_TRUST_SAM_ACCOUNT, 173
SMB_STATUS_NO_USER_SESSION_KEY, 173
SMB_STATUS_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT, 173
SMB_STATUS_NOLOGON_SERVER_TRUST_ACCOUNT, 173
SMB_STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT, 173
SMB_STATUS_NONCONTINUABLE_EXCEPTION, 160
SMB_STATUS_NONE_MAPPED, 163
SMB_STATUS_NONEXISTENT_EA_ENTRY, 162
SMB_STATUS_NONEXISTENT_SECTOR, 160
SMB_STATUS_NOT_A_DIRECTORY, 168
SMB_STATUS_NOT_A_REPARSE_POINT, 177
SMB_STATUS_NOT_CLIENT_SESSION, 174
SMB_STATUS_NOT_COMMITTED, 160
SMB_STATUS_NOT_FOUND, 175
SMB_STATUS_NOT_IMPLEMENTED, 159
SMB_STATUS_NOT_LOCKED, 160
SMB_STATUS_NOT_LOGON_PROCESS, 167
SMB_STATUS_NOT_MAPPED_DATA, 164
SMB_STATUS_NOT_MAPPED_VIEW, 160
SMB_STATUS_NOT_REGISTRY_FILE, 171
SMB_STATUS_NOT_SAME_DEVICE, 166
SMB_STATUS_NOT_SERVER_SESSION, 174
SMB_STATUS_NOT_SUPPORTED, 166
SMB_STATUS_NOT_TINY_STREAM, 175
SMB_STATUS_NOTIFY_CLEANUP, 159
SMB_STATUS_NT_CROSS_ENCRYPTION_REQUIRED, 171
SMB_STATUS_OBJECT_NAME_COLLISION, 161
SMB_STATUS_OBJECT_NAME_INVALID, 161
SMB_STATUS_OBJECT_NAME_NOT_FOUND, 161
SMB_STATUS_OBJECT_PATH_INVALID, 161
SMB_STATUS_OBJECT_PATH_NOT_FOUND, 161
SMB_STATUS_OBJECT_PATH_SYNTAX_BAD, 161
SMB_STATUS_OBJECT_TYPE_MISMATCH, 160
SMB_STATUS_OBJECTID_EXISTS, 175
SMB_STATUS_OK, 158
SMB_STATUS_OPEN_FAILED, 170
SMB_STATUS_OPLOCK_NOT_GRANTED, 167

SMB_STATUS_ORDINAL_NOT_FOUND, 170
SMB_STATUS_PAGEFILE_CREATE_FAILED, 170
SMB_STATUS_PAGEFILE_QUOTA, 159
SMB_STATUS_PAGEFILE_QUOTA_EXCEEDED, 170
SMB_STATUS_PARITY_ERROR, 160
SMB_STATUS_PARTITION_FAILURE, 172
SMB_STATUS_PASSWORD_EXPIRED, 163
SMB_STATUS_PASSWORD_MUST_CHANGE, 175
SMB_STATUS_PASSWORD_RESTRICTION, 163
SMB_STATUS_PATH_NOT_COVERED, 176
SMB_STATUS_PENDING, 158
SMB_STATUS_PIPE_BROKEN, 171
SMB_STATUS_PIPE_BUSY, 165
SMB_STATUS_PIPE_CLOSING, 165
SMB_STATUS_PIPE_CONNECTED, 165
SMB_STATUS_PIPE_DISCONNECTED, 165
SMB_STATUS_PIPE_EMPTY, 167
SMB_STATUS_PIPE_LISTENING, 165
SMB_STATUS_PIPE_NOT_AVAILABLE, 165
SMB_STATUS_PLUGPLAY_NO_DEVICE, 176
SMB_STATUS_PORT_ALREADY_SET, 161
SMB_STATUS_PORT_CONNECTION_REFUSED, 161
SMB_STATUS_PORT_DISCONNECTED, 161
SMB_STATUS_PORT_MESSAGE_TOO_LONG, 161
SMB_STATUS_PORT_UNREACHABLE, 176
SMB_STATUS_POSSIBLE_DEADLOCK, 173
SMB_STATUS_PRINT_CANCELLED, 166
SMB_STATUS_PRINT_QUEUE_FULL, 166
SMB_STATUS_PRIVILEGE_NOT_HELD, 162
SMB_STATUS_PRIVILEGED_INSTRUCTION, 164
SMB_STATUS_PROCEDURE_NOT_FOUND, 163
SMB_STATUS_PROCESS_IS_TERMINATING, 168
SMB_STATUS_PROFILING_AT_LIMIT, 166
SMB_STATUS_PROFILING_NOT_STARTED, 165
SMB_STATUS_PROFILING_NOT_STOPPED, 165
SMB_STATUS_PROPSET_NOT_FOUND, 175
SMB_STATUS_PROTOCOL_UNREACHABLE, 176
SMB_STATUS_PWD_HISTORY_CONFLICT, 176
SMB_STATUS_PWD_TOO_RECENT, 176
SMB_STATUS_PWD_TOO_SHORT, 176
SMB_STATUS_QUOTA_EXCEEDED, 161
SMB_STATUS_QUOTA_LIST_INCONSISTENT, 177
SMB_STATUS_RANGE_NOT_LOCKED, 163
SMB_STATUS_RECOVERY_FAILURE, 175
SMB_STATUS_REDIRECTOR_NOT_STARTED, 168
SMB_STATUS_REDIRECTOR_PAUSED, 166
SMB_STATUS_REDIRECTOR_STARTED, 168
SMB_STATUS_REGISTRY_CORRUPT, 171
SMB_STATUS_REGISTRY_IO_FAILED, 171
SMB_STATUS_REGISTRY_QUOTA_LIMIT, 176
SMB_STATUS_REMOTE_DISCONNECT, 170
SMB_STATUS_REMOTE_NOT_LISTENING, 166
SMB_STATUS_REMOTE_RESOURCES, 170
SMB_STATUS_REMOTE_SESSION_LIMIT, 173
SMB_STATUS_REPLY_MESSAGE_MISMATCH, 174
SMB_STATUS_REQUEST_ABORTED, 176
SMB_STATUS_REQUEST_NOT_ACCEPTED, 166
SMB_STATUS_RESOURCE_DATA_NOT_FOUND, 164
SMB_STATUS_RESOURCE_LANG_NOT_FOUND, 173
SMB_STATUS_RESOURCE_NAME_NOT_FOUND, 164
SMB_STATUS_RESOURCE_NOT_OWNED, 177
SMB_STATUS_RESOURCE_TYPE_NOT_FOUND, 164
SMB_STATUS_RETRY, 175
SMB_STATUS_REVISION_MISMATCH, 162
SMB_STATUS_RXACT_COMMIT_FAILURE, 169
SMB_STATUS_RXACT_INVALID_STATE, 169
SMB_STATUS_SECRET_TOO_LONG, 171
SMB_STATUS_SECTION_NOT_EXTENDED, 164
SMB_STATUS_SECTION_NOT_IMAGE, 161
SMB_STATUS_SECTION_PROTECTION, 162
SMB_STATUS_SECTION_TOO_BIG, 161
SMB_STATUS_SEMAPHORE_LIMIT_EXCEEDED, 161
SMB_STATUS_SERIAL_NO_DEVICE_INITED, 171
SMB_STATUS_SERVER_DISABLED, 163
SMB_STATUS_SERVER_NOT_DISABLED, 163
SMB_STATUS_SERVER_UNAVAILABLE, 177
SMB_STATUS_SHARE_UNAVAILABLE, 177
SMB_STATUS_SHARED_IRQ_BUSY, 172
SMB_STATUS_SHARING_PAUSED, 166
SMB_STATUS_SHARING_VIOLATION, 161
SMB_STATUS_SOME_UNMAPPED, 159
SMB_STATUS_SPECIAL_ACCOUNT, 169
SMB_STATUS_SPECIAL_GROUP, 169
SMB_STATUS_SPECIAL_USER, 169
SMB_STATUS_STACK_OVERFLOW, 168
SMB_STATUS_STACK_OVERFLOW_READ, 175
SMB_STATUS_STOPPED_ON_SYMLINK, 159
SMB_STATUS_SUCCESS, 158
SMB_STATUS_SUSPEND_COUNT_EXCEEDED, 161
SMB_STATUS_SYNCHRONIZATION_REQUIRED, 170

SMB_STATUS_SYSTEM_PROCESS_TERMINATED, 174
SMB_STATUS_THREAD_IS_TERMINATING, 162
SMB_STATUS_THREAD_NOT_IN_PROCESS, 169
SMB_STATUS_TIME_DIFFERENCE_AT_DC, 170
SMB_STATUS_TIMER_NOT_CANCELED, 159
SMB_STATUS_TIMER_RESOLUTION_NOT_SET, 176
SMB_STATUS_TOKEN_ALREADY_IN_USE, 169
SMB_STATUS_TOO_LATE, 173
SMB_STATUS_TOO_MANY_ADDRESSES, 174
SMB_STATUS_TOO_MANY_COMMANDS, 166
SMB_STATUS_TOO_MANY_CONTEXT_IDS, 171
SMB_STATUS_TOO_MANY_GUIDS_REQUESTED, 163
SMB_STATUS_TOO_MANY_LINKS, 177
SMB_STATUS_TOO_MANY_LUIDS_REQUESTED, 163
SMB_STATUS_TOO_MANY_NAMES, 166
SMB_STATUS_TOO_MANY_NODES, 174
SMB_STATUS_TOO_MANY_OPENED_FILES, 169
SMB_STATUS_TOO_MANY_PAGING_FILES, 164
SMB_STATUS_TOO_MANY_SECRETS, 171
SMB_STATUS_TOO_MANY_SESSIONS, 166
SMB_STATUS_TOO_MANY_SIDS, 172
SMB_STATUS_TOO_MANY_THREADS, 169
SMB_STATUS_TRANSACTION_ABORTED, 174
SMB_STATUS_TRANSACTION_INVALID_ID, 174
SMB_STATUS_TRANSACTION_INVALID_TYPE, 174
SMB_STATUS_TRANSACTION_NO_MATCH, 174
SMB_STATUS_TRANSACTION_NO_RELEASE, 174
SMB_STATUS_TRANSACTION_RESPONDED, 174
SMB_STATUS_TRANSACTION_TIMED_OUT, 174
SMB_STATUS_TRUST_FAILURE, 173
SMB_STATUS_TRUSTED_DOMAIN_FAILURE, 173
SMB_STATUS_TRUSTED_RELATIONSHIP_FAILURE, 173
SMB_STATUS_UNABLE_TO_DECOMMIT_VM, 160
SMB_STATUS_UNABLE_TO_DELETE_SECTION, 160
SMB_STATUS_UNABLE_TO_FREE_VM, 160
SMB_STATUS_UNABLE_TO_LOCK_MEDIA, 172
SMB_STATUS_UNABLE_TO_UNLOAD_MEDIA, 172
SMB_STATUS_UNDEFINED_CHARACTER, 171
SMB_STATUS_UNEXPECTED_IO_ERROR, 167
SMB_STATUS_UNEXPECTED_MM_CREATE_ERROR, 167
SMB_STATUS_UNEXPECTED_MM_EXTEND_ERROR, 167
SMB_STATUS_UNEXPECTED_MM_MAP_ERROR, 167
SMB_STATUS_UNEXPECTED_NETWORK_ERROR, 166
SMB_STATUS_UNHANDLED_EXCEPTION, 170
SMB_STATUS_UNKNOWN_PRINTER_DRIVER, 159
SMB_STATUS_UNKNOWN_REVISION, 162
SMB_STATUS_UNMAPPABLE_CHARACTER, 171
SMB_STATUS_UNRECOGNIZED_MEDIA, 160
SMB_STATUS_UNRECOGNIZED_VOLUME, 171
SMB_STATUS_UNSUCCESSFUL, 159
SMB_STATUS_UNSUPPORTED_COMPRESSION, 176
SMB_STATUS_UNWIND, 160
SMB_STATUS_USER_EXISTS, 162
SMB_STATUS_USER_MAPPED_FILE, 176
SMB_STATUS_USER_SESSION_DELETED, 173
SMB_STATUS_VDM_HARD_ERROR, 174
SMB_STATUS_VIRTUAL_CIRCUIT_CLOSED, 166
SMB_STATUS_VOLUME_DISMOUNTED, 177
SMB_STATUS_WORKING_SET_QUOTA, 165
SMB_STATUS_WRONG_PASSWORD, 163
SMB_STATUS_WRONG_PASSWORD_CORE, 171
SMB_STATUS_WRONG_VOLUME, 159
SMB Errors, 144
CLASS_ERR_MODULE, 145
DOS_ERRAlreadyexists, 146
DOS_ERRbadaccess, 146
DOS_ERRbaddata, 146
DOS_ERRbaddirectory, 146
DOS_ERRbaddrive, 146
DOS_ERRbadenv, 146
DOS_ERRbadfid, 146
DOS_ERRbadfile, 145
DOS_ERRbadformat, 146
DOS_ERRbadfunc, 145
DOS_ERRbadmcb, 146
DOS_ERRbadmem, 146
DOS_ERRbadpath, 145
DOS_ERRbadpipe, 146
DOS_ERRbadshare, 146
DOS_ERRdiffdevice, 146
DOS_ERRdirnotempty, 146
DOS_ERRdentsupportipc, 146
DOS_ERRfileexists, 146
DOS_ERRinsufficientbuffer, 146
DOS_ERRinvalidname, 146
DOS_ERRlock, 146
DOS_ERRmoredata, 147

