

```

[Identification]
    OptionType = NetService
[Options]
    DHCP
[FileConstants]
Manufacturer      = "Microsoft"
ProductMajorVersion  = "4"
ProductMinorVersion  = "0"
ProductVersion      = $(ProductMajorVersion)."$(ProductMinorVersion)
ProductOpSupport     = 132
ProductDHCPName     = "DHCPServer"
ProductDHCPImagePath = "%SystemRoot%\System32\tcpsvcs.exe"
ProductDHCPSvcType  = "autoservicesshare"
NetRuleDHCPType     = "dhcpserver tcpipService dhcp"
NetRuleDHCPUse      = "service yes yes"
NetRuleDHCPBindForm = "" "Dhcpserver" "" yes yes container"
NetRuleDHCPClass    = {"dhcp tcpipService yes"}
ProductKeyName      = $(!NTN_SoftwareBase)"\"$(Manufacturer)"\"$(Product$(
Option)Name)"\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 LanguageList = ^(LanguagesSupported, 1)
    Ifcontains(i) $(0) NOT-IN $(LanguageList)
        Return STATUS_NOLANGUAGE
    endif
    Debug-Output "OEMNSVCU.INF: STF_CWDIR is: "$(!STF_CWDIR)
    Debug-Output "OEMNSVCU.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)
    read-syms FileDependentDlg$(!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
        set StartLabel = configureadapter
        Ifstr(i) $(ProductKeyName) == $(!NTN_RegBase)
            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
        endif
    else-Ifstr(i) $(!NTN_InstallMode) == bind
        set StartLabel = bindingadapter
    else
        set StartLabel = installadapter
        set OEM_ABANDON_SOFTWARE = {}
        set OEM_ABANDON_OPTIONS = {}
    endif
endif

```

```

set from = $(fatal)
set to = $(fatal)
goto $(StartLabel)
installadapter = +
  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)
ifstr(i) $(!NTN_InstallMode) == "install"
  Debug-Output "OEMNSVCU.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
set OEM_ABANDON_ON = TRUE
StartWait
set ThisOption = DHCP
Shell $(UtilityInf), AddSoftwareComponent, $(Manufacturer),+
$(Product$(ThisOption)Name),+
$(Product$(ThisOption)Name),+
$(Product$(ThisOption)DisplayName), $(STF_CONTEXTINFNAME),+
$(Product$(ThisOption)ImagePath),+
$(Product$(ThisOption)SvcType),+
"", {"Tcpip"}, "", "%SystemRoot%\System32\dhcpcsvc.dll", 7, "", "", "",+
"%SystemRoot%\System32\kernel32.dll"
ifint $($ShellCode) != $(!SHELL_CODE_OK)
  Debug-Output "OEMNXPTC.INF: ShellCode error, add software component"
  goto ShellCodeError
endif
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
  EndWait
  Debug-Output "OEMNSVDH.INF: Registry error: add software components"
  CloseRegKey $($R1)
  CloseRegKey $($R2)
  CloseRegKey $($R3)
  CloseRegKey $($R4)
  CloseRegKey $($R5)
  goto fatalregistry
endif
Set DHCPVersKeyHandle = $($R1)
set DHCPNetRulesKeyHandle = $($R2)
CloseRegKey $($R3)

```

```

Set DHCPParamKeyHandle = $(R4)
Set DHCPLinkageHandle = $(R5)
set NewValueList = {{OtherDependencies,$(NoTitle),$(!REG_VT_MULTI_SZ),
{"Rpcss","NTLMSSP"}}}
Shell $(UtilityInf), AddValueList, $(DHCPLinkageHandle), $(NewValueList)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNXPTC.INF: ShellCode error."
    goto ShellCodeError
endif
CloseRegKey $(DHCPLinkageHandle)
set NewValueList = {{SoftwareType,$(NoTitle),$(!REG_VT_SZ),"service"},+
    {MajorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMajorVersion)},+
    {MinorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMinorVersion)},+
    {Title,$(NoTitle),$(!REG_VT_SZ),$(Product$
(ThisOption)Title)},+
    {Description,$(NoTitle),$(!REG_VT_SZ),$(Product$
(ThisOption)Description)},+
    {ServiceName,$(NoTitle),$(!REG_VT_SZ),$(Product$
(ThisOption)Name)},+
    {OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupport)},+
    {InstallDate,$(NoTitle),$(!REG_VT_DWORD),*($Now),1}}}
Shell $(UtilityInf), AddValueList, $(DHCPVersKeyHandle), $(NewValueList)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNXPTC.INF: ShellCode error, add value list"
    goto ShellCodeError
endif
set RegistryErrorIndex = $(R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNXPTC.INF: Registry error: add value list."
