LEGAL

VERSION HISTORY

CHANGES IN V7

- Added dialog to display registry entries or export as registry files to use for adding / deleting AddIn keys
- Added support for 64-bit AddIns, restructured and simplified code
- Revised approach for getting CLSID from HKCR before looking in HKCU/HKLM
- Revised query code
- Changed the method used to extract information class and method information from AddIn DLL to work with both 32-bit and 64-bit DLL's from single version of EAII
- Added settings to save column widths and window size
- Changed colours used to highlight observations
- Bug fix - accessing file explorer for DLL file location in treeview
- Removed Added to display EA.exe.Config if present in EA installation directory
- Updated this document included more details on the Assumptions and Approach
- Added button to provide pop-up colour
- Added a green back colour to EA version and location to indicate which is registered for COM and hence providing the EA API
- Added indication in header whether running Admin or not
- Changed the presentation of ClassID source to clarify whether a real entry exists in HKCU or HKLM – so if the Addin entry is a Sparx HKCU32 entry you should expect to see ClassID Source as HKCU32, however if it indicates HKCR then the Class entry is located elsewhere

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

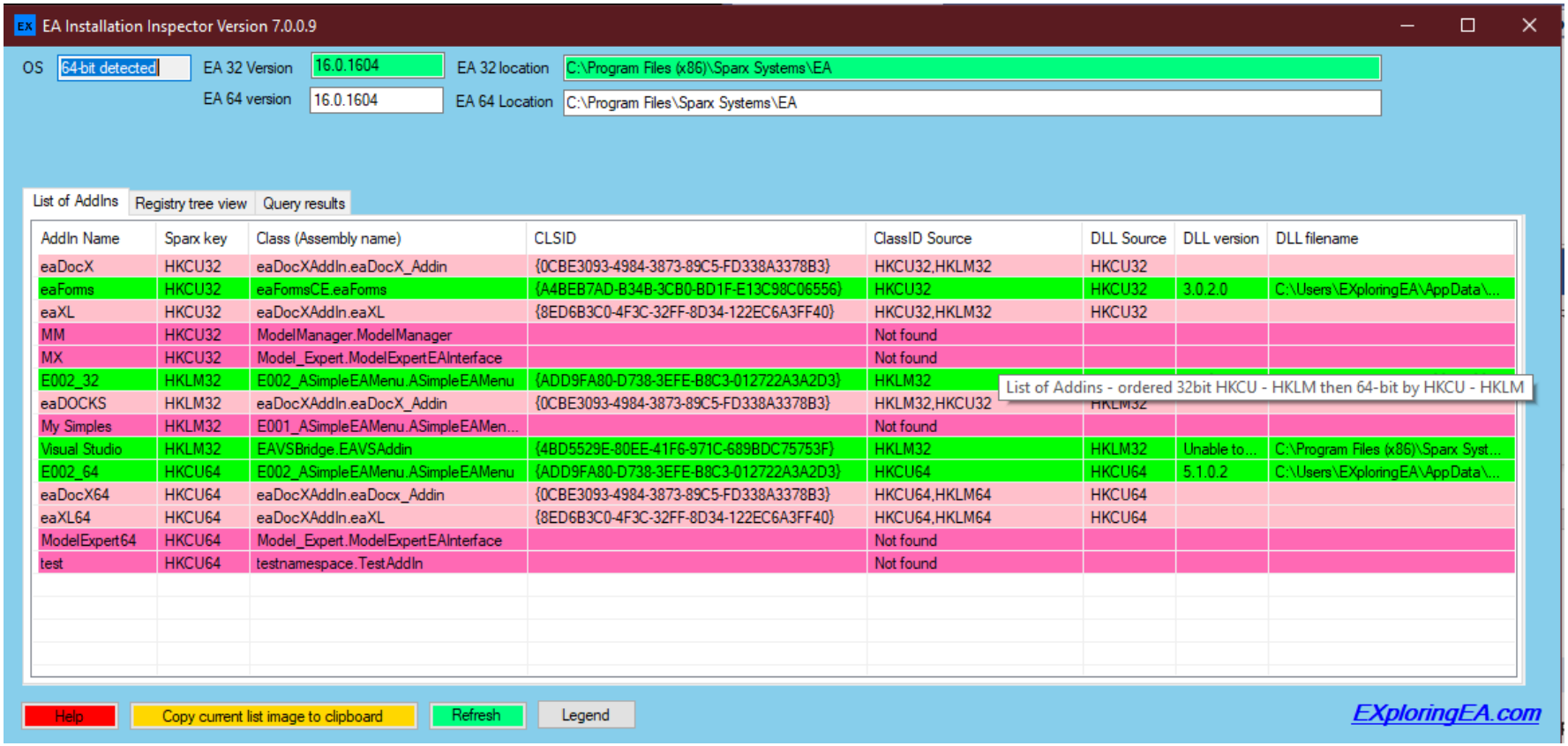
CONTENTS

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 presents the user with basic information about the EA installed environment together with a list of the EA Addin keys found in the registry with details of the relevant classes/DLLs as illustrated in the screen shot below.