DOS_ERRnoaccess, [146](#)
 DOS_ERRnofids, [146](#)
 DOS_ERRnofiles, [146](#)
 DOS_ERRnomem, [146](#)
 DOS_ERRnoshare, [146](#)
 DOS_ERRnotconnected, [146](#)
 DOS_ERRpipebusy, [146](#)
 DOS_ERRpipeclosing, [146](#)
 DOS_ERRremcd, [146](#)
 HRD_ERRFCBUnavail, [148](#)
 HRD_ERRbadcmd, [148](#)
 HRD_ERRbadmedia, [148](#)
 HRD_ERRbadreq, [148](#)
 HRD_ERRbadsector, [148](#)
 HRD_ERRbadshare, [148](#)
 HRD_ERRbadunit, [148](#)
 HRD_ERRdata, [148](#)
 HRD_ERRdiskfull, [148](#)
 HRD_ERRgeneral, [148](#)
 HRD_ERRlock, [148](#)
 HRD_ERRnopaper, [148](#)
 HRD_ERRnotready, [148](#)
 HRD_ERRnowrite, [148](#)
 HRD_ERRread, [148](#)
 HRD_ERRseek, [148](#)
 HRD_ERRsharebufexc, [148](#)
 HRD_ERRwrite, [148](#)
 HRD_ERRwrongdisk, [148](#)
 IS_DOS_ERROR, [145](#)
 IS_HRD_ERROR, [148](#)
 IS_SRV_ERROR, [147](#)
 SMB_DOSERR, [145](#)
 SMB_ERRsuccess, [145](#)
 SRV_ERRaccess, [147](#)
 SRV_ERRbadpermits, [147](#)
 SRV_ERRbadpw, [147](#)
 SRV_ERRcontmpx, [147](#)
 SRV_ERRerror, [147](#)
 SRV_ERRfilespecs, [147](#)
 SRV_ERRinvdevice, [147](#)
 SRV_ERRinvfid, [147](#)
 SRV_ERRinvnetname, [147](#)
 SRV_ERRinvtid, [147](#)
 SRV_ERRinvuid, [147](#)
 SRV_ERRmsgoff, [147](#)
 SRV_ERRnoresource, [147](#)
 SRV_ERRnoroom, [147](#)
 SRV_ERRnosupport, [148](#)
 SRV_ERRpaused, [147](#)
 SRV_ERRqeof, [147](#)
 SRV_ERRqfull, [147](#)
 SRV_ERRqtoobig, [147](#)
 SRV_ERRrmuns, [147](#)
 SRV_ERRsetattrmode, [147](#)
 SRV_ERRsmbcmd, [147](#)
 SRV_ERRsrverror, [147](#)
 SRV_ERRtimeout, [147](#)
 SRV_ERRtoomanyuids, [147](#)
 SRV_ERRusempx, [147](#)
 SRV_ERRusestd, [147](#)
 SMB_DOSERR
 SMB Errors, [145](#)
 SMB_ERRsuccess
 SMB Errors, [145](#)
 SMB_STATUS_ABIOS_INVALID_COMMAND
 SMB Error Codes, [169](#)
 SMB_STATUS_ABIOS_INVALID_LID
 SMB Error Codes, [169](#)
 SMB_STATUS_ABIOS_INVALID_SELECTOR
 SMB Error Codes, [169](#)
 SMB_STATUS_ABIOS_LID_ALREADY_OWNED
 SMB Error Codes, [169](#)
 SMB_STATUS_ABIOS_LID_NOT_EXIST
 SMB Error Codes, [169](#)
 SMB_STATUS_ABIOS_NOT_LID_OWNER
 SMB Error Codes, [169](#)
 SMB_STATUS_ABIOS_NOT_PRESENT
 SMB Error Codes, [168](#)
 SMB_STATUS_ABIOS_SELECTOR_NOT_AVAILABLE
 SMB Error Codes, [169](#)
 SMB_STATUS_ACCESS_DENIED
 SMB Error Codes, [160](#)
 SMB_STATUS_ACCESS_VIOLATION
 SMB Error Codes, [159](#)
 SMB_STATUS_ACCOUNT_DISABLED
 SMB Error Codes, [163](#)
 SMB_STATUS_ACCOUNT_EXPIRED
 SMB Error Codes, [173](#)
 SMB_STATUS_ACCOUNT_LOCKED_OUT
 SMB Error Codes, [175](#)
 SMB_STATUS_ACCOUNT_RESTRICTION
 SMB Error Codes, [163](#)
 SMB_STATUS_ADAPTER_HARDWARE_ERROR
 SMB Error Codes, [166](#)
 SMB_STATUS_ADDRESS_ALREADY_ASSOCIATED
 SMB Error Codes, [175](#)
 SMB_STATUS_ADDRESS_ALREADY_EXISTS
 SMB Error Codes, [174](#)
 SMB_STATUS_ADDRESS_CLOSED
 SMB Error Codes, [174](#)
 SMB_STATUS_ADDRESS_NOT_ASSOCIATED
 SMB Error Codes, [175](#)
 SMB_STATUS_AGENTS_EXHAUSTED
 SMB Error Codes, [164](#)
 SMB_STATUS_ALIAS_EXISTS
 SMB Error Codes, [171](#)
 SMB_STATUS_ALLOCATE_BUCKET
 SMB Error Codes, [175](#)
 SMB_STATUS_ALLOTTED_SPACE_EXCEEDED
 SMB Error Codes, [164](#)
 SMB_STATUS_ALREADY_COMMITTED
 SMB Error Codes, [160](#)
 SMB_STATUS_APP_INIT_FAILURE
 SMB Error Codes, [170](#)
 SMB_STATUS_ARRAY_BOUNDS_EXCEEDED

SMB Error Codes, 164	SMB Error Codes, 162
SMB_STATUS_AUDIT_FAILED	SMB_STATUS_CANT_OPEN_ANONYMOUS
SMB Error Codes, 176	SMB Error Codes, 165
SMB_STATUS_BACKUP_CONTROLLER	SMB_STATUS_CANT_TERMINATE_SELF
SMB Error Codes, 173	SMB Error Codes, 167
SMB_STATUS_BAD_COMPRESSION_BUFFER	SMB_STATUS_CANT_WAIT
SMB Error Codes, 176	SMB Error Codes, 167
SMB_STATUS_BAD_DESCRIPTOR_FORMAT	SMB_STATUS_CHILD_MUST_BE_VOLATILE
SMB Error Codes, 167	SMB Error Codes, 172
SMB_STATUS_BAD_DEVICE_TYPE	SMB_STATUS_CLIENT_SERVER_PARAMETERS_INVALID
SMB Error Codes, 166	SMB Error Codes, 175
SMB_STATUS_BAD_DLL_ENTRYPOINT	SMB_STATUS_COMMITMENT_LIMIT
SMB Error Codes, 176	SMB Error Codes, 170
SMB_STATUS_BAD_FUNCTION_TABLE	SMB_STATUS_CONFLICTING_ADDRESSES
SMB Error Codes, 168	SMB Error Codes, 160
SMB_STATUS_BAD_IMPERSONATION_LEVEL	SMB_STATUS_CONNECTION_ABORTED
SMB Error Codes, 165	SMB Error Codes, 176
SMB_STATUS_BAD_INHERITANCE_ACL	SMB_STATUS_CONNECTION_ACTIVE
SMB Error Codes, 163	SMB Error Codes, 175
SMB_STATUS_BAD_INITIAL_PC	SMB_STATUS_CONNECTION_COUNT_LIMIT
SMB Error Codes, 159	SMB Error Codes, 176
SMB_STATUS_BAD_INITIAL_STACK	SMB_STATUS_CONNECTION_DISCONNECTED
SMB Error Codes, 159	SMB Error Codes, 174
SMB_STATUS_BAD_LOGON_SESSION_STATE	SMB_STATUS_CONNECTION_IN_USE
SMB Error Codes, 168	SMB Error Codes, 168
SMB_STATUS_BAD_MASTER_BOOT_RECORD	SMB_STATUS_CONNECTION_INVALID
SMB Error Codes, 165	SMB Error Codes, 175
SMB_STATUS_BAD_NETWORK_NAME	SMB_STATUS_CONNECTION_REFUSED
SMB Error Codes, 166	SMB Error Codes, 175
SMB_STATUS_BAD_NETWORK_PATH	SMB_STATUS_CONNECTION_RESET
SMB Error Codes, 166	SMB Error Codes, 174
SMB_STATUS_BAD_REMOTE_ADAPTER	SMB_STATUS_CONTROL_C_EXIT
SMB Error Codes, 166	SMB Error Codes, 170
SMB_STATUS_BAD_SERVICE_ENTRYPOINT	SMB_STATUS_CONVERT_TO_LARGE
SMB Error Codes, 176	SMB Error Codes, 175
SMB_STATUS_BAD_STACK	SMB_STATUS_COULD_NOT_INTERPRET
SMB Error Codes, 160	SMB Error Codes, 165
SMB_STATUS_BAD_TOKEN_TYPE	SMB_STATUS_CRC_ERROR
SMB Error Codes, 165	SMB Error Codes, 161
SMB_STATUS_BAD_VALIDATION_CLASS	SMB_STATUS_CTL_FILE_NOT_SUPPORTED
SMB Error Codes, 165	SMB Error Codes, 162
SMB_STATUS_BAD_WORKING_SET_LIMIT	SMB_STATUS_DATA_ERROR
SMB Error Codes, 162	SMB Error Codes, 161
SMB_STATUS_BUFFER_OVERFLOW	SMB_STATUS_DATA_LATE_ERROR
SMB Error Codes, 159	SMB Error Codes, 161
SMB_STATUS_BUFFER_TOO_SMALL	SMB_STATUS_DATA_NOT_ACCEPTED
SMB Error Codes, 160	SMB Error Codes, 174
SMB_STATUS_CANCELLED	SMB_STATUS_DATA_OVERRUN
SMB Error Codes, 169	SMB Error Codes, 161
SMB_STATUS_CANNOT_DELETE	SMB_STATUS_DEBUG_ATTACH_FAILED
SMB Error Codes, 169	SMB Error Codes, 174
SMB_STATUS_CANNOT_IMPERSONATE	SMB_STATUS_DELETE_PENDING
SMB Error Codes, 168	SMB Error Codes, 162
SMB_STATUS_CANNOT_LOAD_REGISTRY_FILE	SMB_STATUS_DEVICE_ALREADY_ATTACHED
SMB Error Codes, 174	SMB Error Codes, 161
SMB_STATUS_CANT_ACCESS_DOMAIN_INFO	SMB_STATUS_DEVICE_CONFIGURATION_ERROR
SMB Error Codes, 167	SMB Error Codes, 172
SMB_STATUS_CANT_DISABLE_MANDATORY	

