DiskView Professional User’s Guide

Scripting with the FSInfo Application Programming Interface
The DiskView FSInfo software component maintains and provides file, folder and disk related information. The component can provide fast lookup of folder sizes given their names, as well as the proportion of the folder size relative to its parents size. This document describes the methods and properties of the DiskView FSInfo component for use with DiskView Professional. The DiskView FSInfo API can be used to write custom scripts for disk management and usage analysis.
DiskView FSInfo Methods and Properties

AddToMap .................................................................................................................................3
CancelScan .................................................................................................................................4
ClearMap .....................................................................................................................................5
GenerateMarkup ............................................................................................................................6
GenerateXML .................................................................................................................................7
GetAllDriveInfo ..........................................................................................................................9
GetCurrentScanInfo ....................................................................................................................10
GetFolderWeight ..........................................................................................................................11
GetScannedSizeAsString ...........................................................................................................12
LoadMap ....................................................................................................................................13
LookupFolderSizeAsString .........................................................................................................14
MapSize ......................................................................................................................................15
RescanLocation ...........................................................................................................................16
SaveMapToFile ............................................................................................................................17
**AddToMap**

Scans the specified folder or network share, and adds information on disk usage to an in-memory hash-map for rapid lookup.

**COM Signature**

HRESULT AddToMap([in] BSTR folderPath);

**VBScript Syntax**

Fsi.AddToMap (sFolderPath)

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fsi</td>
<td>Required. An object expression that evaluates to an FSInfo object.</td>
</tr>
<tr>
<td>sFolderPath</td>
<td>Required. String value indicating Folder path.</td>
</tr>
</tbody>
</table>

**Remarks**

The folder path should be in the form 'C:/Program Files/Microsoft' or '\server\share". 

**Example (VBScript)**

Dim fInfo,
Set fInfo = WScript.CreateObject("DirInfo.FSInfo")
fInfo.AddToMap("C:")
fInfo.AddToMap("\server1\sharename")
### CancelScan

Cancels an ongoing scan. CancelScan method is typically called to interrupt an ongoing scanning task started with the AddToMap method.

#### COM Signature

HRESULT CancelScan(void);

#### VBScript Syntax

Fsi.CancelScan()

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fsi</td>
<td>Required. An object expression that evaluates to an FSInfo object.</td>
</tr>
</tbody>
</table>

**Remarks:** This method is applicable for Visual Basic and C++.

**Example (Visual Basic)**

```vbnet
Private Sub cmdCancel_Click()

    Dim lRet As Long
    If Not StrComp(cmdCancel.Caption, "Finish", vbTextCompare) = 0 Then
        lRet = MsgBox("Are you sure you want to cancel the DiskView Analyzer process?", vbYesNo, "DiskView Analyzer")
        If lRet = vbYes Then
            bCancelled = True
            fsi.CancelScan
            Unload Me
        ElseIf lRet = vbNo Then
            Exit Sub
        End If
    Else
        Unload Me
    End If

End Sub
```
**ClearMap**

Clears the in-memory hash-map containing disk usage information.

**COM Signature**

HRESULT ClearMap(void);

**Syntax**

Fsi.ClearMap()

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Fsi</em></td>
<td>Required. An object expression that evaluates to an <code>FSInfo</code> object.</td>
</tr>
</tbody>
</table>

*Remarks:* ClearMap method can be called to re-initialize the hash-map.

**Example (VBScript)**

```vbnet
Dim finfo, mybox, str
Set finfo = WScript.CreateObject("DirInfo.FSInfo")
finfo.LoadMap()

str = finfo.MapSize
mybox = MsgBox(str, 64, "Number of entries in hash-map")

'Clear the map - MapSize will become 0
finfo.ClearMap()
str = finfo.MapSize
mybox = MsgBox(str, 64, "Number of entries after clearing the hash-map")
```
**GenerateMarkup**

Generates disk usage reports for the folder or network share specified. The report can be generated in HTML, XML or a custom format and is the result of an XSL transformation based on the current XSL stylesheet. If a custom stylesheet has not been defined, this method generates HTML reports showing Pie or Bar Charts of the disk usage.

