

```

[Identification]
    OptionType = NetService
[Options]
    NWWKSTA
[FileConstants]
UtilityInf          = "UTILITY.INF"
subroutineinf       = "SUBROUTN.INF"
SoftwareType        = "service"
Exit_Code           = 0
NetwareEventDLL     = "%SystemRoot%\System32\nwevent.dll"
Manufacturer        = "Microsoft"
ProductMajorVersion = "4"
ProductMinorVersion = "0"
ProductVersion      = "$(ProductMajorVersion)". "$(ProductMinorVersion)"
ProductOpSupport    = 132
ProductNWWKSTAName = "NWCWorkstation"
ProductNWWKSTAImagePath = "%SystemRoot%\System32\services.exe"
NetRuleNWWKSTAType = "nwwsta nwcWorkstation"
NetRuleNWWKSTAClass = {"nwcWorkstation basic"}
NetRuleNWWKSTABindable = {"nwcWorkstation nwlinkipxTransport non non 100"}
NetRuleNWWKSTAUse = $(SoftwareType)" no no"
NetRuleNWWKSTABindForm = """"NwcWorkstation"" yes yes container"
ProductNWRDRName    = "NwRdr"
ProductNWRDRImagePath = "\SystemRoot\System32\drivers\nwrdr.sys"
ProductProviderImagePath = "%SystemRoot%\System32\nwprovau.dll"
ProviderName        = $(ProductNWWKSTAName)
ProductKeyName      = $(!NTN_SoftwareBase)"\"$(Manufacturer)"\"$(Product$(Option)Name)"\CurrentVersion"
ParamKeyName        = $(!NTN_ServiceBase)"\"$(Product$(Option)Name)"\Parameters"
LSAKeyName          = "System\CurrentControlSet\Control\LSA"
[GeneralConstants]
from                = ""
to                  = ""
ExitCodeOk         = 0
ExitCodeCancel     = 1
ExitCodeFatal      = 2
KeyNull            = ""
MAXIMUM_ALLOWED    = 33554432
RegistryErrorIndex = NO_ERROR
KeyProduct         = ""
KeyParameters      = ""
TRUE               = 1
FALSE              = 0
NoTitle            = 0
ExitState          = "Active"
OldVersionExisted  = $(FALSE)
DriverPath         = $(!STF_NTPATH)\drivers
[date]
    Now = {} ? $(!LIBHANDLE) GetSystemDate
[DetectDiskSpace]
    VolumeList      = {} ? $(!LIBHANDLE) GetHardDriveLetters
    VolumeFreeList  = {} ? $(!LIBHANDLE) GetHardDriveFreeSpace
    VolumeFSLList   = {} ? $(!LIBHANDLE) GetHardDriveFileSystems
[DetectSystemMemory]
    SystemMemory    = "" ? $(!LIBHANDLE) GetMemorySize
[Identify]
    read-syms Identification
    set Status      = STATUS_SUCCESSFUL
    set Identifier  = $(OptionType)

```

```

    set Media      = #("Source Media Descriptions", 1, 1)
    Return $(Status) $(Identifier) $(Media)
[ReturnOptions]
    set Status      = STATUS_FAILED
    set OptionList   = {}
    set OptionTextList = {}
    set LanguageList = ^(LanguagesSupported, 1)
    Ifcontains(i) $(($0)) in $(LanguageList)
        goto returnoptions
    else
        set Status = STATUS_NOLANGUAGE
        goto finish_ReturnOptions
    endif
returnoptions = +
    set OptionList   = ^(Options, 1)
    ifstr(i) $(!STF_PRODUCT) == "WINNT"
        set OptionTextList = ^(OptionsWINNTText$(($0)), 1)
    else
        set OptionTextList = ^(OptionsLANMANNTText$(($0)), 1)
    endif
    set Status      = STATUS_SUCCESSFUL
finish_ReturnOptions = +
    Return $(Status) $(OptionList) $(OptionTextList)
[InstallOption]
    set Option      = $(($1))
    set SrcDir      = $(($2))
    set AddCopy     = $(($3))
    set DoCopy      = $(($4))
    set DoConfig    = $(($5))
    set LanguageList = ^(LanguagesSupported, 1)
    Ifcontains(i) $(($0)) NOT-IN $(LanguageList)
        Return STATUS_NOLANGUAGE
    endif
    Debug-Output "OEMNSVNW.INF: STF_CWDDIR is: "$( !STF_CWDDIR)
    Debug-Output "OEMNSVNW.INF: STF_LANGUAGE is: "$( !STF_LANGUAGE)
    set-subst LF = "\n"
    set-subst CR = "\r"
    ifstr(i) $(!STF_PRODUCT) == "WINNT"
        read-syms Workstation$(($0))
    else
        read-syms Gateway$(($0))
    endif
    read-syms GeneralConstants
    read-syms FileConstants
    read-syms DialogConstants$( !STF_LANGUAGE)
    ifstr(i) $(!NTN_Origination) == "NCPA"
        set Continue = $(OK)
    endif
    read-syms FileConstants$( !STF_LANGUAGE)
    detect date
    detect DetectSystemMemory
    detect DetectDiskSpace
    set-title $(FunctionTitle)
    set to = Begin
    set from = Begin
    set CommonStatus = STATUS_SUCCESSFUL
    EndWait
Begin = +
    Ifstr(i) $(!NTN_InstallMode) == deinstall

```

```

        set OEM_ABANDON_OPTIONS = { $(ProductNWWKSTAName), +
            $(ProductNWRDRName)}
        set StartLabel = removeadapter
    else-Ifstr(i) $(!NTN_InstallMode) == Update
        set StartLabel = UpgradeSoftware
    else-Ifstr(i) $(!NTN_InstallMode) == bind
        set StartLabel = bindingadapter
    else-Ifstr(i) $(!NTN_InstallMode) == configure
        Shell $(UtilityInf),RegistryErrorString,CANNOT_CONFIGURE_SOFTWARE
        ifint $($ShellCode) != $(!SHELL_CODE_OK)
            Debug-Output "OEMNSVNW.INF: ShellCode error: cannot get an error
string."
                goto ShellCodeError
        endif
        set Error = $($R0)
        set from = end
        set to = end
        goto nonfatalinfo
    else
        set StartLabel = installadapter
        set OEM_ABANDON_OPTIONS = {}
    endif
    set DoWKSTA          = FALSE
    set DoRDR            = FALSE
    set DoNWLINK        = FALSE
    Ifstr(i) $(Option) == NWWKSTA
        set DoWKSTA      = TRUE
        set DoRDR        = TRUE
        set DoNWLINK     = TRUE
    Else-ifstr(i) $(Option) == RDR
        Set DoRDR = TRUE
    Else
        Debug-Output "OEMNSVWK.INF: Unrecognized option"
    Endif
    set from = $(fatal)
    set to = $(fatal)
    goto $(StartLabel)
installadapter = +
    GetDriveInPath NtDrive $(!STF_WINDOWSSYSPATH)
    Shell "" GetFilesSize
    set RequiredSize = $($R0)
    ForListDo $(VolumeList)
        set CurrentVolume = *$(VolumeList), $(#)
        set CurrentVolumeFree = *$(VolumeFreeList), $(#)
        Ifstr(i) $(CurrentVolume) == $(NtDrive)
            set-mul CurrentVolumeFree = $(CurrentVolumeFree) 1024
            Ifint $(CurrentVolumeFree) < $(RequiredSize)
                Shell $(subroutineinf) SetupMessage, $(!STF_LANGUAGE),
"NONFATAL", $(NotEnoughSpace)
                    ifint $($ShellCode) != $(!SHELL_CODE_OK)
                        goto ShellCodeError
                    endif
                goto end
            else
                goto AfterCheckSpace
            endif
        EndIf
    EndForListDo
    AfterCheckSpace = +

