LEGAL

VERSION HISTORY

CHANGES IN V7.1

- Revised approach by getting CLSID from HKCR before looking in HKCU/HKLM

CHANGES IN V7

- Added support for 64-bit AddIns, restructured and simplified code
- Updated this document included more details on the Assumptions and Approach

CHANGES IN V6.1

- Resolve issue with display of EA Version and program location

CHANGES IN V6

- Added button to display EA.exe.Config if present in EA installation directory

CHANGES IN V5

- Added DLL version to List of AddIns
- Added Registry Tree view complete with registry details
- Added context menu to Registry Tree view items to support queries based on contents of selected item
- Added Query results tab to display results from Registry Tree View queries as well as support user defined queries
- Output of queries to individual log files

CHANGES IN V4

- Added form to display **list of classes and methods** retrieved from the DLL

CHANGES IN V3

- Additional registry locations checked for add-in keys

CHANGES IN V2

- Searches HKLM for Sparx keys as well as HKCU
- Pop-up added to make it easier to read

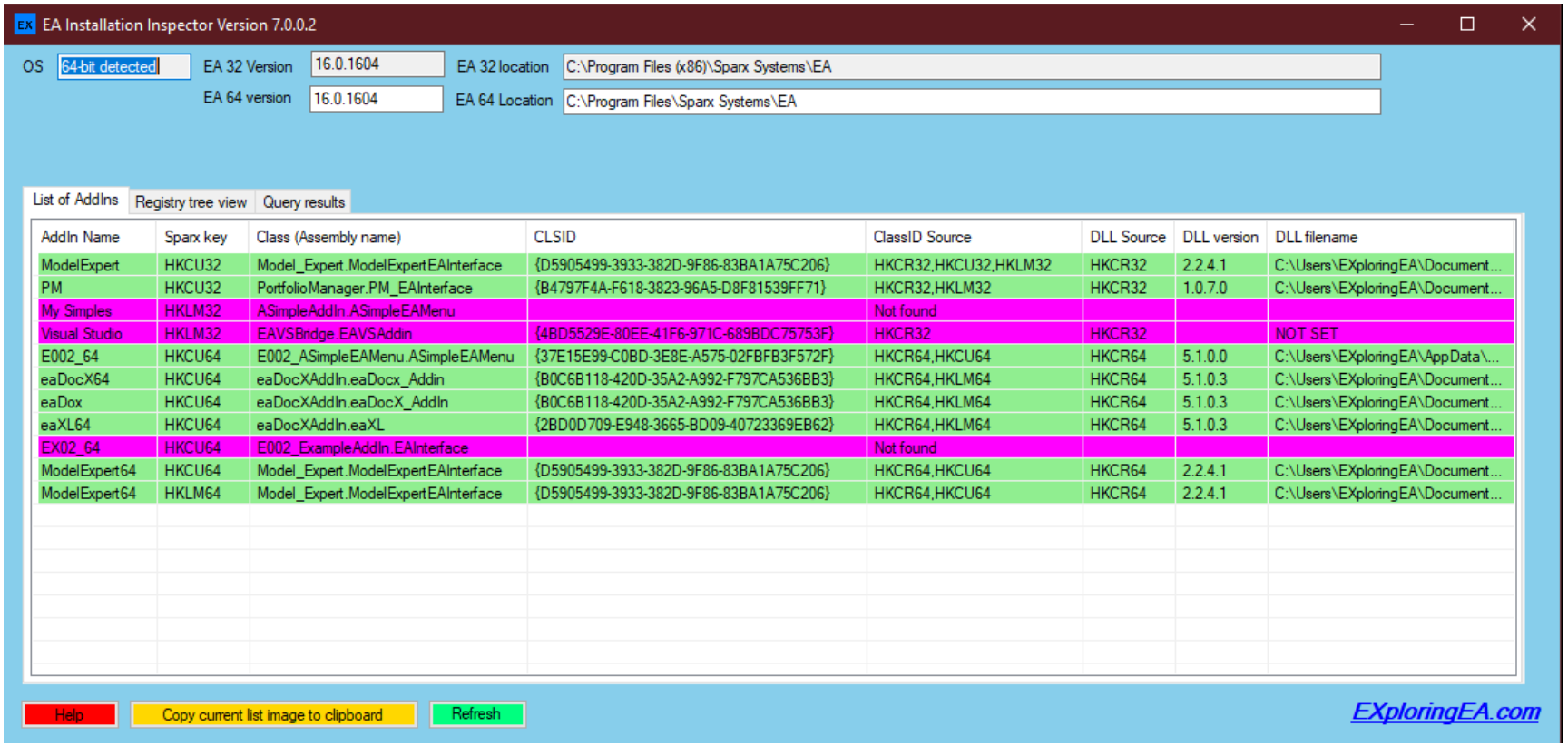
CONTENTS

PURPOSE

EA Installation Inspector is a small utility for developers of EA Addins who wish to check registry information about current EA Addins.

THE RUNNING EA INSTALLATION INSPECTOR

The program is a windows application that will present the user with basic information about the installed environment and version together with a list of the EA Addin keys found in the registry with details of the relevant classes/DLLs - similar to the screen shot below.



Each row represents an AddIn entry – and for a valid entry will display all of:

- The AddIn Name
- The location of the Sparx AddIn key within the windows registry
- Class(Assembly name) - Addin entry class
- (CLSID) Source – the location where the ClassID is defined in the registry
- CLSID – Class ID – as defined when the class was registered
- (DLL) Source – the location where the DLL is defined in the registry
- DLL Version – read from the file information
- DLL – Full file name for the AddIn DLL

In some cases, and as illustrated in the screen shot not all entries are complete; it may be that the AddIn is working or correctly installed and to help see any issues each row is coloured to reflect the status of the entry.