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

- The AddIn Name
- Sparx Addin location within the windows registry
- Class (Assembly name) as specified in the Sparx key Addin entry
- CLSID – Class ID – as defined for the registered class
- ClassID Source – the location where the ClassID is defined in the registry
    - HKCR – if entry in HKCR but not in the relevant 32-bit or 64-bit sections
    - HKCU32 – 32-bit current user
    - HKLM32 – 32-bit local machine
    - HKCU64 – 64-bit current user
    - HKLM64 – 64-bit local machine
- DLL Source – the location where the DLL is defined in the registry (derived from the current record entry)
- DLL Version – read from the DLL file information
- DLL Filename for the AddIn

NOTE: If you double click on an entry a pop-up displays more information – see below.

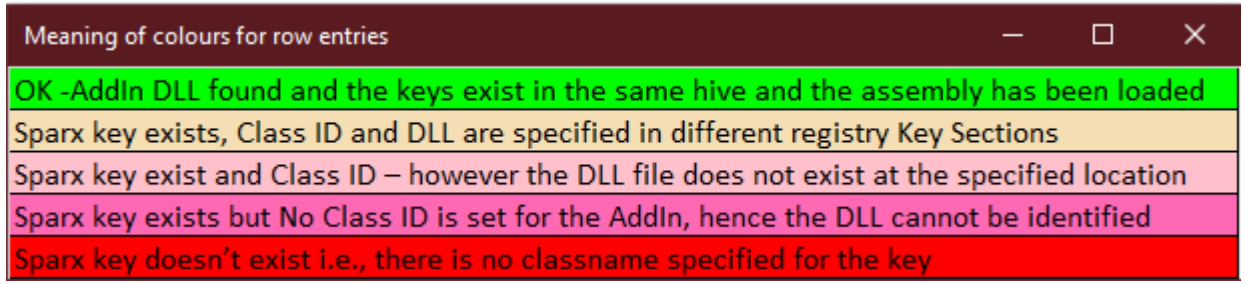
In some cases, and as illustrated in the screen shot not all entries are complete; it may be that the AddIn is correctly installed and working, however this may not always be the case. To help identify issues each row is coloured to reflect the status of the entry

- Green – OK - the AddIn DLL has been found and the keys exist in the same hive and the assembly has been loaded; we assume that AddIn will be found by EA.
- Beige – indicates that CLSID and DLL are specified in different registry Key Sections
- Pink – indicates a Sparx key exist and ClassID – however the DLL file does not exist at the specified location
- Bright pink – means that no Class ID is set for the AddIn, hence the DLL cannot be identified
- Red – means the Sparx key doesn't exist i.e., there is no classname specified for the key

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.

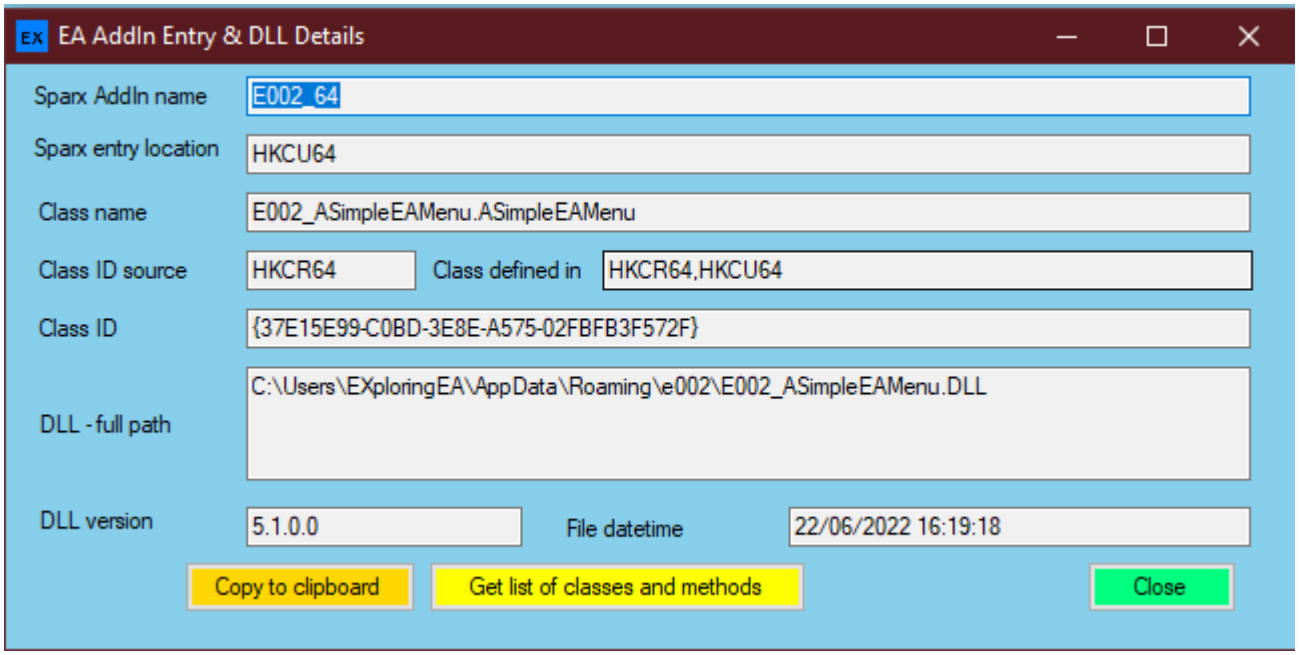
- **Help** – will present this document
- **Copy current list image to clipboard** – will copy the current list as an image to the clipboard for those times when the user may wish to forward to others.
- **Refresh** – will redo the search to reflect any changes that the user many have made to their system
- **Legend -** launches pop-up dialog describing the meaning of the colours – to save check the document