```

```

Ifstr(i) $(DoCopy) == "YES"
  Shell $(UtilityInf), DoAskSource, $(!STF_CWDDIR), $(SrcDir) YES
  Ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Goto ShellCodeError
  Else-Ifstr(i) $($R0) == STATUS_FAILED
    Shell $(UtilityInf) RegistryErrorString "ASK_SOURCE_FAIL"
    ifint $($ShellCode) != $(!SHELL_CODE_OK)
      goto ShellCodeError
    endif
    set Error = $($R0)
    Goto fatal
  Else-Ifstr(i) $($R0) == STATUS_USERCANCEL
    Goto successful
  Endif
  Set SrcDir = $($R1)
Endif
set NwlnkName = "NWLINKIPX"
OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)"\NWLINKIPX" $
(MAXIMUM_ALLOWED) BS_KeyServices
Ifstr $(BS_KeyServices) == $(KeyNull)
  set NwlnkName = "NWLINKIPX"
  OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)"\NWLINKIPX" $
(MAXIMUM_ALLOWED) BS_KeyServices
  Ifstr $(BS_KeyServices) == $(KeyNull)
    Shell "oemnxpip.inf" ReturnOptions $(!STF_LANGUAGE)
    ifstr(i) $($R0) == STATUS_SUCCESSFUL
      set OldStfSrcDir = $(!STF_SRC_DIR_OVERRIDE)
      set !STF_SRC_DIR_OVERRIDE = ""
      set NwlnkName = *($R1),1
      Shell "oemnxpip.inf" InstallOption $(!STF_LANGUAGE), *($R1),1,+
$(SrcDir), $(AddCopy), "NO", $(DoConfig)
      set !STF_SRC_DIR_OVERRIDE = $(OldStfSrcDir)
    else
      Shell $(subroutineinf) SetupMessage, $(!STF_LANGUAGE), "NONFATAL",
$(InstallNWLINKFirst)
      ifint $($ShellCode) != $(!SHELL_CODE_OK)
        goto ShellCodeError
      endif
      goto end
    endif
  else
    CloseRegKey $(BS_KeyServices)
  endif
else
  CloseRegKey $(BS_KeyServices)
endif
OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED) KeyProduct
Ifstr $(KeyProduct) != $(KeyNull)
  CloseRegKey $(KeyProduct)
  Shell $(UtilityInf), VerExistedDlg, $(Product$(Option)Title),+
$(ProductVersion)
  ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVNW.INF: ShellCode error: cannot get an error
string."
    goto ShellCodeError
  endif
  goto end
endif
CloseRegKey $(KeyProduct)

```

```

    OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)"\NetwareWorkstation" $
(MAXIMUM_ALLOWED) BS_KeyServices
    Ifstr $(BS_KeyServices) == $(KeyNull)
        Shell $(UtilityInf), CreateService, +
            "NetwareWorkstation", +
            $(ProductNwwkSTADisplayName), +
            $(ProductNwwkSTAImpagePath), "autoserviceshare", "NetworkProvider", {},
"", +
            $(NetwareEventDLL)
        ifint $($ShellCode) != $(!SHELL_CODE_OK)
            Debug-Output "OEMNSVNW.INF: ShellCode error"
            goto ShellCodeError
        endif
        debug-output $($R0)
        set RegistryErrorIndex = $($R0)
        ifstr(i) $(RegistryErrorIndex) != NO_ERROR
            Shell $(subroutineinf) SetupMessage, $(!STF_LANGUAGE), "NONFATAL", $
(RemoveAndReboot)
            ifint $($ShellCode) != $(!SHELL_CODE_OK)
                goto ShellCodeError
            endif
            goto end
        else
            Shell $(UtilityInf) RemoveService "NetwareWorkstation" "YES"
        endif
    else
        Shell $(subroutineinf) SetupMessage, $(!STF_LANGUAGE), "NONFATAL", $
(RemoveAndReboot)
        ifint $($ShellCode) != $(!SHELL_CODE_OK)
            goto ShellCodeError
        endif
        CloseRegKey $(BS_KeyServices)
        goto end
    endif
    goto nextstep
nextstep = +
    StartWait
    Ifstr(i) $(!NTN_InstallMode) == install
        Ifint $(OldVersionExisted) == $(FALSE)
            goto installproduct
        endif
    endif
    goto writeparameters
installproduct = +
    CloseRegKey $(ParamKeyName)
    ifstr(i) $(!NTN_InstallMode) == "install"
        install "Install-nwcfgdll"
        ifstr(i) $(STF_INSTALL_OUTCOME) != STF_SUCCESS
            Shell $(UtilityInf) RegistryErrorString "UNABLE_COPY_FILE"
            ifint $($ShellCode) != $(!SHELL_CODE_OK)
                goto ShellCodeError
            endif
            set Error = $($R0)
            goto fatal
        endif
    endif
    set OEM_ABANDON_ON = TRUE
    LoadLibrary "nw" $(!STF_CWDDIR)\nwcfg.dll !NWCFG_HANDLE
    Set FLibraryErrCtl = 1

```

```

ifstr(i) $(!NTN_InstallMode) == "install"
  install "Install-Option"
  ifstr(i) $(STF_INSTALL_OUTCOME) != STF_SUCCESS
    Shell $(UtilityInf) RegistryErrorString "UNABLE_COPY_FILE"
    ifint $($ShellCode) != $(!SHELL_CODE_OK)
      goto ShellCodeError
    endif
    set Error = $($R0)
    goto fatal
  endif
endif
LibraryProcedure Result $(!NWCFG_HANDLE), SetupRegistryForNWCS
LibraryProcedure Result $(!NWCFG_HANDLE), AddNetwarePrinterProvider
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\Print\
Providers\"$(ProductPrintProviderName) $(MAXIMUM_ALLOWED) PrintProviderKey
ifstr(i) $(PrintProviderKey) != $(KeyNull)
  SetRegValue $(PrintProviderKey) {"DisplayName",$(NoTitle),$(!REG_VT_SZ),$
(ProductPrintProviderName)}
  CloseRegKey $(PrintProviderKey)
endif
Set FLibraryErrCtl = 0
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\Print\
Providers" $(MAXIMUM_ALLOWED) PrintProviderKey
Ifstr $(PrintProviderKey) != $(KeyNull)
  GetRegValue $(PrintProviderKey), "Order", OrderInfo
  set OrderList = *$(OrderInfo), 4
  set NewOrderList = {$(ProductPrintProviderName)}
  ForListDo $(OrderList)
    ifstr(i) $($) != $(ProductPrintProviderName)
      set NewOrderList = >$(NewOrderList), $($)
    endif
  EndForListDo
  SetRegValue $(PrintProviderKey) {"Order",$(NoTitle),$(!REG_VT_MULTI_SZ),$
(NewOrderList)}
  CloseRegKey $(PrintProviderKey)
endif
OpenRegKey $(!REG_H_LOCAL) "" $(LSAKeyName) $(MAXIMUM_ALLOWED) LSAKey
Ifstr $(LSAKey) != $(KeyNull)
  GetRegValue $(LSAKey), "Authentication Packages", PackagesInfo
  set Packages = *$(PackagesInfo), 4
  ifcontains(i) "nwprovau" in $(Packages)
    debug-output "Authentication Packages already exist."
  else
    ifstr(i) $(Packages) == ""
      set Packages = {"nwprovau"}
    else
      set Packages = >$(Packages), "nwprovau"
    endif
    SetRegValue $(LSAKey) {"Authentication Packages",$(NoTitle),$(!
REG_VT_MULTI_SZ),$(Packages)}
  endif
  CloseRegKey $(LSAKey)
endif
ifstr(i) $(DOWKSTA) == TRUE
  Debug-Output "OEMNSVNW.INF: Install Workstation in registry."
  Set OEM_ABANDON_OPTIONS = >$(OEM_ABANDON_OPTIONS), $(ProductNWWKSTAName)
  Shell $(UtilityInf), AddSoftwareComponent, $(Manufacturer), +
    $(ProductNWWKSTAName), +
    $(ProductNWWKSTAName), +

