

TNTAbstractSecurity.AccessAllowed

Example 1

The following example applies to the component's descendant TNTRegSecurity and demonstrates reading access allowed list.

```
var
  i: integer;
  List: TAccessList;
begin
  NTRegSecurity1.RootKey := HKEY_LOCAL_MACHINE;
  NTRegSecurity1.CurrentPath := 'SOFTWARE\BelCAF';
  List := NTRegSecurity1.AccessAllowed;
  for i := 0 to List.Count - 1 do ListBox1.Items.Add(List[i].UserName);
end;
```

Example 2

The following example applies to the component's descendant TNTShare.

```
var
  i: integer;
  List: TAccessList;
begin
  NTShare1.MachineName := '\\NEPTUNE';
  NTShare1.ShareName := 'Disk C';
  List := NTShare1.AccessAllowed;
  for i := 0 to List.Count - 1 do ListBox1.Items.Add(List[i].UserName);
end;
```

TNTAbstractSecurity.AccessDenied

Example 1

The example applies to component's descendant TNTFileSecurity.

var

i: integer;

List: TAccessList;

begin

FileSecurity1.FileName := 'D:\Program Files\Projects';

List := FileSecurity1.AccessDenied;

for i := 0 **to** List.Count - 1 **do** ListBox1.Items.Add(List[i].UserName);

end;

Example 2

The example applies to component's descendant TNTShare

var

i: integer;

List: TAccessList;

begin

NTShare1.ShareName := 'CD-ROM';

List := NTShare1.AccessDenied;

for i := 0 **to** List.Count - 1 **do** ListBox1.Items.Add(List[i].UserName);

end;

TNTAbstractSecurity.ControlAccess

This example applies to component descendant TFileSecurity and demonstrates how to prohibit an access to the file.

```
FileSecurity1.ControlAccess := false; // erase AccessAllowed and AccessDenied lists.  
FileSecurity1.ControlAccess := true;
```

TNTAbstractSecurity.SystemAudit

The example applies to the component's descendant TNTFileSecurity

```
var
  i: integer;
  List: TAccessList;
begin
  FileSecurity1.FileName := 'D:\Program Files\Projects';
  List := FileSecurity1.SystemAudit;
  for i := 0 to List.Count - 1 do ListBox1.Items.Add(List[i].UserName);
end;
```

TNTEventLog.Active

```
NTEventLogX1.MachineName = "\\PC21"  
NTEventLogX1.SourceName = "Application"  
NTEventLogX1.Active = True
```

TNTEventLog.Add

```
NTEventLogX1.SourceName = "My Application"  
NTEventLogX1.Active = True  
Call NTEventLogX1.Add(EVT_ERROR, "No processor found")  
NTEventLogX1.Active = False
```

TNTEventLog.AddObject

```
NTEventLogX.MachineName = "\\MOON"  
NTEventLogX.Active = True  
  
Set WriteObject = NTEventLogX.WriteObject  
WriteObject.EventId = 1000  
WriteObject.EventType = EVT_ERROR  
WriteObject.EventCategory = 2  
WriteObject.Strings.Text = "The string to create a message from"  
WriteObject.DataBytes.Clear  
For i = 0 To 3  
    WriteObject.DataBytes.Add (i)  
Next i  
  
Call NTEventLogX.AddObject(WriteObject)  
  
NTEventLogX.Active = False
```


TNTEventLog.BackupEventLog

```
' Security section of event log will be saved.  
  NTEventLogX1.SourceName = "Security"  
  NTEventLogX1.Active = True  
  NTEventLogX1.BackupFileName = "c:\backup.evt"  
  NTEventLogX1.BackupEventLog  
  NTEventLogX1.Active = False
```

TNTEventLog.BackupFileName

' open the back-up section of event log

```
NTEventLogX1.BackupFileName = "c:\backup.evt"
```

```
NTEventLogX1.Active = True
```

TNTEventLog.Clear

' The System section of event log will be erased.

```
NTEventLogX1.SourceName = "System"
```

```
NTEventLogX1.Active = True
```

```
NTEventLogX1.Clear
```

```
NTEventLogX1.Active = False
```

