```
[PCIOptions]
[PCIFilename]
[PCIOptionsTextENG]
[MCAOptions]
[MCAFilename]
[MCAOptionsTextENG]
[ISAOptions]
Diehl_DIVA
Diehl_DIVAPRO
Diehl_S2M
Diehl_QUADRO
Diehl_SCOM
Diehl_WAN
ECCARDS
[ISAFilename]
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\EICON\oemsetup.inf
[ISAOptionsTextENG]
Diehl_DIVA  = "Eicon DIVA ISDN ISA Adapter"
Diehl_DIVAPRO = "Eicon DIVA PRO ISDN Adapter with Advanced DSP"
Diehl_S2M   = "Eicon Primary Rate ISDN Adapter"
Diehl_QUADRO = "Eicon QUADRO ISDN Adapter"
Diehl_SCOM  = "Eicon SCOM ISDN Adapter"
Diehl_WAN   = "Eicon Virtual WAN-Miniport ISDN Interface"
ECCARDS         = "Eicon WAN Adapters"
[EISAOptions]
Diehl_DIVA
Diehl_DIVAPRO
Diehl_S2M
Diehl_QUADRO
Diehl_SCOM
Diehl_WAN
ECCARDS
[EISAFilename]
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\DIWAN\oemsetup.inf
drvlib\netcard\alpha\wan\EICON\oemsetup.inf
[EISAOptionsTextENG]
Diehl_DIVA  = "Eicon DIVA ISDN ISA Adapter"
Diehl_DIVAPRO = "Eicon DIVA PRO ISDN Adapter with Advanced DSP"
Diehl_S2M   = "Eicon Primary Rate ISDN Adapter"
Diehl_QUADRO = "Eicon QUADRO ISDN Adapter"
Diehl_SCOM  = "Eicon SCOM ISDN Adapter"
Diehl_WAN   = "Eicon Virtual WAN-Miniport ISDN Interface"
ECCARDS         = "Eicon WAN Adapters"
[Jazz-Internal BusOptions]
[Jazz-Internal BusFilename]
[Jazz-Internal BusOptionsTextENG]
[PCMCIAOptions]
Diehl_DIVAPCM
```

```
[PCMCIAFilename]
drvlib\netcard\alpha\wan\DIWAN\OEMSETUP.INF
[PCMCIAOptionsTextENG]
Diehl_DIVAPCM = "Eicon DIVA PCMCIA ISDN Adapter"
[Identification]
    OptionType = NetAdapter
[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)
        ifstr(i) $($1) == ""
           goto returnoptions
        endif
    else
        set Status = STATUS_NOLANGUAGE
        goto finish_ReturnOptions
    endif
returnoptions = +
    set OptionList     = ^($($1)Options, 1)
    set OptionTextList = ^($($1)OptionsText$($0), 1)
    set Status         = STATUS_SUCCESSFUL
finish_ReturnOptions = +
    Return $(Status) $(OptionList) $(OptionTextList)
[FileConstants]
UtilityInf      = "UTILITY.INF"
ParamInf        = "NCPARAM.INF"
subroutineinf   = "SUBROUTN.INF"
SoftwareType    = "driver"
Exit_Code       = 0
[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
[InstallOption]
    set Option   = $($1)
    set SrcDir   = $($2)
    set AddCopy  = "YES"
    set DoCopy   = "YES"
    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
    ForListDo $(!STF_BUSTYPELIST)
      set OptionList = ^($($)Options,1)
      set FileNameList = ^($($)Filename,1)
      ifint ~($(OptionList),$(Option)) != 0
          goto FoundBus
      endif
    EndForListDo
FoundBus = +
    debug-output *($(FileNameList), ~($(OptionList), $(Option)))
    Shell "ntlanman.inf" RemoveOldInfFile $(Option), "NetAdapter", "oemnad",
"oemnadzz.inf"
    set Num = 0
nextinf = +
    ifint $(Num) < 99
      set OemFileName = "oemnad"$(Num)".inf"
      set FileToCheck = $(!STF_WINDOWSSYSPATH)"\"$(OemFileName)
      install Install-SeeIfFileIsPresent
      ifstr(i) $(STATUS) == "YES"
          set-add Num = $(Num) 1
          goto nextinf
      else
          goto copyinf
      endif
    endif
    goto finish
copyinf = +
    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 finish
    Endif
    Split-String $($R1), "\", lSrcDirParts
    QueryListSize cSrcDirParts $(lSrcDirParts)
    ifstr(i) *($(lSrcDirParts),$(cSrcDirParts)) != "\"
        Set SrcDir = $($R1)"\"
    else
        Set SrcDir = $($R1)
    Endif
    ifint $(cSrcDirParts) >= 3
        debug-output "OEMNADZZ.INF: checking to remove platform"
        ifstr(i) *($(lSrcDirParts),$(cSrcDirParts)) == "\"
            set-sub iStopParts = $(cSrcDirParts), 2
        else
```

```
            set-sub iStopParts = $(cSrcDirParts), 1
        endif
        debug-output "OEMNADZZ.INF: parts were "$(cSrcDirParts)" and stops at "$
(iStopParts)
        set SrcDirMin = ""
        set SkipAppend = FALSE
        ForListDo $(lSrcDirParts)
            ifint $(#) > $(iStopParts)
                ifstr(i) $($) == $(!STF_PLATFORM)
                    set SkipAppend = TRUE
