

```

[Identification]
    OptionType = NetService
[Options]
    RPCLOCATE
    RPCBANYAN
[FileConstants]
UtilityInf      = "UTILITY.INF"
subroutineinf  = "SUBROUTN.INF"
SoftwareType   = "service"
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)
Manufacturer   = "Microsoft"
ProductMajorVersion = "4"
ProductMinorVersion = "0"
ProductVersion = $(ProductMajorVersion)."$(ProductMinorVersion)
ProductOpSupportRPC = 134
ProductOpSupportRPCBANYAN = 132
RpcBanyanSoftKeyName = "SOFTWARE\Microsoft\RpcBanyan"
RpcSoftKeyName = "SOFTWARE\Microsoft\Rpc"
RpcSrvKeyName = "Software\Microsoft\RPCLOCATOR"
RpcSecurityKeyName = "Software\Microsoft\Rpc\SecurityService"
RpcSrvNameRpcss = "RPCSS"
RpcSrvNameLocator = "RPCLOCATOR"
RpcSrvNameBanyan = "RpcBanyan"
RpcSrvNameSSP = "NtLmSsp"
RpcSrvExeRpcss = "%SystemRoot%\System32\RPCSS.EXE"
RpcSrvExeLocator = "%SystemRoot%\System32\LOCATOR.EXE"
RpcSrvExeSSP = "%SystemRoot%\System32\SERVICES.EXE"
ProductKeyName = $(RpcSrvKeyName)"\CurrentVersion"
ServerProto = "\ServerProtocols"
ClientProto = "\ClientProtocols"
RpcSrvDepend= {"LanmanWorkstation","Rdr" }
MSSecurityDLL = "security.dll"
LocatorChoiceList = ^(LocatorData,1)
LocatorEndPointList = ^(LocatorData,2)
LocatorProtocolList = ^(LocatorData,3)
NameServiceKeyName = "Software\Microsoft\Rpc\NameService"
SecurityKeyName = "Software\Microsoft\Rpc\SecurityService"
SecurityChoiceList = ^(SecurityData,1)
SecurityDllList = ^(SecurityData,2)
[LocatorData]
LD_1 = "LocatorWINNT", "\pipe\locator", "ncacn_np"
LD_2 = "LocatorDCE", "", "ncacn_ip_tcp"
[SecurityData]

```

```

SD_1 = "SecurityWINNT","security.dll"
SD_2 = "SecurityDCE","dcesec.dll"
[UpgradeMapConstants]
DLLClientList = { "rpcltc3.dll", "rpcltc6.dll", "rpcltc5.dll", "rpcltc7.dll",
"rpcdgc3.dll" }
DLLServerList = { "rpclts3.dll", "rpclts6.dll", "rpclts5.dll", "rpclts7.dll",
"rpcdgs3.dll" }
[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 $(LanguageList)
        Return STATUS_NOLANGUAGE
    endif
    set-subst LF = "\n"
    Debug-Output "OEMNSVRP.INF: active"
    read-syms GeneralConstants
    read-syms FileConstants
    read-syms UpgradeMapConstants
    detect date
    read-syms DialogConstants$(!STF_LANGUAGE)
    read-syms $(Option)Constants$(!STF_LANGUAGE)
    read-syms FileConstants$(!STF_LANGUAGE)
    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) == bind
    set StartLabel = bindingadapter
else-Ifstr(i) $(!NTN_InstallMode) == configure
    set StartLabel = configureadapter
else
    set StartLabel = installadapter
endif
set from = $(fatal)
set to = $(fatal)
goto $(StartLabel)
configureadapter = +
ifstr(i) $(Option) == "RPCBANYAN"
    Shell $(UtilityInf),RegistryErrorString,CANNOT_CONFIGURE_SOFTWARE
    ifint $($ShellCode) != $(!SHELL_CODE_OK)
        Debug-Output "OEMNSVBH.INF: ShellCode error: cannot get an error
string."