SMB_STATUS_DEVICE_DATA_ERROR	SMB_STATUS_DRIVER_ORDINAL_NOT_FOUND
SMB Error Codes, 164	SMB Error Codes, 177
SMB_STATUS_DEVICE_DOES_NOT_EXIST	SMB_STATUS_DUPLICATE_NAME
SMB Error Codes, 166	SMB Error Codes, 166
SMB_STATUS_DEVICE_NOT_CONNECTED	SMB_STATUS_DUPLICATE_OBJECTID
SMB Error Codes, 164	SMB Error Codes, 175
SMB_STATUS_DEVICE_NOT_PARTITIONED	SMB_STATUS_EA_CORRUPT_ERROR
SMB Error Codes, 172	SMB Error Codes, 162
SMB_STATUS_DEVICE_NOT_READY	SMB_STATUS_EA_TOO_LARGE
SMB Error Codes, 165	SMB Error Codes, 162
SMB_STATUS_DEVICE_OFF_LINE	SMB_STATUS_EAS_NOT_SUPPORTED
SMB Error Codes, 159	SMB Error Codes, 162
SMB_STATUS_DEVICE_POWER_FAILURE	SMB_STATUS_END_OF_FILE
SMB Error Codes, 164	SMB Error Codes, 159
SMB_STATUS_DEVICE_PROTOCOL_ERROR	SMB_STATUS_ENTRYPOINT_NOT_FOUND
SMB Error Codes, 173	SMB Error Codes, 170
SMB_STATUS_DFS_EXIT_PATH_FOUND	SMB_STATUS_ENUMDIR
SMB Error Codes, 164	SMB Error Codes, 158
SMB_STATUS_DFS_UNAVAILABLE	SMB_STATUS_EOM_OVERFLOW
SMB Error Codes, 177	SMB Error Codes, 172
SMB_STATUS_DIRECTORY_NOT_EMPTY	SMB_STATUS_EVENTLOG_CANT_START
SMB Error Codes, 168	SMB Error Codes, 173
SMB_STATUS_DISCONNECT	SMB_STATUS_EVENTLOG_FILE_CHANGED
Internal NT Statuses, 178	SMB Error Codes, 173
SMB_STATUS_DISK_CORRUPT_ERROR	SMB_STATUS_EVENTLOG_FILE_CORRUPT
SMB Error Codes, 161	SMB Error Codes, 173
SMB_STATUS_DISK_FULL	SMB_STATUS_FAIL_CHECK
SMB Error Codes, 163	SMB Error Codes, 175
SMB_STATUS_DISK_OPERATION_FAILED	SMB_STATUS_FILE_CLOSED
SMB Error Codes, 172	SMB Error Codes, 169
SMB_STATUS_DISK_RECALIBRATE_FAILED	SMB_STATUS_FILE_CORRUPT_ERROR
SMB Error Codes, 172	SMB Error Codes, 168
SMB_STATUS_DISK_RESET_FAILED	SMB_STATUS_FILE_DELETED
SMB Error Codes, 172	SMB Error Codes, 169
SMB_STATUS_DLL_INIT_FAILED	SMB_STATUS_FILE_FORCED_CLOSED
SMB Error Codes, 170	SMB Error Codes, 165
SMB_STATUS_DLL_NOT_FOUND	SMB_STATUS_FILE_INVALID
SMB Error Codes, 170	SMB Error Codes, 164
SMB_STATUS_DOMAIN_CONTROLLER_NOT_FOUND	SMB_STATUS_FILE_IS_A_DIRECTORY
SMB Error Codes, 175	SMB Error Codes, 165
SMB_STATUS_DOMAIN_CTRLR_CONFIG_ERROR	SMB_STATUS_FILE_IS_OFFLINE
SMB Error Codes, 171	SMB Error Codes, 177
SMB_STATUS_DOMAIN_EXISTS	SMB_STATUS_FILE_LOCK_CONFLICT
SMB Error Codes, 167	SMB Error Codes, 162
SMB_STATUS_DOMAIN_LIMIT_EXCEEDED	SMB_STATUS_FILE_NOT_AVAILABLE
SMB Error Codes, 167	SMB Error Codes, 177
SMB_STATUS_DOMAIN_TRUST_INCONSISTENT	SMB_STATUS_FILE_RENAMED
SMB Error Codes, 173	SMB Error Codes, 166
SMB_STATUS_DONOTRELEASERESPONSE	SMB_STATUS_FILES_OPEN
Internal NT Statuses, 178	SMB Error Codes, 168
SMB_STATUS_DRIVER_CANCEL_TIMEOUT	SMB_STATUS_FLOAT_DENORMAL_OPERAND
SMB Error Codes, 174	SMB Error Codes, 164
SMB_STATUS_DRIVER_ENTRYPOINT_NOT_FOUND	SMB_STATUS_FLOAT_DIVIDE_BY_ZERO
SMB Error Codes, 177	SMB Error Codes, 164
SMB_STATUS_DRIVER_INTERNAL_ERROR	SMB_STATUS_FLOAT_INEXACT_RESULT
SMB Error Codes, 172	SMB Error Codes, 164
	SMB_STATUS_FLOAT_INVALID_OPERATION
	SMB Error Codes, 164

SMB_STATUS_FLOAT_OVERFLOW	SMB_STATUS_IMAGE_MP_UP_MISMATCH
SMB Error Codes, 164	SMB Error Codes, 176
SMB_STATUS_FLOAT_STACK_CHECK	SMB_STATUS_IN_PAGE_ERROR
SMB Error Codes, 164	SMB Error Codes, 159
SMB_STATUS_FLOAT_UNDERFLOW	SMB_STATUS_INCOMPATIBLE_FILE_MAP
SMB Error Codes, 164	SMB Error Codes, 162
SMB_STATUS_FLOPPY_BAD_REGISTERS	SMB_STATUS_INFO_LENGTH_MISMATCH
SMB Error Codes, 172	SMB Error Codes, 159
SMB_STATUS_FLOPPY_ID_MARK_NOT_FOUND	SMB_STATUS_INSTANCE_NOT_AVAILABLE
SMB Error Codes, 172	SMB Error Codes, 165
SMB_STATUS_FLOPPY_UNKNOWN_ERROR	SMB_STATUS_INSTRUCTION_MISALIGNMENT
SMB Error Codes, 172	SMB Error Codes, 165
SMB_STATUS_FLOPPY_VOLUME	SMB_STATUS_INSUFF_SERVER_RESOURCES
SMB Error Codes, 172	SMB Error Codes, 174
SMB_STATUS_FLOPPY_WRONG_CYLINDER	SMB_STATUS_INSUFFICIENT_BUFFER
SMB Error Codes, 172	SMB Error Codes, 159
SMB_STATUS_FOUND_OUT_OF_SCOPE	SMB_STATUS_INSUFFICIENT_LOGON_INFO
SMB Error Codes, 175	SMB Error Codes, 176
SMB_STATUS_FREE_VM_NOT_AT_BASE	SMB_STATUS_INSUFFICIENT_RESOURCES
SMB Error Codes, 165	SMB Error Codes, 164
SMB_STATUS_FS_DRIVER_REQUIRED	SMB_STATUS_INTEGER_DIVIDE_BY_ZERO
SMB Error Codes, 173	SMB Error Codes, 164
SMB_STATUS_FT_MISSING_MEMBER	SMB_STATUS_INTEGER_OVERFLOW
SMB Error Codes, 171	SMB Error Codes, 164
SMB_STATUS_FT_ORPHANING	SMB_STATUS_INTERNAL_BUFFER_TOO_SMALL
SMB Error Codes, 172	Internal NT Statuses, 178
SMB_STATUS_FULLSCREEN_MODE	SMB_STATUS_INTERNAL_DB_CORRUPTION
SMB Error Codes, 171	SMB Error Codes, 167
SMB_STATUS_GENERIC_NOT_MAPPED	SMB_STATUS_INTERNAL_DB_ERROR
SMB Error Codes, 167	SMB Error Codes, 171
SMB_STATUS_GRACEFUL_DISCONNECT	SMB_STATUS_INTERNAL_ERROR
SMB Error Codes, 175	SMB Error Codes, 167
SMB_STATUS_GROUP_EXISTS	SMB_STATUS_INTERNAL_RELEASE_REQUEST_R- ESPONSE
SMB Error Codes, 162	Internal NT Statuses, 178
SMB_STATUS_GUIDS_EXHAUSTED	SMB_STATUS_INVALID
SMB Error Codes, 164	SMB Error Codes, 158
SMB_STATUS_HANDLE_NOT_CLOSABLE	SMB_STATUS_INVALID_ACCOUNT_NAME
SMB Error Codes, 175	SMB Error Codes, 162
SMB_STATUS_HANDLE_NOT_WAITABLE	SMB_STATUS_INVALID_ACL
SMB Error Codes, 161	SMB Error Codes, 163
SMB_STATUS_HOST_UNREACHABLE	SMB_STATUS_INVALID_ADDRESS
SMB Error Codes, 176	SMB Error Codes, 170
SMB_STATUS_ILL_FORMED_PASSWORD	SMB_STATUS_INVALID_ADDRESS_COMPONENT
SMB Error Codes, 163	SMB Error Codes, 174
SMB_STATUS_ILL_FORMED_SERVICE_ENTRY	SMB_STATUS_INVALID_ADDRESS_WILDCARD
SMB Error Codes, 171	SMB Error Codes, 174
SMB_STATUS_ILLEGAL_CHARACTER	SMB_STATUS_INVALID_BLOCK_LENGTH
SMB Error Codes, 171	SMB Error Codes, 172
SMB_STATUS_ILLEGAL_FLOAT_CONTEXT	SMB_STATUS_INVALID_BUFFER_SIZE
SMB Error Codes, 171	SMB Error Codes, 174
SMB_STATUS_ILLEGAL_FUNCTION	SMB_STATUS_INVALID_CID
SMB Error Codes, 165	SMB Error Codes, 159
SMB_STATUS_ILLEGAL_INSTRUCTION	SMB_STATUS_INVALID_COMPUTER_NAME
SMB Error Codes, 160	SMB Error Codes, 169
SMB_STATUS_IMAGE_ALREADY_LOADED	SMB_STATUS_INVALID_CONNECTION
SMB Error Codes, 168	SMB Error Codes, 170
SMB_STATUS_IMAGE_CHECKSUM_MISMATCH	SMB_STATUS_INVALID_DATATYPE
SMB Error Codes, 175	

