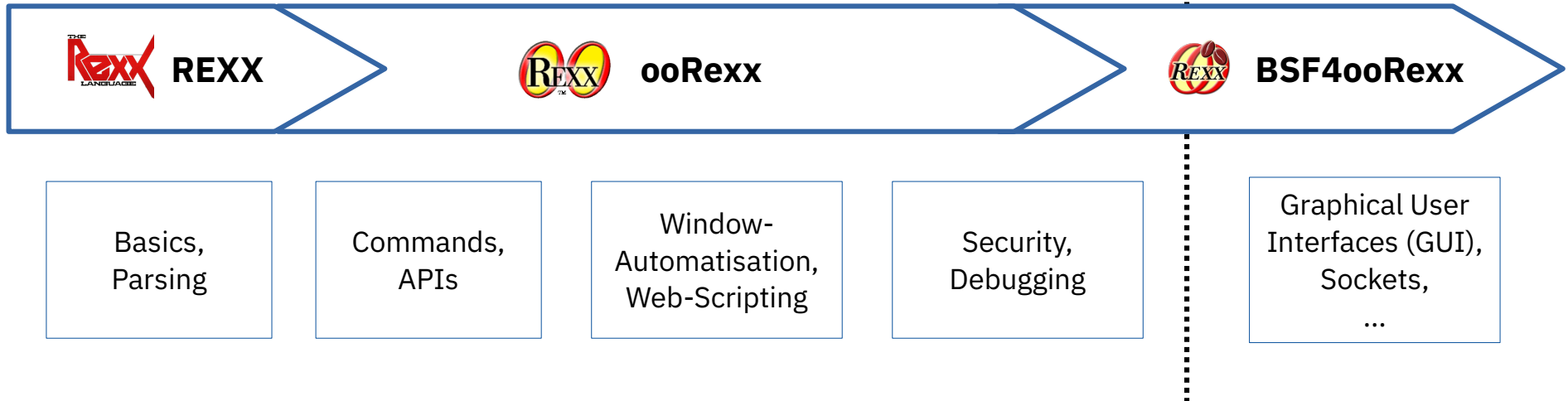


Windows-Automatisation 3

ooRexx Tool "oleinfo": Create HTML OLE Documentation on the Fly

Business Programming 1

Business Programming 2



"oleinfo": Create HTML Documentation



- Information about OLE interfaces sometimes hard to get
- OLE allows querying for information about available
 - *Methods* and their arguments
 - *Properties* and their type
 - *Events* and their arguments
 - *Constants* and their values

if the Windows program publishes this information!

- **OLEObject** can be used to query published information
 - Need to supply the **ProgID** or **CLSID**
 - ooRexx can be used to find all installed **COM** and **OLE** classes with a **ProgID** !

"oleinfo": How to Get Access



- Included in ooRexx 5.1.0 (released: May 2025), location:
 - [oorexx\samples\ole\oleinfo](#)
- For older ooRexx versions (ooRexx < 5.1.0)
 - <https://sourceforge.net/p/oorexx/code-0/HEAD/tree/sandbox/rony/oleinfo/>
 - Zip-archive: <https://wi.wu.ac.at/rgf/wu/lehre/autowin/material/resources>
- Utility programs
 - [listProgIds.rex](#): list all [ProgId](#) (optionally [ClsId](#)) entries in the Windows registry
 - [createOleInfo.rex](#): create HTML documentation for the Windows program with the supplied [ProgId/ClsId](#)
 - [getOleConstants.rex](#): create an ooRexx program with the Window program's constants and which can be loaded via the ooRexx [::requires](#) directive

- Analyzes the Windows registry
- Lists all **COM/OLE** classes with a **ProgId**
 - The result can differ depending on the bitness of ooRexx
 - 32-bit ooRexx will get to see the 32-bit area of the registry
 - 64-bit ooRexx will get to see the 64-bit area of the registry

- Usage (command)

rex **listProgIds** **[0|1]** **[fragment]**

- Optional: "0" (default: show **ProgId** only) or "1" (show **CLSID** in addition)
- Optional: **ProgId** must contain supplied **fragment** (caseless comparison)

rexex listProgIds.rex

Possible output:

32-bit ooRexx (maybe):

```
32-bit #    1/2240: ProgId: [AboveLockAppLaunchCallback]
32-bit #    2/2240: ProgId: [AccClientDocMgr.AccClientDocMgr]
32-bit #    3/2240: ProgId: [AccDictionary.AccDictionary]
32-bit #    4/2240: ProgId: [Access.Application]
... cut ...
32-bit # 2238/2240: ProgId: [xhtmlfile]
32-bit # 2239/2240: ProgId: [XML]
32-bit # 2240/2240: ProgId: [xmlfile]
```