    CloseRegKey $(DHCPVersKeyHandle)
    CloseRegKey $(DHCPNetRulesKeyHandle)
    CloseRegKey $(DHCPParmKeyHandle)
    goto fatalregistry
Endif
set NewValueList = {{type,$(NoTitle),$(!REG_VT_SZ),$(NetRuleDHCPType)},+
    {class,$(NoTitle),$(!REG_VT_MULTI_SZ),$(NetRuleDHCPClass)},
+
    {use,$(NoTitle),$(!REG_VT_SZ),$(NetRuleDHCPUse)}, +
    {bindform,$(NoTitle),$(!REG_VT_SZ),$(NetRuleDHCPBindForm)},
+
    {InfoOption,$(NoTitle),$(!REG_VT_SZ),$(Option)}}}
Shell $(UtilityInf), AddValueList, $(DHCPNetRulesKeyHandle), $(NewValueList)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "ShellCode error."
    goto ShellCodeError
endif
CloseRegKey $(DHCPNetRulesKeyHandle)
set RegistryErrorIndex = $(R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNXPTC.INF: Registry error: add value list."
    CloseRegKey $(DHCPVersKeyHandle)
    CloseRegKey $(DHCPParmKeyHandle)
    goto fatalregistry
Endif

```

```

set NewValueList = {{APIProtocolSupport,$(NoTitle),$(!REG_VT_DWORD),5},+
{DebugFlag,$(NoTitle),$(!REG_VT_DWORD),0},+
{DatabaseName,$(NoTitle),$(!REG_VT_SZ),"dhcp.mdb"},+
{BackupDatabasePath,$(NoTitle),$(!
REG_VT_EXPAND_SZ),"%SystemRoot%\System32\dhcp\backup"},+
{BackupInterval,$(NoTitle),$(!REG_VT_DWORD),60},+
{DatabaseLoggingFlag,$(NoTitle),$(!REG_VT_DWORD),1},+
{DatabaseCleanupInterval,$(NoTitle),$(!REG_VT_DWORD),1440},+
{RestoreFlag,$(NoTitle),$(!REG_VT_DWORD),0},+
{DatabasePath,$(NoTitle),$(!REG_VT_EXPAND_SZ),"%SystemRoot%\
System32\dhcp"}}
Shell $(UtilityInf), AddValueList, $(DHCPParmKeyHandle), $(NewValueList)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNXPTC.INF: ShellCode error, add value list"
    goto ShellCodeError
endif
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNXPTC.INF: Registry error: add value list."
    CloseRegKey $(DHCPVersKeyHandle)
    CloseRegKey $(DHCPParmKeyHandle)
    goto fatalregistry
Endif
CreateRegKey $(DHCPParmsKeyHandle) {"MultiHomeInfo",$(NoTitle),GenericClass} ""
+
    $(MAXIMUM_ALLOWED) "" KeyMultiHomeInfo
CloseRegKey $(KeyMultiHomeInfo)
CreateRegKey $(DHCPParmsKeyHandle) {"Configuration",$(NoTitle),GenericClass} ""
+
    $(MAXIMUM_ALLOWED) "" KeyConfiguration
CloseRegKey $(KeyConfiguration)
CloseRegKey $(DHCPVersKeyHandle)
CloseRegKey $(DHCPParmsKeyHandle)
ifstr(i) $(!STF_INSTALL_TYPE) != "SETUPBOOTED"
    Shell "" AddDHCP
endif
Shell "oemnsvsn.inf" InstallMicrosoftAgent "DHCPServer" "DhcpMibAgent"
"%SystemRoot%\System32\dhcplib.dll"
EndWait
Shell $(UtilityInf) SetMasterComponent Microsoft Tcpip $(STF_CONTEXTINFNAME)
DHCP
ifstr(i) $(!NTN_InstallPhase) != primary
    read-syms DHCPServerDialog
    ui start "Warning"
endif
return STATUS_SUCCESSFUL
configureadapter = +
    goto end
bindingadapter =+
    goto end
removeadapter = +
    Shell $(UtilityInf), RemoveSoftwareComponent, $(Manufacturer), $
(ProductDHCPName)
ifstr(i) $(!STF_INSTALL_TYPE) != "SETUPBOOTED"
    install "Remove-DHCP-Icon"
endif
Shell "oemnsvsn.inf" RemoveMicrosoftAgent "DhcpMibAgent"
goto end

```

```

UpgradeSoftware = +
  OpenRegKey $(!REG_H_LOCAL) "" $(ProductKeyName) $(MAXIMUM_ALLOWED) KeyProduct
  Ifstr $(KeyProduct) != $(KeyNull)
    install "Install-Update"
    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),$(!REG_VT_SZ),$(Product$
(Option)Description)}
    SetRegValue $(KeyProduct) {OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupport)}
    install "Remove-DHCP-Icon"
    install "Install-DHCP-Icon"
    CloseRegKey $(KeyProduct)
    OpenRegKey $(!REG_H_LOCAL) "" "System\CurrentControlSet\Services\
DHCPserver\Parameters" $(MAXIMUM_ALLOWED) DHCPsrvKey
    ifstr(i) $(DHCPsrvKey) != ""
      SetRegValue $(DHCPsrvKey) {BackupInterval,$(NoTitle),$(!