SMB Error Codes, 159	SMB Error Codes, 162
SMB_STATUS_INVALID_DEVICE_REQUEST	SMB_STATUS_INVALID_PAGE_PROTECTION
SMB Error Codes, 159	SMB Error Codes, 161
SMB_STATUS_INVALID_DEVICE_STATE	SMB_STATUS_INVALID_PARAMETER
SMB Error Codes, 172	SMB Error Codes, 159
SMB_STATUS_INVALID_DISPOSITION	SMB_STATUS_INVALID_PARAMETER_1
SMB Error Codes, 160	SMB Error Codes, 167
SMB_STATUS_INVALID_DOMAIN_ROLE	SMB_STATUS_INVALID_PARAMETER_10
SMB Error Codes, 167	SMB Error Codes, 168
SMB_STATUS_INVALID_DOMAIN_STATE	SMB_STATUS_INVALID_PARAMETER_11
SMB Error Codes, 167	SMB Error Codes, 168
SMB_STATUS_INVALID_FILE_FOR_SECTION	SMB_STATUS_INVALID_PARAMETER_12
SMB Error Codes, 160	SMB Error Codes, 168
SMB_STATUS_INVALID_GROUP_ATTRIBUTES	SMB_STATUS_INVALID_PARAMETER_2
SMB Error Codes, 165	SMB Error Codes, 167
SMB_STATUS_INVALID_HANDLE	SMB_STATUS_INVALID_PARAMETER_3
SMB Error Codes, 159	SMB Error Codes, 167
SMB_STATUS_INVALID_HW_PROFILE	SMB_STATUS_INVALID_PARAMETER_4
SMB Error Codes, 177	SMB Error Codes, 167
SMB_STATUS_INVALID_ID_AUTHORITY	SMB_STATUS_INVALID_PARAMETER_5
SMB Error Codes, 164	SMB Error Codes, 168
SMB_STATUS_INVALID_IMAGE_FORMAT	SMB_STATUS_INVALID_PARAMETER_6
SMB Error Codes, 163	SMB Error Codes, 168
SMB_STATUS_INVALID_IMAGE_LE_FORMAT	SMB_STATUS_INVALID_PARAMETER_7
SMB Error Codes, 170	SMB Error Codes, 168
SMB_STATUS_INVALID_IMAGE_NE_FORMAT	SMB_STATUS_INVALID_PARAMETER_8
SMB Error Codes, 169	SMB Error Codes, 168
SMB_STATUS_INVALID_IMAGE_NOT_MZ	SMB_STATUS_INVALID_PARAMETER_9
SMB Error Codes, 170	SMB Error Codes, 168
SMB_STATUS_INVALID_IMAGE_PROTECT	SMB_STATUS_INVALID_PARAMETER_MIX
SMB Error Codes, 170	SMB Error Codes, 161
SMB_STATUS_INVALID_IMAGE_WIN_16	SMB_STATUS_INVALID_PIPE_STATE
SMB Error Codes, 170	SMB Error Codes, 165
SMB_STATUS_INVALID_INFO_CLASS	SMB_STATUS_INVALID_PLUGPLAY_DEVICE_PATH
SMB Error Codes, 159	SMB Error Codes, 177
SMB_STATUS_INVALID_LDT_DESCRIPTOR	SMB_STATUS_INVALID_PORT_ATTRIBUTES
SMB Error Codes, 169	SMB Error Codes, 160
SMB_STATUS_INVALID_LDT_OFFSET	SMB_STATUS_INVALID_PORT_HANDLE
SMB Error Codes, 169	SMB Error Codes, 161
SMB_STATUS_INVALID_LDT_SIZE	SMB_STATUS_INVALID_PRIMARY_GROUP
SMB Error Codes, 169	SMB Error Codes, 162
SMB_STATUS_INVALID_LEVEL	SMB_STATUS_INVALID_QUOTA_LOWER
SMB Error Codes, 171	SMB Error Codes, 161
SMB_STATUS_INVALID_LOCK_RANGE	SMB_STATUS_INVALID_READ_MODE
SMB Error Codes, 173	SMB Error Codes, 165
SMB_STATUS_INVALID_LOCK_SEQUENCE	SMB_STATUS_INVALID_SECURITY_DESCR
SMB Error Codes, 160	SMB Error Codes, 163
SMB_STATUS_INVALID_LOGON_HOURS	SMB_STATUS_INVALID_SERVER_STATE
SMB Error Codes, 163	SMB Error Codes, 167
SMB_STATUS_INVALID_LOGON_TYPE	SMB_STATUS_INVALID_SID
SMB Error Codes, 168	SMB Error Codes, 163
SMB_STATUS_INVALID_MEMBER	SMB_STATUS_INVALID_SUB_AUTHORITY
SMB Error Codes, 172	SMB Error Codes, 163
SMB_STATUS_INVALID_NETWORK_RESPONSE	SMB_STATUS_INVALID_SYSTEM_SERVICE
SMB Error Codes, 166	SMB Error Codes, 160
SMB_STATUS_INVALID_OPLOCK_PROTOCOL	SMB_STATUS_INVALID_TOKEN
SMB Error Codes, 167	SMB Error Codes, 177
SMB_STATUS_INVALID_OWNER	SMB_STATUS_INVALID_UNWIND_TARGET

SMB Error Codes, 160	SMB_STATUS_LOGON_TYPE_NOT_GRANTED
SMB_STATUS_INVALID_USER_BUFFER	SMB Error Codes, 171
SMB Error Codes, 167	SMB_STATUS_LOST_WRITEBEHIND_DATA
SMB_STATUS_INVALID_VARIANT	SMB Error Codes, 175
SMB Error Codes, 175	SMB_STATUS_LPC_REPLY_LOST
SMB_STATUS_INVALID_VIEW_SIZE	SMB Error Codes, 176
SMB Error Codes, 160	SMB_STATUS_LUIDS_EXHAUSTED
SMB_STATUS_INVALID_VOLUME_LABEL	SMB Error Codes, 163
SMB Error Codes, 164	SMB_STATUS_MAPPED_ALIGNMENT
SMB_STATUS_INVALID_WORKSTATION	SMB Error Codes, 174
SMB Error Codes, 163	SMB_STATUS_MAPPED_FILE_SIZE_ZERO
SMB_STATUS_IO_DEVICE_ERROR	SMB Error Codes, 169
SMB Error Codes, 173	SMB_STATUS_MARSHALL_OVERFLOW
SMB_STATUS_IO_PRIVILEGE_FAILED	SMB Error Codes, 175
SMB Error Codes, 170	SMB_STATUS_MEDIA_WRITE_PROTECTED
SMB_STATUS_IO_REPARSE_TAG_NOT_HANDLED	SMB Error Codes, 165
SMB Error Codes, 177	SMB_STATUS_MEMBER_IN_ALIAS
SMB_STATUS_IO_TIMEOUT	SMB Error Codes, 171
SMB Error Codes, 165	SMB_STATUS_MEMBER_IN_GROUP
SMB_STATUS_IP_ADDRESS_CONFLICT1	SMB Error Codes, 163
SMB Error Codes, 176	SMB_STATUS_MEMBER_NOT_IN_ALIAS
SMB_STATUS_IP_ADDRESS_CONFLICT2	SMB Error Codes, 171
SMB Error Codes, 176	SMB_STATUS_MEMBER_NOT_IN_GROUP
SMB_STATUS_KEY_DELETED	SMB Error Codes, 163
SMB Error Codes, 172	SMB_STATUS_MEMBERS_PRIMARY_GROUP
SMB_STATUS_KEY_HAS_CHILDREN	SMB Error Codes, 169
SMB Error Codes, 172	SMB_STATUS_MEMORY_NOT_ALLOCATED
SMB_STATUS_LAST_ADMIN	SMB Error Codes, 165
SMB Error Codes, 163	SMB_STATUS_MESSAGE_NOT_FOUND
SMB_STATUS_LICENSE_QUOTA_EXCEEDED	SMB Error Codes, 168
SMB Error Codes, 176	SMB_STATUS_MISSING_SYSTEMFILE
SMB_STATUS_LINK_FAILED	SMB Error Codes, 170
SMB Error Codes, 170	SMB_STATUS_MORE_ENTRIES
SMB_STATUS_LINK_TIMEOUT	SMB Error Codes, 158
SMB Error Codes, 170	SMB_STATUS_MORE_PROCESSING_REQUIRED
SMB_STATUS_LM_CROSS_ENCRYPTION_REQUIRED	SMB Error Codes, 160
SMB Error Codes, 172	SMB_STATUS_MUTANT_LIMIT_EXCEEDED
SMB_STATUS_LOCAL_DISCONNECT	SMB Error Codes, 173
SMB Error Codes, 170	SMB_STATUS_MUTANT_NOT_OWNED
SMB_STATUS_LOCK_NOT_GRANTED	SMB Error Codes, 161
SMB Error Codes, 162	SMB_STATUS_NAME_TOO_LONG
SMB_STATUS_LOG_FILE_FULL	SMB Error Codes, 168
SMB Error Codes, 173	SMB_STATUS_NET_WRITE_FAULT
SMB_STATUS_LOGIN_TIME_RESTRICTION	SMB Error Codes, 166
SMB Error Codes, 176	SMB_STATUS_NETLOGON_NOT_STARTED
SMB_STATUS_LOGIN_WKSTA_RESTRICTION	SMB Error Codes, 173
SMB Error Codes, 176	SMB_STATUS_NETWORK_ACCESS_DENIED
SMB_STATUS_LOGON_FAILURE	SMB Error Codes, 166
SMB Error Codes, 163	SMB_STATUS_NETWORK_BUSY
SMB_STATUS_LOGON_NOT_GRANTED	SMB Error Codes, 166
SMB Error Codes, 171	SMB_STATUS_NETWORK_CREDENTIAL_CONFLICT
SMB_STATUS_LOGON_SERVER_CONFLICT	SMB Error Codes, 173
SMB Error Codes, 170	SMB_STATUS_NETWORK_NAME_DELETED
SMB_STATUS_LOGON_SESSION_COLLISION	SMB Error Codes, 166
SMB Error Codes, 168	SMB_STATUS_NETWORK_SESSION_EXPIRED
SMB_STATUS_LOGON_SESSION_EXISTS	SMB Error Codes, 177
SMB Error Codes, 167	SMB_STATUS_NETWORK_UNREACHABLE

SMB Error Codes, 175	SMB_STATUS_NO_SUCH_USER
SMB_STATUS_NO_BROWSER_SERVERS_FOUND	SMB Error Codes, 162
SMB Error Codes, 174	SMB_STATUS_NO_TOKEN
SMB_STATUS_NO_CALLBACK_ACTIVE	SMB Error Codes, 163
SMB Error Codes, 176	SMB_STATUS_NO_TRUST_LSA_SECRET
SMB_STATUS_NO_EAS_ON_FILE	SMB Error Codes, 173
SMB Error Codes, 162	SMB_STATUS_NO_TRUST_SAM_ACCOUNT
SMB_STATUS_NO_EVENT_PAIR	SMB Error Codes, 173
SMB Error Codes, 171	SMB_STATUS_NO_USER_SESSION_KEY
SMB_STATUS_NO_GUID_TRANSLATION	SMB Error Codes, 173
SMB Error Codes, 168	SMB_STATUS_NOLOGON_INTERDOMAIN_TRUST_
SMB_STATUS_NO_IMPERSONATION_TOKEN	ACCOUNT
SMB Error Codes, 162	SMB Error Codes, 173
SMB_STATUS_NO_LDT	SMB_STATUS_NOLOGON_SERVER_TRUST_ACCO-
SMB Error Codes, 169	UNT
SMB_STATUS_NO_LOG_SPACE	SMB Error Codes, 173
SMB Error Codes, 172	SMB_STATUS_NOLOGON_WORKSTATION_TRUST-
SMB_STATUS_NO_LOGON_SERVERS	ACCOUNT
SMB Error Codes, 162	SMB Error Codes, 173
SMB_STATUS_NO_MEDIA	SMB_STATUS_NONCONTINUABLE_EXCEPTION
SMB Error Codes, 172	SMB Error Codes, 160
SMB_STATUS_NO_MEDIA_IN_DEVICE	SMB_STATUS_NONE_MAPPED
SMB Error Codes, 160	SMB Error Codes, 163
SMB_STATUS_NO_MEMORY	SMB_STATUS_NONEXISTENT_EA_ENTRY
SMB Error Codes, 160	SMB Error Codes, 162
SMB_STATUS_NO_MORE_ENTRIES	SMB_STATUS_NONEXISTENT_SECTOR
SMB Error Codes, 159	SMB Error Codes, 160
SMB_STATUS_NO_MORE_FILES	SMB_STATUS_NOT_A_DIRECTORY
SMB Error Codes, 159	SMB Error Codes, 168
SMB_STATUS_NO_PAGEFILE	SMB_STATUS_NOT_A_REPARSE_POINT
SMB Error Codes, 170	SMB Error Codes, 177
SMB_STATUS_NO_PREAUTH_INTEGRITY_HASH_	SMB_STATUS_NOT_CLIENT_SESSION
OVERLAP	SMB Error Codes, 174
SMB Error Codes, 177	SMB_STATUS_NOT_COMMITTED
SMB_STATUS_NO_SECURITY_ON_OBJECT	SMB Error Codes, 160
SMB Error Codes, 167	SMB_STATUS_NOT_FOUND
SMB_STATUS_NO_SPOOL_SPACE	SMB Error Codes, 175
SMB Error Codes, 166	SMB_STATUS_NOT_IMPLEMENTED
SMB_STATUS_NO_SUCH_ALIAS	SMB Error Codes, 159
SMB Error Codes, 171	SMB_STATUS_NOT_LOCKED
SMB_STATUS_NO_SUCH_DEVICE	SMB Error Codes, 160
SMB Error Codes, 159	SMB_STATUS_NOT_LOGON_PROCESS
SMB_STATUS_NO_SUCH_DOMAIN	SMB Error Codes, 167
SMB Error Codes, 167	SMB_STATUS_NOT_MAPPED_DATA
SMB_STATUS_NO_SUCH_FILE	SMB Error Codes, 164
SMB Error Codes, 159	SMB_STATUS_NOT_MAPPED_VIEW
SMB_STATUS_NO_SUCH_GROUP	SMB Error Codes, 160
SMB Error Codes, 162	SMB_STATUS_NOT_REGISTRY_FILE
SMB_STATUS_NO_SUCH_JOB	SMB Error Codes, 171
SMB Error Codes, 177	SMB_STATUS_NOT_SAME_DEVICE
SMB_STATUS_NO_SUCH_LOGON_SESSION	SMB Error Codes, 166
SMB Error Codes, 162	SMB_STATUS_NOT_SERVER_SESSION
SMB_STATUS_NO_SUCH_MEMBER	SMB Error Codes, 174
SMB Error Codes, 172	SMB_STATUS_NOT_SUPPORTED
SMB_STATUS_NO_SUCH_PACKAGE	SMB Error Codes, 166
SMB Error Codes, 168	SMB_STATUS_NOT_TINY_STREAM
SMB_STATUS_NO_SUCH_PRIVILEGE	SMB Error Codes, 175
SMB Error Codes, 162	SMB_STATUS_NOTIFY_CLEANUP

