

```

[Identification]
    OptionType = NetService
[Options]
    INETSRV
[FileConstants]
Manufacturer      = "Microsoft"
ProductMajorVersion = "4"
ProductMinorVersion = "0"
ProductVersion    = $(ProductMajorVersion)".$(ProductMinorVersion)
ProductOpSupport  = 134
ProductINETSRVName = "Inetsrv"
ProductKeyName    = $(!INTN_SoftwareBase)"\"$(Manufacturer)"\Inetsrv\CurrentVersion"
[GeneralConstants]
UtilityInf        = "UTILITY.INF"
subroutineinf     = "SUBROUTN.INF"
Exit_Code        = 0
BillboardVisible = 0
from              = ""
to                = ""
ExitCodeOk       = 0
ExitCodeCancel   = 1
ExitCodeFatal    = 2
KeyNull          = ""
MAXIMUM_ALLOWED  = 33554432
SERVICE_NO_CHANGE = 4294967295
RegistryErrorIndex = NO_ERROR
KeyProduct       = ""
KeyParameters    = ""
TRUE             = 1
FALSE            = 0
NoTitle          = 0
ExitState        = "Active"
OldVersionExisted = $(FALSE)
[date]
    Now = {} ? $(!LIBHANDLE) GetSystemDate
[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) $(i) in $(LanguageList)
        goto returnoptions
    else
        set Status = STATUS_NOLANGUAGE
        goto finish_ReturnOptions
    endif
returnoptions = +
    set OptionList      = ^(Options, 1)
    ifstr(i) $(!STF_PRODUCT) == "WINNT"
        set OptionTextList = ^(OptionsWINNTText$(i), 1)
    else
        set OptionTextList = ^(OptionsLANMANNTText$(i), 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 $(LanguageList)
        Return STATUS_NOLANGUAGE
    endif
    set-subst LF = "\n"
    read-syms GeneralConstants
    read-syms FileConstants
    read-syms DialogConstants$(!STF_LANGUAGE)
    ifstr(i) $(!NTN_Origination) == "NCPA"
        set Continue = $(OK)
    endif
    read-syms FileConstants$(!STF_LANGUAGE)
    detect date
    set-title $(FunctionTitle)
    set to = Begin
    set from = Begin
    set CommonStatus = STATUS_NO_EFFECT
    EndWait
Begin = +
    Ifstr(i) $(!NTN_InstallMode) == deinstall
        set StartLabel = removeadapter
    else-Ifstr(i) $(!NTN_InstallMode) == Update
        set StartLabel = UpgradeSoftware
    else-Ifstr(i) $(!NTN_InstallMode) == configure
        Debug-Output "Cannot configure the software."
        set Error = $(Warning1)
        set from = end
        set to = end
        goto nonfatalinfo
    else-Ifstr(i) $(!NTN_InstallMode) == bind
        set StartLabel = bindingadapter
    else
        set StartLabel = installadapter
        set OEM_ABANDON_SOFTWARE = {}
        set OEM_ABANDON_OPTIONS = {}
    endif
    set from = $(fatal)
    set to = $(fatal)
    goto $(StartLabel)
installadapter = +
    Shell "utility.inf" InstallSoftwareProduct "Microsoft" "Inetsrv" "oemnsvin.inf"
    set ISVersion = $($R1)
    set ISNetRule = $($R2)
    ifstr(i) $(!STF_PRODUCT) == "WINNT"
        set ProductTitle = $(FunctionWINNTTitle)
        set ProductDescription = $(ProductSoftwareWINNTDescription)
        set ProductName = $(ProductTitle)
    else
        set ProductTitle = $(FunctionLANMANNTTitle)
        set ProductDescription = $(ProductSoftwareLANMANNTDescription)