```

```

$(ProductNWWKSTADisplayName), $(STF_CONTEXTINFNAME), +
$(ProductNWWKSTAImagePath), "autoserviceshare", "NetworkProvider", {},
"", +
$(NetwareEventDLL)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
  Debug-Output "OEMNSVNW.INF: ShellCode error"
  goto ShellCodeError
endif
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
  EndWait
  Debug-Output "OEMNSVNW.INF: Registry error: add software components"
  CloseRegKey $($R1)
  CloseRegKey $($R2)
  CloseRegKey $($R3)
  CloseRegKey $($R4)
  CloseRegKey $($R5)
  goto fatalregistry
endif
Set SoftProductKey      = $($R1)
Set SoftNetRuleKey     = $($R2)
Set SoftServiceKey     = $($R3)
Set SoftParameterKey  = $($R4)
Set SoftLinkageKey     = $($R5)
set NewValueList = {{SoftwareType,$(NoTitle),$(!REG_VT_SZ),$
(SoftwareType)},+
{MajorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMajorVersion)},+
{MinorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMinorVersion)},+
{Title,$(NoTitle),$(!REG_VT_SZ),$(ProductNWWKSTATitle)},
+
{Description,$(NoTitle),$(!REG_VT_SZ),$
(ProductNWWKSTADescription)},+
{ServiceName,$(NoTitle),$(!REG_VT_SZ),$
(ProductNWWKSTAName)},+
{OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupport)}, +
{InstallDate,$(NoTitle),$(!REG_VT_DWORD),*($Now,1)}}
Shell $(UtilityInf), AddValueList, $(SoftProductKey), $(NewValueList)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
  Debug-Output "OEMNSVNW.INF: ShellCode error."
  goto ShellCodeError
endif
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
  EndWait
  Debug-Output "OEMNSVNW.INF: registry error: add value list."
  CloseRegKey $(SoftProductKey)
  CloseRegKey $(SoftNetRuleKey)
  CloseRegKey $(SoftServiceKey)
  CloseRegKey $(SoftLinkageKey)
  CloseRegKey $(SoftParameterKey)
  goto fatalregistry
endif
set NewValueList = {{type,$(NoTitle),$(!REG_VT_SZ),$(NetRuleNWWKSTAType)},+
{class,$(NoTitle),$(!REG_VT_MULTI_SZ),$
(NetRuleNWWKSTAClass)}, +
{use,$(NoTitle),$(!REG_VT_SZ),$(NetRuleNWWKSTAUse)}, +

```

```

        {bindform,$(NoTitle),$(!REG_VT_SZ),$
(NetRuleNwwkStabBindForm)}, +
        {bindable,$(NoTitle),$(!REG_VT_MULTI_SZ),$
(NetRuleNwwkStabBindable)}, +
        {InfOption,$(NoTitle),$(!REG_VT_SZ),NwwkSta}}
Shell $(UtilityInf), AddValueList, $(SoftNetRuleKey), $(NewValueList)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVNW.INF: ShellCode error."
    goto ShellCodeError
endif
set RegistryErrorIndex = $($R0)
CloseRegKey $(SoftProductKey)
CloseRegKey $(SoftNetRuleKey)
CloseRegKey $(SoftServiceKey)
Shell "" AddParameterKey $(SoftParameterKey)
CloseRegKey $(SoftParameterKey)
CloseRegKey $(SoftLinkageKey)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNSVNW.INF: Registry error: add value list."
    goto fatalregistry
endif
Set FLibraryErrCtl = 1
set FoundAutoExec = TRUE
set AutoExecName = "autoexec.nt"
LibraryProcedure STATUS,$(!LIBHANDLE),CheckFileExistance $(!
STF_WINDOWSSYSPATH)"\"$(AutoExecName)
ifstr(i) $(STATUS) == NO
    set AutoExecName = "autoexec.tmp"
    LibraryProcedure STATUS,$(!LIBHANDLE),CheckFileExistance $(!
STF_WINDOWSSYSPATH)"\"$(AutoExecName)
ifstr(i) $(STATUS) == NO
    set FoundAutoExec = FALSE
endif
endif
ifstr(i) $(FoundAutoExec) == TRUE
    LibraryProcedure Result $(!NWCFG_HANDLE), RemoveSzFromFile, $(!
STF_WINDOWSSYSPATH)"\"$(AutoExecName), $(REM1)$($LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), RemoveSzFromFile, $(!
STF_WINDOWSSYSPATH)"\"$(AutoExecName), "lh %SystemRoot%\system32\nw16"$(LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), RemoveSzFromFile, $(!
STF_WINDOWSSYSPATH)"\"$(AutoExecName), "lh %SystemRoot%\system32\vwipxspx"$(LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), AppendSzToFile, $(!
STF_WINDOWSSYSPATH)"\"$(AutoExecName), $(REM1)$($CR)$($LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), AppendSzToFile, $(!
STF_WINDOWSSYSPATH)"\"$(AutoExecName), "lh %SystemRoot%\system32\nw16"$(CR)$($LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), AppendSzToFile, $(!
STF_WINDOWSSYSPATH)"\"$(AutoExecName), "lh %SystemRoot%\system32\vwipxspx"$(CR)$
(LF)
endif
Set FLibraryErrCtl = 0
endif
ifstr(i) $(DoNWLINK) == TRUE
    Shell "utility.inf", IncrementRefCount, "Software\Microsoft\"$(NwlnkName)"\
CurrentVersion"
endif
ifstr(i) $(DoRDR) == TRUE
    Debug-Output "OEMNSVNW.INF: Install Rdr registry"
    Set OEM_ABANDON_OPTIONS = >$(OEM_ABANDON_OPTIONS), $(ProductNWRDRName)