SMB Error Codes, 159	SMB_STATUS_PIPE_EMPTY
SMB_STATUS_NT_CROSS_ENCRYPTION_REQUIRED	SMB Error Codes, 167
SMB Error Codes, 171	SMB_STATUS_PIPE_LISTENING
SMB_STATUS_OBJECT_NAME_COLLISION	SMB Error Codes, 165
SMB Error Codes, 161	SMB_STATUS_PIPE_NOT_AVAILABLE
SMB_STATUS_OBJECT_NAME_INVALID	SMB Error Codes, 165
SMB Error Codes, 161	SMB_STATUS_PLUGPLAY_NO_DEVICE
SMB_STATUS_OBJECT_NAME_NOT_FOUND	SMB Error Codes, 176
SMB Error Codes, 161	SMB_STATUS_PORT_ALREADY_SET
SMB_STATUS_OBJECT_PATH_INVALID	SMB Error Codes, 161
SMB Error Codes, 161	SMB_STATUS_PORT_CONNECTION_REFUSED
SMB_STATUS_OBJECT_PATH_NOT_FOUND	SMB Error Codes, 161
SMB Error Codes, 161	SMB_STATUS_PORT_DISCONNECTED
SMB_STATUS_OBJECT_PATH_SYNTAX_BAD	SMB Error Codes, 161
SMB Error Codes, 161	SMB_STATUS_PORT_MESSAGE_TOO_LONG
SMB_STATUS_OBJECT_TYPE_MISMATCH	SMB Error Codes, 161
SMB Error Codes, 160	SMB_STATUS_PORT_UNREACHABLE
SMB_STATUS_OBJECTID_EXISTS	SMB Error Codes, 176
SMB Error Codes, 175	SMB_STATUS_POSSIBLE_DEADLOCK
SMB_STATUS_OK	SMB Error Codes, 173
SMB Error Codes, 158	SMB_STATUS_PRINT_CANCELLED
SMB_STATUS_OPEN_FAILED	SMB Error Codes, 166
SMB Error Codes, 170	SMB_STATUS_PRINT_QUEUE_FULL
SMB_STATUS_OPLOCK_NOT_GRANTED	SMB Error Codes, 166
SMB Error Codes, 167	SMB_STATUS_PRIVILEGE_NOT_HELD
SMB_STATUS_ORDINAL_NOT_FOUND	SMB Error Codes, 162
SMB Error Codes, 170	SMB_STATUS_PRIVILEGED_INSTRUCTION
SMB_STATUS_PAGEFILE_CREATE_FAILED	SMB Error Codes, 164
SMB Error Codes, 170	SMB_STATUS_PROCEDURE_NOT_FOUND
SMB_STATUS_PAGEFILE_QUOTA	SMB Error Codes, 163
SMB Error Codes, 159	SMB_STATUS_PROCESS_IS_TERMINATING
SMB_STATUS_PAGEFILE_QUOTA_EXCEEDED	SMB Error Codes, 168
SMB Error Codes, 170	SMB_STATUS_PROFILING_AT_LIMIT
SMB_STATUS_PARITY_ERROR	SMB Error Codes, 166
SMB Error Codes, 160	SMB_STATUS_PROFILING_NOT_STARTED
SMB_STATUS_PARTITION_FAILURE	SMB Error Codes, 165
SMB Error Codes, 172	SMB_STATUS_PROFILING_NOT_STOPPED
SMB_STATUS_PASSWORD_EXPIRED	SMB Error Codes, 165
SMB Error Codes, 163	SMB_STATUS_PROPSET_NOT_FOUND
SMB_STATUS_PASSWORD_MUST_CHANGE	SMB Error Codes, 175
SMB Error Codes, 175	SMB_STATUS_PROTOCOL_UNREACHABLE
SMB_STATUS_PASSWORD_RESTRICTION	SMB Error Codes, 176
SMB Error Codes, 163	SMB_STATUS_PWD_HISTORY_CONFLICT
SMB_STATUS_PATH_NOT_COVERED	SMB Error Codes, 176
SMB Error Codes, 176	SMB_STATUS_PWD_TOO_RECENT
SMB_STATUS_PENDING	SMB Error Codes, 176
SMB Error Codes, 158	SMB_STATUS_PWD_TOO_SHORT
SMB_STATUS_PIPE_BROKEN	SMB Error Codes, 176
SMB Error Codes, 171	SMB_STATUS_QUOTA_EXCEEDED
SMB_ERROR_CODES, 165	SMB Error Codes, 161
SMB_STATUS_PIPE_BUSY	SMB_STATUS_QUOTA_LIST_INCONSISTENT
SMB Error Codes, 165	SMB Error Codes, 177
SMB_STATUS_PIPE_CLOSING	SMB_STATUS_RANGE_NOT_LOCKED
SMB Error Codes, 165	SMB Error Codes, 163
SMB_STATUS_PIPE_CONNECTED	SMB_STATUS_RECOVERY_FAILURE
SMB Error Codes, 165	SMB Error Codes, 175
SMB_STATUS_PIPE_DISCONNECTED	SMB_STATUS_REDIRECTOR_NOT_STARTED
SMB Error Codes, 165	SMB Error Codes, 168

SMB_STATUS_REDIRECTOR_PAUSED	SMB_STATUS_SERVER_NOT_DISABLED
SMB Error Codes, 166	SMB Error Codes, 163
SMB_STATUS_REDIRECTOR_STARTED	SMB_STATUS_SERVER_UNAVAILABLE
SMB Error Codes, 168	SMB Error Codes, 177
SMB_STATUS_REGISTRY_CORRUPT	SMB_STATUS_SHARE_UNAVAILABLE
SMB Error Codes, 171	SMB Error Codes, 177
SMB_STATUS_REGISTRY_IO_FAILED	SMB_STATUS_SHARED_IRQ_BUSY
SMB Error Codes, 171	SMB Error Codes, 172
SMB_STATUS_REGISTRY_QUOTA_LIMIT	SMB_STATUS_SHARING_PAUSED
SMB Error Codes, 176	SMB Error Codes, 166
SMB_STATUS_REMOTE_DISCONNECT	SMB_STATUS_SHARING_VIOLATION
SMB Error Codes, 170	SMB Error Codes, 161
SMB_STATUS_REMOTE_NOT_LISTENING	SMB_STATUS_SOME_UNMAPPED
SMB Error Codes, 166	SMB Error Codes, 159
SMB_STATUS_REMOTE_RESOURCES	SMB_STATUS_SPECIAL_ACCOUNT
SMB Error Codes, 170	SMB Error Codes, 169
SMB_STATUS_REMOTE_SESSION_LIMIT	SMB_STATUS_SPECIAL_GROUP
SMB Error Codes, 173	SMB Error Codes, 169
SMB_STATUS_REPLY_MESSAGE_MISMATCH	SMB_STATUS_SPECIAL_USER
SMB Error Codes, 174	SMB Error Codes, 169
SMB_STATUS_REQUEST_ABORTED	SMB_STATUS_STACK_OVERFLOW
SMB Error Codes, 176	SMB Error Codes, 168
SMB_STATUS_REQUEST_NOT_ACCEPTED	SMB_STATUS_STACK_OVERFLOW_READ
SMB Error Codes, 166	SMB Error Codes, 175
SMB_STATUS_RESOURCE_DATA_NOT_FOUND	SMB_STATUS_STOPPED_ON_SYMLINK
SMB Error Codes, 164	SMB Error Codes, 159
SMB_STATUS_RESOURCE_LANG_NOT_FOUND	SMB_STATUS_SUCCESS
SMB Error Codes, 173	SMB Error Codes, 158
SMB_STATUS_RESOURCE_NAME_NOT_FOUND	SMB_STATUS_SUSPEND_COUNT_EXCEEDED
SMB Error Codes, 164	SMB Error Codes, 161
SMB_STATUS_RESOURCE_NOT_OWNED	SMB_STATUS_SYNCHRONIZATION_REQUIRED
SMB Error Codes, 177	SMB Error Codes, 170
SMB_STATUS_RESOURCE_TYPE_NOT_FOUND	SMB_STATUS_SYSTEM_PROCESS_TERMINATED
SMB Error Codes, 164	SMB Error Codes, 174
SMB_STATUS_RETRY	SMB_STATUS_THREAD_IS_TERMINATING
SMB Error Codes, 175	SMB Error Codes, 162
SMB_STATUS_REVISION_MISMATCH	SMB_STATUS_THREAD_NOT_IN_PROCESS
SMB Error Codes, 162	SMB Error Codes, 169
SMB_STATUS_RXACT_COMMIT_FAILURE	SMB_STATUS_TIME_DIFFERENCE_AT_DC
SMB Error Codes, 169	SMB Error Codes, 170
SMB_STATUS_RXACT_INVALID_STATE	SMB_STATUS_TIMER_NOT_CANCELED
SMB Error Codes, 169	SMB Error Codes, 159
SMB_STATUS_SECRET_TOO_LONG	SMB_STATUS_TIMER_RESOLUTION_NOT_SET
SMB Error Codes, 171	SMB Error Codes, 176
SMB_STATUS_SECTION_NOT_EXTENDED	SMB_STATUS_TOKEN_ALREADY_IN_USE
SMB Error Codes, 164	SMB Error Codes, 169
SMB_STATUS_SECTION_NOT_IMAGE	SMB_STATUS_TOO_LATE
SMB Error Codes, 161	SMB Error Codes, 173
SMB_STATUS_SECTION_PROTECTION	SMB_STATUS_TOO_MANY_ADDRESSES
SMB Error Codes, 162	SMB Error Codes, 174
SMB_STATUS_SECTION_TOO_BIG	SMB_STATUS_TOO_MANY_COMMANDS
SMB Error Codes, 161	SMB Error Codes, 166
SMB_STATUS_SEMAPHORE_LIMIT_EXCEEDED	SMB_STATUS_TOO_MANY_CONTEXT_IDS
SMB Error Codes, 161	SMB Error Codes, 171
SMB_STATUS_SERIAL_NO_DEVICE_INITED	SMB_STATUS_TOO_MANY_GUIDS_REQUESTED
SMB Error Codes, 171	SMB Error Codes, 163
SMB_STATUS_SERVER_DISABLED	SMB_STATUS_TOO_MANY_LINKS
SMB Error Codes, 163	SMB Error Codes, 177