        goto ShellCodeError
    endif
    set Error = $($R0)
    Set CommonStatus = STATUS_USERCANCEL
    Set Severity = STATUS
    set from = "end"
    set to = "end"
    goto nonfatal
endif
set CommonStatus = STATUS_REBOOT
OpenRegKey $(!REG_H_LOCAL) "" $(NameServiceKeyName) $(MAXIMUM_ALLOWED)
KeyNameService
Ifstr(i) $(KeyNameService) == ""
    Set RegistryErrorIndex = UNABLE_ACCESS_SOFTWARE_REG
    Goto fatalregistry
Endif
Set OldVersionExisted = $(TRUE)
GetRegValue $(KeyNameService),"Protocol", ProtocolNameValue
Ifint $(RegLastError) != $(!REG_ERROR_SUCCESS)
    CloseRegKey $(KeyNameService)
    Set RegistryErrorIndex = UNABLE_ACCESS_SOFTWARE_REG
    Goto fatalregistry
Endif
OpenRegKey $(!REG_H_LOCAL) "" $(SecurityKeyName) $(MAXIMUM_ALLOWED) KeySecurity
Ifstr $(KeySecurity) == ""
    Set RegistryErrorIndes = UNABLE_ACCESS_SOFTWARE_REG
    Goto fatalregistry
Endif
GetRegValue $(KeySecurity), "DefaultProvider", ProviderNameValue
Ifint $(RegLastError) != $(!REG_ERROR_SUCCESS)
    CloseRegKey $(KeyNameService)
    CloseRegKey $(KeySecurity)
    Set RegistryErrorIndex = UNABLE_ACCESS_SOFTWARE_REG
    Goto fatalregistry
Endif
GetRegValue $(KeySecurity), "11", DCEProviderDllValue
read-syms SecurityNames$(!STF_LANGUAGE)
Ifstr $(DCEProviderDllValue) == ""
    Set SecurityChoiceList = {"SecurityWINNT"}
    Set SecurityDllList = {"security.dll"}
    Set SecurityListboxNames = {$(SecName1)}

```

```

        Set DCEProviderDll = ""
    Else
        Set DCEProviderDll = *($(DCEProviderDllValue),4)
        Set SecurityChoiceList = {"SecurityWINNT","SecurityDCE"}
        Set SecurityDllList = {"security.dll",$(DCEProviderDll)}
        Set SecurityListboxNames = {$(SecName1),$(SecName2)}
    Endif
    Debug-Output "OEMNSVRP.INF: Configure Name Services"
configureretry = +
    read-syms LocatorNames$(!STF_LANGUAGE)
    Set LocatorListboxNames = {$(LocName1),$(LocName2)}
    Set ProtocolName = *($(ProtocolNameValue),4)
    Set LocatorIndex = ~($(LocatorProtocolList),$(ProtocolName))
    Set LocatorInUse = *($(LocatorChoiceList),$(LocatorIndex))
    Set LocatorChosen = $(LocatorInUse)
    Set LocatorChosenName = *($(LocatorListboxNames),$(LocatorIndex))
    GetRegValue $(KeyNameService),"NetworkAddress", NetAddrValue
    Set NetworkAddress = *($(NetAddrValue),4)
    Set NetworkAddressInUse = $(NetworkAddress)
    Ifstr(i) $(LocatorChosen) == "LocatorWINNT"
        Set NetworkAddressDCE = ""
    Else
        Set NetworkAddressDCE = $(NetworkAddress)
    Endif
    Set SecurityProviderName = *($(ProviderNameValue),4)
    Set SecurityProviderIndex = ~($(SecurityDllList),$(SecurityProviderName))
    Set SecurityProviderInUse = *($(SecurityChoiceList),$(SecurityProviderIndex))
    Set SecurityChosen = $(SecurityProviderInUse)
    Set SecurityChosenName = *($(SecurityListboxNames),$(SecurityProviderIndex))
reinitdialog = +
    Set LocatorLastChosen = $(LocatorChosen)
    Ifstr(i) $(LocatorChosen) == "LocatorWINNT"
        Set NetworkAddress = ""
    Else
        Set NetworkAddress = $(NetworkAddressDCE)
    Endif
    read-syms RpcProviderDlg$(!STF_LANGUAGE)
    ui start "System"
    Ifstr(i) $(DLGEVENT) == "NOTIFY"
        Set LocatorIndex = ~($(LocatorListboxNames),$(Combo1Out))
        Set LocatorChosen = *($(LocatorChoiceList),$(LocatorIndex))
        Set LocatorChosenName = *($(LocatorListboxNames),$(LocatorIndex))
        Debug-Output "OEMNSVRP.INF: NOTIFY: "$(LocatorChosenName)" = "$
(LocatorChosen)
        Set ReInit = YES
        Ifstr(i) $(LocatorChosen) == $(LocatorLastChosen)
            Goto reinitdialog
        Endif
        Ifstr(i) $(LocatorChosen) == "LocatorWINNT"
            Set NetworkAddress = ""
            Set NetworkAddressDCE = $(EditTextOut)
        Endif
        Goto reinitdialog
    Else-Ifstr(i) $(DLGEVENT) != "CONTINUE"
        ui pop 1
        Set CommonStatus = STATUS_USERCANCEL
        Goto configuredone
    Endif
    Set NetworkAddress = $(EditTextOut)

```

```

Set LocatorIndex = ~($(LocatorListboxNames),$(Combo1Out))
Set LocatorChosen = *($(LocatorChoiceList),$(LocatorIndex))
Set LocatorChosenName = *($(LocatorListboxNames),$(LocatorIndex))
Set SecurityProviderIndex = ~($(SecurityListboxNames),$(Combo2Out))
Set SecurityChosen = *($(SecurityChoiceList),$(SecurityProviderIndex))
Set SecurityChosenName = *($(SecurityListboxNames),$(SecurityProviderIndex))
Debug-Output "OEMNSVRP.INF: OK: "$(LocatorChosenName)", "$(LocatorChosen)
Debug-Output "OEMNSVRP.INF: OK: "$(SecurityChosenName)", "$(SecurityChosen)
Ifstr(i) $(LocatorChosen) == "LocatorWINNT"
    Ifstr(i) $(NetworkAddress) != ""
        read-syms NonFatalError2$(!STF_LANGUAGE)
        shell "subroutn.inf" SetupMessage $(!STF_LANGUAGE) "NONFATAL" $
(NonFatal)
    Endif
    Set NetworkAddress = "\\."
