

ELO DIGITAL OFFICE GMBH

EloixAuto

COM Interface for IndexServer to run ELO client
scripts

W. Imig

12.01.2010

The module EloixAuto provides the same COM interface than the ELO client, with the difference that all calls will be handled by the IndexServer instead of the ELO client.

Content

1	Overview.....	3
2	Important: ELO Client Compatibility	3
3	EloixAuto	3
3.1	Installation.....	3
3.1.1	Prerequisites.....	3
3.1.2	Installation.....	3
3.2	Configuration.....	3
3.2.1	Log file	3
3.2.2	Archives or Profiles.....	4
3.2.3	Reference of Profile Entries.....	4
3.3	Usage	4
3.3.1	CreateObject(„EloixAuto.EloixAutoLocalServer“)	4
3.3.2	CreateObject(„EloixAuto.EloixAuto“)	5
3.3.3	Login	5
4	EloixAutoWS	5
4.1	Installation.....	5
4.1.1	Prerequisites.....	5
4.1.2	Prerequisites of IIS.....	5
4.1.3	Installation.....	6
4.1.4	Manual Installation.....	7
4.2	Configuration.....	7
4.2.1	IndexServer.....	7
4.2.2	EloixAutoWS	7
4.3	Usage	8
5	Supported Functions with Remarks	8

1 Overview

The module EloixAuto provides the same COM interface than the ELO client, with the difference that all calls will be handled by the IndexServer instead of the ELO client.

Currently, only functions which do not require interaction with the user can be used. A list with all available functions can be found as attachment.

Part of the module is the web application EloixAutoWS, which allows the IndexServer running ELO client scripts in the workflow.

2 Important: ELO Client Compatibility

Aim of the development of EloixAuto is to emulate the OLE interface as best as possible. However, a similar behavior cannot be reached for a variety of reasons.

Consequently, it cannot be guaranteed that EloixAuto behaves exactly like the ELO Client in all operational scenarios. Depending on the usage, considerable differences may arise.

As far as known the differences are described in chapter 5 „Supported Functions with Remarks“.

In case there are undocumented differences to the ELO Client, differences will either be corrected in EloixAuto or added as a special feature in the documentation.

3 EloixAuto

3.1 Installation

3.1.1 Prerequisites

- .NET Framework 2.0
- IndexServer 7.00.022 or higher
- Windows Script Control
<http://www.microsoft.com/downloads/details.aspx?FamilyId=D7E31492-2595-49E6-8C02-1426FEC693AC&displaylang=en>

3.1.2 Installation

- Run installation packet SetupEloixAuto.msi

3.2 Configuration

The configuration settings in the registry of Windows should be inserted under: HKEY_LOCAL_MACHINE oder HKEY_CURRENT_USER\Software\ELO Digital\EloixAuto.

The settings under HKEY_CURRENT_USER are optional and supplement the values under HKEY_LOCAL_MACHINE.

3.2.1 Log file

„...\\EloixAuto\LogFile“ specifies in a string the complete path of the report file.

3.2.2 Archives or Profiles

The configuration for accessing the archive includes min. IndexServer URL and archive name and is stated in the following **profile**.

Some of the OLE Automation functions can be invoked without login. The IndexServer does not offer comparable options. Consequently, an account needs to be defined which can be used in this case. The account must contain exactly one profile. This makes it the **default profile**, which is being used for all calls, if no login for a normal account was previously defined. The account does not require specific ELO rights and can be created using for example the user name „Guest“.

Profiles are available in the registry editor under keys:

„...\\EloixAuto\\Profiles“

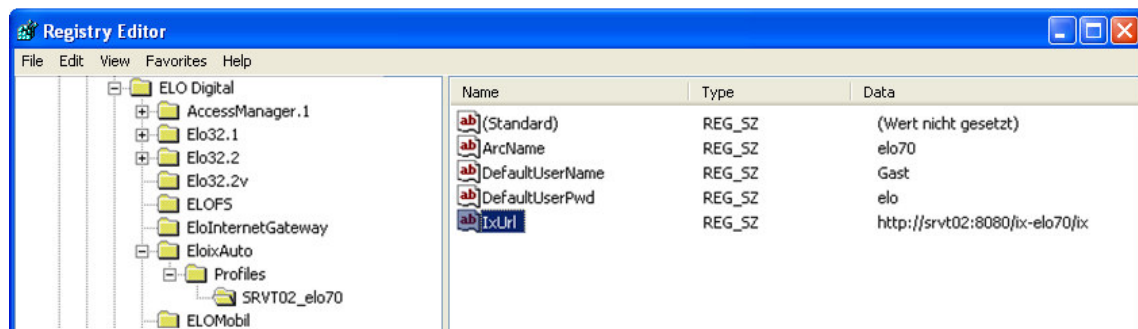


Fig. 1, Example of a profile

As many profiles as required can be created.

3.2.3 Reference of Profile Entries

Name	Description	Required
ArcName	Archive name	Yes
IxUrl	IndexServer URL	Yes
DefaultUserName	This user account is used for function calls, if Login() was not called.	Required for one profile
DefaultUserPwd	Password for DefaultUserName	Required for one profile
SynchronizeIntrayDir	If this value is "true", the server and local intray directories are synchronized when user is logged in or out.	No, default is "false"

3.3 Usage

The EloixAuto components can be instantiated as any other COM component using the CreateObject call in VB scripts. Use as program ID „EloixAuto.EloixAutoLocalServer“ or „EloixAuto.EloixAuto“.

3.3.1 CreateObject(„EloixAuto.EloixAutoLocalServer“)

Using the „EloixAuto.EloixAutoLocalServer“ ID, the COM object starts in a separate process and continues running if the object is not required anymore. Consequently, the initialization of the .NET runtime environment is saved for any further CreateObject calls.

3.3.2 CreateObject(„EloixAuto.EloixAuto“)

Starting the COM object in the address space of the one who invokes, use the ID „EloixAuto.EloixAuto“. However, the one who invokes is a 32-bit program. Furthermore, it should not be an application, which based on an older .NET framework than 2.0.

3.3.3 Login

When logging in, the archive name is not applied but the profile name.

Example:

```
set Elo = CreateObject("EloixAuto.EloixAutoLocalServer")

if (Elo.GetEntryID(-1) < 0) then

    if Elo.Login("Administrator", "elo", "SRVT02_elo70") >= 0 then

        MsgBox "Login successful"

    else

        MsgBox "Login failed"

    end if

end if
```

4 EloixAutoWS

The component EloixAutoWS allows the IndexServer calling VB scripts in the workflow scripts. These scripts actually were written for the ELO client.

Due to the fact that the IndexServer should not depend on the operating system Windows, the scripts are running in an ASP.NET environment in IIS on any other Windows computer in the network. Here, for example, the computer can be used which was selected for FineReader OCR.

4.1 Installation

4.1.1 Prerequisites

- .NET Framework 2.0
- IIS 5.0 or higher
- Windows Script Control
<http://www.microsoft.com/downloads/details.aspx?FamilyId=D7E31492-2595-49E6-8C02-1426FEC693AC&displaylang=en>
- IndexServer 7.00.022 or higher

4.1.2 Prerequisites of IIS

4.1.2.1 Specific settings for 64-bit operating systems

EloixAutoWS is a 32-bit ASP.NET application, which cannot necessarily run on a 64-bit system. The Internet Information Server must be changed to 32-bit. This is valid for all applications in IIS.

Continue as described under the following link:

<http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/405f5bb5-87a3-43d2-8138-54b75db73aa1.mspx?mfr=true>