SMB_STATUS_TOO_MANY_LUIDS_REQUESTED	SMB_STATUS_UNEXPECTED_NETWORK_ERROR
SMB Error Codes, 163	SMB Error Codes, 166
SMB_STATUS_TOO_MANY_NAMES	SMB_STATUS_UNHANDLED_EXCEPTION
SMB Error Codes, 166	SMB Error Codes, 170
SMB_STATUS_TOO_MANY_NODES	SMB_STATUS_UNKNOWN_PRINTER_DRIVER
SMB Error Codes, 174	SMB Error Codes, 159
SMB_STATUS_TOO_MANY_OPENED_FILES	SMB_STATUS_UNKNOWN_REVISION
SMB Error Codes, 169	SMB Error Codes, 162
SMB_STATUS_TOO_MANY_PAGING_FILES	SMB_STATUS_UNMAPPABLE_CHARACTER
SMB Error Codes, 164	SMB Error Codes, 171
SMB_STATUS_TOO_MANY_SECRETS	SMB_STATUS_UNRECOGNIZED_MEDIA
SMB Error Codes, 171	SMB Error Codes, 160
SMB_STATUS_TOO_MANY_SESSIONS	SMB_STATUS_UNRECOGNIZED_VOLUME
SMB Error Codes, 166	SMB Error Codes, 171
SMB_STATUS_TOO_MANY_SIDS	SMB_STATUS_UNSUCCESSFUL
SMB Error Codes, 172	SMB Error Codes, 159
SMB_STATUS_TOO_MANY_THREADS	SMB_STATUS_UNSUPPORTED_COMPRESSION
SMB Error Codes, 169	SMB Error Codes, 176
SMB_STATUS_TRANSACTION_ABORTED	SMB_STATUS_UNWIND
SMB Error Codes, 174	SMB Error Codes, 160
SMB_STATUS_TRANSACTION_INVALID_ID	SMB_STATUS_USER_EXISTS
SMB Error Codes, 174	SMB Error Codes, 162
SMB_STATUS_TRANSACTION_INVALID_TYPE	SMB_STATUS_USER_MAPPED_FILE
SMB Error Codes, 174	SMB Error Codes, 176
SMB_STATUS_TRANSACTION_NO_MATCH	SMB_STATUS_USER_SESSION_DELETED
SMB Error Codes, 174	SMB Error Codes, 173
SMB_STATUS_TRANSACTION_NO_RELEASE	SMB_STATUS_VDM_HARD_ERROR
SMB Error Codes, 174	SMB Error Codes, 174
SMB_STATUS_TRANSACTION_RESPONDED	SMB_STATUS_VIRTUAL_CIRCUIT_CLOSED
SMB Error Codes, 174	SMB Error Codes, 166
SMB_STATUS_TRANSACTION_TIMED_OUT	SMB_STATUS_VOLUME_DISMOUNTED
SMB Error Codes, 174	SMB Error Codes, 177
SMB_STATUS_TRUST_FAILURE	SMB_STATUS_WORKING_SET_QUOTA
SMB Error Codes, 173	SMB Error Codes, 165
SMB_STATUS_TRUSTED_DOMAIN_FAILURE	SMB_STATUS_WRONG_PASSWORD
SMB Error Codes, 173	SMB Error Codes, 163
SMB_STATUS_TRUSTED_RELATIONSHIP_FAILURE	SMB_STATUS_WRONG_PASSWORD_CORE
SMB Error Codes, 173	SMB Error Codes, 171
SMB_STATUS_UNABLE_TO_DECOMMIT_VM	SMB_STATUS_WRONG_VOLUME
SMB Error Codes, 160	SMB Error Codes, 159
SMB_STATUS_UNABLE_TO_DELETE_SECTION	SRV_ERRaccess
SMB Error Codes, 160	SMB Errors, 147
SMB_STATUS_UNABLE_TO_FREE_VM	SRV_ERRbadpermits
SMB Error Codes, 160	SMB Errors, 147
SMB_STATUS_UNABLE_TO_LOCK_MEDIA	SRV_ERRbadpw
SMB Error Codes, 172	SMB Errors, 147
SMB_STATUS_UNABLE_TO_UNLOAD_MEDIA	SRV_ERRcontmpx
SMB Error Codes, 172	SMB Errors, 147
SMB_STATUS_UNDEFINED_CHARACTER	SRV_ERRerror
SMB Error Codes, 171	SMB Errors, 147
SMB_STATUS_UNEXPECTED_IO_ERROR	SRV_ERRfilespecs
SMB Error Codes, 167	SMB Errors, 147
SMB_STATUS_UNEXPECTED_MM_CREATE_ERR	SRV_ERRinvdevice
SMB Error Codes, 167	SMB Errors, 147
SMB_STATUS_UNEXPECTED_MM_EXTEND_ERR	SRV_ERRinvfid
SMB Error Codes, 167	SMB Errors, 147
SMB_STATUS_UNEXPECTED_MM_MAP_ERROR	SRV_ERRinvnetname
SMB Error Codes, 167	SMB Errors, 147

SRV_ERRinvtid	SY_ATTR_SPARSE_FILE
SMB Errors, 147	File Attributes, 128
SRV_ERRinvid	SY_ATTR_SYSTEM
SMB Errors, 147	File Attributes, 127
SRV_ERRmsgoff	SY_ATTR_TEMPORARY
SMB Errors, 147	File Attributes, 128
SRV_ERRnoresource	SY_ATTR_VOLUME
SMB Errors, 147	File Attributes, 127
SRV_ERRnoroom	SY_DEVICEMODE_BITSPERPEL
SMB Errors, 147	Printer Flags, 131
SRV_ERRnosupport	SY_DEVICEMODE_COLLATE
SMB Errors, 148	Printer Flags, 131
SRV_ERRpaused	SY_DEVICEMODE_COLOR
SMB Errors, 147	Printer Flags, 131
SRV_ERRqeof	SY_DEVICEMODE_COPIES
SMB Errors, 147	Printer Flags, 130
SRV_ERRqfull	SY_DEVICEMODE_DEFAULTSOURCE
SMB Errors, 147	Printer Flags, 131
SRV_ERRqtoobig	SY_DEVICEMODE_DISPLAYFLAGS
SMB Errors, 147	Printer Flags, 131
SRV_ERRrmuns	SY_DEVICEMODE_DISPLAYFREQUENCY
SMB Errors, 147	Printer Flags, 131
SRV_ERRsetattrmode	SY_DEVICEMODE_DITHERTYPE
SMB Errors, 147	Printer Flags, 131
SRV_ERRsmbcmd	SY_DEVICEMODE_DUPLEX
SMB Errors, 147	Printer Flags, 131
SRV_ERRsrverror	SY_DEVICEMODE_FORMNAME
SMB Errors, 147	Printer Flags, 131
SRV_ERRtimeout	SY_DEVICEMODE_ICMINTENT
SMB Errors, 147	Printer Flags, 131
SRV_ERRtoomanyuids	SY_DEVICEMODE_ICMMETHOD
SMB Errors, 147	Printer Flags, 131
SRV_ERRusempx	SY_DEVICEMODE_LOGPIXELS
SMB Errors, 147	Printer Flags, 131
SRV_ERRusestd	SY_DEVICEMODE_MEDIATYPE
SMB Errors, 147	Printer Flags, 131
SY_ATTR_ARCHIVE	SY_DEVICEMODE_NUP
File Attributes, 127	Printer Flags, 130
SY_ATTR_COMPRESSED	SY_DEVICEMODE_ORIENTATION
File Attributes, 128	Printer Flags, 130
SY_ATTR_DEVICE	SY_DEVICEMODE_PANNINGHEIGHT
File Attributes, 127	Printer Flags, 131
SY_ATTR_DIRECTORY	SY_DEVICEMODE_PANNINGWIDTH
File Attributes, 127	Printer Flags, 131
SY_ATTR_ENCRYPTED	SY_DEVICEMODE_PAPERLENGTH
File Attributes, 128	Printer Flags, 130
SY_ATTR_HIDDEN	SY_DEVICEMODE_PAPERSIZE
File Attributes, 127	Printer Flags, 130
SY_ATTR_NORMAL	SY_DEVICEMODE_PAPERWIDTH
File Attributes, 127	Printer Flags, 130
SY_ATTR_NOT_CONTENT_INDEXED	SY_DEVICEMODE_PELHEIGHT
File Attributes, 128	Printer Flags, 131
SY_ATTR_OFFLINE	SY_DEVICEMODE_PELWIDTH
File Attributes, 128	Printer Flags, 131
SY_ATTR_READONLY	SY_DEVICEMODE_POSITION
File Attributes, 127	Printer Flags, 130
SY_ATTR_REPARSE_POINT	SY_DEVICEMODE_PRINTQUALITY
File Attributes, 128	Printer Flags, 131

SY_DEVICEMODE_SCALE	Printer Flags, 130	SY_PRINTERCONTROL_RESUME	Printer Control Commands, 140
SY_DEVICEMODE_TTOPTION	Printer Flags, 131	SY_PRINTERCONTROL_SETSTATUS	Printer Control Commands, 140
SY_DEVICEMODE_YRESOLUTION	Printer Flags, 131	SY_PRINTEROSVERSION_WIN	Printer, 129
SY_PRINTERATTR_BIDI	Printer Attribute Flags, 135	SY_PRINTERSTATUS_BUSY	Printer Status Flags, 132
SY_PRINTERATTR_CONTAINER	Printer Attribute Flags, 135	SY_PRINTERSTATUS_DOOROPEN	Printer Status Flags, 133
SY_PRINTERATTR_DEFAULT	Printer Attribute Flags, 134	SY_PRINTERSTATUS_ERROR	Printer Status Flags, 132
SY_PRINTERATTR_DEVQUERY	Printer Attribute Flags, 135	SY_PRINTERSTATUS_INITIALIZING	Printer Status Flags, 133
SY_PRINTERATTR_DIRECT	Printer Attribute Flags, 134	SY_PRINTERSTATUS_IOACTIVE	Printer Status Flags, 132
SY_PRINTERATTR_EXPAND	Printer Attribute Flags, 135	SY_PRINTERSTATUS_MANUALFEED	Printer Status Flags, 132
SY_PRINTERATTR_ICON1	Printer Attribute Flags, 135	SY_PRINTERSTATUS_NOTAVAILABLE	Printer Status Flags, 133
SY_PRINTERATTR_ICON2	Printer Attribute Flags, 135	SY_PRINTERSTATUS_NOTONER	Printer Status Flags, 133
SY_PRINTERATTR_ICON3	Printer Attribute Flags, 135	SY_PRINTERSTATUS_OFFLINE	Printer Status Flags, 132
SY_PRINTERATTR_ICON4	Printer Attribute Flags, 135	SY_PRINTERSTATUS_OUTOFMEMORY	Printer Status Flags, 133
SY_PRINTERATTR_ICON5	Printer Attribute Flags, 135	SY_PRINTERSTATUS_OUTPUTBINFULL	Printer Status Flags, 133
SY_PRINTERATTR_ICON6	Printer Attribute Flags, 136	SY_PRINTERSTATUS_PAGEPUNT	Printer Status Flags, 133
SY_PRINTERATTR_ICON7	Printer Attribute Flags, 136	SY_PRINTERSTATUS_PAPERJAM	Printer Status Flags, 132
SY_PRINTERATTR_ICON8	Printer Attribute Flags, 136	SY_PRINTERSTATUS_PAPEROUT	Printer Status Flags, 132
SY_PRINTERATTR_KEEPJOBS	Printer Attribute Flags, 135	SY_PRINTERSTATUS_PAPERPROBLEM	Printer Status Flags, 132
SY_PRINTERATTR_LIFO	Printer Attribute Flags, 135	SY_PRINTERSTATUS_PAUSED	Printer Status Flags, 132
SY_PRINTERATTR_LOCAL	Printer Attribute Flags, 135	SY_PRINTERSTATUS_PENDINGDELETION	Printer Status Flags, 132
SY_PRINTERATTR_NETWORK	Printer Attribute Flags, 134	SY_PRINTERSTATUS_POWERSAVE	Printer Status Flags, 133
SY_PRINTERATTR_OFFLINE	Printer Attribute Flags, 135	SY_PRINTERSTATUS_PRINTING	Printer Status Flags, 133
SY_PRINTERATTR_PUBLISHED	Printer Attribute Flags, 135	SY_PRINTERSTATUS_PROCESSING	Printer Status Flags, 133
SY_PRINTERATTR_QUEUED	Printer Attribute Flags, 134	SY_PRINTERSTATUS_SERVERUNKNOWN	Printer Status Flags, 133
SY_PRINTERATTR_RAWONLY	Printer Attribute Flags, 135	SY_PRINTERSTATUS_TONERLOW	Printer Status Flags, 133
SY_PRINTERATTR_RESERVED	Printer Attribute Flags, 135	SY_PRINTERSTATUS_USERINTERVENTION	Printer Status Flags, 133
SY_PRINTERATTR_SHARED	Printer Attribute Flags, 134	SY_PRINTERSTATUS_WAITING	Printer Status Flags, 133
SY_PRINTERCONTROL_PAUSE	Printer Control Commands, 140	SY_PRINTERSTATUS_WARMINGUP	Printer Status Flags, 133
SY_PRINTERCONTROL_PURGE	Printer Control Commands, 140	SY_PRINTFORMFLAG_BUILTIN	Printer Form Flags, 142