**COM Signature**

HRESULT GenerateMarkup([in] BSTR folderPath, [out, retval] BSTR * markupStr);

**VBScript Syntax**

```vbnet
markup = Fsi.GenerateMarkup(sFolderPath)
```

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Fsi</em></td>
<td>Required. An object expression that evaluates to an FSInfo object.</td>
</tr>
<tr>
<td><em>sFolderPath</em></td>
<td>Required. String value indicating Folder path in the form file:///C:/path/to/folder</td>
</tr>
</tbody>
</table>

**Return Value:** String containing disk usage report in HTML (or custom) format.

**Remarks:**

Custom stylesheet to be used can be specified using the Advanced Tab of DiskView Options Dialog. Optionally, the stylesheet can be specified using the following registry keys:

- HKCU/Software/VizExp/UseStyle DWORD(1)
- HKCU/Software/VizExp/MapStyle String("C:\path\to\stylesheet.xsl")

Set the ActiveTab property to 0 to get Pie chart report when using the default DiskView Stylesheet. To get Bar chart report, set ActiveTab property to 2

**Example (VBScript)**

```vbnet
'The following script generates HTML Pie and Bar Chart reports for C:\WINDOWS folder
Set finfo = WScript.CreateObject("DirInfo.FSInfo")
Set fs = CreateObject("Scripting.FileSystemObject")

Set htmlfilepie = fs.CreateTextFile("DiskUsageCPie.html", True)
Set htmlfilebar = fs.CreateTextFile("DiskUsageCBar.html", True)

finfo.LoadMap()
finfo.ActiveTab=0
strpie = finfo.GenerateMarkup("file:///C:/WINDOWS/")
htmlfilepie.Write(strpie)

finfo.ActiveTab=2
strbar = finfo.GenerateMarkup("file:///C:/WINDOWS/")
htmlfilebar.Write(strbar)

htmlfilepie.Close
htmlfilebar.Close
```
**GenerateXML**

Generates XML disk usage reports for the folder or network share specified.

**COM Signature**

HRESULT GenerateXML([in] BSTR folderPath,[out, retval] BSTR * xmlStr);

**VBScript Syntax**

```vb
xml = Fsi.GenerateXML(sFolderPath)
```

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fsi</strong></td>
<td>Required. An object expression that evaluates to an FSInfo object.</td>
</tr>
<tr>
<td><strong>sFolderPath</strong></td>
<td>Required. String value indicating Folder path in the form file:///C:/path/to/folder</td>
</tr>
</tbody>
</table>

**Return Value:**

String containing disk usage report for the specified folder or network share in XML format.

**Remarks:**

XML uses the following tags to indicate file attributes:

- FILE_ATTRIBUTE_HIDDEN  H="1"
- FILE_ATTRIBUTE_ARCHIVE  A="1"
- FILE_ATTRIBUTE_COMPRESSED C="1"
- FILE_ATTRIBUTE_ENCRYPTED E="1"
- FILE_ATTRIBUTE_OFFLINE  O="1"
- FILE_ATTRIBUTE_READONLY  R="1"
- FILE_ATTRIBUTE_REPARSE_POINT RP="1"
- FILE_ATTRIBUTE_sparse_file SP="1"
- FILE_ATTRIBUTE_SYSTEM   S="1"
- FILE_ATTRIBUTE_TEMPORARY T="1"

Additional tags use the following mapping:

- Directory  DIR
- Drives     DRIVE
- Subdirectory DIRNODE TYPE="DIR"
- File       DIRNODE TYPE="FILE"

Attributes for the file system objects include the following tags:

- Time Created  CT
- Time Last Accessed  AT
- Time Last Written  WT
- Total Bytes  TB
- Type of Drive  TS
- Used Bytes  UB
- Free Bytes  FB
- Percentage Filled  P
- File Fragments  FF
Example (VBScript)

'The following script generates XML DiskUsage report for “C:\Documents and Settings” folder
Set finfo = WScript.CreateObject("DirInfo.FSInfo")
Set fs = CreateObject("Scripting.FileSystemObject")

Set xmlfile = fs.CreateTextFile("DiskUsageDocs.xml", True)

finfo.LoadMap()
finfo.ActiveTab=0
xmlstring = finfo.GenerateXML("file://C:/Documents%20and%20Settings/")
xmlfile.Write(xmlstring)

xmlfile.Close
**GetAllDriveInfo**

Returns real-time information about all local and mapped network drives in an XML format.

