

```

[Identification]
    OptionType = NetService
[Options]
    IPRIP
[FileConstants]
Manufacturer      = "Microsoft"
ProductMajorVersion = "4"
ProductMinorVersion = "0"
SoftwareType      = "service"
ProductVersion     = $(ProductMajorVersion)".$(ProductMinorVersion)
ProductOpSupport   = 132
ProductIPRIPIName = "IpRip"
ProductIPRIPIImagePath = "%SystemRoot%\System32\router.exe"
ProductIPRIPEventFile = "%SystemRoot%\System32\iprip.dll"
ProductSoftwareConfigDLL = tcpcfg.dll
ProductKeyName    = $(!NTN_SoftwareBase)"\"$(Manufacturer)"\IpRip\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) $($0) in $(LanguageList)
        goto returnoptions
    else
        set Status = STATUS_NOLANGUAGE
        goto finish_ReturnOptions
    endif
returnoptions = +
    set OptionList      = ^(Options, 1)
    set OptionTextList = ^(OptionsText$($0), 1)

```

```

    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 InstallFrom   = $($6)
    ifstr(i) $(InstallFrom) == ""
        set InstallFrom = "IPRIP"
    endif
    set LanguageList = ^(LanguagesSupported, 1)
    Ifcontains(i) $($0) NOT-IN $(LanguageList)
        Return STATUS_NOLANGUAGE
    endif
    Debug-Output "OEMNSVIR.INF: STF_CWDIR is: "$(!STF_CWDIR)
    Debug-Output "OEMNSVIR.INF: STF_LANGUAGE is: "$(!STF_LANGUAGE)
    set-subst LF = "\n"
    read-syms GeneralConstants
    read-syms FileConstants
    read-syms DialogConstants$(!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_SUCCESSFUL
    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."
        Shell $(UtilityInf),RegistryErrorString,CANNOT_CONFIGURE_SOFTWARE
        ifint $($ShellCode) != $(!SHELL_CODE_OK)
            Debug-Output "ShellCode error: cannot get an error string."
            goto ShellCodeError
        endif
        set Error = $($R0)
        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 = +
  set OEM_ABANDON_ON = TRUE
  StartWait
  OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED) KeySoftware
  Ifstr(i) $(KeySoftware) == $(KeyNull)
    ifstr(i) $(!NTN_InstallMode) == "install"
      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
    OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)"\TCP/IP" $
    (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 = ""
        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
      endif
    else
      CloseRegKey $(BS_KeyServices)
    endif
    Debug-Output "OEMNSVIR.INF: installadapter: installing [Install-
Option]"
    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
  OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_ServiceBase)"\IPRIP" $
  (MAXIMUM_ALLOWED) KeyService
  Ifstr(i) $(KeyService) == $(KeyNull)

```

```

Shell $(UtilityInf), AddSoftwareComponent, "Microsoft", +
    $(ProductIPRIPName), +
    $(ProductIPRIPName), +
    $(ProductIPRIPDisplayName), +
    $(STF_CONTEXTINFNAME), +
    $(ProductIPRIPIImagePath), +
    "autoserviceshare" , "", {"TCPIP"}, "", $(ProductIPRIPEventFile)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVIR.INF: ShellCode error"
    return STATUS_FAILED
endif
set RegistryErrorIndex = $($R0)
set ProductKey = $($R1)
set RuleKey = $($R2)
CloseRegKey $($R3)
set ParamKey = $($R4)
CloseRegKey $($R5)
Ifstr(i) $(RegistryErrorIndex) != "NO_ERROR"
    EndWait
    Debug-Output "OEMNSVIR.INF: Registry Error "$(RegistryErrorIndex)
    CloseRegKey $($R1)
    CloseRegKey $($R2)
    CloseRegKey $($R4)
    goto fatalregistry
endif
set NewValueList = {{SoftwareType,$(NoTitle),$(!REG_VT_SZ),$
(SoftwareType)},+
    {MajorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMajorVersion)},+
    {Title,$(NoTitle),$(!REG_VT_SZ),$
(ProductIPRIPTitle)},+
    {Description,$(NoTitle),$(!REG_VT_SZ),$
(ProductIPRIPDescription)},+
    {MinorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMinorVersion)},+
    {OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupport)}, +
    {InstallDate,$(NoTitle),$(!REG_VT_DWORD),*($
(Now),1)} +
    {Review,$(NoTitle),$(!REG_VT_DWORD),1}}
Shell $(UtilityInf), AddValueList, $(ProductKey), $(NewValueList)
CloseRegKey $(ProductKey)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "utility.INF: ShellCode error. Cannot add IPRIP router
service"
    goto ShellCodeError
endif
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNSVIR.INF: error: create service"
    set Status = STATUS_FAILED
endif
set NewValueList = +
    {{class, $(NoTitle), $(!REG_VT_SZ), "ipripRouter basic"},+
    {type,$(NoTitle),$(!REG_VT_SZ),"iprip ipripRouter"},+
    {InfoOption,$(NoTitle),$(!REG_VT_SZ),"IPRIP"}, +
    {bindform,$(NoTitle),$(!REG_VT_SZ),"""iprip"" yes yes simple"}, +
    {bindable,$(NoTitle),$(!REG_VT_MULTI_SZ),+

