We're no longer updating this content regularly. Check the Microsoft Product Lifecycle for information about how this product, service, technology, or API is supported.

Return to main site

Filter by title

IISDB_CDT::GetTableDescriptorByIndex method (Windows)
ExitDuration Element
ButtonLocations Element (Child of ScenesMenu6)
Create Keywords for a Binary Index (Windows)
CD3D11_VIEWPORT class (Windows)
Error element (Windows)
ITransportParameters::TrackInformation method (Windows)
ApnID (Windows)
WS_OPAQUE_WINDOWS_CREDENTIAL structure (Windows)
GetBestAlternate function (Windows)
IInputPersonalizationManager::Reserved3 method (Windows)
SIZETToLongPtr function (Windows)
SSIZETToSizeT function (Windows)
SSIZETToUIntPtr function (Windows)
ITablet::GetPropertyMetrics method (Windows)
Message element (Windows)
DeviceControllerFinderHandler function (Windows)
TrackInformation structure (Windows)
D2D1CreateFactory(D2D1_FACTORY_TYPE,REFIID,void**) function (Windows)
XMUSHORT2.XMUSHORT2(const float*) constructor (Windows)
PlaySpeed structure (Windows)
HTML Help API Reference (Windows)
operator XMVECTOR method (Windows)
How to Open a Rooted View of a Junction Point Through a Shortcut File (Windows)
MFCheckContentProtectionDevice function (Windows)
ISpatialAudioMetadataReader::ReadItemCountInFrames method (Windows)
ISpatialAudioRenderStreamForHrtf::Reset method (Windows)
DVDTransitions Element
SceneButtonTFXToken Element
Property Management Helper Functions

… / Previous Versions / Windows / Managed Object Format

Win32_NTEventlogFile class

Article • 08/31/2015

In this article

The **Win32_NTEventlogFile** WMI class represents a logical file or directory of operating system events. The file is also known as the event log.

The following syntax is simplified from Managed Object Format (MOF) code and includes all of the inherited properties. Properties and methods are in alphabetic order, not MOF order.

Syntax

Copy

```
[Provider("MS_NT_EVENTLOG_PROVIDER"), Dynamic]
class Win32_NTEventlogFile : CIM_DataFile
{
  uint32   AccessMask;
  boolean  Archive;
  string   Caption;
  boolean  Compressed;
  string   CompressionMethod;
  string   CreationClassName;
  datetime CreationDate;
  string   CSCreationClassName;
  string   CSName;
  string   Description;
  string   Drive;
  string   EightDotThreeFileName;
  boolean  Encrypted;
  string   EncryptionMethod;
  string   Extension;
  string   FileName;
  uint64   FileSize;
  string   FileType;
  string   FSCreationClassName;
  string   FSName;
  boolean  Hidden;
  datetime InstallDate;
  uint64   InUseCount;
  datetime LastAccessed;
  datetime LastModified;
  string   LogfileName;
  string.  Manufacturer;
  uint32   MaxFileSize;
  string   Name;
  uint32   NumberOfRecords;
  uint32   OverwriteOutDated;
  string   OverWritePolicy;
  string   Path;
  boolean  Readable;
  string   Sources[];
  string   Status;
  boolean  System;
  string   Version;
  boolean  Writeable;
};
```

Members

The **Win32_NTEventlogFile** class has these types of members:

- Methods
- Properties

Methods

The **Win32_NTEventlogFile** class has these methods.