TNTEventLog.Count

```
NTEventLogX1.SourceName = "Application"  
NTEventLogX1.Active = True  
MsgBox ("Number of items in Application section: " + Format(NTEventLogX1.Count,  
"00000"))  
NTEventLogX1.Active = False
```

TNTEventLog.Handle

NTEventLogX1.Active = True

GetOldestEventLogRecord(NTEventLogX1.Handle, OldestRecord)

TNTEventLog.IncludeUserName

```
NTEventLogX1.SourceName = "Application"  
NTEventLogX1.Active = True  
NTEventLogX1.IncludeUserName = True  
Call NTEventLogX1.Add(EVT_INFORMATION, Text1.Text)  
NTEventLogX1.Active = False
```

TNTEventLog.Items

```
NTEventLogX1.SourceName = "Application"  
NTEventLogX1.Active = True  
Text1.Text = NTEventLogX1.Items(0)  
NTEventLogX1.Active = False
```

TNTEventLog.Objects

```
NTEventLogX1.SourceName = "Application"  
NTEventLogX1.Active = True  
Text1.Text = "Message has " + Format(NTEventLogX1.Objects(0).DataBytes.Count,  
"@@@@") + " bytes of data"  
NTEventLogX1.Active = False
```


TNTEventLog.OnChange

```
Private Sub NTEventLogX1_OnChange()  
    MsgBox ("Event log has changed")  
End Sub
```

TNTEventLog.SourceName

```
EventLog.SourceName = "Security"  
EventLog.Active = True
```

TNTFileSecurity.FileName

```
FileSecurity1.FileName := 'D:\temp';  
if not (FileSecurity1.FileSystem) = 'NTFS' then Exit;
```

TNTFileSecurity.FileSystem

```
if not (FileSecurity1.FileSystem) = 'NTFS' then Exit;  
FileSecurity1.ControlAccess := true;
```

TNTPersistent.LogonAs

```
NTPrivilegeX1.LogonAs.UserName = "DomainAdmin"  
NTPrivilegeX1.LogonAs.Password = "aBigSecret"  
NTPrivilegeX1.MashineName = "\\PDC_Server"
```

TNTPrivilege.Account

NTPrivilegeX.Account = "Administrators"

S = "Administrators have " + Format(NTPrivilegeX.Privileges.Count, "00") + " privileges"

MsgBox (S)

TNTPrivilege.Accounts

The following example deprives user Everyone of ability to log on locally.

```
NTPrivilegeX.Privilege = "SeInteractiveLogonRight"  
Set PList = NTPrivilegeX.Accounts  
If Not PList.IndexOf("Everyone") = -1 Then  
    PList.Delete (PList.IndexOf("Everyone"))  
End If
```

TNTPrivilege.Enabled

The following example enables the current process to change security attributes of files.

```
NTPrivilegeX.Privilege = "SeSecurityPrivilege"  
NTPrivilegeX.Enabled = True
```


TNTPrivilege.Privilege

The following example prohibits changing time on remote computer:

```
NTPrivilegeX.MachineName = "\\MOON"  
NTPrivilegeX.Privilege = "SeSystemtimePrivilege"  
NTPrivilegeX.Accounts.Clear
```

TNTPrivilege.Privileges

The following example grant the permission of changing time to "Guests" group;

```
NTPrivilegeX.Account = "Guests"  
NTPrivilegeX.Privileges.Add ("SeSystemtimePrivilege")
```

TNTRegSecurity.CurrentPath

```
NTRegSecurity1.MachineName := '\\PC21';  
NTRegSecurity1.Rootkey := HKEY_LOCAL_MACHINE;  
NTRegSecurity1.CurrentPath := 'SOFTWARE\BelCAF\OperaWizard';
```

TNTRegSecurity.KeyOwner

NTResSecurity1.KeyOwner := 'PC21\Administrator'

TNTRegSecurity.RootKey

```
NTRegSecurity1.MachineName := '\\PC21';  
NTRegSecurity1.Rootkey := HKEY_CURRENT_USER;
```

TNTService.ActiveManager

NTServiceX.MachineName = "\\PC21"