```



```

Shell $(UtilityInf), CreateService, $(ProductNWRDRName), +
$(ProductNWRDRDisplayName), $(ProductNWRDRImagePath), +
"system", "Network", {}, "", +
$(NetwareEventDLL)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
  Debug-Output "OEMNSVNW.INF: ShellCode error"
  goto ShellCodeError
endif
set RegistryErrorIndex = $($R0)
set NWRdrServiceKey = $($R1)
CloseRegKey $($R2)
CloseRegKey $($R3)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
  EndWait
  Debug-Output "OEMNSVNW.INF: Registry error: add software components"
  goto fatalregistry
endif
CreateRegKey $(NWRdrServiceKey) {"Performance",$(NoTitle),GenericClass} ""
+
$(MAXIMUM_ALLOWED) "" KeyPerformance
set NewValueList = {{Library,$(NoTitle),$(!REG_VT_SZ),"Perfnw.dll"},+
{Open,$(NoTitle),$(!
REG_VT_SZ),"OpenNetWarePerformanceData"},+
{Collect,$(NoTitle),$(!
REG_VT_SZ),"CollectNetWarePerformanceData"},+
{Close,$(NoTitle),$(!
REG_VT_SZ),"CloseNetWarePerformanceData"}}
Shell $(UtilityInf), AddValueList, $(KeyPerformance), $(NewValueList)
set RegistryErrorIndex = $($R0)
CloseRegKey $(KeyPerformance)
CloseRegKey $(NWRdrServiceKey)
LibraryProcedure Result $(!NWCFG_HANDLE), GetKernelVersion
ifint *($Result),3) < 529
  Set FLibraryErrCtl = 1
  LibraryProcedure STATUS $(!NWCFG_HANDLE), lodctr $(!
STF_WINDOWSSYSPATH)\nwperf.ini
  Set FLibraryErrCtl = 0
endif
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
  EndWait
  Debug-Output "OEMNSVNW.INF: Registry error: add performance
information"
  goto fatalregistry
endif
Debug-Output "OEMNSVNW.INF: Add NetworkProvider"
Shell $(UtilityInf), AddNetworkProvider, $(ProviderName), +
$(ProductProviderImagePath), $(ProductProviderDisplayName), "nwrdr"
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
  EndWait
  Debug-Output "OEMNSVNW.INF: Registry error: add software components"
  goto fatalregistry
endif
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\
NetworkProvider\Order" $(MAXIMUM_ALLOWED) NetProviderKey
Ifstr $(NetProviderKey) != $(KeyNull)
  GetRegValue $(NetProviderKey),"ProviderOrder",OrderInfo
  set OrderList = *($OrderInfo),4)
  Split-String $(OrderList), ",", ProviderList

```

```

set NewOrderList = $(ProductNwwkSTAName)
ForListDo $(ProviderList)
    ifstr(i) $($) != ","
        ifstr(i) $($) != $(ProductNwwkSTAName)
            set NewOrderList = $(NewOrderList)"", $($)
        endif
    endif
EndForListDo
SetRegValue $(NetProviderKey) {"ProviderOrder",$(NoTitle),$(!
REG_VT_SZ),$(NewOrderList)}
CloseRegKey $(NetProviderKey)
endif
Shell "" AddShortName
Shell $(UtilityInf) AddServerRpcProtocol "ncadg_ipx" "RpcLtScm.Dll"
Shell $(UtilityInf) AddServerRpcProtocol "ncacn_spx" "RpcLtScm.Dll"
Ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNXPTC.INF: ShellCode error adding RPC procotol"
    goto ShellCodeError
Endif
Set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != "NO_ERROR"
    Debug-Output "OEMNXPTC.INF: ERROR adding RPC protocol data"
    EndWait
    goto fatalregistry
Endif
OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)"\"$(ProviderName)"\
networkprovider" $(MAXIMUM_ALLOWED) ProviderKey
ifstr(i) $(ProviderKey) != ""
    set NewValueList = {{Class,$(NoTitle),$(!REG_VT_DWORD),11},+
                        {NWCompatibleAuthentication,$(NoTitle),$(!
REG_VT_DWORD),1}}
    Shell $(UtilityInf) AddValueList $(ProviderKey) $(NewValueList)
endif
CloseRegKey $(ProviderKey)
LibraryProcedure Status1, $(!LIBHANDLE), CheckFileExistance, $(!
STF_WINDOWSSYSPATH)"\novell.cpl"
ifstr(i) $(Status1) == "YES"
    LibraryProcedure Status2, $(!LIBHANDLE), RenFile, $(!
STF_WINDOWSSYSPATH)"\novell.cpl", $(!STF_WINDOWSSYSPATH)"\novell.bak"
endif
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\Wow" $
(MAXIMUM_ALLOWED) WowKey
ifstr(i) $(WowKey) != ""
    GetRegValue $(WowKey),"KnownDLLs", KnownDLLsInfo
    Split-String *($KnownDLLsInfo),4, " ", KnownDLLsList
    Ifcontains(i) "netware.driv" not-in $(KnownDLLsList)
        ifstr(i) *($KnownDLLsInfo),4 == ""
            set KnownDLLs = "netware.driv"
        else
            set KnownDLLs = *($KnownDLLsInfo),4" netware.driv"
        endif
    SetRegValue $(WowKey) {"KnownDLLs",$(NoTitle),$(!REG_VT_SZ),$
(KnownDLLs)}
endif
CloseRegKey $(WowKey)
endif
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Services\
LanmanServer\Parameters" $(MAXIMUM_ALLOWED) ServerParmKey
ifstr(i) $(ServerParmKey) != ""

```

```

        SetRegValue $(ServerParmKey) {"EnableSharedNetDrives",$(NoTitle),$(!
REG_VT_DWORD),1}
        CloseRegKey $(ServerParmKey)
    endif
endif
writeparameters = +
    FreeLibrary $(!NWCFG_HANDLE)
    EndWait
    goto successful
ConfigureBrowser = +
    goto successful
bindingadapter =+
    set Error = "Binding: Sorry, not yet implemented."
    goto fatal
removeadapter = +
    OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\Wow" $
(MAXIMUM_ALLOWED) WowKey
    ifstr(i) $(WowKey) != ""
        GetRegValue $(WowKey),"KnownDLLs", KnownDLLsInfo
        Split-String *($(KnownDLLsInfo),4), " ", KnownDLLsList
        set KnownDLLs = ""
        Ifcontains(i) "netware.driv" in $(KnownDLLsList)
            ForListDo $(KnownDLLsList)
                ifstr(i) $($) != " "
                    ifstr(i) $($) != "netware.driv"
                        ifstr(i) $(KnownDLLs) == ""
                            set KnownDLLs = $($)
                        else
                            set KnownDLLs = $(KnownDLLs)" "$($)
                        endif
                    endif
                endif
            EndForListDo
        else
            set KnownDLLs = *($(KnownDLLsInfo),4)
        endif
        SetRegValue $(WowKey) {"KnownDLLs",$(NoTitle),$(!REG_VT_SZ),$(KnownDLLs)}
        CloseRegKey $(WowKey)
    endif
    OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Services\LanmanServer\
Parameters" $(MAXIMUM_ALLOWED) ServerParmKey
    ifstr(i) $(ServerParmKey) != ""
        SetRegValue $(ServerParmKey) {"EnableSharedNetDrives",$(NoTitle),$(!
REG_VT_DWORD),0}
        CloseRegKey $(ServerParmKey)
    endif
    LoadLibrary "nw" $(!STF_CWDDIR)\nwcfg.dll !NWCFG_HANDLE
    Set FLibraryErrCtl = 1
    OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\Print\
Providers" $(MAXIMUM_ALLOWED) PrintProviderKey
    ifstr(i) $(PrintProviderKey) != ""
        GetRegValue $(PrintProviderKey) "Order" OrderInfo
        ifcontains(i) $(OldPrintProviderName) in *($(OrderInfo),4)
            LibraryProcedure Result $(!NWCFG_HANDLE), DeleteNetwarePrinterProvider
$(OldPrintProviderName)
        endif
        ifcontains(i) $(OldPrintProviderName2) in *($(OrderInfo),4)
            LibraryProcedure Result $(!NWCFG_HANDLE), DeleteNetwarePrinterProvider
$(OldPrintProviderName2)