Or 64-bit ooRexx (maybe):

```
64-bit #    1/1902: ProgId: [AADJCSP]
64-bit #    2/1902: ProgId: [ABIT.ABIT]
64-bit #    3/1902: ProgId: [AboveLockAppLaunchCallback]
... cut ...
64-bit # 1900/1902: ProgId: [ZeroConfigService.ZDevice]
64-bit # 1901/1902: ProgId: [ZeroConfigService.ZService]
64-bit # 1902/1902: ProgId: [ZipSend.CourierService]
```

listProgIds.rex, 3



rex `listProgIds.rex explorer`

Output:

32-bit ooRexx (maybe):

```
32-bit # 316/2240: ProgId: [ClassicExplorer.ClassicCopyExt]
32-bit # 317/2240: ProgId: [ClassicExplorer.ExplorerBand]
32-bit # 318/2240: ProgId: [ClassicExplorer.ExplorerBH0]
32-bit # 319/2240: ProgId: [ClassicExplorer.ShareOverlay]
32-bit # 755/2240: ProgId: [InternetExplorer.Application]
32-bit # 1561/2240: ProgId: [Shell.Explorer]
```

Or 64-bit ooRexx (maybe):

```
64-bit # 306/1902: ProgId: [ClassicExplorer.ClassicCopyExt]
64-bit # 307/1902: ProgId: [ClassicExplorer.ExplorerBand]
64-bit # 308/1902: ProgId: [ClassicExplorer.ExplorerBH0]
64-bit # 309/1902: ProgId: [ClassicExplorer.ShareOverlay]
64-bit # 665/1902: ProgId: [InternetExplorer.Application]
64-bit # 1273/1902: ProgId: [Shell.Explorer]
```

createOleInfo.rex, 1



- Creates an **OLEObject** instance from the ProgId or CLSID
- Queries all interfaces for methods, properties, events and constants
- Creates an HTML text file
- Displays the results in the default web browser
- Usage (command)

```
rexx createOleInfo.rex ProgId | CLSID [0|1] [0|1]
```

- ProgId or CLSID (either must be supplied)
- Optional: "0" (default: long format) or "1" (short reference format)
- Optional: "0" (do not display result) or "1" (default: display result in web browser)

- `createOleInfo.rex` can be called from within a ooRexx program
 - The first argument is an `OLEObject` object returned by a Windows program
 - Allows one to document OLE objects for which no `ProgId` may exist

- Usage from within an ooRexx program

```
call createOleInfo.rex OleObject, [Title], [0|1], [0|1]
```

- `OleObject`
- Optional `Title` (an informative string) for the generated HTML document
- Optional: "0" (default: long format) or "1" (short reference format)
- Optional: "0" (do not display result) or "1" (default: display result in web browser)

createOleInfo.rex, 3a



rexex createOleInfo.rex *InternetExplorer.Application*

InternetExplorer.Application		
Definitions from typelib: [SHDocVw] with the brief documentation: [Microsoft Internet Controls] (These published interfaces got retrieved using the .OLEInfo class from ooRexx with the exact name: "REXX-ooRexx_5.0.0(MT)_32-bit 6.05 21 Apr 2022")		
show ☒	16 Method[s]	
No.	Name	Documentation, Argument[s], Return Value
1	ClientToWindow	<i>Converts client sizes into window sizes.</i> arg: # 1: pcx in/out VT_PTR # 2: pcy in/out VT_PTR returns: VT_VOID (NO RETURN VALUE)
2	ExecWB	<i>IoleCommandTarget::Exec</i> # 1: cmdID in VT_USERDEFINED # 2: cmdexeopt in VT_USERDEFINED arg: # 3: [pvaln] in [optional] VT_PTR # 4: [pvaOut] in/out [optional] VT_PTR returns: VT_VOID (NO RETURN VALUE)
3	GetProperty	<i>Retrieve the Associated value for the property vtValue in the context of the object.</i> arg: # 1: Property in VT_BSTR returns: VT_VARIANT
4	GoBack	<i>Navigates to the previous item in the history list.</i> returns: VT_VOID (NO RETURN VALUE)
5	GoForward	<i>Navigates to the next item in the history list.</i> returns: VT_VOID (NO RETURN VALUE)