NTServiceX.ManagerAccess = M_CONNECT + M_ENUMERATE_SERVICE

NTServiceX.ActiveManager = True

TNTService.ActiveService

Example 1

```
NTServiceX.ActiveManager = True  
NTServiceX.ServiceName = "Schedule"  
NTServiceX.ServiceAccess = S_ALL_ACCESS  
NTServiceX.ActiveService = True  
NTServiceX.ErrorControl = ERROR_NORMAL
```

' affects database

Example 2

```
NTServiceX.ActiveManager = True  
NTServiceX.ServiceName = "Schedule"  
NTServiceX.ServiceAccess = S_ALL_ACCESS  
NTServiceX.ErrorControl = ERROR_NORMAL  
NTServiceX.ActiveService = True
```

' useless

TNTService.BinaryPathName

```
NTServiceX.ActiveManager = True  
NTServiceX.ServiceName = "Schedule"  
NTServiceX.ActiveService = True  
Text1.Text = NTServiceX.BinaryPathName
```


TNTService.ControlService

```
NTServiceX.ManagerAccess = M_CONNECT + M_ENUMERATE_SERVICE  
NTServiceX.ServiceAccess = S_ALL_ACCESS  
NTServiceX.ActiveManager = True  
NTServiceX.ServiceName = "NetLogon"  
NTServiceX.ActiveService = True  
NTServiceX.ControlService (CONTROL_STOP) 'require to stop the service  
NTServiceX.ActiveService = False  
NTServiceX.ActiveManager = False
```

TNTService.Createservice

```
NTServiceX.ManagerAccess = M_CONNECT + M_CREATE_SERVICE
NTServiceX.ServiceType = WIN32_OWN_PROCESS
NTServiceX.ErrorControl = ERROR_NORMAL
NTServiceX.ServiceName = "MyService"
NTServiceX.DisplayName = "My Service"
NTServiceX.BinaryPathName = "c:\temp\myserv.exe"
NTServiceX.StartType = AUTO_START
NTServiceX.ActiveManager = True
NTServiceX.Createservice
NTServiceX.ActiveManager = False
```

TNTService.DBLocked

```
NTServiceX.ManagerAccess = NTServiceX.ManagerAccess + M_LOCK
NTServiceX.ActiveManager = True
NTServiceX.DBLocked = True
NTServiceX.ServiceName = "Schedule"
NTServiceX.ActiveService = True
NTServiceX.StartType = DEMAND_START
NTServiceX.ActiveService = False
NTServiceX.DBLocked = False
NTServiceX.ActiveManager = False
```

TNTService.DeleteService

```
NTServiceX.ServiceAccess = S_ALL_ACCESS  
NTServiceX.ServiceName = "MyService"  
NTServiceX.ActiveManager = True  
NTServiceX.ActiveService = True  
NTServiceX.DeleteService  
NTServiceX.ActiveManager = False
```

TNTService.Dependencies

NTServiceX.Dependencies.Text = Text1.Text

TNTService.DisplayName

```
NTServiceX.ServiceName = "Schedule"  
NTServiceX.ActiveService = True  
NTServiceX.DisplayName = "TimeTable"  
NTServiceX.ActiveService = False
```

TNTService.GetDependentServiceList

```
NTServiceX.ActiveManager = True
NTServiceX.ServiceName = "LanmanWorkstation"
NTServiceX.ServiceAccess = S_ALL_ACCESS
NTServiceX.ActiveService = True
Set List = NTServiceX.GetDependentServiceList(STATE_ACTIVE +
STATE_INACTIVE)
Caption = Format(List.Count, "000") + " dependent services"
NTServiceX.ActiveManager = False
```

TNTService.GetServiceDisplayName

```
NTServiceX.ActiveManager = True  
Caption = NTServiceX.GetServiceDisplayName("PlugPlay")  
NTServiceX.ActiveManager = False
```


TNTService.GetServiceKeyName

NTServiceX.ActiveManager = True

Caption = NTServiceX.GetServiceKeyName("Plug and Play")