```

```

        endif
    endif
    OpenRegKey $(!REG_H_LOCAL) "" "Software\Microsoft\NwSapAgent" $
(MAXIMUM_ALLOWED) SapAgentKey
    ifstr(i) $(SapAgentKey) == ""
    else
        CloseRegKey $(SapAgentKey)
    endif
    LibraryProcedure Result $(!NWCFG_HANDLE), DeleteNetwarePrinterProvider
    Set FLibraryErrCtl = 0
    LibraryProcedure Result $(!NWCFG_HANDLE), GetKernelVersion
    ifint *$(Result),3 < 529
        Set FLibraryErrCtl = 1
        LibraryProcedure STATUS $(!NWCFG_HANDLE), unloadctr nwrdr
        Set FLibraryErrCtl = 0
    endif
    OpenRegKey $(!REG_H_LOCAL) "" $(LSAKeyName) $(MAXIMUM_ALLOWED) LSAKey
    Ifstr $(LSAKey) != $(KeyNull)
        GetRegValue $(LSAKey),"Authentication Packages", PackagesInfo
        set Packages = *$(PackagesInfo), 4
        ifcontains(i) "nwprovau" not-in $(Packages)
            debug-output "Authentication Packages does not exist.."
        else
            set NewPackages = {}
            ForListDo $(Packages)
                ifstr(i) $($) != "nwprovau"
                    ifstr(i) $(NewPackages) == {}
                        set NewPackages = {$(}$)}
                    else
                        set NewPackages = >$(NewPackages),$(}$)
                    endif
                endif
            EndForListDo
            SetRegValue $(LSAKey) {"Authentication Packages",$(NoTitle),$(!
REG_VT_MULTI_SZ),$(NewPackages)}
        endif
        CloseRegKey $(LSAKey)
    endif
    ifcontains(i) $(ProductNWWKSTAName) in $(OEM_ABANDON_OPTIONS)
        Shell $(UtilityInf), RemoveSoftwareComponent, $(Manufacturer), +
$(ProductNWWKSTAName)
        ifint $$ShellCode != $(!SHELL_CODE_OK)
            Debug-Output "OEMNSVNW.INF: ShellCode error"
            goto ShellCodeError
        endif
        set RegistryErrorIndex = $($R0)
        Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
            goto fatalregistry
        endif
    endif
    ForListDo {$(ProductNWRDRName)}
        ifcontains(i) $($) in $(OEM_ABANDON_OPTIONS)
            Shell $(UtilityInf), RemoveService, $($), "YES"
            ifint $$ShellCode != $(!SHELL_CODE_OK)
                Debug-Output "OEMNSVNW.INF: ShellCode error"
                goto ShellCodeError
            endif
            set RegistryErrorIndex = $($R0)
            Ifstr(i) $(RegistryErrorIndex) != NO_ERROR

```

```

        goto fatalregistry
    endif
endif
EndForListDo
set NwlnkName = "NWLINKIPX"
OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)\NWLINKIPX" $(
(MAXIMUM_ALLOWED) BS_KeyServices
Ifstr $(BS_KeyServices) == $(KeyNull)
    set NwlnkName = "NWLNKIPX"
endif
Shell "UTILITY.INF", DecrementRefCount, "Software\Microsoft\"$(NwlnkName)\
CurrentVersion"
Set FLibraryErrCtl = 1
set FoundAutoExec = TRUE
set AutoExecName = "autoexec.nt"
LibraryProcedure STATUS,$(!LIBHANDLE),CheckFileExistance $(!
STF_WINDOWSSYSPATH)\\"$(AutoExecName)
ifstr(i) $(STATUS) == NO
    set AutoExecName = "autoexec.tmp"
    LibraryProcedure STATUS,$(!LIBHANDLE),CheckFileExistance $(!
STF_WINDOWSSYSPATH)\\"$(AutoExecName)
ifstr(i) $(STATUS) == NO
    set FoundAutoExec = FALSE
endif
endif
ifstr(i) $(FoundAutoExec) == TRUE
    LibraryProcedure Result $(!NWCFG_HANDLE), RemoveSzFromFile, $(!
STF_WINDOWSSYSPATH)\\"$(AutoExecName), $(REM1)$\LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), RemoveSzFromFile, $(!
STF_WINDOWSSYSPATH)\\"$(AutoExecName), "lh %SystemRoot%\system32\nw16"$\LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), RemoveSzFromFile, $(!
STF_WINDOWSSYSPATH)\\"$(AutoExecName), "lh %SystemRoot%\system32\vwipxspx"$\LF)
endif
Set FLibraryErrCtl = 0
Shell $(UtilityInf) RemoveNetworkProvider $(ProviderName)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVNW.INF: ShellCode error"
    goto ShellCodeError
endif
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    goto fatalregistry
endif
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\
NetworkProvider\ShortName" $(MAXIMUM_ALLOWED) ShortNameKey
ifstr $(ShortNameKey) != $(KeyNull)
    DeleteRegValue $(ShortNameKey) "NW"
else
    debug-output "something wrong..."
endif
Set FLibraryErrCtl = 1
LibraryProcedure Result $(!NWCFG_HANDLE), CleanupRegistryForNWCS
Set FLibraryErrCtl = 0
FreeLibrary $(!NWCFG_HANDLE)
Install "RestoreFiles"
Set FLibraryErrCtl = 1
AddfileToDeleteList $(!STF_WINDOWSSYSPATH)\nwc.cpl"
AddfileToDeleteList $(!STF_WINDOWSSYSPATH)\nwapi32.dll"
LibraryProcedure Status1, $(!LIBHANDLE), DelFile, $(!STF_WINDOWSSYSPATH)\

```

```

nwc.cpl"
LibraryProcedure Status1, $(!LIBHANDLE), DelFile, $(!STF_WINDOWSSYSPATH)"\
nwapi32.dll"
Set FLibraryErrCtl = 0
Shell "" RemoveBackupFiles
goto end
UpgradeSoftware = +
LibraryProcedure STATUS,$(!LIBHANDLE),CheckFileExistance $(SrcDir)"\nwc.2a"
ifstr(i) $(STATUS) == YES
Shell $(subroutineinf) SetupMessage, $(!STF_LANGUAGE), "NONFATAL", $
(UpgradeBetaOnTopOfDaytona)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
goto ShellCodeError
endif
goto end
endif
LibraryProcedure STATUS,$(!LIBHANDLE),CheckFileExistance $(SrcDir)"\nwc.2b"
ifstr(i) $(STATUS) == YES
Shell $(subroutineinf) SetupMessage, $(!STF_LANGUAGE), "NONFATAL", $
(UpgradeBetaOnTopOfDaytona)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
goto ShellCodeError
endif
goto end
endif
Install "Install-Update"
ifstr(i) $(STF_INSTALL_OUTCOME) != STF_SUCCESS
goto fatal
endif
LoadLibrary "nw" $(!STF_CWDDIR)\nwcfg.dll !NWC_CFG_HANDLE
OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED) KeyProduct
Ifstr $(KeyProduct) != $(KeyNull)
ifstr(i) $(!STF_NTUPGRADE) != "YES"
ifstr(i) $(STF_CONTEXTINFNAME) != "oemnswnw.inf"
OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName)"\NetRules" $
(MAXIMUM_ALLOWED) KeyNetRules
Ifstr $(KeyNetRules) != $(KeyNull)
GetRegValue $(KeyNetRules),"InfName", InfNameInfo
set !UG_Filename = *$(InfNameInfo), 4)
CloseRegKey $(KeyNetRules)
else
Split-String $(!NTN_Infname), "\", FilenameList
QueryListSize ListSize $(FilenameList)
set !UG_Filename = *$(FilenameList), $(ListSize))
endif
install "Install-Update-Inf"
endif
endif
ifstr(i) $(STF_INSTALL_OUTCOME) != STF_SUCCESS
goto fatal
endif
SetRegValue $(KeyProduct) {MajorVersion,$(NoTitle),$(!REG_VT_SZ),$
(ProductMajorVersion)}
SetRegValue $(KeyProduct) {MinorVersion,$(NoTitle),$(!REG_VT_SZ),$
(ProductMinorVersion)}
SetRegValue $(KeyProduct) {Description,$(NoTitle),$(!REG_VT_SZ),$
(ProductNWWKSTADescription)}
SetRegValue $(KeyProduct) {OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupport)}