Expand table

Method	Description
BackupEventLog	Saves the specified event log to a backup file.
ChangeSecurityPermissions	Class method that changes the security permissions for the logical file specified in the **Name** property.
ChangeSecurityPermissionsEx	Class method that changes the security permissions for the logical file specified in the **Name** property.
ClearEventLog	Clears the specified event log.
Compress	Class method that compresses the logical file (or directory) specified in the **Name** property.
CompressEx	Class method that uses NTFS compression to compress the logical file (or directory) specified in the **Name** property.
Copy	Class method that copies the logical file or directory specified in the **Name** property to the location specified by the input parameter.
CopyEx	Class method that copies the logical file or directory specified in the **Name** property to the location specified by the *FileName* parameter.
Delete	Class method that deletes the logical file (or directory) specified in the **Name** property.
DeleteEx	Class method that deletes the logical file (or directory) specified in the **Name** property.
GetEffectivePermission	Class method that determines whether the caller has the aggregated permissions specified by the *Permission* argument not only on the file object, but on the share the file or directory resides on (if it is on a share).
Rename	Class method that renames the logical file (or directory) specified in the **Name** property.
TakeOwnerShip	Class method that obtains ownership of the logical file specified in the **Name** property.
TakeOwnerShipEx	Class method that obtains ownership of the logical file specified in the **Name** property.
Uncompress	Class method that uncompresses the logical file (or directory) specified in the **Name** property.
UncompressEx	Class method that uncompresses the logical file (or directory) specified in the **Name** property.

Properties

The **Win32_NTEventlogFile** class has these properties.

AccessMask

Data type: **uint32**

Access type: Read-only

Bitmask that represents the access rights required to access or perform specific operations on the event log file. For bit values, see File and Directory Access Rights Constants .

Note On FAT volumes, the **FULL_ACCESS** value is returned instead, which indicates no security has been set on the object.

Archive

Data type: **boolean**

Access type: Read-only

If **True**, a file that contains Windows events should be archived.

Caption

Data type: **string**

Access type: Read-only

Short description of the object.

Compressed

Data type: **boolean**

Access type: Read-only

If **True**, a file that contains Windows events is compressed.

CompressionMethod

Data type: **string**

Access type: Read-only

Algorithm or tool used to compress the logical file that contains Windows events.

CreationClassName

Data type: **string**

Access type: Read-only

Qualifiers: Key , Dynamic , MaxLen (256) , Dynamic

Name of the first concrete class to appear in the inheritance chain used in the creation of an instance. When used with the other key properties of the class, this property allows all instances of this class and its subclasses to be uniquely identified.

CreationDate

Data type: **datetime**

Access type: Read-only

Date that the file that contains Windows events was created.

CSCreationClassName

Data type: **string**

Access type: Read-only

Class of the computer system.

CSName

Data type: **string**

Access type: Read-only

Name of the computer system.

Description

Data type: **string**

Access type: Read-only

Description of the object.

Drive

Data type: **string**

Access type: Read-only

Drive letter (including colon) of the file that contains Windows events.

Example: "c:"

EightDotThreeFileName

Data type: **string**

Access type: Read-only

DOS-compatible file name for the file that contains Windows events.

Example: "c:\progra~1"

Encrypted

Data type: **boolean**

Access type: Read-only

File that contains Windows events is encrypted.

EncryptionMethod

Data type: **string**

Access type: Read-only

Algorithm or tool used to encrypt the logical file.

Extension

Data type: **string**

Access type: Read-only

File name extension (without the dot) of the file that contains Windows events.

Example: "txt", "mof", "mdb"

FileName

Data type: **string**

Access type: Read-only

File name (without extension) of the file that contains Windows events.

Example: "autoexec"

FileSize

Data type: **uint64**

Access type: Read-only

Size of the file that contains Windows events (in bytes).

For more information about using **uint64** values in scripts, see Scripting in WMI.

FileType

Data type: **string**

Access type: Read-only

File type (indicated by the **Extension** property).

FSCreationClassName

Data type: **string**

Access type: Read-only

Class of the file system.

FSName

Data type: **string**

Access type: Read-only

Name of the file system.

Hidden

Data type: **boolean**

Access type: Read-only

If **True**, a file that contains Windows events is hidden.

InstallDate

Data type: **datetime**

Access type: Read-only

Object is installed. This property does not need a value to indicate that the object is installed.

InUseCount

Data type: **uint64**

Access type: Read-only

Number of "file opens" that are currently active against the file that contains Windows events.

For more information about using **uint64** values in scripts, see Scripting in WMI.