Else-ifstr(i) $(NetworkAddress) == ""
    read-syms NonFatalError1$(!STF_LANGUAGE)
    shell "subroutn.inf" SetupMessage $(!STF_LANGUAGE) "NONFATAL" $(NonFatal)
    Goto configureretry
Endif
Ifstr(i) $(SecurityChosen) == "SecurityDCE"
    Ifstr(i) $(DCEProviderDll) == ""
        read-syms NonFatalError3$(!STF_LANGUAGE)
        shell "subroutn.inf" SetupMessage $(!STF_LANGUAGE) "NONFATAL" $
(NonFatal)
    Goto configureretry
    Endif
Endif
Ifstr(i) $(LocatorChosen) == $(LocatorInUse)
    Ifstr(i) $(NetworkAddress) == $(NetworkAddressInUse)
        Ifstr(i) $(SecurityChosen) == $(SecurityProviderInUse)
            ui pop 1
            Set CommonStatus = STATUS_USERCANCEL
            Goto configuredone
        Endif
    Endif
Endif
ui pop 1
Set ProtocolName = *($(LocatorProtocolList),$(LocatorIndex))
Set EndPointName = *($(LocatorEndPointList),$(LocatorIndex))
Set NewValueList = {{Protocol,$(NoTitle),$(!REG_VT_SZ),$(ProtocolName)},+
{NetworkAddress,$(NoTitle),$(!REG_VT_SZ),$
(NetworkAddress)},+
{ServerNetworkAddress,$(NoTitle),$(!REG_VT_SZ),$
(NetworkAddress)},+
{Endpoint,$(NoTitle),$(!REG_VT_SZ),$(EndPointName)}}
Shell $(UtilityInf), AddValueList, $(KeyNameService), $(NewValueList)
Ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVRP.INF: ShellCode error."
    Goto ShellCodeError
Endif
set RegistryErrorIndex = $(%R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    Debug-Output "OEMNSVRP.INF: registry error: add value list."
    Goto fatalregistry
Endif
Set SecurityDllName = *($(SecurityDllList),$(SecurityProviderIndex))
Set NewValueList = {{DefaultProvider,$(NoTitle),$(!REG_VT_SZ),$
(SecurityDllName)}}

```

```

Debug-Output "OEMNSVRP.INF: adding security provider list: "$(SecurityDllName)
Shell $(UtilityInf), AddValueList, $(KeySecurity), $(NewValueList)
Ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVRP.INF: ShellCode error."
    Goto ShellCodeError
Endif
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    Debug-Output "OEMNSVRP.INF: registry error: add value list."
    Goto fatalregistry
Endif
configuredone = +
CloseRegKey $(KeyNameService)
CloseRegKey $(KeySecurity)
Goto successful
installadapter = +
ifstr(i) $(Option) == "RPCBANYAN"
    OpenRegKey $(!REG_H_LOCAL) "" $(RpcBanyanSoftKeyName) $(MAXIMUM_ALLOWED)
KeyProduct
    Ifstr $(KeyProduct) != $(KeyNull)
        CloseRegKey $(KeyProduct)
        Shell $(UtilityInf), VerExistedDlg, $(ProductRPCBanyanTitle),+
            $(ProductVersion)
        ifint $($ShellCode) != $(!SHELL_CODE_OK)
            Debug-Output "ShellCode error: cannot get an error string."