- Green – OK - the AddIn DLL has been found and the keys exist in the same hive; we assume that AddIn will be found by EA.
- Cyan – indicates that all the keys look fine but the DLL file does not exist at the specified location
- Magenta – means that no Class ID is set for the AddIn, hence the DLL cannot be indentified
- Red – indicates that CLSID and DLL are specified in different registry Key Sections
- Yellow – means that the DLL path is not set so cannot be found

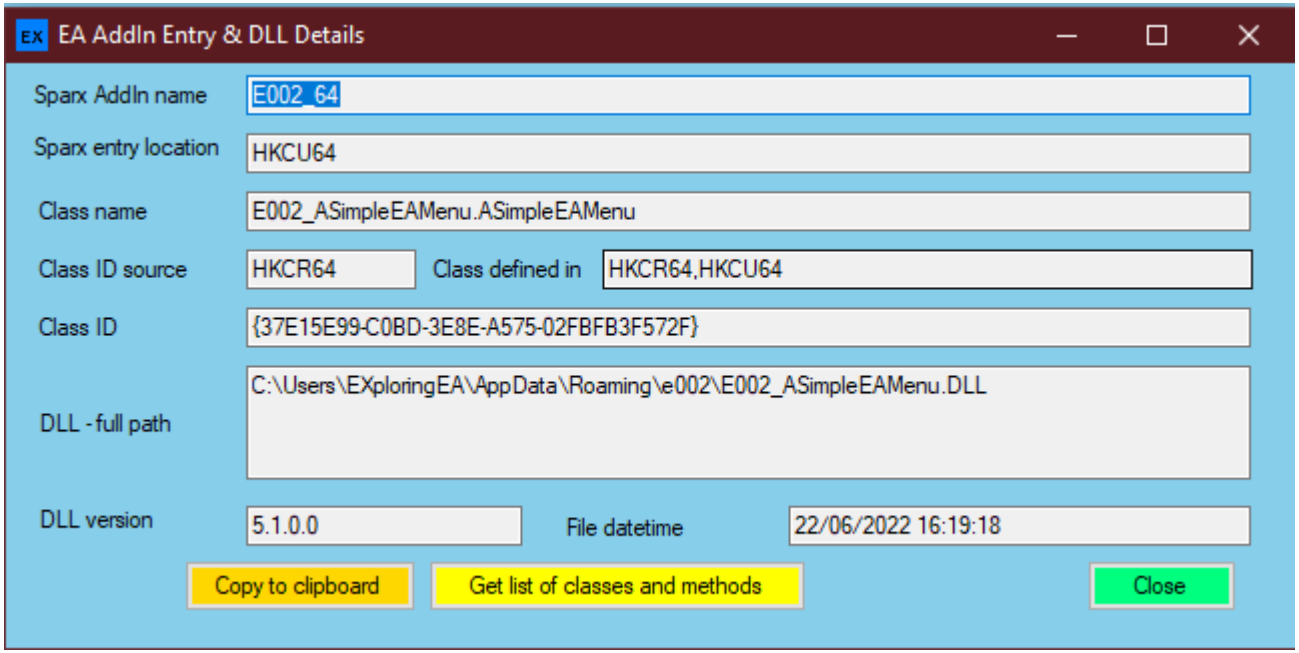
FUNCTIONS

The program will automatically perform the search and present the list of AddIn's when run. The following functions are available by accessing the buttons at the bottom of the screen dialog.

- **Help** – will present this document
- **Refresh** – will redo the search to reflect any changes that the user many have made to their system
- **Copy current list image to clipboard** – will copy the current list as an image to the clipboard for those times when the user may wish to forward to others.
- **EXploringEA.com –** will launch the default web browser with our blog page. You can also find contact information on this site should you wish to contact us

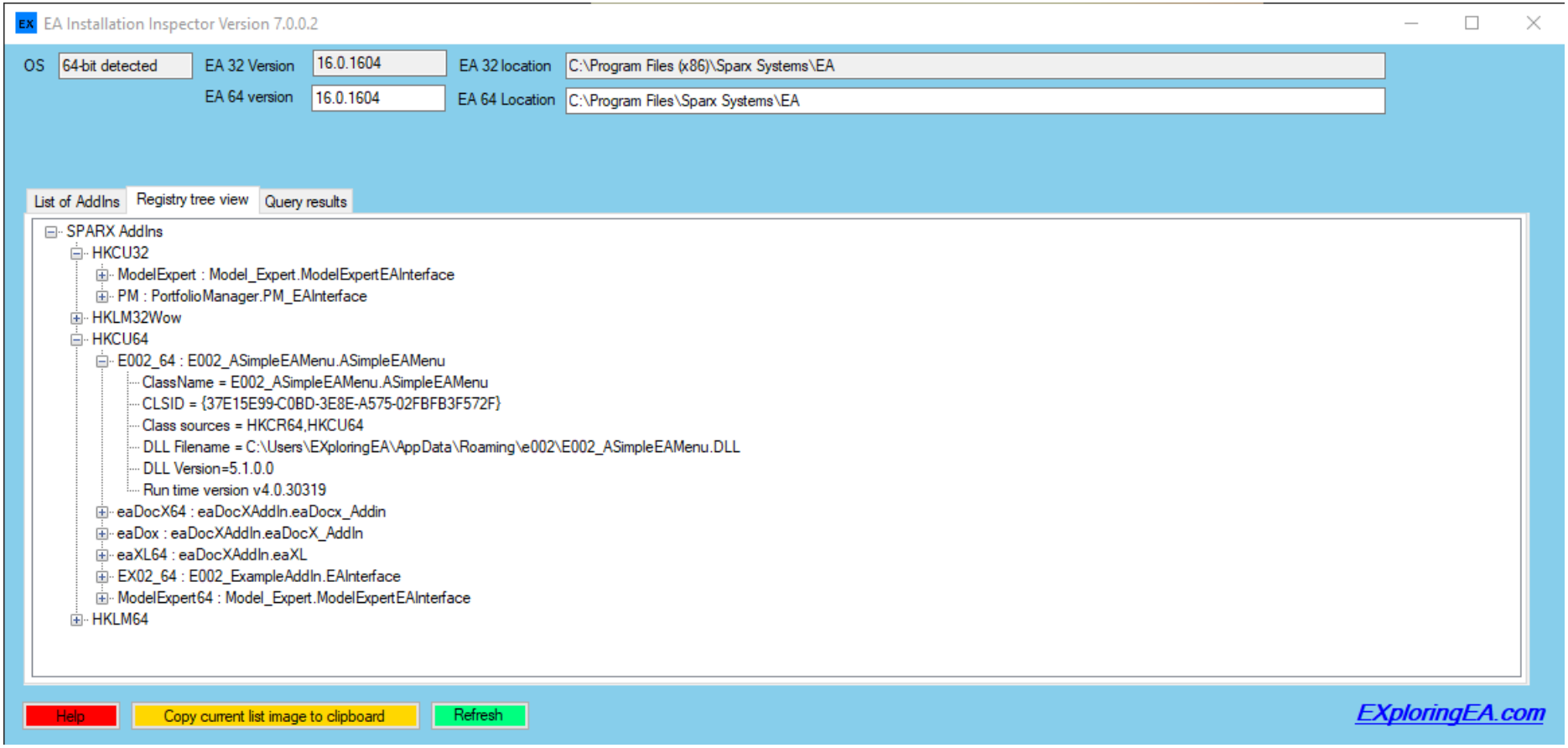
POP-UP ENTRY DETAILS FORM

It can sometimes be difficult to see all the information in a row so you can now ***double_click*** an entry and a form (similar to that illustrated below) is presented with the values more readily seen.