SY_PRINTFORMFLAG_PRINTER
Printer Form Flags, [142](#)

SY_PRINTFORMFLAG_USER
Printer Form Flags, [142](#)

SY_PRINTJOB_DEFPRRIORITY
Printer Priority Job, [139](#)

SY_PRINTJOB_MAXPRRIORITY
Printer Priority Job, [139](#)

SY_PRINTJOB_MINPRRIORITY
Printer Priority Job, [139](#)

SY_PRINTJOB_COM_CANCEL
Printer Job Control Commands, [141](#)

SY_PRINTJOB_COM_DELETE
Printer Job Control Commands, [141](#)

SY_PRINTJOB_COM_LASTPAGEEJECTED
Printer Job Control Commands, [141](#)

SY_PRINTJOB_COM_PAUSE
Printer Job Control Commands, [141](#)

SY_PRINTJOB_COM_RESTART
Printer Job Control Commands, [141](#)

SY_PRINTJOB_COM_RESUME
Printer Job Control Commands, [141](#)

SY_PRINTJOB_COM_SENDDTOPRINTER
Printer Job Control Commands, [141](#)

SY_PRINTJOBSTATUS_BLOCKED
Printer Status Job, [138](#)

SY_PRINTJOBSTATUS_DELETED
Printer Status Job, [138](#)

SY_PRINTJOBSTATUS_DELETING
Printer Status Job, [137](#)

SY_PRINTJOBSTATUS_ERROR
Printer Status Job, [137](#)

SY_PRINTJOBSTATUS_OFFLINE
Printer Status Job, [137](#)

SY_PRINTJOBSTATUS_PAPERPROBLEM
Printer Status Job, [137](#)

SY_PRINTJOBSTATUS_PAUSED
Printer Status Job, [137](#)

SY_PRINTJOBSTATUS_PRINTED
Printer Status Job, [137](#)

SY_PRINTJOBSTATUS_PRINTING
Printer Status Job, [137](#)

SY_PRINTJOBSTATUS_RESTART
Printer Status Job, [138](#)

SY_PRINTJOBSTATUS_SPOOLING
Printer Status Job, [137](#)

SY_PRINTJOBSTATUS_USERINTERVENTION
Printer Status Job, [138](#)

SYDeviceMode, [251](#)
bitsPerPel, [252](#)
collate, [252](#)
color, [252](#)
copies, [252](#)
defaultSource, [252](#)
displayFlags, [252](#)
displayFrequency, [252](#)
ditherType, [253](#)
driverExtraData, [252](#)
driverExtraLength, [252](#)
driverVersion, [252](#)
duplex, [252](#)
fields, [252](#)
formName, [252](#)
icmIntent, [252](#)
icmMethod, [252](#)
logPixels, [252](#)
mediaType, [253](#)
orientation, [252](#)
panningHeight, [253](#)
panningWidth, [253](#)
paperLength, [252](#)
paperSize, [252](#)
paperWidth, [252](#)
pelsHeight, [252](#)
pelsWidth, [252](#)
printQuality, [252](#)
reserved1, [253](#)
reserved2, [253](#)
scale, [252](#)
size, [252](#)
specVersion, [252](#)
tOption, [252](#)
yResolution, [252](#)

SYFileInformation, [253](#)
allocSizeHigh, [254](#)
allocSizeLow, [254](#)
attributes, [254](#)
creationTime, [253](#)
fileIdHigh, [254](#)
fileIdLow, [254](#)
isDeleted, [254](#)
lastAccessTime, [254](#)
lastChangeTime, [253](#)
lastWriteTime, [254](#)
numLinks, [254](#)
sizeHigh, [254](#)
sizeLow, [254](#)

SYPrintFormInfo, [258](#)
flags, [258](#)
id, [258](#)
imageableArea, [259](#)
name, [258](#)
size, [259](#)

SYPrintJobInfo, [259](#)
documentName, [259](#)
id, [259](#)
pStatus, [259](#)
pagesPrinted, [260](#)
position, [260](#)
priority, [260](#)
securityDescriptor, [260](#)
securityDescriptorLength, [260](#)
size, [260](#)
startTime, [260](#)
status, [260](#)
submitTime, [260](#)

- time, [260](#)
- totalPages, [260](#)
- untilTime, [260](#)
- user, [261](#)
- SYPrintRect, [261](#)
 - bottom, [261](#)
 - left, [261](#)
 - right, [261](#)
 - top, [261](#)
- SYPrintSize, [262](#)
 - height, [262](#)
 - width, [262](#)
- SYPrinterDriver, [255](#)
 - attributes, [256](#)
 - configFile, [255](#)
 - configVersion, [256](#)
 - dataFile, [255](#)
 - defaultDataType, [255](#)
 - dependentFiles, [255](#)
 - driverDate, [255](#)
 - driverPath, [255](#)
 - driverVersion, [256](#)
 - driverVersions, [255](#)
 - hardwareID, [256](#)
 - helpFile, [255](#)
 - manufacturer, [256](#)
 - manufacturerURL, [256](#)
 - monitorName, [255](#)
 - name, [255](#)
 - osVersion, [255](#)
 - previousNames, [255](#)
 - provider, [256](#)
- SYPrinterInfo, [256](#)
 - attributes, [257](#)
 - averagePpm, [258](#)
 - buildVersion, [258](#)
 - cJobs, [257](#)
 - cSetPrinter, [258](#)
 - comment, [257](#)
 - dataType, [257](#)
 - defaultPriority, [257](#)
 - devMode, [257](#)
 - deviceNotSelectedTimeout, [258](#)
 - driverName, [257](#)
 - flags, [257](#)
 - globalCounter, [257](#)
 - location, [257](#)
 - majorVersion, [257](#)
 - parameters, [257](#)
 - portName, [257](#)
 - printProcessor, [257](#)
 - printerErrors, [258](#)
 - priority, [257](#)
 - securityDescriptor, [258](#)
 - securityDescriptorLength, [258](#)
 - sepFile, [257](#)
 - sessionCounter, [258](#)
 - startTime, [257](#)
 - status, [257](#)
 - totalBytes, [257](#)
 - totalJobs, [257](#)
 - totalPages, [257](#)
 - transmissionRetryTimeout, [258](#)
 - untilTime, [257](#)
- SYTimeFragments, [262](#)
 - day, [263](#)
 - hour, [263](#)
 - min, [263](#)
 - month, [263](#)
 - sec, [263](#)
 - year, [263](#)
- SYVolumeInformation, [263](#)
 - blockSizeHigh, [264](#)
 - blockSizeLow, [264](#)
 - blocksPerUnitHigh, [264](#)
 - blocksPerUnitLow, [264](#)
 - creationTimeHigh, [264](#)
 - creationTimeLow, [264](#)
 - fileSystemIdHigh, [264](#)
 - fileSystemIdLow, [264](#)
 - freeUnitsHigh, [265](#)
 - freeUnitsLow, [265](#)
 - serialNumberHigh, [264](#)
 - serialNumberLow, [264](#)
 - totalUnitsHigh, [265](#)
 - totalUnitsLow, [264](#)
- scale
 - SYDeviceMode, [252](#)
- Search Operations, [33](#)
 - ccFindClose, [35](#)
 - ccFindFirstFile, [34](#)
 - ccFindFirstFileA, [33](#)
 - ccFindNextFile, [35](#)
 - ccFindNextFileA, [34](#)
 - FindFileDataA_t, [33](#)
- sec
 - SYTimeFragments, [263](#)
- sectorsPerCluster
 - CCVolumeInfo, [237](#)
- Security Mechanisms, [10](#)
 - AM_MECH_KERBEROS, [10](#)
 - AM_MECH_NTLMSSP, [10](#)
- Security Password Type, [11](#)
 - AM_CREDENTIALS_PASSWORD_MD4, [11](#)
 - AM_CREDENTIALS_PASSWORD_PLAINTEXT, [11](#)
- securityDescriptor
 - SYPrinterInfo, [258](#)
 - SYPrintJobInfo, [260](#)
- securityDescriptorLength
 - SYPrinterInfo, [258](#)
 - SYPrintJobInfo, [260](#)
- sepFile
 - SYPrinterInfo, [257](#)
- serialNumber
 - CCVolumeInfo, [237](#)

- ul style="list-style-type: none; padding-left: 0;">
- serialNumberHigh
 - SYVolumeInformation, [264](#)
- serialNumberLow
 - SYVolumeInformation, [264](#)
- Server, [106](#)
- Server capabilities, [24](#)
 - CC_CAPABILITY_DFS, [24](#)
 - CC_CAPABILITY_DIRECTORY_LEASING, [24](#)
 - CC_CAPABILITY_ENCRYPTION, [24](#)
 - CC_CAPABILITY_LARGE_MTU, [24](#)
 - CC_CAPABILITY_LEASING, [24](#)
 - CC_CAPABILITY_MULTI_CHANNEL, [24](#)
 - CC_CAPABILITY_PERSISTENT_HANDLES, [24](#)
- serverCapabilities
 - CCServerInfo, [236](#)
- serverIP
 - CMResolverRegisteredMethodDescription, [242](#)
- serverInfo
 - CCMountInfo, [231](#)
 - CCMountInfoA, [232](#)
- serverName
 - CCMountInfo, [231](#)
 - CCMountInfoA, [232](#)
- serverSigningPolicy
 - CCServerInfo, [236](#)
- sessionCounter
 - SYPrinterInfo, [258](#)
- sha256
 - CMCrypterList, [239](#)
- sha512
 - CMCrypterList, [240](#)
- shareName
 - EVCSOpenFileEntry, [245](#)
 - EVCSShareConnectionEntry, [245](#)
 - UDShareAccessEvent, [267](#)
- Shares Management, [108](#)
 - csAddShare, [108](#)
 - csAddShareA, [108](#)
 - csRemoveShare, [109](#)
 - csRemoveShareA, [109](#)
 - csSetShareEncryption, [110](#)
 - csSetShareEncryptionA, [110](#)
- sign
 - NQ_INT64, [247](#)
- signing policy, [25](#)
 - CC_SIGNING_ENABLED, [25](#)
 - CC_SIGNING_REQUIRED, [25](#)
- size
 - SYDeviceMode, [252](#)
 - SYPrintFormInfo, [259](#)
 - SYPrintJobInfo, [260](#)
- sizeHigh
 - SYFileInformation, [254](#)
 - UDFileAccessEvent, [266](#)
- sizeLow
 - SYFileInformation, [254](#)
 - UDFileAccessEvent, [266](#)
- specVersion
 - SYDeviceMode, [252](#)
- Special Operations, [70](#)
 - ccGetMaxSecurityLevel, [71](#)
 - ccGetMinSecurityLevel, [71](#)
 - ccGetSigningPolicy, [70](#)
 - cclsExclusiveAccessToFile, [73](#)
 - cclsExclusiveAccessToFileA, [72](#)
 - ccSetExclusiveAccessToFile, [72](#)
 - ccSetExclusiveAccessToFileA, [71](#)
 - ccSetMaxSecurityLevel, [71](#)
 - ccSetMinSecurityLevel, [70](#)
 - ccSetSigningPolicy, [70](#)
- startTime
 - SYPrinterInfo, [257](#)
 - SYPrintJobInfo, [260](#)
- statistics
 - CCMountInfo, [231](#)
 - CCMountInfoA, [232](#)
- status
 - SYPrinterInfo, [257](#)
 - SYPrintJobInfo, [260](#)
- submitTime
 - SYPrintJobInfo, [260](#)
- sylnit
 - System API, [124](#)
- syStop
 - System API, [124](#)
- syapi.h, [277](#)
- sycommon.h, [278](#)
- System API, [124](#)
 - sylnit, [124](#)
 - syStop, [124](#)
- System Common API, [125](#)
- TCP and NetBIOS Layer Error Codes, [191](#)
 - NQ_ERR_ATTRERROR, [194](#)
 - NQ_ERR_CAPTUREZERO, [193](#)
 - NQ_ERR_ICONVERROR, [195](#)
 - NQ_ERR_ICONVOPEN, [195](#)
 - NQ_ERR_INVALIDUINT32SIZE, [193](#)
 - NQ_ERR_KERBEROSERROR, [194](#)
 - NQ_ERR_NBBUFFEROVERFLOW, [193](#)
 - NQ_ERR_NBCANCELLISTENFAIL, [192](#)
 - NQ_ERR_NBDDCOMMUNICATIONERROR, [192](#)
 - NQ_ERR_NBHOSTNAMENOTRESOLVED, [192](#)
 - NQ_ERR_NBILLEGALDATAGRAMDESTINATION, [192](#)
 - NQ_ERR_NBILLEGALDATAGRAMSOURCE, [192](#)
 - NQ_ERR_NBILLEGALDATAGRAMTYPE, [192](#)
 - NQ_ERR_NBILLEGALSOCKETSLLOT, [191](#)
 - NQ_ERR_NBINTERNALERROR, [192](#)
 - NQ_ERR_NBINVALIDPARAMETER, [192](#)
 - NQ_ERR_NBLISTENFAIL, [193](#)
 - NQ_ERR_NBNEGATIVEVERESONSE, [192](#)
 - NQ_ERR_NBNOBINDBEFORELISTEN, [192](#)
 - NQ_ERR_NBNOTNETBIOSNAME, [191](#)
 - NQ_ERR_NBRELEASENAMEFAIL, [193](#)
 - NQ_ERR_NBSOCKETOVERFLOW, [192](#)
 - NQ_ERR_NBTIMEOUT, [192](#)