```

```

CloseRegKey $(KeyProduct)
OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)"\Nwrdr\Performance" $
(MAXIMUM_ALLOWED) KeyPerformance
Ifstr $(KeyPerformance) != $(KeyNull)
    CloseRegKey $(KeyPerformance)
    Set FLibraryErrCtl = 1
    ifstr(i) $(!STF_PRODUCT) == "WINNT"
        LibraryProcedure Result $(!NWCFG_HANDLE), CleanupRegistryForNWCS
    endif
    Set FLibraryErrCtl = 0
    LibraryProcedure Result $(!NWCFG_HANDLE), GetKernelVersion
    ifint *$(Result),3 < 529
        Set FLibraryErrCtl = 1
        LibraryProcedure STATUS $(!NWCFG_HANDLE), unlodctr nwrdr
        LibraryProcedure STATUS $(!NWCFG_HANDLE), lodctr $(!
STF_WINDOWSSYSPATH)\nwperf.ini
        Set FLibraryErrCtl = 0
    endif
else
    Set FLibraryErrCtl = 1
    LibraryProcedure Result $(!NWCFG_HANDLE), RemoveSzFromFile, $(!
STF_WINDOWSSYSPATH)\autoexec.nt", $(REM1)$$(LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), RemoveSzFromFile, $(!
STF_WINDOWSSYSPATH)\autoexec.nt", "lh %SystemRoot%\system32\nw16"$(LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), RemoveSzFromFile, $(!
STF_WINDOWSSYSPATH)\autoexec.nt", "lh %SystemRoot%\system32\vwpixsp"$(LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), AppendSzToFile, $(!
STF_WINDOWSSYSPATH)\autoexec.nt", $(REM1)$$(CR)$$(LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), AppendSzToFile, $(!
STF_WINDOWSSYSPATH)\autoexec.nt", "lh %SystemRoot%\system32\nw16"$(CR)$$(LF)
    LibraryProcedure Result $(!NWCFG_HANDLE), AppendSzToFile, $(!
STF_WINDOWSSYSPATH)\autoexec.nt", "lh %SystemRoot%\system32\vwpixsp"$(CR)$$(LF)
    Set FLibraryErrCtl = 0
    OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)"\nwrdr" $
(MAXIMUM_ALLOWED) KeyNwrdr
    CreateRegKey $(KeyNwrdr) {"Performance",$(NoTitle),GenericClass} "" +
$(MAXIMUM_ALLOWED) "" KeyPerformance
    set NewValueList = {{Library,$(NoTitle),$(!REG_VT_SZ),"Perfnw.dll"},+
{Open,$(NoTitle),$(!
REG_VT_SZ),"OpenNetWarePerformanceData"},+
{Collect,$(NoTitle),$(!
REG_VT_SZ),"CollectNetWarePerformanceData"},+
{Close,$(NoTitle),$(!
REG_VT_SZ),"CloseNetWarePerformanceData"}}
    Shell $(UtilityInf), AddValueList, $(KeyPerformance), $(NewValueList)
    set RegistryErrorIndex = $(R0)
    CloseRegKey $(KeyPerformance)
    CloseRegKey $(KeyNwrdr)
    LibraryProcedure Result $(!NWCFG_HANDLE), GetKernelVersion
    ifint *$(Result),3 < 529
        Set FLibraryErrCtl = 1
        LibraryProcedure STATUS $(!NWCFG_HANDLE), lodctr $(!
STF_WINDOWSSYSPATH)\nwperf.ini
        Set FLibraryErrCtl = 0
    endif
    Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
        EndWait
        Debug-Output "OEMNSVNW.INF: Registry error: add performance
information"

```

```

        goto fatalregistry
    endif
endif
endif
LibraryProcedure Result $(!NWCFG_HANDLE), SetupRegistryForNWC
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Services\
NWCWorkstation\networkprovider" $(MAXIMUM_ALLOWED) ProviderKey
ifstr(i) $(ProviderKey) != ""
    SetRegValue $(ProviderKey) {Devicename,$(NoTitle),$(!REG_VT_SZ),"\Device\
nwrdr"}
    SetRegValue $(ProviderKey) {Name,$(NoTitle),$(!REG_VT_SZ),$
(ProductProviderDisplayName)}
    CloseRegKey $(ProviderKey)
endif
LibraryProcedure Status1, $(!LIBHANDLE), CheckFileExistance, $(!
STF_WINDOWSSYSPATH)"\novell.cpl"
ifstr(i) $(Status1) == "YES"
    LibraryProcedure Status2, $(!LIBHANDLE), RenFile, $(!STF_WINDOWSSYSPATH)"\
novell.cpl", $(!STF_WINDOWSSYSPATH)"\novell.bak"
endif
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\WOW" $
(MAXIMUM_ALLOWED) WowKey
ifstr(i) $(WowKey) != ""
    GetRegValue $(WowKey),"KnownDLLs", KnownDLLsInfo
    Split-String *($KnownDLLsInfo),4, " ", KnownDLLsList
    Ifcontains(i) "netware.driv" not-in $(KnownDLLsList)
        ifstr(i) *($KnownDLLsInfo),4 == ""
            set KnownDLLs = "netware.driv"
        else
            set KnownDLLs = *($KnownDLLsInfo),4" netware.driv"
        endif
    SetRegValue $(WowKey) {"KnownDLLs",$(NoTitle),$(!REG_VT_SZ),$
(KnownDLLs)}
endif
CloseRegKey $(WowKey)
endif
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Services\LanmanServer\
Parameters" $(MAXIMUM_ALLOWED) ServerParmKey
ifstr(i) $(ServerParmKey) != ""
    SetRegValue $(ServerParmKey) {"EnableSharedNetDrives",$(NoTitle),$(!
REG_VT_DWORD),1}
    CloseRegKey $(ServerParmKey)
endif
OpenRegKey $(!REG_H_LOCAL) "" "Software\Microsoft\NWCWorkstation\
CurrentVersion" $(MAXIMUM_ALLOWED) CurrentVersionKey
ifstr(i) $(CurrentVersionKey) != ""
    SetRegValue $(CurrentVersionKey) {"Title",$(NoTitle),$(!REG_VT_SZ),$
(NWType)}
    CloseRegKey $(CurrentVersionKey)
endif
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Services\
NWCWorkstation" $(MAXIMUM_ALLOWED) nwcKey
ifstr(i) $(nwcKey) != ""
    SetRegValue $(nwcKey) {"DisplayName",$(NoTitle),$(!REG_VT_SZ),$(NWType)}
    SetRegValue $(nwcKey) {"ImagePath",$(NoTitle),$(!REG_VT_EXPAND_SZ),$
(ProductNWWKSTAImpath)}
    CloseRegKey $(nwcKey)
endif
Set FLibraryErrCtl = 1