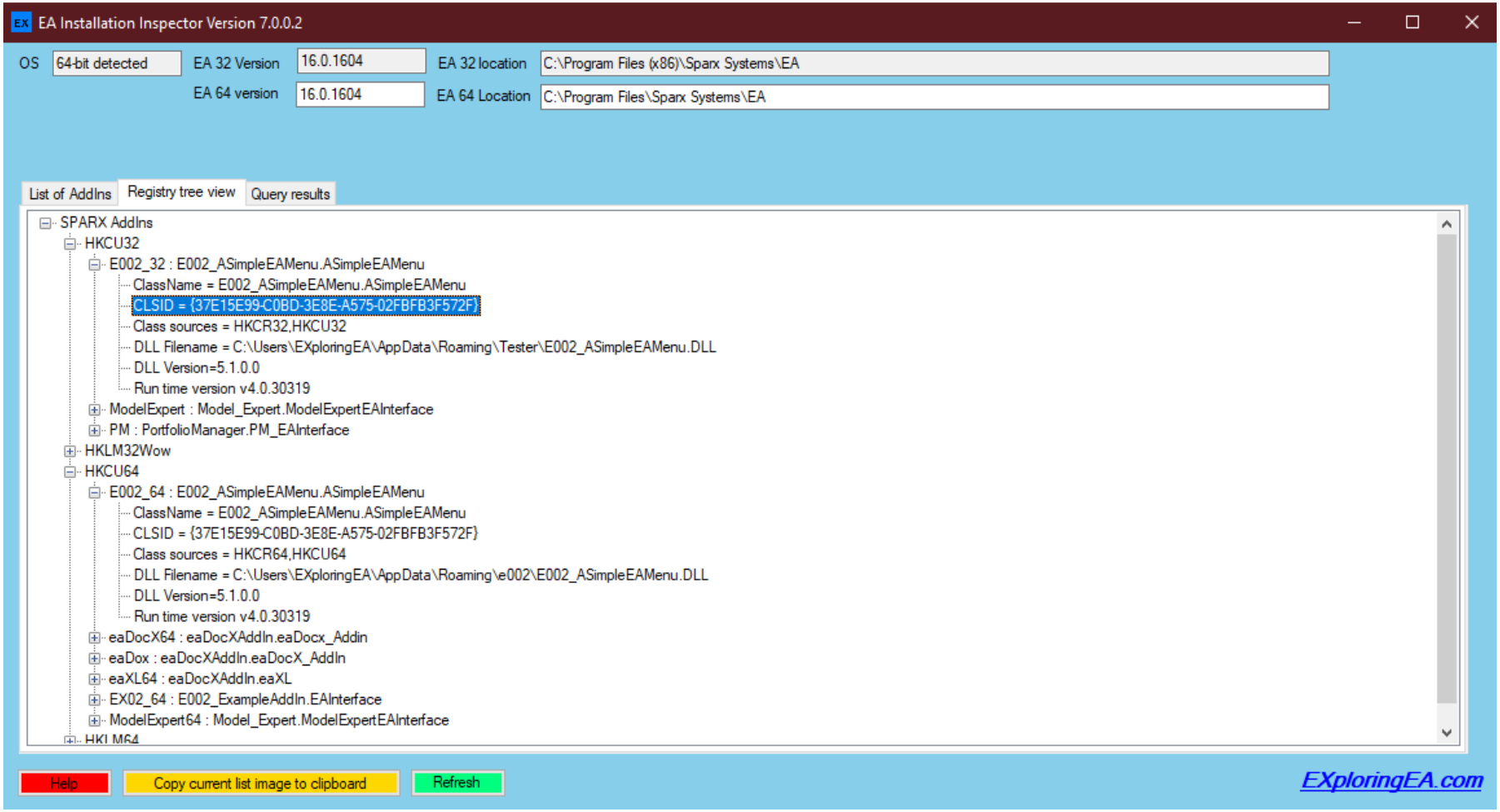


REGISTRY TREE VIEW

This tab presents the AddIn information as a tree format, based on the location of the information stored within the registry



This means that if there are entries for an AddIn which for example has been installed both for 32-bit and 64-bit use then it can be seen – See E002 which has been installed for 32-bit and 64-bit use; worth noting that the classID's are the same although but they refere to different dll's in different locations.



REGISTRY TREE VIEW QUERIES

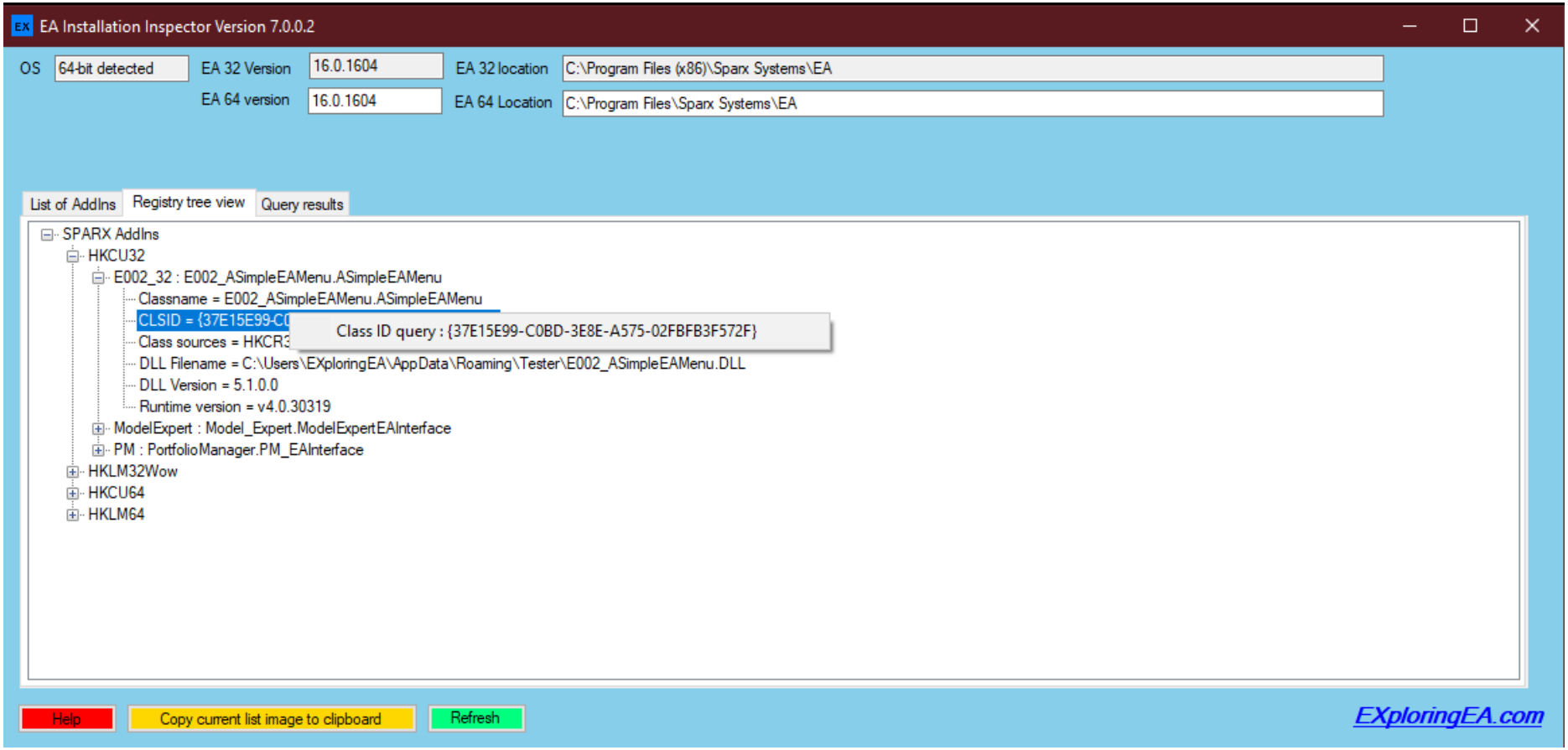
The user can select the context menu for some items:

- CLSID
- Classname
- Filename
- ProgID

to initiate a query for value within the selected line i.e. GUID, classname, filename, classname respectively. Menu items that provide an immediate response will have a pop-up dialog, whilst those that query the registry will be initiated in the background (so other actions can continue) - with the current registry query present in the Query info text box and a "Query active" indicator present on its right hand side, and results presented in the Query tab, where each query executed is listed before its results are displayed.

CLASSNAME QUERY

For CLISD and Classname entries a classname query can be performed. This will search the registry and output details of ANY key which contain the classname.



ilename entries:

- Open file location in windows explorer - will open windows explorer at the location specified in the selected key

- FileInformation - will present a pop-up window displaying information about the DLL.
- Filename query - will initiate a query across all the registry for the specific DLL.

An example of the query is Query started: reg query HKLM\SOFTWARE\CLASSES /reg:32 /s /f EAFeaturesTester.dll The results are display in the query results tab. Note some queries may take some time - they will execute in the background.

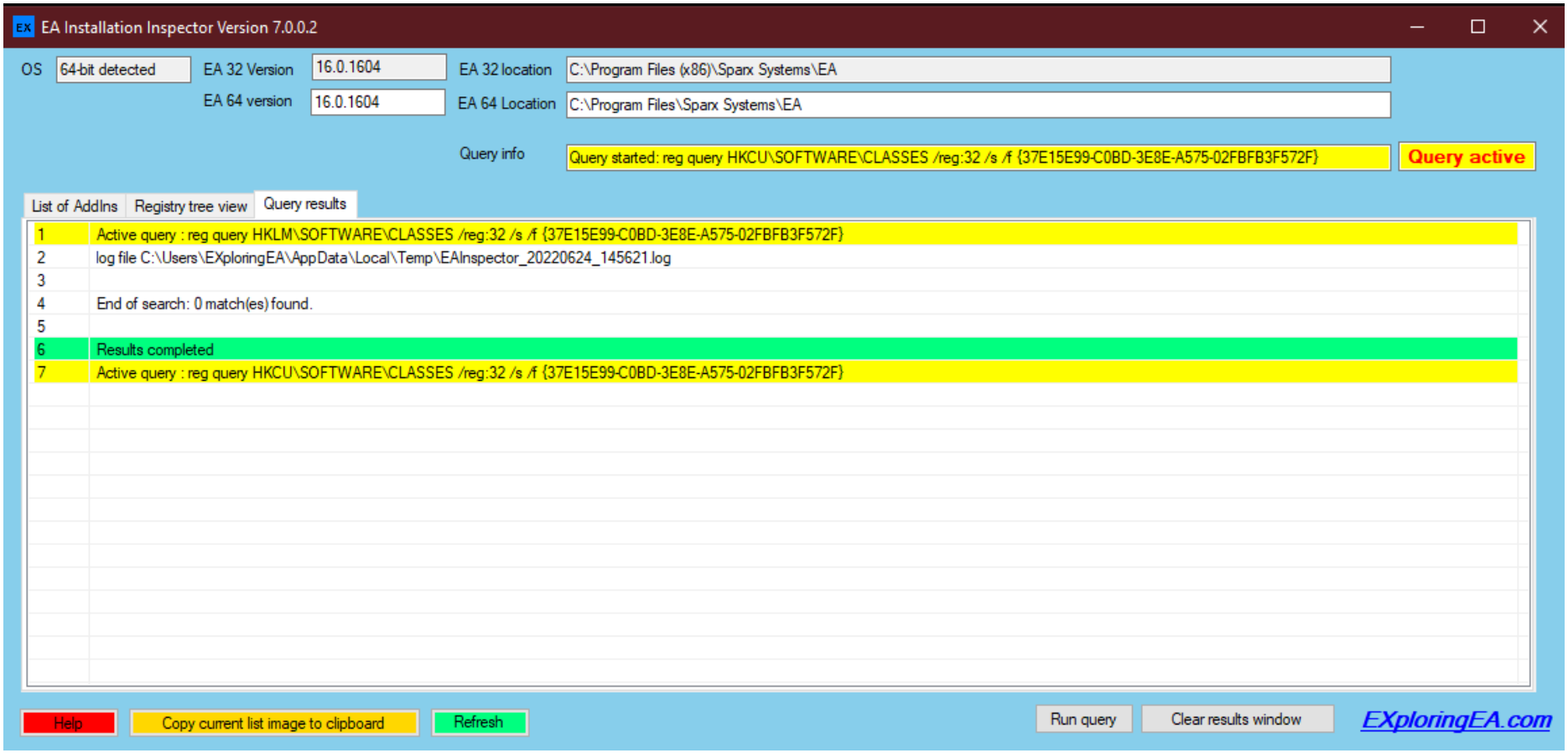
PROG ID ENTRIES

The Prog ID query will search the whole registry for ProgID's that match the selected entry. An example of the query is **Query started: reg query HKCU\SOFTWARE\CLASSES /reg:32 /s /f EAFeaturesTester.EA_Interface**