            goto ShellCodeError
        endif
        set CommonStatus = STATUS_USERCANCEL
        goto end
    endif
    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
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
Shell $(UtilityInf), InstallSoftwareProduct, $(Manufacturer), $
(RpcSrvNameBanyan), "oemnsvrp.inf"
Set SoftProductKey = $($R1)
Set SoftNetRuleKey = $($R2)

```

```

        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),$
(ProductRPCBanyanTitle)},+
        {Description,$(NoTitle),$(!REG_VT_SZ),$
(ProductRPCBaynanSoftDescription)},+
        {OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupportRPCBANYAN)}, +
        {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 $(SoftNetRuleKey)
        goto fatalregistry
    endif
    set NewValueList = {{InfoOption,$(NoTitle),$(!REG_VT_SZ),RPCBANYAN}}
    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)
    OpenRegKey $(!REG_H_LOCAL) "" $(RpcSoftKeyName)$(ServerProto) $
(MAXIMUM_ALLOWED) KeyProduct
    Ifstr $(KeyProduct) == $(KeyNull)
        Set RegistryErrorIndex = UNABLE_ACCESS_SOFTWARE_REG
        Goto fatalregistry
    endif
    SetRegValue $(KeyProduct) +
        {"ncacn_vns_spp",$(NoTitle),$(!REG_VT_SZ),"rpclts8.dll" }
    CloseRegKey $(KeyProduct)
    OpenRegKey $(!REG_H_LOCAL) "" $(RpcSoftKeyName)$(ClientProto) $
(MAXIMUM_ALLOWED) KeyProduct
    Ifstr $(KeyProduct) == $(KeyNull)
        Set RegistryErrorIndex = UNABLE_ACCESS_SOFTWARE_REG
        Goto fatalregistry
    endif
    SetRegValue $(KeyProduct) +
        {"ncacn_vns_spp",$(NoTitle),$(!REG_VT_SZ),"rpcltc8.dll" }
    CloseRegKey $(KeyProduct)
    Goto successful
endif
OpenRegKey $(!REG_H_LOCAL) "" $(RpcSrvKeyName) $(MAXIMUM_ALLOWED) KeyProduct
Ifstr $(KeyProduct) != $(KeyNull)
    CloseRegKey $(KeyProduct)
    Shell $(UtilityInf), VerExistedDlg, $(ProductRPCTitle),+
        $(ProductVersion)

```

```

    ifint $($ShellCode) != $(!SHELL_CODE_OK)
        Debug-Output "ShellCode error: cannot get an error string."
        goto ShellCodeError
    endif
    set CommonStatus = STATUS_USERCANCEL
    goto end
endif
StartWait
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
    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
Set RpcSecurityServiceKey = $(KeyNull)
Set RpcKey = $(KeyNull)
OpenRegKey $(!REG_H_LOCAL) "" $(RpcSoftKeyName) $(MAXIMUM_ALLOWED) RpcKey
Ifstr(i) $(RpcKey) == $(KeyNull)
    Debug-Output "OEMNSVRP.INF: Error opening RPC key"
    Set RegistryErrorIndex = UNABLE_ACCESS_SOFTWARE_REG
    Goto fatalregistry
Endif
CreateRegKey $(RpcKey) {"SecurityService",$(NoTitle),GenericClass} "" $
(MAXIMUM_ALLOWED) "" RpcSecurityServiceKey
Ifstr(i) $(RpcSecurityServiceKey) == $(KeyNull)
    OpenRegKey $(!REG_H_LOCAL) "" $(RpcSecurityKeyName) $(MAXIMUM_ALLOWED)
RpcSecurityServiceKey
Endif
Ifstr(i) $(RpcSecurityServiceKey) == $(KeyNull)
    Debug-Output "OEMNSVRP.INF: Registry error creating subordinate RPC keys"
    CloseRegKey $(RpcKey)
    Goto fatalregistry
Endif
Set NewValueList = {{DefaultProvider,$(NoTitle),$(!REG_VT_SZ),$
(MSSecurityDLL)},+
                    {10,$(NoTitle),$(!REG_VT_SZ),$(MSSecurityDLL)},+
                    {DefaultAuthLevel,$(NoTitle),$(!REG_VT_DWORD),0}}

```



```

Shell $(UtilityInf), AddValueList, $(RpcSecurityServiceKey), $(NewValueList)
Ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVRP.INF: ShellCode error."