```



```

OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\Print\
Providers" $(MAXIMUM_ALLOWED) PrintProviderKey
ifstr(i) $(PrintProviderKey) != ""
    GetRegValue $(PrintProviderKey) "Order" OrderInfo
    ifcontains(i) $(OldPrintProviderName) in *($OrderInfo),4)
        set oldname = $(OldPrintProviderName)
        goto changePrintProvider
    endif
    ifcontains(i) $(OldPrintProviderName2) in *($OrderInfo),4)
        set oldname = $(OldPrintProviderName2)
        goto changePrintProvider
    endif
    goto nochangePrintProvider
changePrintProvider = +
DeleteRegKey $(PrintProviderKey) $(oldname)
OpenRegKey $(PrintProviderKey) "" $(ProductPrintProviderName) $
(MAXIMUM_ALLOWED) ProductProviderKey
ifstr $(ProductProviderKey) == $(KeyNull)
    CreateRegKey $(PrintProviderKey) {$(ProductPrintProviderName),$
(NoTitle),GenericClass} "" $(MAXIMUM_ALLOWED) "" ProductProviderKey
endif
ifstr $(ProductProviderKey) != $(KeyNull)
    SetRegValue $(ProductProviderKey) {"Name",$(NoTitle),$(!
REG_VT_SZ),"nwprovau.dll"}
    SetRegValue $(ProductProviderKey) {"DisplayName",$(NoTitle),$(!
REG_VT_SZ),$(ProductPrintProviderName)}
endif
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\Print\
Providers" $(MAXIMUM_ALLOWED) PrintProviderKey
Ifstr $(PrintProviderKey) != $(KeyNull)
    GetRegValue $(PrintProviderKey),"Order",OrderInfo
    set OrderList = *($OrderInfo),4)
    set NewOrderList = {$(ProductPrintProviderName)}
    ForListDo $(OrderList)
        ifstr(i) $($) != $(oldname)
            ifstr(i) $($) != $(ProductPrintProviderName)
                set NewOrderList = >($NewOrderList),$(%)
            endif
        endif
    EndForListDo
    SetRegValue $(PrintProviderKey) {"Order",$(NoTitle),$(!
REG_VT_MULTI_SZ),$(NewOrderList)}
    CloseRegKey $(PrintProviderKey)
endif
nochangePrintProvider = +
endif
Set FLibraryErrCtl = 0
OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)"\"$(ProviderName)"\
networkprovider" $(MAXIMUM_ALLOWED) ProviderKey
ifstr(i) $(ProviderKey) != ""
    set NewValueList = {{Class,$(NoTitle),$(!REG_VT_DWORD),11},+
{NWCompatibleAuthentication,$(NoTitle),$(!
REG_VT_DWORD),1}}
    Shell $(UtilityInf) AddValueList $(ProviderKey) $(NewValueList)
    CloseRegKey $(ProviderKey)
endif
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\Print\
Providers\"$(ProductPrintProviderName) $(MAXIMUM_ALLOWED) PrintProviderKey
ifstr(i) $(PrintProviderKey) != $(KeyNull)

```

```

        SetRegValue $(PrintProviderKey) {"DisplayName",$(NoTitle),$(!REG_VT_SZ),$
(ProductPrintProviderName)}
        CloseRegKey $(PrintProviderKey)
    endif
    Shell "" AddShortName
    Shell $(UtilityInf) AddServerRpcProtocol "ncadg_ipx" "RpcLtScm.Dll"
    Shell $(UtilityInf) AddServerRpcProtocol "ncacn_spx" "RpcLtScm.Dll"
    FreeLibrary $(!NWCFG_HANDLE)
    goto end
successful = +
    goto end
warning = +
    Shell $(subroutineinf) SetupMessage, $(!STF_LANGUAGE), "WARNING", $(Error)
    ifint $($ShellCode) != $(!SHELL_CODE_OK)
        goto ShellCodeError
    endif
    ifstr(i) $($R1) == "OK"
        goto $(to)
    else-ifstr(i) $($R1) == "CANCEL"
        goto $(from)
    else
        goto end
    endif
nonfatalinfo = +
    Set CommonStatus = STATUS_USERCANCEL
    Set Severity = STATUS
    goto nonfatalmsg
nonfatal = +
    Set Severity = NONFATAL
    goto nonfatalmsg
nonfatalmsg = +
    ifstr(i) $(Error) == ""
        Set Severity = NONFATAL
        Shell $(UtilityInf) RegistryErrorString "SETUP_FAIL"
        ifint $($ShellCode) != $(!SHELL_CODE_OK)
            goto ShellCodeError
        endif
        set Error = $($R0)
    endif
    Shell $(subroutineinf) SetupMessage, $(!STF_LANGUAGE), $(Severity), $(Error)
    ifint $($ShellCode) != $(!SHELL_CODE_OK)
        goto ShellCodeError
    endif
    ifstr(i) $($R1) == "OK"
        goto $(from)
    else
        goto "end"
    endif
fatalregistry = +
    Shell $(UtilityInf) RegistryErrorString $(RegistryErrorIndex)
    ifint $($ShellCode) != $(!SHELL_CODE_OK)
        goto ShellCodeError
    endif
    set Error = $($R0)
    goto fatal
fatal = +
    ifstr(i) $(Error) == ""
        Shell $(UtilityInf) RegistryErrorString "SETUP_FAIL"
        ifint $($ShellCode) != $(!SHELL_CODE_OK)

```

```

        goto ShellCodeError
    endif
    set Error = $($R0)
endif
Shell $(subroutineinf) SetupMessage, $(!STF_LANGUAGE), "FATAL", $(Error)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    goto ShellCodeError
endif
goto setfailed
ShellCodeError = +
    set DlgType      = "MessageBox"
    set STF_MB_TITLE = "Error: "$(FunctionTitle)
    set STF_MB_TEXT  = $(ShellCodeErrorText)
    set STF_MB_TYPE  = 1
    set STF_MB_ICON  = 3
    set STF_MB_DEF   = 1
    ui start "Error Message"
    goto setfailed
setfailed = +
    set CommonStatus = STATUS_FAILED
    ifstr(i) $(OEM_ABANDON_ON) == TRUE
        set OEM_ABANDON_ON = FALSE
        goto removeadapter
    endif
    goto end
end = +
    goto term
term = +
    Return $(CommonStatus)
[RestoreFiles]
    AddSectionFilesToCopyList Restore-Oldexe $(!STF_WINDOWSSYSPATH) $(!
STF_WINDOWSSYSPATH)
    AddSectionFilesToCopyList Restore-Oldsys $(!STF_WINDOWSSYSPATH)\drivers $(!
STF_WINDOWSSYSPATH)\drivers
    set !STF_NCPA_FLUSH_COPYLIST = TRUE
    CopyFilesInCopyList
    Exit
[Install-nwcfgdll]
    set STF_VITAL = ""
    ifstr(i) $(AddCopy) == "YES"
        AddSectionFilesToCopyList Files-nwcfgdll $(SrcDir) $(!STF_WINDOWSSYSPATH)
    endif
    ifstr(i) $(DoCopy) == "YES"
        set !STF_NCPA_FLUSH_COPYLIST = TRUE
        CopyFilesInCopyList
    endif
    Exit
[Install-Option]
    set STF_VITAL = ""
    ifstr(i) $(AddCopy) == "YES"
        AddSectionFilesToCopyList Files-$(Option) $(SrcDir) $(!STF_WINDOWSSYSPATH)
        AddSectionFilesToCopyList Files-Drivers $(SrcDir) $(!STF_WINDOWSSYSPATH)\
drivers
    endif
    ifstr(i) $(DoCopy) == "YES"
        set !STF_NCPA_FLUSH_COPYLIST = TRUE
        CopyFilesInCopyList
    endif
    ifstr(i) $(DoConfig) == "YES"