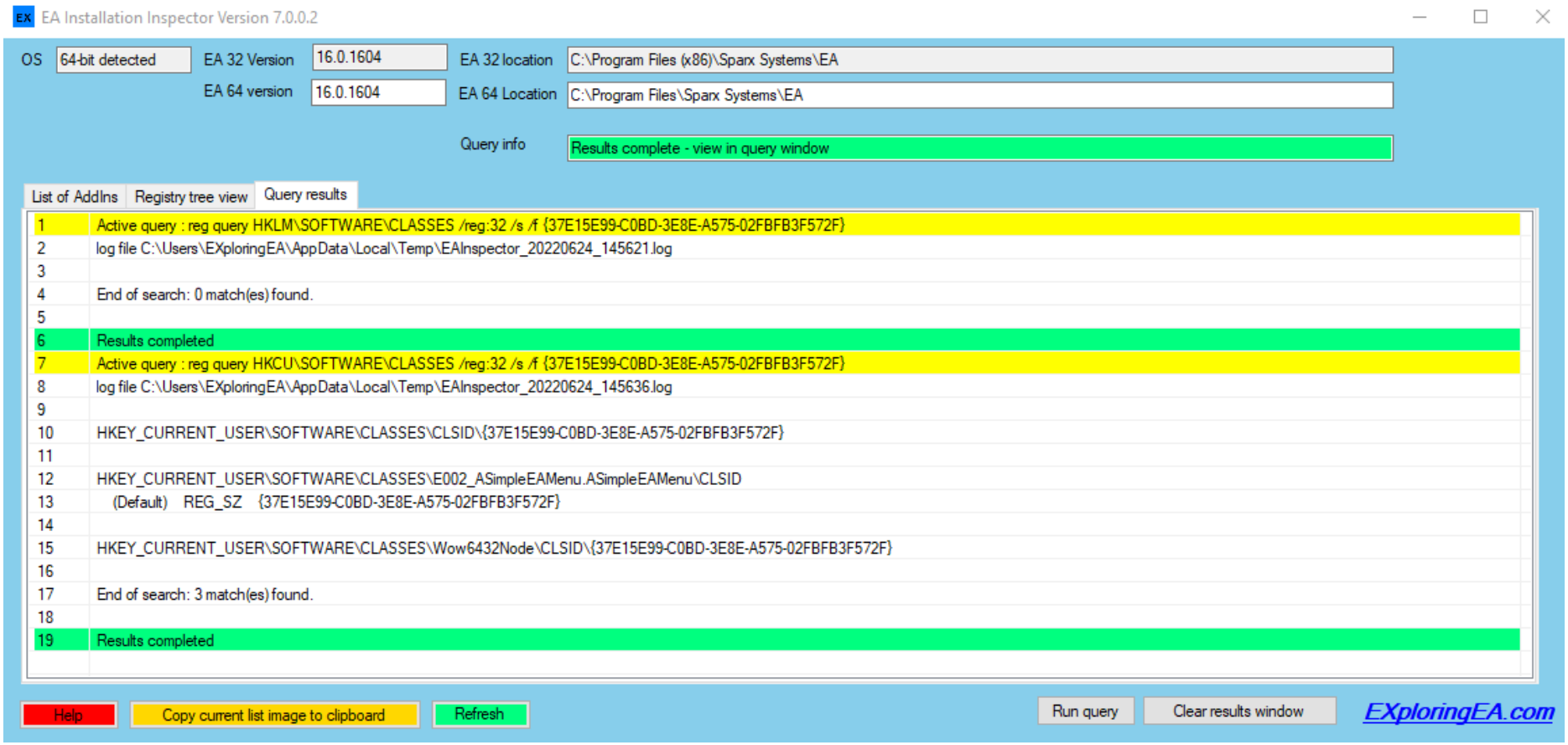
QUERY RESULTS TAB

The main function of this tab is to display results from queries. The output is display in the main area as shown below.

NB: Queries of the registry can take a lot of time hence each query is queued and the results output as they are obtained. The current query is displayed in the **query info** box which will have a yellow background with red text whilst queries are running, also a **Query active** label to the right of the query info box will be present. When all queries are completed then the **query info** box background turns green and the **query active** label disappears: Note this can take some time – the query will contin ue in the background allowing you to do other things if needed.



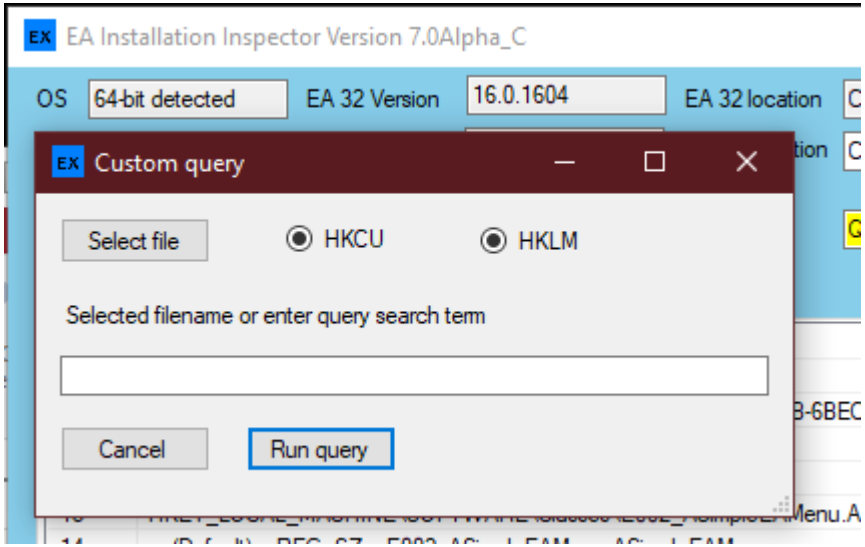
Below is a screenshot of all queries complete - also note the name of each log file should the user wish to get a hard copy of the results for a specific query.



Note: Unless the user specifically clears the results, they are appended to the page so there may be a need to scroll down.

The **Clear results window** button can be used to clear the contents.

In addition to queries initiated by the context menu on the **Registry Tree View** a user can create their own query using the **Run query** button.



QUERY LOG FILE

A log file for each query is produced and saved in the users AppData area in the directory for example:

C:\USERS\USERNAME\APPDATA\LOCAL\TEMP\EAINSPECTOR_DATE_TIME.LOG

OTHER BUTTONS

The program will automatically perform the search and present the list of Add-in's when run. The following functions are available by accessing the buttons at the bottom of the screen dialog.

- Help – the local help file may not be up to date.
- Refresh – will redo the search to reflect any changes that the user many have made to their system
- Copy current list image to clipboard – will copy the current list as an image to the clipboard for those times when the user may wish to forward to others.