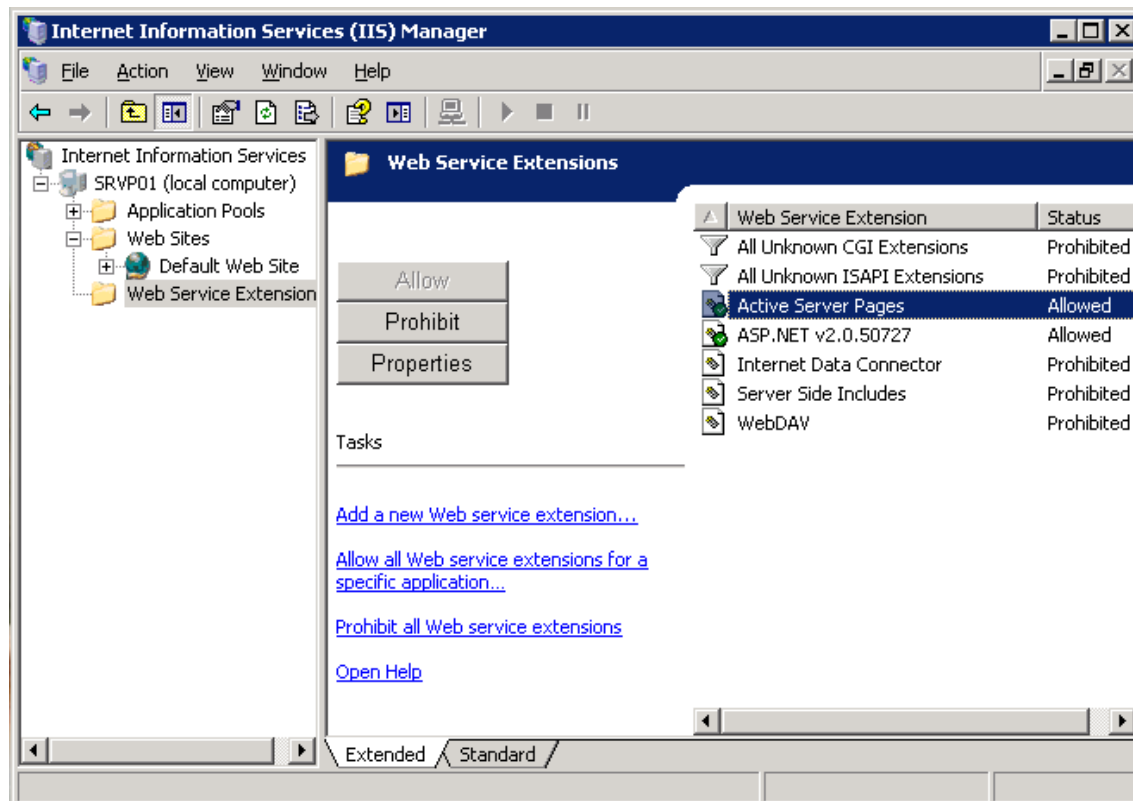
Note: the 32-bit limitation implies that the Windows Script Control is not available for 64-bit systems.

4.1.2.2 Register ASP.NET for IIS

Run the following command in the command line:

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe /i

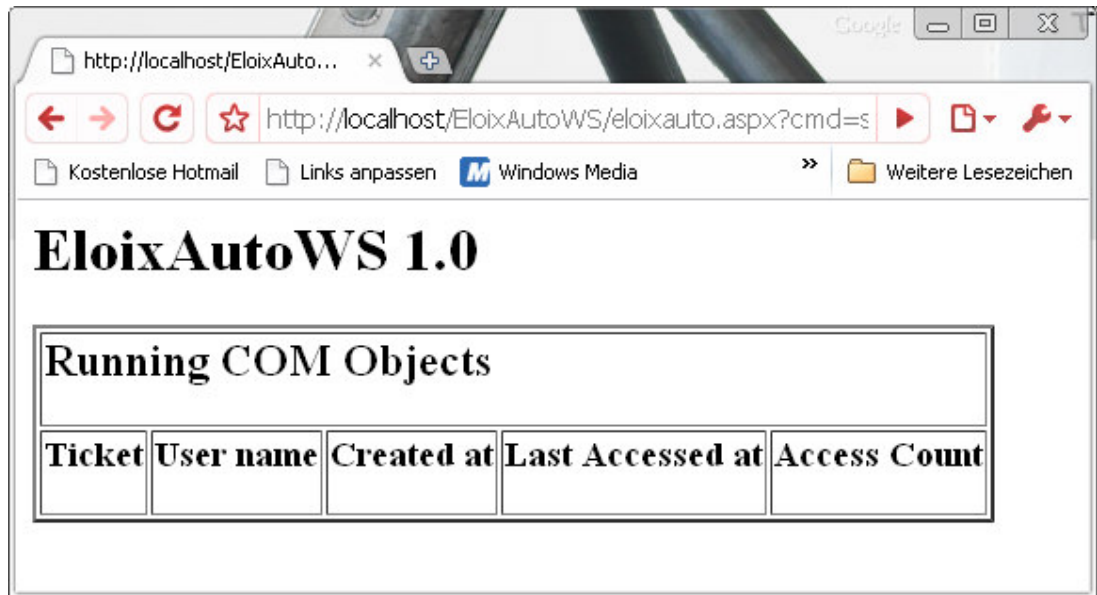
4.1.2.3 Allow Active Server Pages and ASP.NET (IIS 6.0)