```

```

        {"ipripRouter ndisDriver non non 100"}}}
Shell $(UtilityInf), AddValueList, $(RuleKey), $(NewValueList)
CloseRegKey $(RuleKey)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "utility.INF: ShellCode error. Cannot add IPRIP
service"
        goto ShellCodeError
endif
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNSVIR.INF: Registry error: create service"
    set Status = STATUS_FAILED
endif
set Result = 0
ifstr(i) $(!STF_GUI_UNATTENDED) != "YES"
    LoadLibrary "x" $(!STF_CWDDIR)$(ProductSoftwareConfigDLL) !
hDialogDLL
        ifstr(i) $(!hDialogDLL) != NULL
            set FLibraryErrCtl = 1
            LibraryProcedure ResultList, $(!hDialogDLL),
TcpEnableRipSilentMode
            set FLibraryErrCtl = 0
            set Result = *$(ResultList), 1
        endif
    endif
SetRegValue $(ParamKey) {SilentRip, $(NoTitle), $(!REG_VT_DWORD), $
(Result)}
    CloseRegKey $(ParamKey)
endif
Else
    Debug-Output "OEMNSVIR.INF: already installed InstallFrom = "$(InstallFrom)
    GetRegValue $(KeySoftware) "MajorVersion" VersionInfo
    set VersionMajor = *$(VersionInfo), 4
    GetRegValue $(KeySoftware) "MinorVersion" VersionInfo
    set VersionMinor = *$(VersionInfo), 4
    set InstalledVersion = $(VersionMajor)."$(VersionMinor)
    ifstr(i) $(InstallFrom) == "IPRIP"
        read-syms VerExists$(!STF_LANGUAGE)
        set Text = $(Product$(Option)Title)$(Ver)$(ProductVersion)+
            $(Text1)
        Shell $(subroutineinf), SetupMessage $(!STF_LANGUAGE) "NONFATAL" $
(Text)
        Ifint $($ShellCode) != $(!SHELL_CODE_OK)
            Goto fatal
        Else-Ifstr(i) $($R0) == STATUS_FAILED
            Goto fatal
        Else-Ifstr(i) $($R0) == STATUS_USERCANCEL
            Goto end
        Endif
    else
        Shell "" AddIPRIPUsed $(InstallFrom)
    endif
    goto end
EndIf
Ifstr(i) $(KeySoftware) != $(KeyNull)
    CloseRegKey $(KeySoftware)
endif
Shell "" AddIPRIPUsed $(InstallFrom)