    CloseRegKey $(RpcSecurityServiceKey)
    goto ShellCodeError
Endif
set RegistryErrorIndex = $($R0)
CloseRegKey $(RpcSecurityServiceKey)
CloseRegKey $(RpcKey)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNSVRP.INF: registry error: add value list."
    goto fatalregistry
Endif
createrpcservices =+
Ifstr(i) $(!STF_PRODUCT) == "LANMANNNT"
    Set RpcLocatorStartType = "serviceauto"
Else
    Set RpcLocatorStartType = "service"
Endif
Shell $(UtilityInf), AddSoftwareComponent, $(Manufacturer), +
    $(RpcSrvNameLocator), +
    $(RpcSrvNameLocator), +
    $(RpcSrvDisplayNameLocator), $(STF_CONTEXTINFNAME), +
    $(RpcSrvExeLocator), $(RpcLocatorStartType), "", $(RpcSrvDepend), "", ""
set RegistryErrorIndex = $($R0)
Ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVRP.INF: ShellCode error creating RPCLOCATOR service: "$
($ShellCode)
    Goto ShellCodeError
Endif
Set SoftProductKey = $($R1)
Set SoftNetRuleKey = $($R2)
CloseRegKey $($R3)
Set RpcLocatorParametersKey = $($R4)
CloseRegKey $($R5)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNSVRP.INF: Registry error: creating RPCLOCATE software
product"
    CloseRegKey $(SoftProductKey)
    CloseRegKey $(SoftNetRuleKey)
    CloseRegKey $(RpcLocatorParametersKey)
    Goto fatalregistry
Endif
Set NewValueList = {{SoftwareType,$(NoTitle),$(!REG_VT_SZ),$(SoftwareType)},+
    {MajorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMajorVersion)},+
    {MinorVersion,$(NoTitle),$(!REG_VT_DWORD),$
(ProductMinorVersion)},+
    {Title,$(NoTitle),$(!REG_VT_SZ),$(ProductRPCTitle)},+
    {Description,$(NoTitle),$(!REG_VT_SZ),$
(ProductSoftwareDescription)},+
    {OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupportRPC)}, +
    {InstallDate,$(NoTitle),$(!REG_VT_DWORD),*($Now),1}}
Shell $(UtilityInf), AddValueList, $(SoftProductKey), $(NewValueList)
Set RegistryErrorIndex = $($R0)
CloseRegKey $(SoftProductKey)

```

```

ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "ShellCode error."
    goto ShellCodeError
endif
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNSVRP.INF: Registry error: add value list to RPCLOCATE
product"
    CloseRegKey $(SoftNetRuleKey)
    CloseRegKey $(RpcLocatorParametersKey)
    goto fatalregistry
endif
Set NewValueList = {{InfoOption,$(NoTitle),$(!REG_VT_SZ),$(Option)}}
Shell $(UtilityInf), AddValueList, $(SoftNetRuleKey), $(NewValueList)
set RegistryErrorIndex = $($R0)
CloseRegKey $(SoftNetRuleKey)
ifint $($ShellCode) != $(!SHELL_CODE_OK)
    CloseRegKey $(RpcLocatorParametersKey)
    Debug-Output "ShellCode error."
    goto ShellCodeError
endif
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    CloseRegKey $(RpcLocatorParametersKey)
    Debug-Output "OEMNSVSV.INF: Registry error: add value list."
    goto fatalregistry
Endif
Set NewValueList = {{ExpirationAge,$(NoTitle),$(!REG_VT_DWORD),3600}}
Shell $(UtilityInf), AddValueList, $(RpcLocatorParametersKey), $(NewValueList)
Ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVRP.INF: ShellCode error."
    goto ShellCodeError
Endif
set RegistryErrorIndex = $($R0)
CloseRegKey $(RpcLocatorParametersKey)
Ifstr(i) $(!STF_PRODUCT) == "LANMANNNT"
    Set AclIndex = 4
Else
    Set AclIndex = 3
Endif
Shell $(UtilityInf), CreateService, $(RpcSrvNameSSP), +
    $(RpcSrvDisplayNameSSP), $(RpcSrvExeSSP), "serviceshare", "", {}, "", ""
Ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVRP.INF: ShellCode error creating RPCSS service: "$
($ShellCode)
    goto ShellCodeError
endif
Set RegistryErrorIndex = $($R0)
CloseRegKey $($R1)
CloseRegKey $($R2)
CloseRegKey $($R3)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    EndWait
    Debug-Output "OEMNSVRP.INF: Registry error: creating RPCSS service"
    Goto fatalregistry
Endif
LibraryProcedure NCPA_RESULT $(!NCPA_HANDLE), CPlSetup $(!STF_HWND),+
    SECURESVC, $(RpcSrvNameLocator), $(AclIndex)
LibraryProcedure NCPA_RESULT $(!NCPA_HANDLE), CPlSetup $(!STF_HWND),+

```

```

        SECURESVC, $(RpcSrvNameRpcss), $(AcIIndex)
LibraryProcedure NCPA_RESULT $(!NCPA_HANDLE), CPLSetup $(!STF_HWND),+
        SECURESVC, $(RpcSrvNameSSP), $(AcIIndex)
EndWait
goto successful
bindingadapter =+
    set Error = "RPC INF file invoked during bindings review."