rex createOleInfo.rex *InternetExplorer.Application*

Internet Explorer			show ☒			14 Read-only Properties		
No.	Name	Documentation, Return Value	No.	Name	Documentation, Return Value	No.	Name	Value
1	ClientToWindow	Converts client sizes into window sizes. arg: # 1: pcx in /out VT_UI4 # 2: pcy in /out VT_UI4 returns: VT_VOID (NO RETURN VALUE)	39	WindowSetTop	Fired when the host window should change its Top coordinate arg: # 1: Top in VT_UI4	1	CSC_NAVIGATEBACK	2
2	ExecWB	IOleCommandTarget::ExecuteCommand # 1: cmdID in VT_UI4 # 2: cmdexect in VT_UI4 # 3: [pvaln] in [optional] VT_UI4 # 4: [pvaOut] in /out [optional] VT_UI4 returns: VT_VOID (NO RETURN VALUE)	40	WindowStateChanged	Fired to indicate that the browser window's visibility or enabled state has changed. arg: # 1: dwWindowStateFlags in VT_UI4 # 2: dwValidFlagsMask in VT_UI4	2	CSC_NAVIGATEFORWARD	1
3	GetProperty	Retrieve the Associated value for the property vtValue arg: # 1: Property in VT_BSTR returns: VT_VARIANT	41			3	CSC_UPDATECOMMANDS	-1
4	GoBack	Navigates to the previous item in the history list. returns: VT_VOID (NO RETURN VALUE)				4	OLECMDEXEPT_DODEFAULT	0
5	GoForward	Navigates to the next item in the history list. returns: VT_VOID (NO RETURN VALUE)				5	OLECMDEXEPT_DONTPROMPTUSER	2
						6	OLECMDEXEPT_PROMPTUSER	1
						7	OLECMDEXEPT_SHOWHELP	3

createOleInfo.rex, 4



rex createOleInfo.rex *InternetExplorer.Application* **1**

InternetExplorer.Application

Definitions from typelib: [[SHDocVw](#)] with the brief documentation: [[Microsoft Internet Controls](#)]

(These published interfaces got retrieved using the .OLEInfo class from ooRexx with the exact name: "REXX-ooRexx_5.0.0(MT)_32-bit 6.05 21 Apr 2022")

16 Method[s]

No.	Name	Argument[s], Documentation
1	ClientToWindow	(pcx <i>VT_PTR</i> in/out , pcy <i>VT_PTR</i> in/out) <i>Converts client sizes into window sizes.</i>
2	ExecWB	(cmdID <i>VT_USERDEFINED</i> , cmdexcopt <i>VT_USERDEFINED</i> , [pvaIn <i>VT_PTR</i>] , [pvaOut <i>VT_PTR</i> in/out]) <i>IOleCommandTarget::Exec</i>
3	<i>VT_VARIANT</i> GetProperty	(Property <i>VT_BSTR</i>) <i>Retrieve the Associated value for the property vtValue in the context of the object.</i>
4	GoBack	<i>Navigates to the previous item in the history list.</i>
5	GoForward	<i>Navigates to the next item in the history list.</i>

- Creates an ooRexx program that contains all published constants
 - Program saves all constants in the `.local` directory with the index `OLE.CONST`
 - Any entry in `.local` or `.environment` can be referenced with an environment symbol
 - `.ole.const` returns therefore the stringtable with the saved OLE constants
 - Sending the name of the constant to `.ole.const` returns its value
 - E.g. `.ole.const~RGBBROWN` may return the value `2763429` for Excel
 - It is possible to require different generated constant programs
 - All constants will always be saved in a stringtable named `.ole.const`
- Usage (command)

```
rexex getOleConstants.rex ProgId | CLSID [filename]
```

 - `ProgId` or `CLSID` (either must be supplied)
 - Optional: `filename` to store the generated Rexx program, otherwise writes to `stdout`

getOleConstants.rex, 2



- Example

```
rexex getOleConstants.rex Excel.Application constExcel.rex
```

- Queries Excel for all its published constants
 - Creates a REXX program that stores all constants into the `.ole.const` stringtable
 - Saves that REXX program in a file named `constExcel.rex`
- Any ooREXX program can then require `constExcel.rex` and access `.ole.const`

```
/* demonstrate how to access an Excel constant */  
say "Excel constant for the color brown:" .ole.const~RgbBrown /* yields: 2763429 */  
  
::requires "constExcel.rex" -- make Excel constants available via .ole.const
```

Possible Output:

```
Excel constant for the color brown: 2763429
```