LastAccessed

Data type: **datetime**

Access type: Read-only

Date and time that the file that contains Windows events was last accessed.

LastModified

Data type: **datetime**

Access type: Read-only

Date and time that the file that contains Windows events was last modified.

LogfileName

Data type: **string**

Access type: Read-only

Name of the file that contains Windows events. Standard log file names include: Application, System, and Security.

To return the actual path and file name of the event log (for example, C:\Windows\System32\Config\Sysevent.evt), use the Name property instead.

Manufacturer

Data type: **string.**

Access type: Read-only

Manufacturer from version resource, if one is present.

MaxFileSize

Data type: **uint32**

Access type: Read/write

Maximum size (in bytes) permitted for the file that contains Windows events. If the file exceeds its maximum size, its contents are moved to another file and the primary file is emptied. A value of zero indicates no size limit. WMI retrieves the **Maxsize** value from the Event Log Service registry values.

Although event logs can be sized as large as 4 gigabytes, in practice they should be limited to no more than 300 megabytes. Event logs larger than that can be difficult to analyze because of the number of events contained within the log and because event logs are not optimized for data retrieval.

Name

Data type: **string**

Access type: Read-only

Qualifiers: **Key**, **Dynamic**

Inherited name that serves as a key of a logical file instance that contains Windows events within a file system. Full path names should be provided.

Example: "c:\winnt\system\win.ini"

NumberOfRecords

Data type: **uint32**

Access type: Read-only

Number of records in the file that contains Windows events. This value is determined by calling the Windows function **GetNumberOfEventLogRecords**.

OverwriteOutDated

Data type: **uint32**

Access type: Read/write

Qualifiers: **Units** (Days) , **Dynamic**

Number of days after which an event can be overwritten.

Possible values for **OverwriteOutDated** include the following.

Expand table

Value	Meaning
0 (0x0)	Any record can be overwritten if necessary. If necessary, all existing events in the event log can be overwritten to make room for new events.
1...365	**Windows Server 2003 and Windows XP:** Possible values for **OverwriteOutDated** include the following. Events older than the specified number of days can be overwritten as needed. If the event log does not contain any records older than the value specified, no new events will be recorded until the log has been cleared.
4294967295 (0xFFFFFFFF)	No records can be overwritten. If the log reaches its maximum size, no new events will be recorded until the log has been cleared.

OverWritePolicy

Data type: **string**

Access type: Read-only

Current overwrite policy the Event Log service employs for this log file. Data can be never overwritten, or can be overwritten when necessary or when outdated. When data is outdated depends on the **OverwriteOutDated** value.

Expand table

Value	Meaning
WhenNeeded	The value of **OverwriteOutDated** equals 0 (zero). Any record can be overwritten to make room for new records.
OutDated	The value of **OverwriteOutDated** ranges from 1 to 365. Records older than a specified number of days can be overwritten to make room for new records.
Never	The value of **OverwriteOutDated** equals 4294967295. Old records are never overwritten.

Path

Data type: **string**

Access type: Read-only

Path of the file that contains Windows event. This includes leading and trailing backslashes.

Example: "\windows\system\"

Readable

Data type: **boolean**

Access type: Read-only

If **True**, a file that contains Windows events can be read.

Sources

Data type: **string** array

Access type: Read-only

List of applications that are registered to log into this log file.

Status

Data type: **string**

Access type: Read-only

Current status of the object.

The values are:

"OK"

"Error"

"Degraded"

"Unknown"

"Pred Fail"

"Starting"

"Stopping"

"Service"

"Stressed"

"NonRecover"

"No Contact"

"Lost Comm"

System

Data type: **boolean**

Access type: Read-only

If **True**, a file that contains Windows event is a system file.

Version

Data type: **string**

Access type: Read-only

Version string from version resource if one is present.

Writeable

Data type: **boolean**

Access type: Read-only

If **True**, a file that contains Windows events can be written.

Remarks

The **Win32_NTEventlogFile** class is derived from CIM_DataFile.