- EXploringEA.com – will launch the default web browser with our blog page. You can also find contact information on this site should you wish to contact us

ASSUMPTIONS AND APPROACH

CHANGES WITH OPTIONS FOR 32-BIT AND 64-BIT VERSIONS OF EA

The initial versions of the EA Installation Inspector only supported 32-bit Add-ins. This meant that an approach which simply trawled the registry for AddIn information could be used.

With the introduction of a 64-bit version of EA, which can coexist with the 32-bit version, there is the possibility that there are both 32-bit and 64-bit versions of the same AddIn installed and hence the aim is to display information that relates to each item. And although there will be differences between the two versions, it is reasonable to have an AddIn which has the same ClassName and Class ID(CLSID) especially if they are built from the same code base. Windows recognises when a call to a class is made from a 32-bit or 64-bit application and is able to select the relevant DLL.

Within this section details are provided on how the information relating to AddIns is found and the relevant registry entries that are inspected.

Note that there are difference when running 32-bit and 64-bit versions of EA as well as differences when running on 32-bit or 64-bit operating systems.

BTW: If all else fails whilst reviewing the table or treeview and you are looking for problems associated with misplaced keys use can be made of the Query function to look for stray keys!

ADDIN'S

We don’t have access to the Sparx Code which loads the Add-ins, but assume that EA identifies AddIns by inspection of the following registry keys during startup. *The abbreviation in () is the value that is used within the EA Installation Inspector.*

32-BIT EA

- "HKCU\SOFTWARE\Sparx Systems\EAAddins" (HKCU32)
- "HKLM\ SOFTWARE\Sparx Systems\EAAddins" (HKLM32) if running on a 32-bit operating system

OR

- "HKLM\ SOFTWARE\WOW6432NODE\Sparx Systems\EAAddins" (HKLM32Wow) if running on a 64-bit operarting system

64-BIT EA

- "HKCU\SOFTWARE\Sparx Systems\EAAddins64" (HKCU64)
- "HKLM\ SOFTWARE\Sparx Systems\EAAddins64" (HKLM64)

Within these folders the name of the keys is how EA refers to the AddIn e.g. in the Manage Add-Ins dialog, and the value of the key is the class name which provides the interface to EA i.e. menus etc.

CLASS IDENTIFY

EA will attempt to load the class for a specific Add-In as defined in the Sparx Add-In registry entries (above). EA can make a request to access the class and interact using the API that Sparx defines for Add-Ins. To make this possible each AddIn must be registered so that Windows has information about the class library (dll).

As an Add-In developer you may wish to inspect the information that windows has about your Add-In(s) and this is the main purpose of the installation Inspector. It can also be used to check the installation to verify which files are being used.

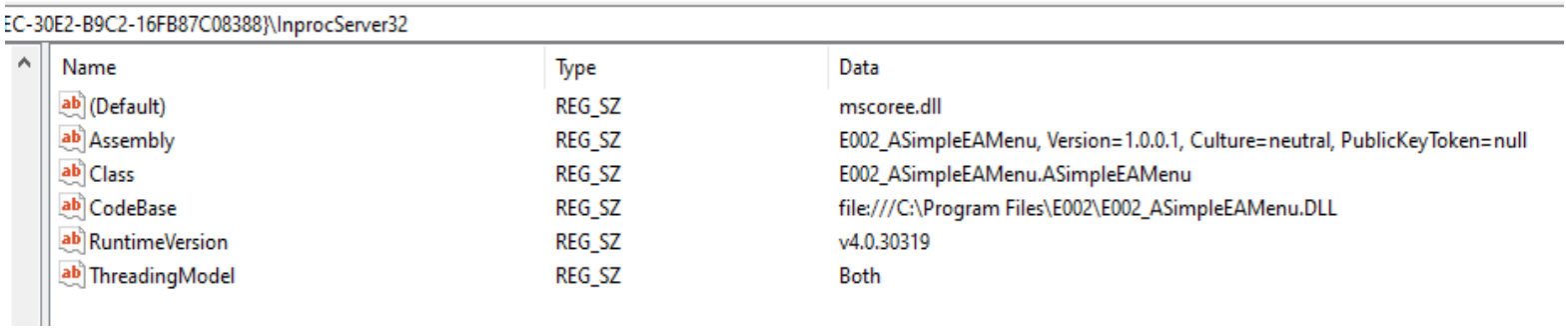
1. Inspect the Sparx AddIn registry entires and obtain a list of AddIns and their EAInterface class name

2. Using the class name inspect the various locations which contain information about the AddIn based on the
    a. Version(s) of EA - 32-bit or 64-bit
    b. Operating system - 32-bit or 64-bit
    c. Installation type – current user or local machine (all users)
3. Present the information found

There are rules that define how to get from the classname to the relevant registry entry for the AddIn.

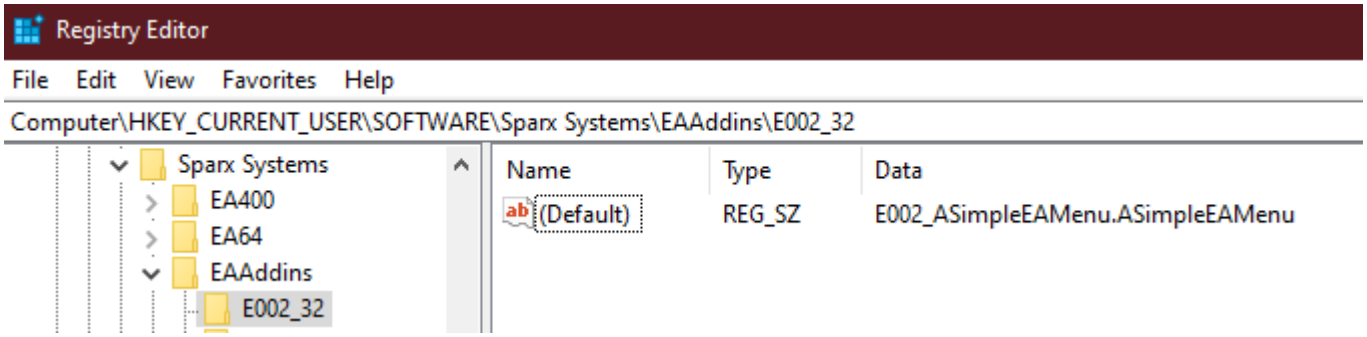
32-BIT REGISTRY KEYS

- Class name (MyAddIn.EAInterface) from Computer\HKEY_CURRENT_USER\SOFTWARE\Sparx Systems\EAAddins\MyAddIn
- ClassID ({MyClassID}) from Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\MyAddIn.EAInterface\CLSID
- Information on 64-bit system DLL located in Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\Wow6432Node\CLSID\{MyClassID}
    - See below the Information available from the InprocServer32 subfolder – note it's InprocServer32 for 64-bit classes as well!
        - Filename
        - Assembly version
        - .NET version

