

# **Network Resources ActiveX Control for Microsoft® Windows™**

**Copyright © Magneto Software  
All rights reserved**

## Contents

<b>1. Overview .....</b>	<b>3</b>
1.1. <i>Introduction</i> .....	3
1.2. <i>Usage</i> .....	3
1.3. <i>Interface Summary</i> .....	3
1.3.1. <i>_DSKNetResource</i> .....	3
<i>AboutBox</i> .....	3
<i>NetConnectionEnum</i> .....	3
<i>NetFileEnum</i> .....	3
<i>NetSessionEnum</i> .....	3
1.4. <i>Method Summary</i> .....	3
<i>AboutBox</i> .....	3
<i>NetConnectionEnum</i> .....	4
<i>NetFileEnum</i> .....	4
<i>NetFileClose</i> .....	4
<b>2. Methods.....</b>	<b>4</b>
2.1. <i>AboutBox</i> .....	4
2.2. <i>NetConnectionEnum</i> .....	5
2.3. <i>NetFileEnum</i> .....	6
2.4. <i>NetSessionEnum</i> .....	7
2.5. <i>NetRemoteSharedResourceEnum</i> .....	8
2.6. <i>NetShareEnum</i> .....	8
2.7. <i>NetFileClose</i> .....	9
2.8. <i>NetSessionDel</i> .....	10
2.9. <i>NetShareDel</i> .....	11
2.10. <i>NetRemoteSharedResourceClose</i> .....	11
<b>3. Examples.....</b>	<b>12</b>

## 1. Overview

### 1.1. Introduction

The Magneto Software Network Resources ActiveX control (sknetresource.ocx) provides the ability to manage network resources from 23-Bit or 64-Bit applications. It is a lightweight and powerful control that allows developers to retrieve some vital network resources information in real-time.

It can be used from any Windows-based applications development environment, including Visual Studio.

The control comes with documentation, sample code, and working demo programs

### 1.2. Usage

Sknetresource Info ActiveX Control provides the following information:

- Lists all connections made to a shared resource on the local server or all connections established from a particular computer.
- Returns information about some or all-open files on a server.
- Provides information about sessions established on a server.
- Retrieves information about each shared resource on a server or on a particular computer.
- Manages shared files, connections and sessions or a particular computer.

Note: Remote computer resources management for authorized users only.

### 1.3. Interface Summary

#### 1.3.1. \_DSKNetResource

Specifies a collection of logically grouped methods to retrieve and manage network resources.

[AboutBox](#)

[NetConnectionEnum](#)

[NetFileEnum](#)

[NetSessionEnum](#)

[NetRemoteSharedResourceEnum](#)

[NetShareEnum](#)

[NetFileClose](#)

[NetSessionDel](#)

[NetShareDel](#)

[NetRemoteSharedResourceClose](#)

### 1.4. Method Summary

[AboutBox](#)

Display a dialog box with Sknetresources ActiveX control license and version information.

### [NetConnectionEnum](#)

Lists all connections made to a shared resource on the server

### [NetFileEnum](#)

Returns information about some or all open files on a server.

### [NetSessionEnum](#)

Method provides information about sessions established on a server.

### [NetRemoteSharedResourceEnum](#)

Enumerates remote shared resources

### [NetShareEnum](#)

Retrieves information about each shared resource on a server.

### [NetFileClose](#)

Method forces a resource to close.

### [NetSessionDel](#)

Ends a network session between a server and a client

### [NetShareDel](#)

Deletes a share name from a server's list of shared resources, disconnecting all connections to the shared resource

### [NetRemoteSharedResourceClose](#)

Closes remote shared resources

## **2. Methods**

### **2.1. AboutBox**

#### **Summary**

Display a dialog box with Sknetresources ActiveX control license and version information.

#### **Syntax**

```
void AboutBox();
```

#### **Description**

This method could be used to display version license information or to register sknetresources.ocx control.

## Parameters

None.

## Return value:

This function does not return a value.

## 2.2. NetConnectionEnum

### Summary

Lists all connections made to a shared resource on the server

### Syntax

```
long NetConnectionEnum(BSTR strServerName,  
                      BSTR strQualifier,  
                      VARIANT* pvarNetConnectionInfo);
```

### Description

Lists all connections made to a shared resource on the server or all connections established from a particular computer. If there is more than one user using this connection, then it is possible to get more than one structure for the same connection, but with a different user name

### Parameters

*strServerName*

[in] A string that specifies the DNS or NetBIOS name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used.

*strQualifier*

[in] A string that specifies a share name or computer name for the connections of interest. If it is a share name, then all the connections made to that share name are listed. If it is a computer name (for example, it starts with two backslash characters), then [NetConnectionEnum](#) lists all connections made from that computer to the server specified.

*pvarNetConnectionInfo*

[out] Pointer to a variant, containing two-dimensional SAFEARRAY. Each element of this SAFEARRAY is a VARIANT.