```

```

    OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Services\Tcpip\
Parameters" $(MAXIMUM_ALLOWED) KeyTcpParameter
    ifstr(i) $(KeyTcpParameter) != ""
        GetRegValue $(KeyTcpParameter) "IPEnableRouter" RouterInfo
        ifint *$(RouterInfo),4) == 0
            SetRegValue $(KeyTcpParameter) {"IPEnableRouter", $(NoTitle), $(!
REG_VT_DWORD), 1}
        endif
    endif
    goto successful
configureadapter = +
    goto end
bindingadapter =+
    goto end
removeadapter = +
    set IpRipKeyName = "System\CurrentControlSet\Services\IpRip"
    set ProductIPRIPName = "IpRip"
    Shell "" RemoveIPRIPUsed $(InstallFrom)
    OpenRegKey $(!REG_H_LOCAL) "" $(IpRipKeyName) $(MAXIMUM_ALLOWED) BS_KeyServices
    Ifstr $(BS_KeyServices) != $(KeyNull)
        CloseRegKey $(BS_KeyServices)
        Shell "" NumOfUsed
        set NumOfUsed = $($R0)
        ifint $(NumOfUsed) == 1
            Shell "" FindUsed "IPRIP"
            ifstr(i) $($R0) == TRUE
                set NumOfUsed = 0
            endif
        endif
        ifint $(NumOfUsed) == 0
            Shell $(UtilityInf), RemoveSoftwareComponent, $(Manufacturer), $
(ProductIPRIPName)
        else
            ifstr(i) $(InstallFrom) == "IPRIP"
                read-syms InUse$(!STF_LANGUAGE)
                set from = end
                set to = end
                goto warning
            endif
        endif
    endif
    goto end
UpgradeSoftware = +
    OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED) KeyProduct
    Ifstr(i) $(KeyProduct) != $(KeyNull)
        install "Install-Update"
        ifstr(i) $(STF_INSTALL_OUTCOME) != STF_SUCCESS
            goto fatal
        endif
        SetRegValue $(KeyProduct) {SoftwareType,$(NoTitle),$(!REG_VT_SZ),$
(SoftwareType)}
        SetRegValue $(KeyProduct) {MajorVersion,$(NoTitle),$(!REG_VT_SZ),$
(ProductMajorVersion)}
        SetRegValue $(KeyProduct) {MinorVersion,$(NoTitle),$(!REG_VT_SZ),$
(ProductMinorVersion)}
        SetRegValue $(KeyProduct) {Description,$(NoTitle),$(!REG_VT_SZ),$
(ProductIPRIPDescription)}
        SetRegValue $(KeyProduct) {OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupport)}

```

```

        LibraryProcedure Result, $(!LIBHANDLE), SetupChangeServiceConfig, "IpRip",
$(SERVICE_NO_CHANGE), $(SERVICE_NO_CHANGE), $(SERVICE_NO_CHANGE), "%SystemRoot%\
System32\router.exe", "", "", "", "", ""
        CloseRegKey $(KeyProduct)
    endif
    goto end
successful = +
    goto end
infomsg =+
    read-syms InfoDlg
    ui start "Warning"
    set CommonStatus = STATUS_USERCANCEL
    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
    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)
[GetFilesSize]
    set FileSizeList = ^(Files-TCPPRINT,3)
    set TotalSize = 0
    ForListDo $(FileSizeList)
        Split-String $($) "=" SplitString
        set Size = *$(SplitString),3)
        set-add TotalSize = $(TotalSize) $(Size)
    EndForListDo
    set-div SizeInK = $(TotalSize) 1024
    return $(SizeInK)
[Install-Option]
    set STF_VITAL = NO
    ifstr(i) $(AddCopy) == "YES"
        AddSectionFilesToCopyList Files-IPRIP $(SrcDir) $(!STF_WINDOWSSYSPATH)
    endif
    ifstr(i) $(DoCopy) == "YES"
        set !STF_NCPA_FLUSH_COPYLIST = TRUE
        CopyFilesInCopyList
    endif
    ifstr(i) $(DoConfig) == "YES"
    endif
    Exit
[Install-Update]
    set STF_VITAL = NO