NTServiceX.ActiveManager = False

TNTService.GetServiceList

```
NTServiceX.ManagerAccess = M_CONNECT + M_ENUMERATE_SERVICE
NTServiceX.ActiveManager = True
Set List = NTServiceX.GetServiceList(STATE_ACTIVE, PROCESS)
i = List.IndexOf("Spooler")
If i <> -1 Then Caption = "Spooler is active"
```

TNTService.LoadOrder

NTServiceX.LoadOrder = "SpoolerGroup"

TNTService.ManagerAccess

```
NTServiceX.ManagerAccess = M_CONNECT + M_ENUMERATE_SERVICE
```

```
NTServiceX.ActiveManager = True
```

```
' do something
```

```
NTServiceX.ActiveManager = False
```

TNTService.ManagerHandle

Dim S As String

Dim BufLen As Long

S = String(50, " ")

BufLen = 50

NTServiceX.ActiveManager = True

B = GetServiceDisplayNameW(NTServiceX.ManagerHandle, "Schedule", S, BufLen)

TNTService.Password

NTServiceX.Password = "JohnSmith"

NTServiceX.ServiceStartName = "PC21\Administrator"

TNTService.QueryServiceLockStatus

```
Dim S As String

NTServiceX.ManagerAccess = M_QUERY_LOCK_STATUS + M_CONNECT
NTServiceX.ActiveManager = True
Set Status = NTServiceX.QueryServiceLockStatus
If Status.flrLocked > 0 Then
    S = Status.lfLockOwner
    S = "User " + Format(S, "&&&&&&&&&") + " has been locking service database for
"
    S = S + Format(Status.dwLockDuration, "000") + " seconds"
    Text1.Text = S
End If
NTServiceX.ActiveManager = False
```

TNTService.QueryServiceStatus

```
NTServiceX.ActiveManager = True
NTServiceX.ServiceName = "PlugPlay"
NTServiceX.ServiceAccess = S_ALL_ACCESS
NTServiceX.ActiveService = True
Set Status = NTServiceX.QueryServiceStatus
If Status.ControlsAccepted And ACCEPT_PAUSE > 0 Then
    btnStop.Enabled = True
Else: btnStop.Enabled = False
End If
NTServiceX.ActiveManager = False
```


TNTService.ServiceAccess

```
NTServiceX.ServiceAccess = S_ALL_ACCESS  
NTServiceX.ActiveService = True
```

TNTService.ServiceHandle

DeleteService(NTServiceX.ServiceHandle)

TNTService.ServiceName

```
NTServiceX.ActiveManager = True  
NTServiceX.ServiceAccess = S_ALL_ACCESS  
NTServiceX.ServiceName = "PlugPlay"  
NTServiceX.ActiveService = True
```

TNTService.ServiceStartName

NTServiceX.Password = "JohnSmith"

NTServiceX.ServiceStartName = "PC21\Administrator"

TNTService.ServiceType

NTServiceX.ServiceType = NTServiceX.ServiceType+ INTERACTIVE_PROCESS

TNTService.StartType

NTServiceX.StartType = DEMAND_START

TNTService.Startservice

```
NTServiceX.ServiceAccess = S_ALL_ACCESS  
NTServiceX.ServiceName = "MyService"  
NTServiceX.ActiveManager = True  
NTServiceX.ActiveService = True  
NTServiceX.Startservice  
NTServiceX.ActiveManager = False
```

TNTShare.Connections

```
var  
  i: integer;  
  PConnections: TConnectionList;  
begin  
  PConnections := Share1.Connections;  
  for i := 0 to PConnections.Count - 1 do  
    ListBox1.Items.Add(PConnections[i].LocalName);  
end;
```


TNTShare.CurrentUsers

```
Share1.MachineName := '\\PC21';  
ShareName = 'Database';  
try  
Edit1.Text := IntToStr(Share1.CurrentUsers);  
except  
Edit1.Text := 'Access denied';  
end;
```

TNTShare.MaxUsers

```
Share1.MachineName := '\\PC21';  
ShareName = 'Database';  
Share1.MaxUsers := 5;
```

TNTShare.Path

```
Share1.MachineName := '\\PC21';  
ShareName = 'Database';  
try  
Edit1.Text := Share1.Path;  
except  
Edit1.Text := 'Access denied';  
end;
```