4.1.3 Installation

- Run installation packet SetupEloixAutoWS.msi

- Verify if the installation was successfully:
<http://servername/EloixAutoWS/eloixauto.aspx?cmd=status>



4.1.4 Manual Installation

If the installation using the MSI file does not work – for whatever reason- you can follow the manual installation instructions below:

- Create virtual directory using IIS Manager. Alias: „EloixAutoWS“, Directory: „C:\inetpub\wwwroot\EloixAutoWS“, Access rights: Read, Run scripts
- Copy files from „DeployManually“ to the physical directory „C:\inetpub\wwwroot\EloixAutoWS“
- Verify installation: <http://servername/EloixAutoWS/eloixauto.aspx?cmd=status>

4.2 Configuration

4.2.1 IndexServer

The IndexServer receives the address of the EloixAutoWS application through the parameter „eloixAutoWebServiceUrl“.

The EloixAutoWS application requires conversely access to the IndexServer. The IndexServer URL is transferred to the application before running the workflow script. The configuration value “eloixAutoIxUrl” specifies the URL of the IndexServer in view of the EloixAutoWS application.

<input type="checkbox"/>	_ALL	eloixAutoIxUrl	http://192.168.1.155:8084/ix-elo70/ix
<input type="checkbox"/>	_ALL	eloixAutoWebServiceUrl	http://srv02:9100/EloixAutoWS/eloixauto.aspx

4.2.2 EloixAutoWS

The registry value for the log file is evaluated only: „HKEY_LOCAL_MACHINE\Software\ELO Digital\EloixAuto\LogFile“.

This should be a file name under C:\Windows\Temp, to avoid that the web application cannot write the file due to missing access rights.

If VB scripts should run on the same server using the component EloixAuto, then one of the two applications cannot write the log file. Due to this fact, it is possible to create a value of the same name (with another log file) under HKEY_CURRENT-USER. Meaning the user account that runs VB scripts.

4.3 Usage

Please verify if the scripts to be run in the workflow are available in this folder:

"\Administration\IndexServer Scripting Base_ALL\EloixAuto Scripts"

Logical copies – such as from ELOscripts directory – are allowed.

The OLE functions are called as usual via the „ELO.professional“ created object.

EloixAuto overwrites this ID with its own ID before running. Consequently, the client scripts need not to be changed.

The script names of the start and end scripts are stated without file extension as in the ELO client.

5 Supported Functions with Remarks

Function	Remarks
ActivateFlow	Only ActiveUserId supported for parameter FlowUser
ActivateFlowNode	
ActivePostFile	
ActiveUserId	
AddLink	
AddNote	See AddNoteEx3
AddNoteEx	See AddNoteEx3
AddNoteEx2	See AddNoteEx3
AddNoteEx3	Returns -2 instead of -4, if destination object is not a document.
AddPostboxFile	
AddSignature	
AddSw	
AttId	Read-only
CheckFile	
CheckFileHash	Does not show a dialog. Return value 1, if document exists. Return value -2 not supported. Return value >1 not supported.
CheckIn	Does not show a dialog. Calls CheckInEx with empty strings for version and comment. See CheckInEx.
CheckInEx	Returns -9 instead of -1 ... -10
CheckInOutFileName	
CheckInOutObjID	
CheckObjAcl	
CheckOut	Parameter iMode=2 is handled as iMode=1 (no dialog for confirmation).