```



```

set STF_OVERWRITE      = "VERIFYSOURCEOLDER"
AddSectionFilesToCopyList Files-IPRIP $(SrcDir) $(!STF_WINDOWSSYSPATH)
Exit
[AddIPRIPUsed]
  read-syms GeneralConstants
  read-syms FileConstants
  read-syms FileConstants$(!STF_LANGUAGE)
  set KeyName = $($0)
  OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED) KeyService
  ifstr(i) $(KeyService) != ""
    OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName)\UseRef" $(MAXIMUM_ALLOWED)
KeyUseRef
  ifstr(i) $(KeyUseRef) == ""
    CreateRegKey $(KeyService) {"UseRef",$(NoTitle),GenericClass} "" $
(MAXIMUM_ALLOWED) "" KeyUseRef
  endif
  ifstr(i) $(KeyUseRef) != ""
    SetRegValue $(KeyUseRef) {$(KeyName),$(NoTitle),$(!REG_VT_SZ),"1"}
    CloseRegKey $(KeyUseRef)
  endif
  CloseRegKey $(KeyService)
else
endif
return
[RemoveIPRIPUsed]
  read-syms GeneralConstants
  read-syms FileConstants
  read-syms FileConstants$(!STF_LANGUAGE)
  set KeyName = $($0)
  OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED) KeyService
  ifstr(i) $(KeyService) != ""
    OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName)\UseRef" $(MAXIMUM_ALLOWED)
KeyUseRef
  ifstr(i) $(KeyUseRef) == ""
    CreateRegKey $(KeyService) {"UseRef",$(NoTitle),GenericClass} "" $
(MAXIMUM_ALLOWED) "" KeyUseRef
  endif
  ifstr(i) $(KeyUseRef) != ""
    DeleteRegValue $(KeyUseRef) $(KeyName)
    CloseRegKey $(KeyUseRef)
  endif
  CloseRegKey $(KeyService)
else
endif
return
[NumOfUsed]
  read-syms GeneralConstants
  read-syms FileConstants
  read-syms FileConstants$(!STF_LANGUAGE)
  OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED) KeyService
  ifstr(i) $(KeyService) != ""
    OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName)\UseRef" $(MAXIMUM_ALLOWED)
KeyUseRef
  ifstr(i) $(KeyUseRef) == ""
    CreateRegKey $(KeyService) {"UseRef",$(NoTitle),GenericClass} "" $
(MAXIMUM_ALLOWED) "" KeyUseRef
  endif
  ifstr(i) $(KeyUseRef) != ""
    EnumRegValue $(KeyUseRef) UsageList

```

```

        set Num = 0
        ForListDo $(UsageList)
            set-add Num = $(Num), 1
        EndForListDo
        CloseRegKey $(KeyUseRef)
    endif
    CloseRegKey $(KeyService)
else
endif
return $(Num)
[FindUsed]
read-syms GeneralConstants
read-syms FileConstants
read-syms FileConstants$(!STF_LANGUAGE)
set KeyName = $($0)
set Find = FALSE
OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED) KeyService
ifstr(i) $(KeyService) != ""
    OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName)"\UseRef" $(MAXIMUM_ALLOWED)
KeyUseRef
    ifstr(i) $(KeyUseRef) == ""
        CreateRegKey $(KeyService) {"UseRef",$(NoTitle),GenericClass} "" $
(MAXIMUM_ALLOWED) "" KeyUseRef
    endif
    ifstr(i) $(KeyUseRef) != ""
        GetRegValue $(KeyUseRef),$(KeyName), KeyInfo
        ifstr(i) $(RegLastError) == $(!REG_ERROR_SUCCESS)
            set Find = TRUE
        endif
        CloseRegKey $(KeyUseRef)
    endif
    CloseRegKey $(KeyService)
else
endif
return $(Find)
[Source Media Descriptions]
1 = "Windows NT Service Pack CD-ROM" , TAGFILE = spcdrom.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-IPRIP]
1,IPRIP.DLL , SIZE=161884
1,ROUTER.EXE , SIZE=52712
1,ROUTETAB.DLL , SIZE=55832
1,TRACE.DLL , SIZE=57844
[LanguagesSupported]
ENG
[OptionsTextENG]
IPRIP = "RIP for Internet Protocol"
[FileConstantsENG]
ProCaption = "Windows NT 001000"
ProCancel = "0000"

```