Knowing the properties of your event logs can be useful in planning management activities such as backing up and clearing the logs. For example, knowing both the maximum allowable size and the current size of an event log tells you how much space is available in the log. In turn, this helps you decide whether the log needs to be backed up and cleared.

In addition, tracking the number of records in each log is a simple metric that can often trigger alarms regarding potential problems. For example, suppose routine checks of the number of records in an event log show that a specific computer typically records 100 events a day. Today, however, this routine check shows that the computer has recorded 500 events. This might indicate a serious problem that warrants further investigation.

Scripts that retrieve information about the event logs on a computer do not retrieve information about the Security event log unless those scripts include the Security privilege. The ability to manipulate the Security event log is provided by the Manage auditing and security logs user right, which must be explicitly assigned. To manipulate the Security event log, you must include this privilege as part of the GetObject moniker, even if you are an administrator and have been assigned this right by default.

The Security privilege does not grant you the ability to manage auditing and security logs. You must already possess this right (typically assigned through Group Policy), or the script will fail. To access information from or about the Security event log, you must possess the Manage auditing and security logs user right, and the script must include the Security privilege. The following table indicates the results of querying event logs without including the Security privilege.

Expand table

If You Attempt to Access -	You Will Retrieve
All the event logs on a computer	Data for all the event logs except the Security event log
Security event log plus a second event log	Data for only the second event log
Only the Security event log	No data

No special user rights are required to access any of the other event logs on a computer.

Examples

The following VBScript sample retrieves the number of records in and the maximum file size of the Security event log.

VB

Copy

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:" _
 & "{impersonationLevel=impersonate,(Security)}!\\" & _
 strComputer & "\root\cimv2")
Set colLogFiles = objWMIService.ExecQuery _
 ("SELECT * FROM Win32_NTEventLogFile WHERE LogFileName='Security'")
For Each objLogFile in colLogFiles
 Wscript.Echo objLogFile.NumberOfRecords
 Wscript.Echo "Maximum Size: " & objLogfile.MaxFileSize
Next
```

The following VBScript code sample demonstrates how to retrieve the info about the event log files on the local machine from instances of **Win32_NTEventlogFile**.

Note This script only applies to NT-based systems since Win9x does not support event logs.

VB

Copy

```
Set LogFileSet = GetObject("winmgmts:").InstancesOf ("Win32_NTEventLogFile")

for each Logfile in LogFileSet
    WScript.Echo " Log Name: " & Logfile.LogfileName & Chr(13), _
        "Number of Records: " & Logfile.NumberOfRecords & Chr(13), _
        "Max Size: " & Logfile.MaxFileSize & " bytes" & Chr(13), _
        "File name: " & Logfile.Name
next
```

The following Perl code sample demonstrates how to retrieve the info about the event log files on the local machine from instances of **Win32_NTEventlogFile**.

Note This script only applies to NT-based systems since Win9x does not support event logs.

Copy

```
use strict;
use Win32::OLE;

my ( $LogFileSet, $LogFile );

eval { $LogFileSet = Win32::OLE->GetObject("winmgmts:{impersonationLevel=impersc
                        InstancesOf("Win32_NTEventLogFile"); };

unless ($@)
{
    print "\n";
    foreach $LogFile (in $LogFileSet)
    {
        print "Log Name: ", $LogFile->{LogfileName}, "\n";
        if(defined ($LogFile->{NumberOfRecords}))
        {
            print "Number of Records: ", $LogFile->{NumberOfRecords}, "\n";
        }
        else
        {
            print "Number of Records: \n";
        }
        print "Max Size: ", $LogFile->{MaxFileSize}, " bytes", "\n";
        print "File name: ", $LogFile->{Name}, "\n";
        print "\n";
    }
}
else
{
    print STDERR Win32->LastError, "\n";
}
```

Requirements

Expand table

Minimum supported client	Windows XP
Minimum supported server	Windows Server 2003
Namespace	Root\CIMV2
MOF	Ntevt.mof
DLL	Ntevt.dll

See also

Operating System Classes

WMI Tasks: Event Logs

© Microsoft 2024