	On error, EloixAuto returns a string containing a negative integer. The exact values are undefined, as they are undefined of the ELO Client.
CheckUpdate	As there is no user interface, this function does nothing.
CollectChildList	
CollectLinks	
CollectWv	Calls CollectWvEx with parameter Filter=""
CollectWvEx	Only tasks of the current user can be collected. The parameter Filter is only applied on short description, memo, node name and workflow name. Until the IndexServer supports activities completely, only reminders and workflows should be collected. The parameter Prio is interpreted as: 0=Priority A, 1=Priority A and B, 2=Priority A, B and C. Return value -2 is used for all errors returned from IndexServer. Value -3 is returned, if the parameter UserId does not refer to the current user.
ColorInfo	
ColorName	
ColorNo	
ConfirmWorkflowEx	Only tasks of the current user can be forwarded. Returns -2 instead of -5.
CreateCounter	
CreateStructure	
DateToInt	
DebugOut	
DeleteFlow	
DeleteMask	On error, property TextParam contains the IndexServer error message instead of an array of object IDs.
DeleteNote	
DeleteObj	
DeleteSwl	
DeleteWv	Parameter UserOwner is ignored.
DocId	Read-only
DocKind	
DocPath	
DocTPath	
DoInvisibleSearch	
DoSearch	Calls DoSearchEx with parameters SelectObject="" and Mode=1
DoSearchEx	
DoSearchSel	Calls DoSearchEx with paramter Mode=1
EloWindow	Since there is no user interface, this function does nothing.
FindFirstNote	
FindFirstWv	
FindFlowTemplate	
FindNextNote	

FindNextWv	
FindUser	
FindUserEx	
GetArcName	Returns the archive name from the current profile.
GetChildNode	
GetCookie	
GetCounter	
GetCounterList	
GetDocExt	
GetDocFromObj	
GetDocRefComment	
GetDocumentExt	
GetDocumentPath	Calls GetDocumentPathVersion with parameter Version=0
GetDocumentPathName	
GetDocumentPathVersion	Parameter Status, Bit 0 (write access): the document file is copied into a temporary directory and is deleted on logout. Parameter Status, Bit 2 (Fulltext representation file): not supported.
GetDocumentSize	
GetEntryId	Unsupported for reminder or workflow tasks. Unsupported parameter Line=-10 ... -13.
GetEntryName	
GetGuidFromObj	
GetHistDoc	
GetHistObj	
GetLastVersionTimestamp	
GetListEntry	Return value "Id1" is always 0 for search result items (attachment ID is not returned). Return value "Text" is always empty for search result items.
GetMD5Hash	
GetNodeList	
GetObjAttrib	
GetObjAttribFlags	
GetObjAttribKey	
GetObjAttribMax	
GetObjAttribMin	
GetObjAttribName	
GetObjAttribType	
GetObjFromDoc	
GetObjFromDocEx	
GetObjFromGuid	
GetObjRef	
GetPostDir	
GetTreePath	
GetUILanguage	
Gotold	This function prepares the internal buffers so that subsequent calls to GetEntryId return the children of the

	object. Signum of parameter ObjektId is ignored. Returns value -2 instead of -3.
GotoPath	This function prepares the internal buffers so that subsequent calls to GetEntryId return the children of the object.
ImportScript	
InsertDocAttachment	
InsertDocAttachmentEx	
InsertRef	Parameter CheckTypes ignored, types are always checked.
IntToDate	
LoadUserName	
LockObject	Parameters ActionCode=8 and 9 not supported. The timestamp is always updated. ActionCode=8 is mapped to 0, ActionCode=9 is mapped to 1.
Login	Return value is always 0 if login succeeded. Connection numbers are not returned.
LookupDelimiter	
LookupDocType	
LookupHistMD5	
LookupHistMD5Ext	
LookupIndex	Return value is 0 for parameter ObjIndex="". Parameter ObjIndex=*Schlüsseltext (???) is unsupported.
LookupKeyName	
LookupMaskName	
LookupUserName	
MaskFlags	
MaskKey	Returns always -1 like newer versions of ELO Client. Read-only.
MinDocLevel	Returns always 1 like newer versions of ELO Client. Read-only.
MovePostboxFile	
MovePostboxFile2	Parameter sReportParam is ignored. No report entry is written.
MoveToArchive	
MoveToArchiveEx	
NodeAction	
NodeActivateDate	
NodeActivateScript	
NodeAlertFromBegin	
NodeAlertUser	
NodeAlertWait	
NodeAllowActivate	
NodeAvailable	
NodeComment	
NodeDelayDate	
NodeDepartment	
NodeFlowID	