**COM Signature**

HRESULT GetAllDriveInfo([out, retval] BSTR * allDriveXML);

**VBScript Syntax**

`driveinfo = Fsi. GetAllDriveInfo()`

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Fsi</em></td>
<td>Required. An object expression that evaluates to an <a href="#">FSInfo object</a>.</td>
</tr>
</tbody>
</table>

**Remarks**

The tags used by the XML are described in the documentation for the GenerateXML method.

**Example (VBScript)**

```vbscript
Set finfo = WScript.CreateObject("DirInfo.FSInfo")
str = finfo.GetAllDriveInfo()
mybox = MsgBox(str, 64, "All Drives Info")
```
GetCurrentScanInfo

Returns statistics about an on-going scan. It returns the Returns real-time information about all local and mapped network drives in an XML format.

**COM Signature**

```
HRESULT GetCurrentScanInfo([out, retval] BSTR * scanInfo);
```

**VB Syntax**

```
currentscaninfo = Fsi.GetCurrentScanInfo
```

### Part Description

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Fsi</em></td>
<td>Required. An object expression that evaluates to an <em>FSInfo</em> object.</td>
</tr>
</tbody>
</table>

**Remarks**

This method is The tags used by the XML are described in the documentation for the GenerateXML method.

**Example (Visual Basic)**

```vbnet
Private Sub tmrProgress_Timer()

sStatusStr = Split(fsi.GetCurrentScanInfo(), ",", , vbTextCompare)
sStatusStr(0) = left$(sStatusStr(0), Len(sStatusStr(0)) - 1)
lblFolderName.Caption = CompactedPath(sStatusStr(0), lblFolderName.Width \ Screen.TwipsPerPixelX, Me.hDC)
lblFileFldCount.Caption = Replace(sStatusStr(1), "scanned", ",", , , vbTextCompare)
cCurrentScanned_temp = CCur(fsi.GetScannedSizeAsString)
'If cCurrentDiskSpace = 0 Then
nDiskProgVal = Round((cCurrentScanned_temp / cCurrentDiskSpace) * 100)

If nDiskProgVal > nPct Then
  pbDiskProgress.Value = nPct
Else
  pbDiskProgress.Value = nDiskProgVal
End If

nOverallProgVal = Round(((cCurrentScanned + cCurrentScanned_temp) / cOverallDiskSpace) * 100)

If nOverallProgVal > nPct Then
  pbOverallProgress.Value = nPct
Else
  pbOverallProgress.Value = nOverallProgVal
End If
If bCancelled = True Then
  StopTimer
  Unload Me
  Exit Sub
End If

End Sub
```
**GetFolderWeight**

Returns the proportion of the specified folder’s size to the total used size on disk

**COM Signature**
HRESULT GetFolderWeight([in] BSTR folderPath);

**VBScript Syntax**
Fsi.GetFolderWeight (sFolderPath)

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fsi</td>
<td>Required. An object expression that evaluates to an FSInfo object.</td>
</tr>
<tr>
<td>sFolderPath</td>
<td>Required. String value indicating Folder path.</td>
</tr>
</tbody>
</table>

**Remarks**
The folder path should be in the form "file:///C:/Documents%20and%20Settings/Username"
Returns -2.0 if the folder being queried is not scanned, or not present in the hash-map containing folder size information. This could also be returned if there is a typo in the Folder name, different capitalization or incorrect format of folder path. For root drives(like file:///C:), this method returns the proportion of the occupied drive space.

**Example (VBScript)**
Set finfo = WScript.CreateObject("DirInfo.FSInfo")
finfo.LoadMap()
str = finfo.getFolderWeight("file:///C:/WINDOWS")
mybox = MsgBox(str, 64, "C:"
GetScannedSizeAsString

Returns the total size of all folders scanned so far while a scan is in progress.

**COM Signature**
HRESULT GetScannedSizeAsString([out, retval] BSTR *sizeAsString);

**Visual Basic Syntax**
cCurrentScanned = CCur(fsi.GetScannedSizeAsString)

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fsi</td>
<td>Required. An object expression that evaluates to an FSInfo object.</td>
</tr>
</tbody>
</table>

**Remarks**
The size is returned as a string.

**Example (Visual Basic)**
Dim cCurrentScanned as Currency
cCurrentScanned = CCur(fsi.GetScannedSizeAsString)
**LoadMap**

Loads a previously cached database of file system information from disk. This method hydrates the FSInfo object with file system information from a previous scan. The database is cached when the SaveMapToFile method is called.