TNTShare.Resources

```
var  
  i: integer;  
  PResources: TResourceList;  
begin  
  PResources := Share1.Resources;  
  for i := 0 to PResources.Count - 1 do  
    ListBox1.Items.Add(PResources[i].ShareName);  
end;
```

TNTShare.Sessions

var

i: integer;

PSessions: TSessionList;

begin

PSessions := Share1.Sessions;

for i := 0 **to** PSessions.Count - 1 **do** ListBox1.Items.Add(PSessions[i].ClientName);

end;

TNTShare.ShareComment

```
Share1.MachineName := '\\PC15';  
Share1.ShareName    := 'Transfer'  
Share1.ShareComment := 'For file transfer';
```

TNTShare.ShareName

```
Share1.MachineName := '\\PC25';  
Share1.ShareName := 'COMMON';  
try  
Edit1.Text := Share1.ShareComment;  
except  
Edit1.Text := 'Resource \\PC25\\COMMON does not exist';  
end;
```

TNTShare.Usages

var

i: integer;

PUsages: TUsageList;

begin

PUsages := Share1.Usages;

for i := 0 **to** PUsages.Count - 1 **do** ListBox1.Items.Add(PUsages[i].PathName);

end;

TNTUserMan.GlobalGroupComment

NTUserManX1.MachineName = "\\BELCAF"

NTUserManX1.GlobalGroupName = "DomainGuests"

NTUserManX1.GlobalGroupComment = "All guests of domain BelCAF"

TNTUserMan.GlobalGroupId

```
NTUserManX1.MachineName = "\\MOON"  
NTUserManX1.GlobalGroupName = "PISA"  
Caption = Str(NTUserManX1.GlobalGroupId)
```

TNTUserMan.GlobalGroupMembers

Example 1 - replacement of the list of members

```
NTUserManX1.GlobalGroupName = "Users"  
NTUserManX1.GlobalGroupMembers.Text = "Administrator" + Chr(13) + Chr(10) +  
"Guest"
```

Example 2 - how to add a new member to the local group

```
NTUserManX1.GlobalGroupName = "Users"  
NTUserManX1.GlobalGroupMembers.Add ("BELCAF\Sergei")
```

Example 3 - how to delete a member from the local group

```
NTUserManX1.GlobalGroupName = "Users"  
NTUserManX1.GlobalGroupMembers.Remove (0)
```

TNTUserMan.GlobalGroupName

NTUserManX1.MachineName = "\\MOON"

NTUserManX1.GlobalGroupName = "IMAP4 project"

TNTUserMan.GlobalGroups

Example 1 - replacement of the list of global groups

```
NTUserManX1.MachineName = "\\Server"
```

```
NTUserManX1.GlobalGroups.Text = "Administration" + Chr(13) + Chr(10) + "Internet"
```

Example 2 - how to add a new global group

```
NTUserManX1.MachineName = "\\Server"
```

```
NTUserManX1.GlobalGroups.Add = ("Users")
```

Example 3 - how to delete a global group

```
NTUserManX1.MachineName = "\\Server"
```

```
NTUserManX1.GlobalGroups.Remove (0)
```

TNTUserMan.LocalGroupComment

NTUserManX1.MachineName = "\\PC21"

NTUserManX1.LocalGroupName = "Guests"

NTUserManX1.LocalGroupComment = "All guests of domain BelCAF"

TNTUserMan.LocalGroupMembers

Example 1 - replacement of the list of members

```
NTUserManX1.LocalGroupName = "Users"  
NTUserManX1.LocalGroupMembers.Text = "Administrator" + Chr(13) + Chr(10) +  
"Guest"
```

Example 2 - how to add a new member to the local group

```
NTUserManX1.LocalGroupName = "Users"  
NTUserManX1.LocalGroupMembers.Add ("PC21\Sergei")
```

Example 3 - how to delete a member from the local group

```
NTUserManX1.LocalGroupName = "Users"  
NTUserManX1.LocalGroupMembers.Remove (0)
```