                    debug-output "OEMNADZZ.INF: skipping appended"$($)
                    goto skip_appending_list_item
                else
                    ifstr(i) $(SkipAppend) == TRUE
                        debug-output "OEMNADZZ.INF: skipping appended"$($)
                        goto skip_appending_list_item
                    endif
                endif
            endif
            set NewSrcDir = $(SrcDirMin)$($)
            set SrcDirMin = $(NewSrcDir)
            debug-output "OEMNADZZ.INF: appended items "$(SrcDirMin)
skip_appending_list_item = +
        EndForListDo
    else
        debug-output "OEMNADZZ.INF: path too small to have platform"
        set SrcDirMin = *($(lSrcDirParts),1)"\"
    endif
    debug-output "OEMNADZZ.INF: SrcDirMin : "$(SrcDirMin)
    set DiskLocation = $(SrcDirMin)
    debug-output "OEMNADZZ.INF: SrcDir : "$(SrcDir)
    set OemSrc = $(SrcDir)*($(FileNameList), ~($(OptionList), $(Option)))
    set OemDst = $(!STF_WINDOWSSYSPATH)"\"$(OemFileName)
    set FileToCheck = $(OemSrc)
    install Install-SeeIfFileIsPresent
    ifstr(i) $(STATUS) != "YES"
        set File1 = $(OemSrc)
        set OemSrc = $(SrcDirMin)*($(FileNameList), ~($(OptionList), $(Option)))
        set FileToCheck = $(OemSrc)
        install Install-SeeIfFileIsPresent
        ifstr(i) $(STATUS) != "YES"
            set FileToCheck = $(DiskLocation)"oemsetup.inf"
              install Install-SeeIfFileIsPresent
              ifstr(i) $(STATUS) == "YES"
                  set OemSrc = $(FileToCheck)
                  goto InstallINF
            endif
            set FileToCheck = $(DiskLocation)"oemsetnt.inf"
              install Install-SeeIfFileIsPresent
              ifstr(i) $(STATUS) == "YES"
                  set OemSrc = $(FileToCheck)
                  goto InstallINF
              endif
              set File2 = $(OemSrc)
              set File3 = $(DiskLocation)"\oemsetup.inf"
              set File4 = $(DiskLocation)"\oemsetnt.inf"
              read-syms ErrorMsg$(!STF_LANGUAGE)
              set DlgText = $(CannotFindOemsetup)
              Shell "Subroutn.Inf" SetupMessage $(!STF_LANGUAGE) "NONFATAL" $
```

```
(DlgText)
            Set !STF_SRCDIR_OVERRIDE = ""
              goto copyinf
        endif
    endif
InstallINF = +
    install Install-OemINF
    Split-String $(OemSrc), "\", OemSrcInfo
    QueryListSize ListSize $(OemSrcInfo)
    set-sub ListSize = $(ListSize) 2
    set Count = 0
    set SrcDir = ""
SrcDirLoop = +
    set-add Count = $(Count) 1
    set SrcDir = $(SrcDir)*($(OemSrcInfo),$(Count))
    ifint $(Count) != $(ListSize)
      goto SrcDirLoop
    endif
    debug-output $(SrcDir)
    set !STF_SRCDIR_OVERRIDE = $(SrcDir)
    Shell $(OemDst) "InstallOption" +
      $(!STF_LANGUAGE) $(Option) +
      $(SrcDir) $(AddCopy) $(DoCopy) $(DoConfig)
finish = +
    return STATUS_SUCCESSFUL
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"
setfailed = +
    set CommonStatus = STATUS_FAILED
term = +
    Return $(CommonStatus)
[Install-SeeIfFileIsPresent]
LibraryProcedure STATUS,$(!LIBHANDLE), CheckFileExistance $(FileToCheck)
exit
[Install-OemINF]
LibraryProcedure STATUS,$(!NCPA_HANDLE), CopySingleFile $(!STF_HWND) $(OemSrc) $
(OemDst)
exit
[Source Media Descriptions]
    1 = "Windows NT Workstation CD-ROM" , TAGFILE = cdrom_w.40
[Signature]
    FileType = MICROSOFT_FILE
[GetSignature]
    read-syms Signature
    return $(FileType)
[ProductType]
STF_PRODUCT  = WinNT
STF_PLATFORM = Alpha
[Files-Inf]
2,    oemsetup.inf,      SIZE=1000, RENAME=$(!UG_Filename)
[LanguagesSupported]
    ENG
[FileConstantsENG]
```

```
ProCaption   = "Windows NT ��ı���"
ProCancel    = "����"
ProCancelMsg = "Windows NT ȯ����������İ����Ă��· ���B"+
               "�����~���· ���?"
ProCancelCap = "ȯ�� ��ı��� γ����"
ProText1     = "����:"
ProText2     = "����:"
ShellCodeErrorTitle     = "�װ: "
ShellCodeErrorText      = "��� ���� �װ�ł��B"
[ErrorMsgENG]
CannotFindOemsetup = "��ı���� "$(File1)"�A"$(File2)"�A"$(File3)"�A�· ��� "$(File4)"
����������Б��ł��· ���B����������������������ǂ����m�F���Ă���������B"
```