**COM Signature**

```csharp
HRESULT LoadMap();
```

**VBScript Syntax**

```vbscript
Fsi.LoadMap()
```

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fsi</td>
<td>Required. An object expression that evaluates to an [FSInfo object].</td>
</tr>
</tbody>
</table>

**Remarks**

This method loads the last successfully scanned database of file system information. DiskView Analyzer writes to the same database as well, so if LoadMap is called after running DiskView Analyzer, it will contain file system information collected by the Analyzer.

**Example (VBScript)**

```vbscript
Set finfo = WScript.CreateObject("DirInfo.FSInfo")
finfo.LoadMap()
str = finfo.MapSize
mybox = MsgBox(str, 64, "Loaded")
```
**LookupFolderSizeAsString**

Returns the total size on disk occupied by a folder and all files and folders contained within it.

**COM Signature**

HRESULT LookupFolderSizeAsString([in] BSTR folderPath, [out, retval] BSTR * sizeAsString);

**VBScript Syntax**

str = Fsi.LookupFolderSizeAsString(“file:///C:/WINDOWS”)

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Fsi</em></td>
<td>Required. An object expression that evaluates to an <strong>FSInfo object</strong>.</td>
</tr>
<tr>
<td><em>sFolderPath</em></td>
<td>Required. String value indicating Folder path.</td>
</tr>
</tbody>
</table>

**Remarks**

The size is returned as a string in units of bytes. The folder path should be in the form “file:///C:/Documents%20and%20Settings/Username”. Returns -2.0 if the folder being queried is not scanned, or not present in the in-memory database containing folder size information.

**Example (VBScript)**

Set finfo = WScript.CreateObject(“DirInfo.FSInfo”)  
finfo.LoadMap()  
str = finfo.LookupFolderSizeAsString(“file:///C:/WINDOWS”)  
mybox = MsgBox(str, 64, “Size of C:\WINDOWS in Bytes”)
**MapSize**
Returns the number of folders for which information is available using the FSInfo object.

**COM Signature**
HRESULT MapSize([out, retval] long *pVal);

**VBScript Syntax**
Fsi.MapSize

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fsi</td>
<td>Required. An object expression that evaluates to an FSInfo object.</td>
</tr>
</tbody>
</table>

**Remarks**
The MapSize is reset to 0 when ClearMap method is called.

**Example (VBScript)**
Option Explicit
On Error Resume Next

Dim finfo, mybox, str
Set finfo = WScript.CreateObject("DirInfo.FSInfo")
finfo.AddToMap("C:"
'finfo.AddToMap("\servername\sharename")
finfo.SaveMapToFile()
mybox = MsgBox (finfo.MapSize, 64, "finfo.MapSize")
**RescanLocation**
Rescans the specified folder or network share, and refreshes information on disk usage accordingly.

**COM Signature**
HRESULT RescanLocation([in] BSTR sFolderPath);

**VBScript Syntax**
Fsi.RescanLocation ("file:///C:/WINDOWS")

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fsi</td>
<td>Required. An object expression that evaluates to an FSInfo object.</td>
</tr>
<tr>
<td>sFolderPath</td>
<td>Required. String value indicating Folder path.</td>
</tr>
</tbody>
</table>

**Remarks**
The folder path should be in the form "file:///C:/Documents%20and%20Settings/Username". This method does not save the results of the re-scan in the cache file. For this, the SaveMapToFile method must be called.

**Example (VBScript)**
Set finfo = WScript.CreateObject("DirInfo.FSInfo")
finfo.LoadMap()
finfo. RescanLocation ("file:///C:/WINDOWS")
**SaveMapToFile**

Persists the in memory database of file system information to a cache file that can be later loaded using the LoadMap method.

**COM Signature**

HRESULT SaveMapToFile();

**VBScript Syntax**

Fsi.SaveMapToFile()

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fsi</td>
<td>Required. An object expression that evaluates to an <em>FSInfo object</em>.</td>
</tr>
</tbody>
</table>

**Example (VBScript)**

Option Explicit
On Error Resume Next

Dim finfo, mybox, str
Set finfo = WScript.CreateObject("DirInfo.FSInfo")
finfo.AddToMap("C:")
finfo.AddToMap("\servername\sharename")
finfo.SaveMapToFile()
mybox = MsgBox (finfo.MapSize, 64, "finfo.MapSize")