NQ_ERR_RAPACCESSDENIED, 194
 NQ_ERR_RAPBADTRANSACTCONFIG, 194
 NQ_ERR_RAPDEVICENOTCONNECTED, 194
 NQ_ERR_RAPGENERALERROR, 194
 NQ_ERR_RAPMOREDATA, 194
 NQ_ERR_RAPNOBROWSERVERSERVERSFOUND, 195
 NQ_ERR_RAPNOTACTIVEINDOMAIN, 194
 NQ_ERR_RAPSERVERNOTSTARTED, 194
 NQ_ERR_RAPSERVICENOTINSTALLED, 194
 NQ_ERR_RECONNECTREQUIRED, 193
 NQ_ERR_SEEKERROR, 194
 NQ_ERR_SIZEERROR, 194
 NQ_ERR_SOCKETADDRESS, 193
 NQ_ERR_SOCKETBIND, 193
 NQ_ERR_SOCKETCREATE, 193
 NQ_ERR_SOCKETNAME, 193
 NQ_ERR_SOCKETSEND, 193
 NQ_ERR_SOCKETSETOPTION, 193
 UD_NQ_MAXPWDLEN, 195
 tid
 UDFileAccessEvent, 266
 UDShareAccessEvent, 267
 time
 SYPrintJobInfo, 260
 timeHigh
 CCFileTime, 227
 timeLow
 CCFileTime, 227
 timeout
 CMResolverRegisteredMethodDescription, 242
 Timeout Operations, 55
 ccConfigGetTimeout, 55
 ccConfigSetTimeout, 55
 ccSetGetBackupListTimeout, 55
 toAnsi
 CMCodepage, 238
 toUnicode
 CMCodepage, 238
 toUpper
 CMCodepage, 238
 top
 SYPrintRect, 261
 totalBytes
 SYPrinterInfo, 257
 totalClusters
 CCVolumeInfo, 237
 totalJobs
 SYPrinterInfo, 257
 totalPages
 SYPrinterInfo, 257
 SYPrintJobInfo, 260
 totalUnitsHigh
 SYVolumeInformation, 265
 totalUnitsLow
 SYVolumeInformation, 264
 transmissionRetryTimeout
 SYPrinterInfo, 258
 ttOption
 SYDeviceMode, 252
 type
 AMCredentials, 221
 AMCredentialsA, 222
 CCNetDomainItem, 233
 CCNetServerItem, 234
 CCNetShareItem, 234
 CCNetShareItemA, 235
 UD_DNS_SERVERSTRINGSIZE
 udapi.h, 297
 UD_LOG_CLASS_CONNECTION
 Event Log Constants, 218
 UD_LOG_CLASS_FILE
 Event Log Constants, 217
 UD_LOG_CLASS_GEN
 Event Log Constants, 217
 UD_LOG_CLASS_SHARE
 Event Log Constants, 218
 UD_LOG_CLASS_USER
 Event Log Constants, 218
 UD_LOG_CONNECTION_CONNECT
 Event Log Constants, 220
 UD_LOG_CONNECTION_DISCONNECT
 Event Log Constants, 220
 UD_LOG_FILE_ATTRIBGET
 Event Log Constants, 219
 UD_LOG_FILE_ATTRIBSET
 Event Log Constants, 219
 UD_LOG_FILE_CLOSE
 Event Log Constants, 218
 UD_LOG_FILE_CREATE
 Event Log Constants, 218
 UD_LOG_FILE_DELETE
 Event Log Constants, 218
 UD_LOG_FILE_LOCK
 Event Log Constants, 219
 UD_LOG_FILE_OPEN
 Event Log Constants, 218
 UD_LOG_FILE_QUERYDIRECTORY
 Event Log Constants, 219
 UD_LOG_FILE_RENAME
 Event Log Constants, 218
 UD_LOG_FILE_SEEK
 Event Log Constants, 219
 UD_LOG_FILE_SIZESET
 Event Log Constants, 219
 UD_LOG_FILE_UNLOCK
 Event Log Constants, 219
 UD_LOG_FILE_VOLUMEINFO
 Event Log Constants, 219
 UD_LOG_GEN_NAMECONFLICT
 Event Log Constants, 218
 UD_LOG_GEN_NAMEREGFAIL
 Event Log Constants, 218
 UD_LOG_GEN_START
 Event Log Constants, 218
 UD_LOG_GEN_STOP

Event Log Constants, 218	Functions, 201
UD_LOG_MODULE_CC	udGetDnsParams
Event Log Constants, 217	Functions, 201
UD_LOG_MODULE_CS	udGetDomain
Event Log Constants, 217	Functions, 199
UD_LOG_SHARE_CONNECT	udGetDriverName
Event Log Constants, 219	Functions, 215
UD_LOG_SHARE_DISCONNECT	udGetFileSystemName
Event Log Constants, 219	Functions, 211
UD_LOG_USER_LOGOFF	udGetGlobalEncryption
Event Log Constants, 219	Functions, 214
UD_LOG_USER_LOGON	udGetHostName
Event Log Constants, 219	Functions, 215
UD_NQ_MAXPWDLEN	udGetInternalCapture
TCP and NetBIOS Layer Error Codes, 195	Functions, 214
UDFileAccessEvent, 265	udGetInternalTrace
access, 266	Functions, 215
before, 266	udGetLogFileBaseFolder
fileName, 265	Functions, 215
infoLevel, 266	udGetMaxDiskSize
newName, 265	Functions, 216
offsetHigh, 266	udGetMessageSigningPolicy
offsetLow, 266	Functions, 214
rid, 266	udGetNextMount
sizeHigh, 266	Functions, 214
sizeLow, 266	udGetNextShareEx
tid, 266	Functions, 214
UDShareAccessEvent, 266	udGetPassword
ipc, 267	Functions, 201
printQueue, 267	udGetPort
rid, 267	Functions, 204
shareName, 267	udGetScopeID
tid, 267	Functions, 199
UDUserAccessEvent, 267	udGetSecurityDescriptor
rid, 268	Functions, 213
udAllocateBuffer	udGetServerComment
Functions, 211	Functions, 203
udCifsServerClosed	udGetSmbError
Functions, 199	Functions, 202
udCifsServerStarted	udGetTaskPriorities
Functions, 198	Functions, 215
udCreateUser	udGetTransportPriority
Functions, 207	Functions, 202
udDeleteUserByRid	udGetUserCount
Functions, 207	Functions, 205
udEventLog	udGetUserInfo
Functions, 211	Functions, 205
udGetCaptureFileBaseFolder	udGetUserNameByRid
Functions, 215	Functions, 205
udGetCodePage	udGetUserRidByName
Functions, 204	Functions, 205
udGetComputerId	udGetWins
Functions, 204	Functions, 199
udGetComputerSecret	udInit
Functions, 209	Functions, 198
udGetComputerSecretByDomain	udLoadShareSecurityDescriptor
Functions, 209	Functions, 213
udGetCredentials	udNetBiosDaemonClosed

Functions, [199](#)
 udNetBiosDaemonStarted
 Functions, [199](#)
 udNetBiosDataIn
 Functions, [203](#)
 udNetBiosError
 Functions, [208](#)
 udNqToSystemError
 Functions, [202](#)
 udReleaseBuffer
 Functions, [211](#)
 udRemoveShare
 Functions, [208](#)
 udResetServerParams
 Functions, [198](#)
 udSaveShareInformation
 Functions, [207](#)
 udSaveShareSecurityDescriptor
 Functions, [213](#)
 udServerDataIn
 Functions, [203](#)
 udServerShareConnect
 Functions, [203](#)
 udServerShareDisconnect
 Functions, [203](#)
 udSetCodePage
 Functions, [204](#)
 udSetComputerSecret
 Functions, [210](#)
 udSetComputerSecretByDomain
 Functions, [210](#)
 udSetSecurityDescriptor
 Functions, [213](#)
 udSetUserAsAdministrator
 Functions, [206](#)
 udSetUserInfo
 Functions, [206](#)
 udStop
 Functions, [198](#)
 udapi.h, [280](#)
 CMIOBufBlob, [297](#)
 NQ_BOOL, [298](#)
 NQ_BYTE, [298](#)
 NQ_CHAR, [298](#)
 NQ_COUNT, [298](#)
 NQ_FLOAT, [298](#)
 NQ_HANDLE, [298](#)
 NQ_INDEX, [298](#)
 NQ_INT, [298](#)
 NQ_INT16, [298](#)
 NQ_INT32, [298](#)
 NQ_IOBuf, [297](#)
 NQ_IOBuffer, [297](#)
 NQ_IPADDRESS4, [298](#)
 NQ_IPADDRESS6, [298](#)
 NQ_PORT, [298](#)
 NQ_STATUS, [298](#)
 NQ_TIME, [298](#)
 NQ_UINT, [298](#)
 NQ_UINT16, [298](#)
 NQ_UINT32, [298](#)
 NQ_ULONG, [298](#)
 NQ_WCHAR, [298](#)
 UD_DNS_SERVERSTRINGSIZE, [297](#)
 untilTime
 SYPrinterInfo, [257](#)
 SYPrintJobInfo, [260](#)
 user
 AMCredentials, [221](#)
 AMCredentialsA, [222](#)
 SYPrintJobInfo, [261](#)
 User Credentials, [76](#)
 ccResetCredentails, [76](#)
 ccResetCredentailsA, [76](#)
 ccValidateUser, [77](#)
 ccValidateUserA, [77](#)
 ccValidateUserByCredentials, [78](#)
 ccValidateUserByCredentialsA, [77](#)
 User Defined, [143](#)
 User logon type, [23](#)
 CC_LOGON_TYPE_ANONYMOUS, [23](#)
 CC_LOGON_TYPE_GUEST, [23](#)
 CC_LOGON_TYPE_REGULAR, [23](#)
 userLogonType
 CCMountInfo, [231](#)
 CCMountInfoA, [232](#)
 userName
 CSFileData, [244](#)
 EVCSSOpenFileEntry, [245](#)
 EVCSSShareConnectionEntry, [245](#)
 FileDataA_t, [246](#)
 Utils, [123](#)
 nqEnableTraceLog, [123](#)

 v4
 NQ_IPADDRESS, [249](#)
 NQ_IPADDRESSVR, [250](#)

 v6
 NQ_IPADDRESS, [249](#)
 NQ_IPADDRESSVR, [250](#)

 version
 CMPProductInfo, [241](#)
 NQ_IPADDRESS, [249](#)
 NQ_IPADDRESSVR, [250](#)

 volumeld
 CCFileInfo, [225](#)

 volumeSerialNumber
 CCFileInfo_t, [226](#)

 vr
 NQ_IPADDRESS, [249](#)

 width
 SYPrintSize, [262](#)

 writeAmount
 CCSStatistics, [236](#)

 yResolution

SYDeviceMode, [252](#)
year
SYTimeFragments, [263](#)