NodeFlowName	
NodeID	
NodeName	
NodeObjectID	
NodePrio	
NodeTerminateDate	
NodeTerminateScript	
NodeType	
NodeUser	
NodeYesNoCondition	
NotelId	
NoteOwner	
NoteParentId	
NoteText	
NoteType	
ObjAcl	
ObjFlags	
ObjGuid	
ObjGuiid	
ObjIDate	
ObjIndex	Value „*Schlüsseltext (???)“ is unsupported.
ObjInfo	This value is not set by an import or an automatic archive synchronisation, because this functions cannot be performed with EloixAuto.
ObjLock	
ObjMainParent	Read-only. Use function InsertRef to move an archive entry.
ObjMaskNo	
ObjMemo	
ObjMemoInfo	
ObjMName	
ObjOwner	Main administration rights are required to modify this value in the database. Otherwise a following UpdateObject will fail.
ObjSDate	
ObjShort	
ObjSIDate	
ObjSReg	
ObjStatus	
ObjSVDate	
ObjType	
ObjTypeEx	
ObjVDate	
ObjXDate	
OpenChildNodes	
PostBoxLineSelected	
PrepareObject	
PrepareObjectEx	

PromoteAcl	Parameter iShowResult is ignored.
ReadColorInfo	
ReadObjMask	
ReadSwl	
ReadUser	
ReadWv	
RemoveRef	
ResetFlowNode	
RunEloScript	COM Program IDs of ELO Client („ELO.professional<n>“, etc.) are automatically replaced by a reference to “this” object.
SaveObject	
SelectAllPostBoxLines	
SelectCurrentNode	
SelectNode	
SelectPostBoxLine	
SelectPostBoxLineEx	
SelectRootNode	
SelectView	
SetCookie	
SetObjAttrib	
SetObjAttribFlags	
SetObjAttribKey	
SetObjAttribMax	
SetObjAttribMin	
SetObjAttribName	
SetObjAttribType	
Status	The status text is written into the log file.
StoreDirect	
StoreKeyword	If the indexing information is missing, the intray documents are stored using the default storage mask for documents. The file name (without extension) becomes the short description. Because no dialogs can be displayed, the function ignores the document, if the archive destination cannot be evaluated based on the indexing information.
StoreMulti	Because no dialogs can be displayed, the function ignores a document, if the archive destination cannot be evaluated based on the indexing information.
TextParam	
UnSelectAllPostBoxLines	
UnSelectPostBoxLine	
UpdateDocument	Calls UpdateDocumentEx with empty strings for parameters Version and Comment.
UpdateDocumentEx	Parameter Status is ignored. Access permissions are always checked. Returns value -2 instead of -3...-6.
UpdateNote	
UpdateObject	
UpdatePostbox	

UpdateSw	
UserFlags	
UserGroup	
UserGroups	
UserId	
UserKeys	
UserName	
UserPwd	
UserTerminate	
Version	
WFComment	
WFFlowName	
WFName	
WFOwner	
WFStatus	
WriteColorInfo	
WriteObjMask	The maximum number of characters in a mask line of type text will be set to 255 (resp. column width of objkeys.okeydata), if SetObjAttribMax was called with 0.
WriteUser	Returns -3 instead of -5.
WriteWv	
WvCreateDate	
WvDate	
WvDesc	
WvDueDate	
WvIdent	
WvListInvalidate	
WvParent	
WvParentType	
WvParentTypeEx	
WvPrio	
WvShort	
WvUserFrom	
WvUserOwner	