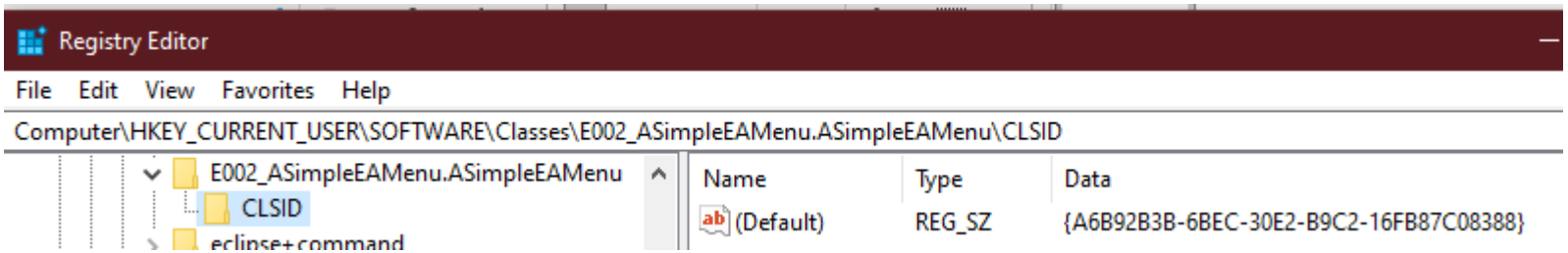


EXAMPLE E002_32

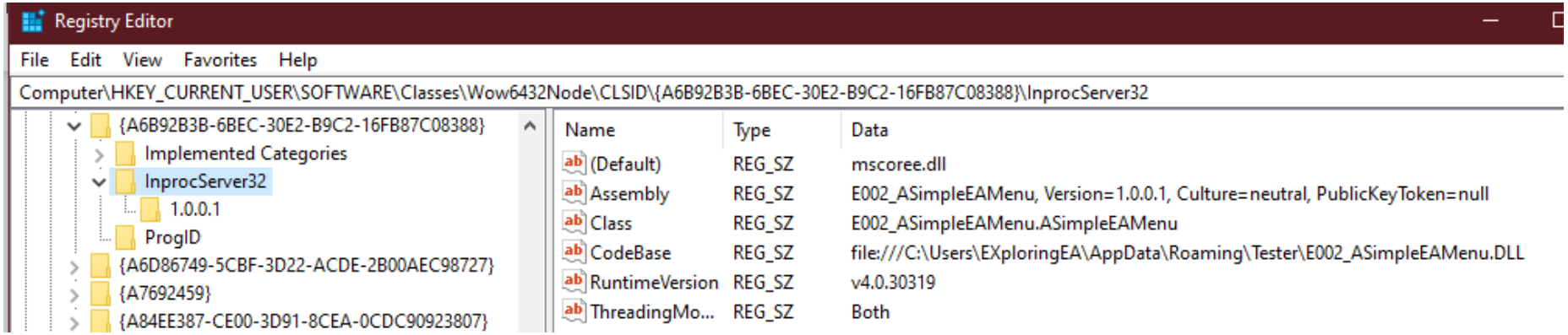
Sparx entry for AddIn **E002_32**, whose class name is **E002_ASimpleEAMenu.ASimpleEAMenu**



Class ID from HKCU Classes using the class name as the key



Class Information for HKCU WOW6432Node (i.e. 32-bit DLL in 64-bit system)

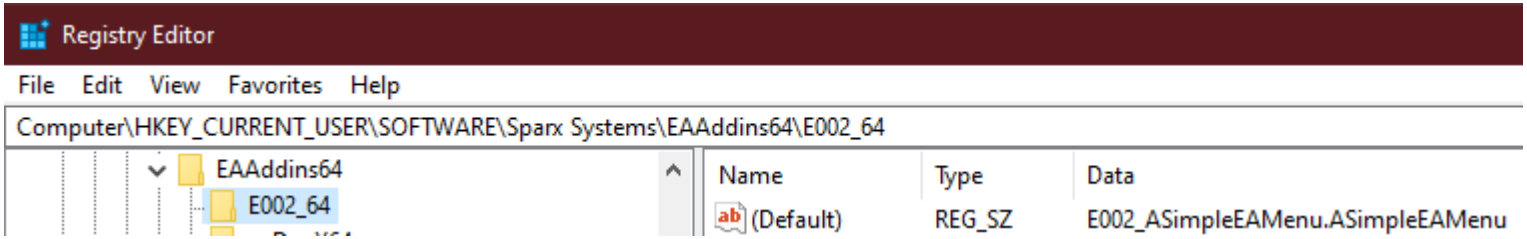


64-BIT EXAMPLE

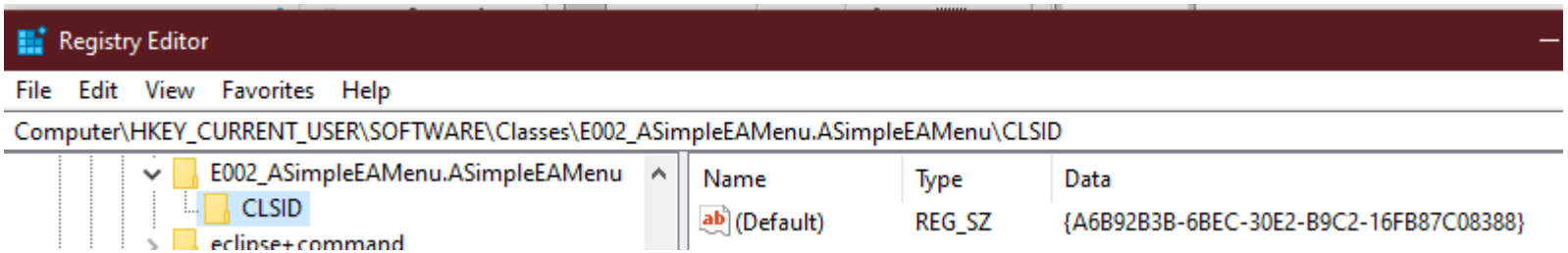
- Class name (MyAddIn.EAInterface) from Computer\HKEY_CURRENT_USER\SOFTWARE\Sparx Systems\EAAddins64\MyAddIn
- ClassID ({MyClassID}) from Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\MyAddIn.EAInterface\CLSID
- Information on DLL located in Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID\{MyClassID}
    - See below the Information available from the InprocServer32 subfolder – note it's InprocServer32 for 64-bit classes as well!
        - Filename
        - Assembly version
        - .NET version