    goto fatal
removeadapter = +
    ifstr(i) $(Option) == "RPCBANYAN"
        Debug-Output "OEMNSVRP.INF: Remove component RpcBanyan"
        Shell $(UtilityInf), RemoveSoftwareComponent, $(Manufacturer), $
(RpcSrvNameBanyan), FALSE
        OpenRegKey $(!REG_H_LOCAL) "" $(RpcSoftKeyName)$(ServerProto) $
(MAXIMUM_ALLOWED) KeyProduct
        Ifstr $(KeyProduct) == $(KeyNull)
            Set RegistryErrorIndex = UNABLE_ACCESS_SOFTWARE_REG
            Goto fatalregistry
        endif
        DeleteRegValue $(KeyProduct) "ncacn_vns_spp"
        CloseRegKey $(KeyProduct)
        OpenRegKey $(!REG_H_LOCAL) "" $(RpcSoftKeyName)$(ClientProto) $
(MAXIMUM_ALLOWED) KeyProduct
        Ifstr $(KeyProduct) == $(KeyNull)
            Set RegistryErrorIndex = UNABLE_ACCESS_SOFTWARE_REG
            Goto fatalregistry
        endif
        DeleteRegValue $(KeyProduct) "ncacn_vns_spp"
        CloseRegKey $(KeyProduct)
        Goto successful
    endif
    Debug-Output "OEMNSVRP.INF: Remove component "$(RpcSrvNameLocator)
    Shell $(UtilityInf), RemoveSoftwareComponent, $(Manufacturer), $
(RpcSrvNameLocator), FALSE
    Debug-Output "OEMNSVRP.INF: Remove service "$(RpcSrvNameSSP)
    Shell $(UtilityInf), RemoveService, $(RpcSrvNameSSP), "YES"
    goto end
UpgradeSoftware = +
    ifstr(i) $(Option) == "RPCBANYAN"
        OpenRegKey $(!REG_H_LOCAL) "" $(RpcBanyanSoftKeyName) $(MAXIMUM_ALLOWED)
KeyProduct
        Ifstr $(KeyProduct) != $(KeyNull)
            SetRegValue $(KeyProduct) {MajorVersion,$(NoTitle),$(!REG_VT_SZ),$
(ProductMajorVersion)}
            SetRegValue $(KeyProduct) {MinorVersion,$(NoTitle),$(!REG_VT_SZ),$
(ProductMinorVersion)}
            SetRegValue $(KeyProduct) {Description,$(NoTitle),$(!REG_VT_SZ),$
(ProductRPCBanyanSoftDescription)}
            SetRegValue $(KeyProduct) {OperationsSupport,$(NoTitle),$(!
REG_VT_DWORD),$(ProductOpSupportRPCBANYAN)}
            CloseRegKey $(KeyProduct)
        endif
        Goto successful
    endif
    set RpcLocatorService = "SYSTEM\CurrentControlSet\Services\"$
(RpcSrvNameLocator)
    OpenRegKey $(!REG_H_LOCAL) "" $(RpcLocatorService) $(MAXIMUM_ALLOWED)
KeyService
    ifstr $(KeyService) != $(KeyNull)

```

```

        SetRegValue $(KeyService) +
            {"DependOnService",$(NoTitle),$(!REG_VT_MULTI_SZ),$
(RpcSrvDepend) }
        endif
        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),$(!REG_VT_SZ),$
(ProductSoftwareDescription)}
            SetRegValue $(KeyProduct) {OperationsSupport,$(NoTitle),$(!REG_VT_DWORD),$
(ProductOpSupportRPC)}
            OpenRegKey $(!REG_H_LOCAL) "" $(SecurityKeyName) $(MAXIMUM_ALLOWED)
KeySecurity
            Ifstr $(KeySecurity) == ""
                Shell $(UtilityInf), CreateService, $(RpcSrvNameSSP), +
                    $(RpcSrvDisplayNameSSP), $(RpcSrvExeSSP), "serviceshare", +
                    "", {}, "", ""
                Ifint $(ShellCode) != $(!SHELL_CODE_OK)
                    Debug-Output "OEMNSVRP.INF: ShellCode error creating RPCSS service:
"$(ShellCode)
                    goto ShellCodeError
                endif
                Set RegistryErrorIndex = $($R0)