```

```

        set ProductName = $(ProductTitle)
    endif
    set NewValueList = {{SoftwareType,$(NoTitle),$(!REG_VT_SZ),"service"},+
    {MajorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMajorVersion)},+
    {MinorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMinorVersion)},+
    {Title,$(NoTitle),$(!REG_VT_SZ),$(ProductTitle)},+
    {Description,$(NoTitle),$(!REG_VT_SZ),$
(ProductDescription)},+
    {ServiceName,$(NoTitle),$(!REG_VT_SZ),$(ProductName)},+
    {OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupport)}, +
    {InstallDate,$(NoTitle),$(!REG_VT_DWORD),*$(Now),1}}
    Shell $(UtilityInf), AddValueList, $(ISVersion), $(NewValueList)
    ifint $($ShellCode) != $(!SHELL_CODE_OK)
        Debug-Output "OEMNSVNW.INF: ShellCode error."
        goto ShellCodeError
    endif
    set NewValueList = {{"InfOption",$(NoTitle),$(!REG_VT_SZ),"Inetsrv"}}
    Shell $(UtilityInf), AddValueList, $(ISNetRule), $(NewValueList)
    ifint $($ShellCode) != $(!SHELL_CODE_OK)
        Debug-Output "OEMNSVNW.INF: ShellCode error."
        goto ShellCodeError
    endif
    ifstr(i) $(!NTN_InstallPhase) == primary
        goto end
    endif
    OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_SoftwareBase)"\Microsoft\INetStp" $
(MAXIMUM_ALLOWED) BS_INetStp
    ifstr $(BS_INetStp) != $(KeyNull)
        ifstr(i) $(!STF_PRODUCT) == "WINNT"
            set Error = $(Warning3Winnt)
        else
            set Error = $(Warning3LanmanNt)
        endif
        set from = end
        set to = end
        goto warning
    else
        ifstr(i) $(!NTN_Origination) == "install"
            ifstr(i) $(!NTN_InstallPhase) == "secondary"
                ifstr(i) $(!STF_PRODUCT) == "WINNT"
                    set Error = $(Warning4Winnt)
                else
                    set Error = $(Warning4LanmanNt)
                endif
            endif
            Shell "utility.inf" RemoveSoftwareComponent "Microsoft" "Inetsrv"
"FALSE"
            set from = end
            set to = end
            goto nonfatal
        endif
    endif
    set FWaitForProcess = 1
    RunProgram RC "" "" $(!STF_WINDOWSSYSPATH)"\inetins.exe"
    set FWaitForProcess = 0
    OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_SoftwareBase)"\Microsoft\INetStp" $
(MAXIMUM_ALLOWED) INetStpKey

```

```

        ifstr $(INetStpKey) == $(KeyNull)
            Shell "utility.inf" RemoveSoftwareComponent "Microsoft" "Inetsrv"
"FALSE"
        endif
    endif
    set CommonStatus = STATUS_NO_EFFECT
    goto successful
configureadapter = +
    goto end
bindingadapter =+
    goto end
removeadapter = +
    OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_SoftwareBase)"\Microsoft\INetStp" $
(MAXIMUM_ALLOWED) BS_INetStp
    ifstr $(BS_INetStp) == $(KeyNull)
        Shell "utility.inf" RemoveSoftwareComponent "Microsoft" "Inetsrv" "FALSE"
        set CommonStatus = STATUS_NO_EFFECT
    else
        ifstr(i) $(!NTN_InstallPhase) == primary
            goto end
        endif
        GetRegValue $(BS_INetStp),"InstallPath",InstallPath
        set FWaitForProcess = 1
        RunProgram RC "" "" *$(InstallPath),4"\setup.exe" "/R"
        set FWaitForProcess = 0
        OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_SoftwareBase)"\Microsoft\INetStp" $
(MAXIMUM_ALLOWED) INetStpKey
        ifstr $(INetStpKey) == $(KeyNull)
            Shell "utility.inf" RemoveSoftwareComponent "Microsoft" "Inetsrv"
"FALSE"
        endif
    endif
    goto end
UpgradeSoftware = +
    ifstr(i) $(!STF_PRODUCT) == "WINNT"
        set ProductDescription = $(ProductSoftwareWINNTDescription)
    else
        set ProductDescription = $(ProductSoftwareLANMANNTDescription)
    endif
    OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED)
KeyCurrentVersion
    Ifstr $(KeyCurrentVersion) != $(KeyNull)
        SetRegValue $(KeyCurrentVersion) {Description,$(NoTitle),$(!REG_VT_SZ),$
(ProductDescription)}
        SetRegValue $(KeyCurrentVersion) {OperationsSupport,$(NoTitle),$(!
REG_VT_DWORD),$(ProductOpSupport)}
        CloseRegKey $(KeyCurrentVersion)
    endif
    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
    Ifint $(BillboardVisible) != 0
        Shell "subroutn.inf" PopBillboard
        Set BillboardVisible = 0
    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 = $(ShellCodeErrorTitle)
    set STF_MB_TEXT = $(ShellCodeErrorText)
    set STF_MB_TYPE = 1
    set STF_MB_ICON = 3

```