EXAMPLE E002_64

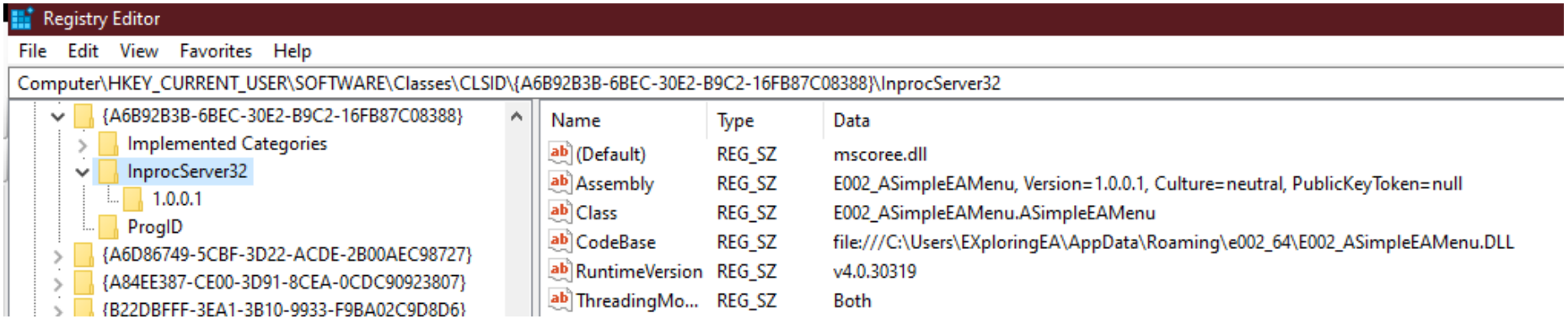
The same AddIn installed for 64-bit EA and hence as a 64-bit class library, where the Sparx entry is listed as a 64-bit Addin



ClassID in HKCU – Note that it is identical to the 32-bit version above



Class Information for HKCU



EXPECTED LOCATIONS OF KEYS AND CLASS INFORMATION

The information used by the EA Installation Inspector is obtained from the windows registry, which contains information about both programs and class libraries, amongst many other things. The following sections summarise the registry key location from where information is retrieved.

SPARX ADDIN KEYS

The following table details the locations where AddIn keys are stored

Type of AddIn	Registry Location	Abbreviation
eaHKCU32AddInKey	Computer\HKEY_CURRENT_USER\SOFTWARE\Sparx Systems\EAAddins	HKCU32
eaHKCU64AddInKey	Computer\HKEY_CURRENT_USER\SOFTWARE\Sparx Systems\EAAddins64	HKCU64
eaHKLM32AddInKey	Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Sparx Systems\EAAddins	HKLM32
eaHKLM64AddInKey	Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Sparx Systems\EAAddins64	HKLM64

CLASS INFORMATION IS CONTAINED WITHIN THE FOLLOWING AREAS.

The location of the class information varies depending on the operating system.

```
  ' For 32-bit addins runnin on a 64-bit OS there are a few locations
    ' \HKEY_CURRENT_USER\SOFTWARE\WOW6432Node\ -> Not a lot under this so no need to check
    ' \HKEY_CURRENT_USER\SOFTWARE\Classes\WOW6432Node\CLSID
    ' \HKEY_LOCAL_MACHINE\SOFTWARE\Classes\WOW6432Node\CLSID
    ' \HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Classes\CLSID

    ' for 64-bit on 64-bit os and 32-bit on 32-bit os
    ' HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID
    ' HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID
```

Both the class name e.g. myAddIn.EAInterface and it's class guid are located under the CLSID subkey folder

Class Locations Abbreviations	Registry Location
HKCU_Classes	Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\
HKCUWOW_Classes	Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\Wow6432Node\
HKLM_Classes	Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\
HKLMWOW1_Classes	Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\WOW6432Node\

HKLMWOW2_Classes	Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Classes\

CLASS INFORMATION 32-BIT OS - 32-BIT EA ONLY - 32-BIT ADDIN

The following areas are inspected for 32-bit addins running on 32-bit OS

Sparx Key	Class location abbreviation	Registry location
eaHKCU32AddInKey	HKCU_Classes	Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\
eaHKLM32AddInKey	HKLM_Classes	Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\

CLASS INFORMATION 64-BIT OS - 32 BIT EA - 32-BIT ADDIN

The following areas are inspected for 32-bit addins running on 32-bit or 64-bit OS

Sparx Key	Class location abbreviation	Registry location
eaHKCU32AddInKey	HKCUWOW_Classes	Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\Wow6432Node\
eaHKLM32AddInKey	HKLMWOW1_Classes or HKLMWOW2_Classes	Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\WOW6432Node\ Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Classes\

CLASS INFORMATION 64-BIT OS - 64 BIT EA - 64-BIT ADDIN

The following areas are inspected for 64-bit addins running on 64-bit OS

Sparx Key	Class location abbreviation	Registry location
eaHKCU64AddInKey	HKCU_Classes	Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\
eaHKLM64AddInKey	HKLM_Classes	Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\

OTHER LOCATIONS

The tables above indicate the expected locations where information about the relevant AddIn Class libraries are located. The output the EA Installation Inspector indicates where the class library is found and sometimes this is not as expected. For example a 32-bit addin may have been installed for the current user so the Sparx key is in HKCU but the class library has been registered for all users (HKLM).

It has been known that for reasons unknown the class library is registered in a completely different location and as such may not appear in the list of AddIns. In this case the query features of the tool can be used to do a search of the registry to see what entries contain information about an addin class. For example, if you have an AddIn whose interface class is MyAddIn.EAInterface enter this string and inspect the results from the search. For more details on this feature see "Query Tab"

PROGRAM APPROACH

For information the program steps are:

1. Identify all the Add-ins from the Sparx registry keys
2. Find the CLSID for the class name from in HKCR or HKCR WOW6432Node
3. Use the CLSID get the assembly details for the AddIn from within HKCR:
    - Filename and location
    - File datestamp
    - Assembly version
    - DotNet runtime version
4. If the AddIn exists check the other HIVEs to see if other entries exist. As HKCR is a merge of HKCU and HKLM it would be expected that there should be an entry. Locate class name in the expected hive and get the ClassID
    a. 32-bit EA and OS
        i. HKCU
        ii. HKLM
    b. 32-bit EA, 64-bit OS

    i. HKCU32WOW
    ii. HKLM32WOW
c. 64-bit EA, 64-bit OS
    i. HKCU
    ii. HKLM