                CloseRegKey $($R1)
                CloseRegKey $($R2)
                CloseRegKey $($R3)
                Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
                    EndWait
                    Debug-Output "OEMNSVRP.INF: Registry error: creating RPCSS service"
                    Goto fatalregistry
                Endif
                OpenRegKey $(!REG_H_LOCAL) "" $(RpcSoftKeyName) $(MAXIMUM_ALLOWED)
RpcKey
                Ifstr(i) $(RpcKey) == $(KeyNull)
                    Debug-Output "OEMNSVRP.INF: Error opening RPC key"
                    Set RegistryErrorIndex = UNABLE_ACCESS_SOFTWARE_REG
                    Goto fatalregistry
                Endif
                CreateRegKey $(RpcKey) {"SecurityService",$(NoTitle),GenericClass}
                "" $(MAXIMUM_ALLOWED) "" KeySecurity
                Ifstr(i) $(KeySecurity) == $(KeyNull)
                    OpenRegKey $(!REG_H_LOCAL) "" $(RpcSecurityKeyName) $
(MAXIMUM_ALLOWED) KeySecurity
                Endif
                CloseRegKey $(RpcKey)
                Ifstr(i) $(KeySecurity) == $(KeyNull)
                    Debug-Output "OEMNSVRP.INF: Registry error creating subordinate RPC
keys"
                    Goto fatalregistry
                Endif
                Set NewValueList = {{DefaultProvider,$(NoTitle),$(!REG_VT_SZ),$
(MSSecurityDLL)},+

```

```

        {10,$(NoTitle),$(!REG_VT_SZ),$(MSSecurityDLL)},+
        {DefaultAuthLevel,$(NoTitle),$(!REG_VT_DWORD),0}}
Shell $(UtilityInf), AddValueList, $(KeySecurity), $(NewValueList)
Ifint $($ShellCode) != $(!SHELL_CODE_OK)
    Debug-Output "OEMNSVRP.INF: ShellCode error."
    CloseRegKey $(KeySecurity)
    goto ShellCodeError
Endif
set RegistryErrorIndex = $($R0)
Ifstr(i) $(RegistryErrorIndex) != NO_ERROR
    Debug-Output "OEMNSVRP.INF: registry error: add value list."
    goto fatalregistry
Endif
Endif
OpenRegKey $(!REG_H_LOCAL) "" $(RpcSoftKeyName)$(ClientProto) $
(MAXIMUM_ALLOWED) KeyClient
Ifstr $(KeyClient) != $(KeyNull)
    EnumRegValue $(KeyClient) ValueList
    ForListDo $(ValueList)
        set ValueItemList = $($)
        Ifcontains(i) *$(ValueItemList),4) in $(DLLClientList)
            set NewValueList = {*($(ValueItemList),1),0,$(!
REG_VT_SZ),"rpc1tccm.dll" }
            SetRegValue $(KeyClient) $(NewValueList)
            Ifint $(RegLastError) != $(!REG_ERROR_SUCCESS)
                Debug-Output "OEMNSVRP.INF: registry error: setting new
client protocol dll name:*($(ValueItemList),1):"$(RegLastError)
                goto fatalregistry
            endif
        endif
    EndForListDo
    CloseRegKey $(KeyClient)
else
    Debug-Output "OEMNSVRP.INF: registry error: opening client protocols
for dll rename"
    goto fatalregistry
endif
OpenRegKey $(!REG_H_LOCAL) "" $(RpcSoftKeyName)$(ServerProto) $
(MAXIMUM_ALLOWED) KeyServer
Ifstr $(KeyServer) != $(KeyNull)
    EnumRegValue $(KeyServer) ValueList
    ForListDo $(ValueList)
        set ValueItemList = $($)
        Ifcontains(i) *$(ValueItemList),4) in $(DLLServerList)
            set NewValueList = {*($(ValueItemList),1),0,$(!