REG_VT_DWORD),60}
      GetRegValue $(DHCPsrvKey),"DatabaseCleanupInterval", IntervalInfo
      ifint *$(IntervalInfo),4) == 86400
        SetRegValue $(DHCPsrvKey) {DatabaseCleanupInterval,$(NoTitle),$(!
REG_VT_DWORD),1440}
      endif
    endif
    CloseRegKey $(DHCPsrvKey)
  endif
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)
[GetFileSize]
set FileSizeList = ^(Files-DHCP,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 = ""
ifstr(i) $(AddCopy) == "YES"
    AddSectionFilesToCopyList Files-DHCP $(SrcDir) $(!STF_WINDOWSSYSPATH)
endif
ifstr(i) $(DoCopy) == "YES"
    set !STF_NCPA_FLUSH_COPYLIST = TRUE
    CopyFilesInCopyList
else
    LibraryProcedure STATUS,$(!NCPA_HANDLE), CopySingleFile $(!STF_HWND) $
(SrcDir)\DHCPADMN.EXE $(!STF_WINDOWSSYSPATH)\DHCPADMN.EXE
endif
ifstr(i) $(DoConfig) == "YES"
endif
Exit
[Install-Update]
set STF_VITAL = ""
set STF_OVERWRITE = "VERIFYSOURCEOLDER"
AddSectionFilesToCopyList Files-DHCP $(SrcDir) $(!STF_WINDOWSSYSPATH)
Exit
[Install-DHCP-Icon]
CreateCommonProgManGroup $(GroupName), ""
RemoveCommonProgManItem $(GroupName), $(IconName)
CreateCommonProgManItem $(GroupName), $(IconName), dhcpadmn.exe, dhcpadmn.exe,
""
ShowCommonProgManGroup $(GroupName), 6
exit
[Remove-DHCP-Icon]
RemoveCommonProgManItem $(OldGroupName), $(IconName)
RemoveCommonProgManItem $(GroupName), $(IconName)
exit
[AddDHCP]
read-syms FileConstants$(!STF_LANGUAGE)
read-syms GeneralConstants
OpenRegKey $(!REG_H_LOCAL) "" $(!NTN_SoftwareBase)"\Microsoft\DHCP\Server" $
(MAXIMUM_ALLOWED) ProductKey
ifstr(i) $(ProductKey) != ""
    install "Install-DHCP-Icon"
    CloseRegKey $(ProductKey)
endif
return
[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)
[LanguagesSupported]
ENG
[DialogConstantsENG]
Help = "0000(&H)"
Exit = "00:00"
OK = "OK"
HelpContext = ""
Continue = "000s"
Cancel = "00:00"
[FileDependentDlgENG]
ProCaption = "Windows NT 001000"
ProCancel = "00:00"
ProCancelMsg = "Windows NT 0000000000i0000000. 000B"+
"00:00-000. 000?"
ProCancelCap = "000 001000 v0000"
ProText1 = "0000:"
ProText2 = "0000:"
[FileConstantsENG]
ProductDHCPDisplayName = "Microsoft DHCP Server"
ProductDHCPTitle = "Microsoft DHCP 000"
ProductDHCPDescription = "DHCP (000Ivč\00000010) 000000 000"0000s0000s0000000?AIP 000
00000I0Q000000000. 000"
OldGroupName = "00000000"
GroupName = "00000"
IconName = "DHCP 000"
[OptionsTextENG]
DHCP = "Microsoft DHCP 000"
[DHCPServerDialog]
DHCPTxt = "0000 DHCP 00g000 IP 000:0000000000000000A"+
"000000I IP 000:000g0p00000K0v0000000. 000000.0 000:00 [0SPH000] 00"+
"0-0000000 TCP/IP 00000000e 00A00\0000000A00000000000. 000B"
STF_MB_TITLE = $(DHCP)
DlgType = "MessageBox"
STF_MB_TEXT = $(DHCPTxt)
STF_MB_TYPE = 1
STF_MB_ICON = 2
STF_MB_DEF = 1
```