TNTUserMan.LocalGroupName

NTUserManX1.MachineName = "\\MOON"

NTUserManX1.LocalGroupName = "Power users"

TNTUserMan.LocalGroups

Example 1 - how to add a new local group

NTUserManX1.LocalGroups.Add ("MyFamily")

Example 2 - how to delete a local group

NTUserManX1.LocalGroups.Remove (0)

TNTUserMan.MemberOfGlob

Example 1 - replacement of the list of membership

```
NTUserManX1.MachineName = "\\MOON"  
NTUserManX1.Username = "Administrator"  
NTUserManX1.MemberOfGlob.Text = "Administrators" + Chr(13) + Chr(10) + "Power  
users"
```

Example 2 - how to add a membership in a global group PISA

```
NTUserManX1.MachineName = "\\MOON"  
NTUserManX1.Username = "Administrator"  
NTUserManX1.MemberOfGlob.Add ("PISA")
```

Example 3 - how to delete a membership in the first global group

```
NTUserManX1.MachineName = "\\MOON"  
NTUserManX1.Username = "Administrator"  
NTUserManX1.MemberOfGlob.Remove (0)
```

TNTUserMan.UserInfo

```
NTUserManX1.MachineName = ""  
NTUserManX1.Username = "Administrator"  
Text1.Text = NTUserManX1.UserInfo.Comment
```

TNTUserMan.Users

Example 1 - how to add a user

```
NTUserManX1.Users.Add ("Intrnet")
```

Example 2 - how to delete a user

```
NTUserManX1.Users.Remove (0)
```

TNTUserman.MemberOfLocal

Example 1 - replacement of the list of membership

```
NTUserManX1.MachineName = ""  
NTUserManX1.Username = "Administrator"  
NTUserManX1.MemberOfLocal.Text = "Administrators" + Chr(13) + Chr(10) + "Power  
users"
```

Example 2 - how to add a membership in a local group

```
NTUserManX1.Username = "Administrator"  
NTUserManX1.MemberOfLocal.Add ("Users")
```

Example 3 - how to delete a membership in the local group

```
NTUserManX1.Username = "Administrator"  
NTUserManX1.MemberOfLocal.Remove (0)
```

TLogonHours.Days

The following examples are equivalent

Example 1

```
NTUserManX1.Username = "Guest"  
NTUserManX1.UserInfo.LogonHours.Sun.Clear
```

Example 2

```
NTUserManX1.Username = "Guest"  
NTUserManX1.UserInfo.LogonHours.Days(0).Clear
```

TLogonHours.Sun

The following examples are equivalent

Example 1

NTUserManX1.Username = "Guest"

NTUserManX1.UserInfo.LogonHours.Sun.Hours(22) = False

Example 2

NTUserManX1.Username = "Guest"

NTUserManX1.UserInfo.LogonHours.Days(0).Hours(22) = False

TDayHours.SetAll

The following examples are equivalent

Example 1

```
NTUserManX1.UserInfo.LogonHours.Days(0).SetAll
```

Example 2

```
For i = 0 To 23
```

```
NTUserManX1.UserInfo.LogonHours.Days(0).Hours(i) = True
```

```
Next i
```


TDayHours.Hours

NTUserManX1.UserInfo.LogonHours.Wed.Hours(18) = True

TUserInfo.FullName

```
NTUserManX1.Users.Add ("JS")  
NTUserManX1.Username = "JS"  
NTUserManX1.UserInfo.FullName = "John Smith"
```

TUserInfo.Options

```
NTUserManX1.Users.Add ("JS")  
NTUserManX1.Username = "JS"  
NTUserManX1.UserInfo.Options = F_SCRIPT + F_PASSWD_CANT_CHANGE +  
F_DONT_EXPIRE_PASSWD
```

TUserInfo.Privilege

```
NTUserManX1.Username = "JS"  
If NTUserManX1.UserInfo.Privilege <> USR_PRIV_ADMIN Then  
    MsgBox ("The user does not have ADMINISTRATOR permissions")  
End If
```

TUserInfo.Workstations

NTUserManX1.Username = "JS"

NTUserManX1.UserInfo.Workstations = "PC21,PC15"