*pvarNetConnectionInfo* receives a list of all connections and their properties. The SAFEARRAY dimension definitions are as follows:

[1<sup>st</sup> dimension] index of the connection in the array

[2<sup>nd</sup> dimension] describes connection properties.

1. Connection id

2. Connection type
3. Connection time
4. Connected user

**Return value:**

If the function succeeds, the return value is zero.

### 2.3. NetFileEnum

**Summary**

Returns information about some or all-open files on a server, depending on the parameters specified.

**Syntax**

```
long NetFileEnum(BSTR strServerName,  
                BSTR strUserName,  
                VARIANT* pvarNetFileInfo);
```

**Description**

Returns information about some or all open files on a server, depending on the parameters specified

**Parameters**

*strServerName*

[in] A string that specifies the DNS or NetBIOS name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used.

*strUserName*

[in] A string that specifies the name of the user. If this parameter is specified, *strUserName* serves as a qualifier for the enumeration. The files returned are limited to those that have user names matching the qualifier. If this parameter is NULL, no user-name qualifier is used.

*pvarNetFileInfo*

[out] Pointer to a variant, containing two-dimensional SAFEARRAY. Each element of this SAFEARRAY is a VARIANT.

*pvarNetFileInfo* receives a list of open files. The SAFEARRAY dimension definitions are as follows:

[1<sup>st</sup> dimension] index of the connection in the array

[2<sup>nd</sup> dimension] describes connection properties:

1. Identification number assigned to the resource when it is opened.

2. Access permissions associated with the opening application.
3. The number of file locks on the file.
4. Path of the opened resource, which user (on servers that have user-level security) or which computer (on servers that have share-level security) opened the resource

**Return value:**

If the function succeeds, the return value is zero.

## 2.4. NetSessionEnum

### Summary

Method provides information about sessions established on a server.

### Syntax

```
long NetSessionEnum(BSTR strServerName,  
                   BSTR strUserName,  
                   VARIANT* pvarNetSessionInfo);
```

### Description

Method provides information about sessions established on a server.

### Parameters

*strServerName*

[in] String that specifies the DNS or NetBIOS name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used.

*strUserName*

[in] A string that specifies the name of the user. If this parameter is specified, *strUserName* serves as a qualifier for the enumeration. The files returned are limited to those that have user names matching the qualifier. If this parameter is NULL, no user-name qualifier is used.

*pvarNetSessionInfo*

[out] Pointer to a variant, containing two-dimensional SAFEARRAY. Each element of this SAFEARRAY is a VARIANT.

*pvarNetSessionInfo* receives a list of open sessions files. The SAFEARRAY dimension definitions are as follows:

[1<sup>st</sup> dimension] index of the connection in the array

[2<sup>nd</sup> dimension] describes session properties:

1. User name.
2. Number of connections made during the session.

3. Number of files opened during the session.
4. Number of seconds the session has been idle,

**Return value:**

If the function succeeds, the return value is zero.

## 2.5. NetRemoteSharedResourceEnum

### Summary

Method provides information about sessions established on a server.

### Syntax

```
long NetRemoteSharedResourceEnum (VARIANT* pvarRemoteSharedResourceInfo);
```

### Description

Method provides information about sessions established on a server.

### Parameters

*pvarRemoteSharedResourceInfo*

[out] Pointer to a variant, containing two-dimensional SAFEARRAY. Each element of this SAFEARRAY is a VARIANT.

*pvarNetSessionInfo* receives a list of open sessions files. The SAFEARRAY dimension definitions are as follows:

[1<sup>st</sup> dimension] index of the connection in the array

[2<sup>nd</sup> dimension] describes session properties:

1. The type of the shared resources.
2. Local Path to the shared resource
3. Share name of a resource.
4. String containing optional comment about the resource

**Return value:**

If the function succeeds, the return value is zero.

## 2.6. NetShareEnum

### Summary

Retrieves information about each shared resource on a server.

### Syntax

```
long NetShareEnum(BSTR strServerName,  
                  VARIANT* pvarNetShareInfo);
```



## Description

Retrieves information about each shared resource on a server.

### Parameters

*strServerName*

[in] String that specifies the DNS or NetBIOS name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used.

*pvarNetShareInfo*

[out] Pointer to a variant, containing two-dimensional SAFEARRAY. Each element of this SAFEARRAY is a VARIANT.

*pvarNetShareInfo* receives a list of shared resources. The SAFEARRAY dimension definitions are as follows:

[1<sup>st</sup> dimension] index of the resource in the array

[2<sup>nd</sup> dimension] describes session properties:

1. The share name of a resource.
2. The type of the shared resource.
3. An optional comment about the shared resource.
4. The shared resource's permissions for servers running with share-level security.
5. The maximum number of concurrent connections that the shared resource can accommodate.
6. The local path for the shared resource.
7. The number of current connections to the resource

### Return value:

If the function succeeds, the return value is zero.