```

```

endif
Exit
[Install-Update]
set STF_VITAL = ""
AddSectionFilesToCopyList Files-$(Option) $(SrcDir) $(!STF_WINDOWSSYSPATH)
AddSectionFilesToCopyList Files-nwcfgdll $(SrcDir) $(!STF_WINDOWSSYSPATH)
AddSectionFilesToCopyList Files-Drivers $(SrcDir) $(!STF_WINDOWSSYSPATH)\drivers
ifstr(i) $(DoCopy) == "YES"
    set !STF_NCPA_FLUSH_COPYLIST = TRUE
    CopyFilesInCopyList
endif
Exit
[AddShortName]
read-syms GeneralConstants
read-syms FileConstants
read-syms FileConstants$(!STF_LANGUAGE)
OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Control\
NetworkProvider" $(MAXIMUM_ALLOWED) NetProviderKey
ifstr $(NetProviderKey) != $(KeyNull)
    OpenRegKey $(NetProviderKey) "" "ShortName" $(MAXIMUM_ALLOWED) ShortNameKey
    ifstr $(ShortNameKey) == $(KeyNull)
        CreateRegKey $(NetProviderKey) {"ShortName",$(NoTitle),GenericClass} ""
$(MAXIMUM_ALLOWED) "" ShortNameKey
    endif
    ifstr $(ShortNameKey) != $(KeyNull)
        SetRegValue $(ShortNameKey) {"NW",$(NoTitle),$(!REG_VT_SZ),$
(ProductProviderDisplayName)}
    endif
else
    debug-output "something wrong..."
endif
return
[AddParameterKey]
read-syms GeneralConstants
set SoftParameterKey = $($0)
ifstr(i) $(!STF_GUI_UNATTENDED) == "YES"
    shell $(!STF_UNATTENDED) ReadDefaultData $(!STF_UNATTENDED_SECTION)
    SetRegValue $(SoftParameterKey) {DefaultLocation,$(NoTitle),$(!REG_VT_SZ),$
(!DefaultLocation)}
    SetRegValue $(SoftParameterKey) {DefaultScriptOptions,$(NoTitle),$(!
REG_VT_DWORD),$(!DefaultScriptOptions)}
endif
CreateRegKey $(SoftParameterKey) {"Logon",$(NoTitle),GenericClass} "" $
(MAXIMUM_ALLOWED) "" LogonKey
CreateRegKey $(SoftParameterKey) {"Option",$(NoTitle),GenericClass} "" $
(MAXIMUM_ALLOWED) "" OptionKey
Set FLibraryErrCtl = 1
LibraryProcedure Result $(!NWCFG_HANDLE), SetEverybodyPermission, $(OptionKey),
2
Set FLibraryErrCtl = 0
CloseRegKey $(LogonKey)
CloseRegKey $(OptionKey)
Return
[GetFilesSize]
set FileSizeList = >(^((Files-Drivers,3),^(Files-NWWKSTA,3))
set TotalSize = 0
ForListDo $(FileSizeList)
    ForListDo $($)
        Debug-Output $($)

```

```

        Split-String $($) "=" SplitString
        set Size = *($($SplitString),3)
        set-add TotalSize = $(TotalSize) $(Size)
    EndForListDo
EndForListDo
set-div SizeInK = $(TotalSize) 1024
return $(SizeInK)
[Install-Update-Inf]
AddSectionFilesToCopyList Files-Inf $(SrcDir) $(!STF_WINDOWSSYSPATH)
set !STF_NCPA_FLUSH_COPYLIST = TRUE
CopyFilesInCopyList
exit
[RemoveBackupFiles]
return
[Restore-Oldexe]
[Restore-Oldsys]
[Source Media Descriptions]
1 = "Windows NT Workstation CD-ROM" , TAGFILE = cdrom_w.40
[Signature]
FileType = MICROSOFT_FILE
[GetSignature]
read-syms Signature
return $(FileType)
[ProductType]
STF_PRODUCT = Winnt
STF_PLATFORM = I386
[Files-Inf]
2, oemsetup.inf, SIZE=1000, RENAME=$(!UG_Filename)
[Files-Drivers]
1,NWRDR.SYS , SIZE=999
[Files-nwcfgdll]
1,NWAPI32.DLL , SIZE=999
[Files-NWWWKSTA]
1,NETWARE.DRV , SIZE=999
1,NW16.EXE , SIZE=3270
1,NWAPI16.DLL , SIZE=999
1,NWC.CPL , SIZE=999
1,NWDOC.CNT , SIZE=999
1,NWDOC.HLP , SIZE=999
1,NWEVENT.DLL , SIZE=999
1,NWSCRIPT.EXE , SIZE=999
1,NWWWKS.DLL , SIZE=999
1,PERFNW.DLL , SIZE=999
1,VWIPXSPX.DLL , SIZE=999
1,VWIPXSPX.EXE , SIZE=1158
[LanguagesSupported]
ENG
[WorkstationENG]
NWType = "Klient dla NetWare"
NWDescription = "Pozwala na logowanie z tego komputera do serwerów NetWare i na
dostęp do ich zasobów."
[GatewayENG]
NWType = "Gateway Service for NetWare"
NWDescription = "Enables other Windows NT computers to access NetWare servers
without running NetWare Client software."
[OptionsWINNTTextENG]
NWWWKSTA = "Klient dla NetWare"
[OptionsLANMANNTTextENG]
NWWWKSTA = "Brama (i klient) dla NetWare"

```

```

[FileConstantsENG]
NotEnoughSpace = "Brak miejsca na dysku. Usuń niepotrzebne pliki przed instalacją
"$(NWType)"."
UpgradeBetaOnTopOfDaytona = "You cannot update the beta version of "$(NWType)" over
Windows NT 3.51's "$(NWType)"."
InstallNWLINKFirst = "Zainstaluj transport zgodny z NwLink IPX/SPX przed instalacją
"$(NWType)"."
InstallNTWKSTAFirst = "Zainstaluj usługę Stacja robocza Windows NT przed instalacją
"$(NWType)"."
RemoveAndReboot = "Usuń wszystkie istniejące składowiki "$(NWType)" i uruchom
ponownie komputer przed instalacją ostatniej wersji "$(NWType)"."
UpgradeFirst = "Zainstaluj uaktualnienie Windows NT przed instalacją "$(NWType)"."
ProCaption = "Instalator Windows NT"
ProCancel = "Anuluj"
ProCancelMsg = "Sieć Windows NT nie jest zainstalowana poprawnie. "+
"Czy na pewno chcesz anulować kopiowanie plików?"
ProCancelCap = "Komunikat Instalatora sieci"
ProText1 = "Kopiowanie:"
ProText2 = "Do:"
FunctionTitle = "$(NWType)" Instalator"
ProductNWWKSTADescription = "$(NWDescription)"
ProductNWRDRDescription = "Usługa zgodna z readresatorem NetWare"
ProviderRDRName = "NetWare Compatible Redirector"
ProductNWWKSTATitle = "$(NWType)"
ProductNWWKSTADisplayName = "$(NWType)"
ProductNWRDRDisplayName = "NetWare Rdr"
ProductNWRDRTitle = "Readresator NetWare"
ProductProviderName = "System NetWare"
ProductProviderDisplayName = "System NetWare lub zgodny"
ProductPrintProviderName = "System NetWare lub zgodny"
OldPrintProviderName = "System Netware(R)"
OldPrintProviderName2 = "System NetWare"
REM1 = "REM Install network redirector"
[DialogConstantsENG]
Help = "Pomo&c"
Exit = "Anuluj"
OK = "OK"
HelpContext = ""
Continue = "Kontynuuj"
Cancel = "Anuluj"
ShellCodeErrorText = "Błąd kodu powłoki"
[FileDependentDlgENG]

```