REG_VT_SZ),"rpc1tscm.dll" }
            SetRegValue $(KeyServer) $(NewValueList)
            Ifint $(RegLastError) != $(!REG_ERROR_SUCCESS)
                Debug-Output "OEMNSVRP.INF: registry error: setting new
server protocol dll name:*($(ValueItemList),1):"$(RegLastError)
                goto fatalregistry
            endif
        endif
    Endif
    EndForListDo
    CloseRegKey $(KeyServer)
else
    Debug-Output "OEMNSVRP.INF: registry error: opening server protocols
for dll rename"
    goto fatalregistry
endif

```

```

        endif
        CloseRegKey $(KeySecurity)
        CloseRegKey $(KeyProduct)
    else
        goto fatalregistry
    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
nonfatal = +
    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), "NONFATAL", $(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 = $(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)
[Install-Option]
    set STF_VITAL = ""
    ifstr(i) $(AddCopy) == "YES"
        AddSectionFilesToCopyList Files-$(Option) $(SrcDir) $(!STF_WINDOWSSYSPATH)
    endif
    ifstr(i) $(DoCopy) == "YES"
        set !STF_NCPA_FLUSH_COPYLIST = TRUE
        CopyFilesInCopyList
    endif
    ifstr(i) $(DoConfig) == "YES"
    endif
    Exit
[Install-Update]
    set STF_VITAL      = ""
    set STF_OVERWRITE  = "VERIFYSOURCEOLDER"
    AddSectionFilesToCopyList Files-$(Option) $(SrcDir) $(!STF_WINDOWSSYSPATH)
    exit
[Source Media Descriptions]
    1 = "Windows NT Workstation CD-ROM" , TAGFILE = cdrom_w.40
[Signature]
    FileType = MICROSOFT_FILE
[GetSignature]
    read-syms Signature
    return $(FileType)
[ProductType]
STF_PRODUCT = Winnt
STF_PLATFORM = I386
[Files-Inf]
2, oemsetup.inf,      SIZE=1000, RENAME=$(!UG_Filename)
[Files-RPCBANYAN]
1,RPCLTC8.DLL , SIZE=15120
1,RPCLTS8.DLL , SIZE=17168
[Files-RPCLOCATE]
1,LOCATOR.EXE , SIZE=118032
[LanguagesSupported]
    ENG
[OptionsTextENG]
    RPCLOCATE = "RPC Configuration"
    RPCBANYAN = "RPC support for Banyan"
[LocatorNamesENG]

```

```

LocName1 = "Windows NT Locator"
LocName2 = "DCE Cell Directory Service"
[SecurityNamesENG]
SecName1 = "Windows NT Security Service"
SecName2 = "DCE Security Service"
[FileConstantsENG]
ProCaption = "Windows NT Setup"
ProCancel = "Cancel"
ProCancelMsg = "Windows NT Networking is not correctly installed. "+
               "Are you sure you want to cancel copying files?"
ProCancelCap = "Network Setup Message"
ProText1 = "Copying:"
ProText2 = "To:"
ProductSoftwareDescription = "Remote Procedure Call service enables RPC-
written programs to perform procedures on multiple computers. Includes the endpoint
mapper and other RPC services."
ProductRPCTitle = "RPC Configuration"
ProductRPCBanyanTitle = "RPC support for Banyan"
ProductRPCBaynanSoftDescription = "RPC support for Banyan"
RpcSrvDisplayNameRpcss = "Remote Procedure Call (RPC) Service"
RpcSrvDisplayNameLocator = "Remote Procedure Call (RPC) Locator"
RpcSrvDisplayNameSSP = "NT LM Security Support Provider"
RpcSuiteDisplayName = "Remote Procedure Call Configuration"
[RPCLOCATEConstantsENG]
FunctionTitle = "RPC Configuration"
ShellCodeErrorTitle = "Error: "$(FunctionTitle)
ShellCodeErrorText = "Shell Code Error."
[RPCBANYANConstantsENG]
FunctionTitle = "RPC support for Banyan"
ShellCodeErrorTitle = "Error: "$(FunctionTitle)
ShellCodeErrorText = "Shell Code Error."
[DialogConstantsENG]
Help = "&Help"
Exit = "Cancel"
OK = "OK"
HelpContext = ""
Continue = "C&ontinue"
Cancel = "C&ancel"
[RpcProviderDlgENG]
Caption = "RPC Configuration"
Edit1Label = "Network &Address:"
Combo1Label = "&Name Service Provider:"
Combo2Label = "&Security Service Provider:"
Help = "&Help"
Continue = "OK"
Cancel = "Cancel"
DlgType = Combination
DlgTemplate = "RPCPROVIDE"
EditTextIn = $(NetworkAddress)
EditFocus = "ALL"
EditTextLim = 50
Combo1List = $(LocatorListboxNames)
Combo1Out = $(LocatorChosenName)
Combo2List = $(SecurityListboxNames)
Combo2Out = $(SecurityChosenName)
ComboListItemsIn = {Combo1List, Combo2List}
ComboListItemsOut = {Combo1Out, Combo2Out}
HelpContext = $(!IDH_DB_RPCLOCATE_INS)
ExitState = "Active"

```



```
NotifyFields = { YES }  
[NonFatalError1ENG]  
NonFatal = "No network address was entered."  
[NonFatalError2ENG]  
NonFatal = "Network address ignored for Windows NT Locator."  
[NonFatalError3ENG]  
NonFatal = "The DCE Security Service is not installed on this computer."
```