

10Tec xDir ActiveX Library 2.1

What's New in the Library

Tags used to classify changes:

- [New] – a totally new feature;
- [Change] – a change in a member functionality or interactive behavior;
- [Fixed] – a fixed bug or solved problem;
- [Removed] – a member was completely removed;
- [Enhancement] – some functionality was enhanced;
- [Optimization] – a feature has speed improvements;
- [Renaming] – a member was renamed;
- [Code-Upgrade] – indicates a change existing code may need when upgrading from the previous version.

v2.10, build 0012 | 2016-Feb-15 (Release of xDir 2.1)

1. [Enhancement] This release of xDir no longer contains code related to the outdated Windows ANSI platforms like Windows 95, 98 and Me, and can be used only on Windows NT platforms. The component code works faster as the check of the OS platform (Windows 9x or NT) has been removed.
2. [New] The new Boolean property **WOW64FSRedirection** has been implemented. It allows you to control whether file system redirection is enabled in 64-bit versions of Microsoft Windows for some special system folders like %windir%\System32.

The default value of this property is True, which means that the redirection is enabled. In this case, if you try to process the folder %windir%\System32 in a 64-bit edition of Windows with xDir, you will get the contents of the folder %windir%\SysWOW64.

For more info about the system redirection, read the File System Redirector topic in MSDN:

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa384187\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa384187(v=vs.85).aspx)

3. [New][Change] xDir implements the new Boolean **TraverseJunctionPoints** property that indicates whether the contents of junction points should be processed. The default value of this property is False, which corresponds to the recommendation by Microsoft (in order to avoid endless circular references and content duplication):

[https://msdn.microsoft.com/en-us/library/windows/desktop/bb968829\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb968829(v=vs.85).aspx)

Note that the previous versions of xDir always enumerated the contents of junction points, and there was no way to turn this behavior off. In this release of xDir you need to enable this feature explicitly by setting the new **TraverseJunctionPoints** property to True.

4. [New] The new **IXDFolderEnumError/IXDPFolderEnumError** interface has been implemented. Its only **Proc** method is called by xDir each time the component cannot enumerate the contents of a folder because of a system error like "Access denied". The method has the following syntax:

```
Sub Proc( _  
    ByVal ParentFullPath As String, _  
    ByVal ItemName As String, _  
    ByVal ItemID As Long, _  
    ByVal SystemErrorCode As Long, _  
    ByVal SystemErrorMessage As String, _  
    ByVal UserValue As Long)
```

Similar to other xDir interfaces, this sub has some standard parameters used in other interfaces of xDir:

- The **ParentFullPath** and **ItemName** parameters are used to pass the location of the folder to your code.
- The **ItemID** parameter is the same parameter passed to the **Proc** sub of the **IXDFoundItemEx** interface and gives you the unique numeric identifier of the problem folder in the current enumeration process.
- The **UserValue** parameter contains the optional user value specified in the **ProcessFolder** call.

The other parameters are specific to this sub and tell you about the problem with the folder:

- The **SystemErrorCode** parameter contains the numeric code of the folder access error (all possible system error codes can be found in the MSDN in the article “System Error Codes” at <https://msdn.microsoft.com/en-us/library/windows/desktop/ms681381%28v=vs.85%29.aspx>).
- The **SystemErrorMessage** contains the equivalent string message for the error code passed in **SystemErrorCode**. Note that this error description is automatically localized as it is provided for the current OS language.

xDir guarantees that the **Proc** method of the **IXDFolderEnumError** interface is called for a folder only after the **Proc** method of the **IXDFoundItem** or **IXDFoundItemEx** interface has been called for it. This means that you can always access the folder already enumerated in **IXDFolderEnumError_Proc** and change its text or place any other error message next to it if xDir could not enumerate its contents. For instance, if you populate a TreeView control with folder structure and want to display folder enumeration errors, your code could look like the following:

```
Private Sub IXDFolderEnumError_Proc(ByVal ParentFullPath As String, ByVal
ItemName As String, ByVal ItemID As Long, ByVal SystemErrorCode As Long,
ByVal SystemErrorMessage As String, ByVal UserValue As Long)

    ' If it is the root folder and we do not include it into enumeration,
    ' we do not have it in the tree:
    If ItemID = 0 Then Exit Sub

    ' Modify the existing node to indicate enumeration error
    ' (xDir guarantees that IXDFoundItemEx_Proc has been already
    ' called for this node):
    Dim oTreeNode As Node
    Set oTreeNode = TreeView1.Nodes.Item("K" & ItemID)
    oTreeNode.Text = oTreeNode.Text + " --> ENUM ERROR: " + SystemErrorMessage

End Sub
```

Note that the **ItemID** parameter can be zero if the root folder isn't included in the enumeration and xDir cannot access it.