## 2.7. NetFileClose

### Summary

The [NetFileClose](#) function forces a resource to close

### Syntax

```
long NetFileClose(BSTR strServerName,  
                 long dwFileId);
```

### Description

The [NetFileClose](#) function forces a resource to close. This function can be used when an error prevents closure by any other means. You should use [NetFileClose](#) with caution because it does not write data cached on the client system to the file before closing the file.

### Parameters

*strServerName*

[in] A string that specifies the DNS or NetBIOS name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used.

*dwFileId*

[in] Specifies the file identifier of the opened resource instance to close.

### Return value:

If the function succeeds, the return value is zero.

## 2.8. NetSessionDel

### Summary

Method ends a network session between a server and a workstation

### Syntax

```
long NetSessionDel(  
    BSTR strServerName,  
    BSTR strClientName,  
    short nSessionId);
```

### Description

Method ends a network session between a server and a workstation

### Parameters

*strServerName*

[in] A string that specifies the DNS or NetBIOS name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used.

*strClientName*

[in] A string that specifies the name of the user whose session is to be terminated. If this parameter is NULL, all users' sessions are to be terminated.

*nSessionId*

identification of the session to delete

### Return value:

If the function succeeds, the return value is zero.

## 2.9. NetShareDel

### Summary

Deletes a share name from a server's list of shared resources, disconnecting all connections to the shared resource

### Syntax

```
long NetShareDel(BSTR strServerName,  
                BSTR strNetName);
```

### Description

Deletes a share name from a server's list of shared resources, disconnecting all connections to the shared resource

### Parameters

*strServerName*

[in] String that specifies the DNS or NetBIOS name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used.

*strNetName*

[in] string that specifies the name of the share to delete..

### Return value:

If the function succeeds, the return value is zero.

## 2.10. NetRemoteSharedResourceClose

### Summary

Deletes a share name from a server's list of shared resources, disconnecting all connections to the shared resource

### Syntax

```
long NetRemoteSharedResourceClose( BSTR strResourceName,  
                                   short bForce);
```

### Description

Deletes a share name from a server's list of shared resources, disconnecting all connections to the shared resource

### Parameters

*strResourceName*

[in] String that specifies the DNS or NetBIOS name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used.

*bForce*

[in] string that specifies the name of the share to delete..

**Return value:**

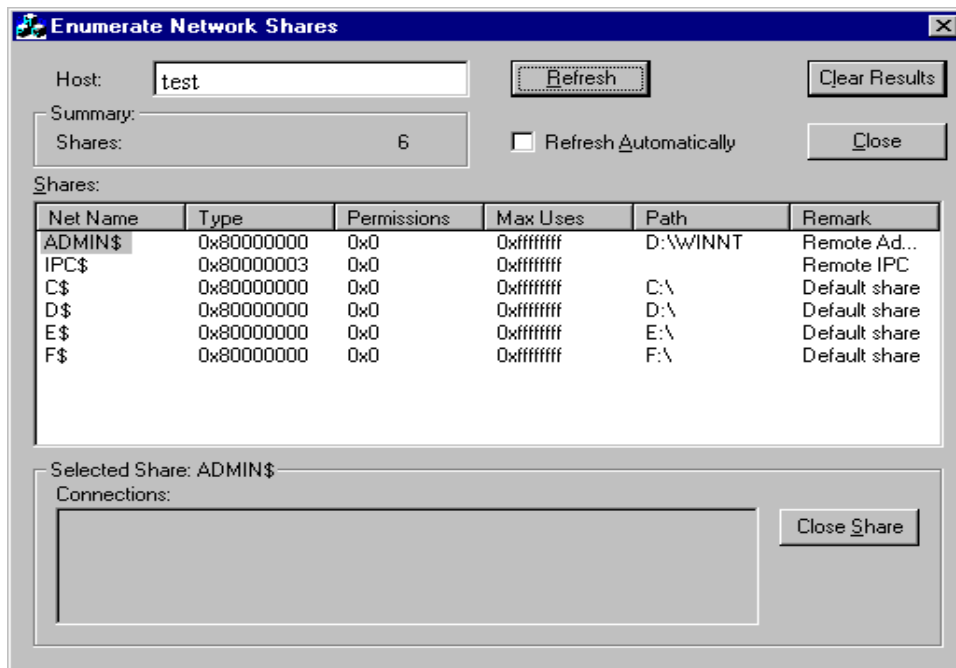
If the function succeeds, the return value is zero.

**3. Examples**

sknetresource ActiveX Control comes with complete documentation, VC++/VB/ASP sample code, and working demo programs distributed during install time.

Demo programs below were written using sknetresource control to retrieve information about each shared resource on a server or client machine.

**Figure 1 Enumerate network shares by using NetShareEnum method**



**Figure 2 Enumerate Remote Shared Resources by using NetRemoteSharedResourceEnum method**