5. [Enhancement][Change] If the specified root folder does not exist or cannot be enumerated by another reason (access denied, etc.), the **Proc** method of the new **IXDFolderEnumError/IXDPFolderEnumError** interface is called by xDir and the **SystemErrorCode/SystemErrorMessage** parameters indicate the reason why the enumeration failed.

In the previous builds the **ProcessFolder** method returned -1 in this case to indicate that a problem with accessing the root folder occurred, but no other detailed information about the problem was reported. In the new implementation **ProcessFolder** returns 0, but the **Proc** method of the **IXDFolderEnumError/IXDPFolderEnumError** interface provides you with the detailed information about the problem.

In the Pro version of xDir the **IXDPEnumCompleted_Proc** method is now also called if the enumeration fails because of a problem with accessing the root folder. This is a useful and only way to notify you that xDir has finished its work if the asynchronous enumeration mode is on.

6. [Enhancement][Fixed] The context search algorithm has been rewritten for better performance. The new implementation also solves the problem with processing big files with a size greater than 2Gb – the context search did not work for them in the previous versions of xDir.
7. [New][Code-Upgrade] The two new parameters, **SystemErrorCode** and **SystemErrorMessage**, were added to the **Proc** sub of the **IXDFileOpenError/IXDPFileOpenError** interface called when xDir cannot open a file for context search. These parameters contain the system error code and the corresponding localized string message that tell you the extended description of the problem why the file cannot be read for context search.
8. [Enhancement][Code-Upgrade] The size of a file is passed as a Currency value in the **IXDItemCheckEx/IXDItemCheckEx** and **IXDFoundItemEx/IXDFoundItemEx** interfaces now. In the previous versions of xDir file sizes were passed as Decimal values packed into the parameter of the variant data type, which caused extra performance degradation and may have caused other potential problems with untyped Variant values.

The type of the **SizeFrom** and **SizeTo** properties of the **FileCriteria** object property of the xDir object (the **CFileCriteria** class) has been changed to Currency as well.

9. [Enhancement][Code-Upgrade] The **ByVal/ByRef** modifiers of some parameters of the **Proc** methods in the xDir interfaces have been changed to get the best performance and to have the identical definitions of the corresponding interfaces in both editions of xDir. Now almost all parameters are passed by value; only the parameters that can be changed to control the behavior of xDir (such as **CancelEnum**) are passed by reference.
10. [Renaming] The **NodeKey** and **ParentNodeKey** parameters in **IXDFoundItem/IXDFoundItemEx** interfaces were renamed to **ItemID** and **ParentItemID** respectively as these values can be used not only to build the hierarchy of tree nodes.

The **ItemCount** parameter of the **Proc** sub in the **IXDPEnumCompleted** interface was renamed to **FoundItemCount** to reflect the fact that this is the number of found but not all enumerated items.

The existing source code does not require changes as the binary compatibility has not been broken.

11. [Code-Upgrade] The class name of the xDir component in the standard edition was changed from xDir200_75B4A91C to xDir210Std_10Tec. The class name of the xDir component in the professional edition was changed from xDir200Pro_75B4A91C to xDir210Pro_10Tec.
12. [Fixed] The **Attributes** parameter of the **Proc** method of the **IXDFoundItemEx (IXDPFoundItemEx)** interface contained incorrect values when the **Sort** property of the **xDir (xDirPro)** object was set to **xdstTree**.
13. [Fixed] The **FileSize** parameter of the **Proc** method of the **IXDFoundItemEx (IXDPFoundItemEx)** interface always contained zero for files when the **Sort** property of the **xDir (xDirPro)** object was set to **xdstList**.
14. [Fixed] xDir did not detect the file/folder encryption attribute correctly because of an error with describing the corresponding **FILE_ATTRIBUTE_ENCRYPTED** constant in the early versions of MSDN. The value of the **xdiaEncrypted** item in the **EItemAttributes** enumeration was changed to the correct value (&H4000) to fix this problem.