- **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

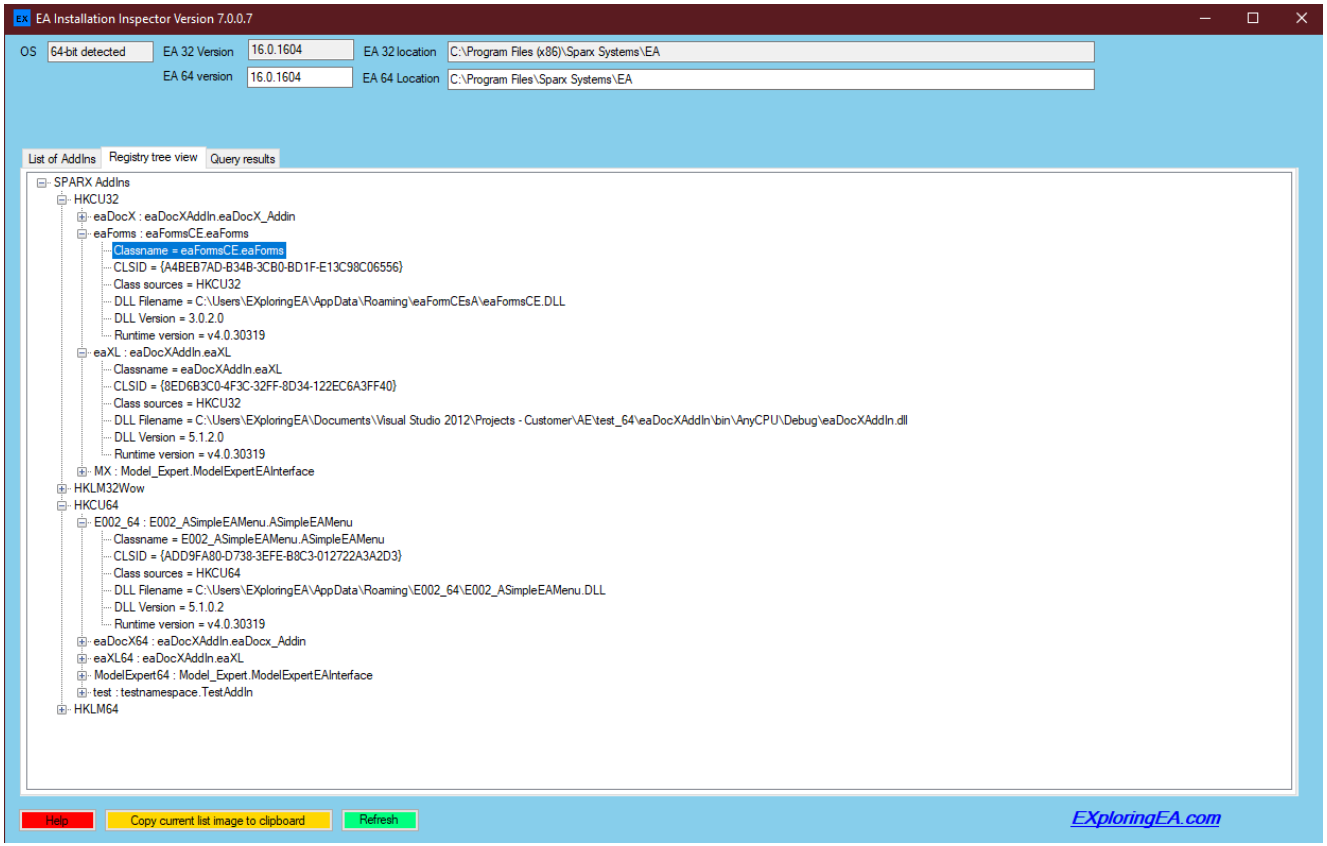
POP-UP ENTRY DETAILS FORM

It can sometimes be difficult to see all the information in a row. If you ***double_click*** on an entry a pop-up form (like that illustrated below) presents the Addin information and provides options to get more information.



REGISTRY TREE VIEW

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

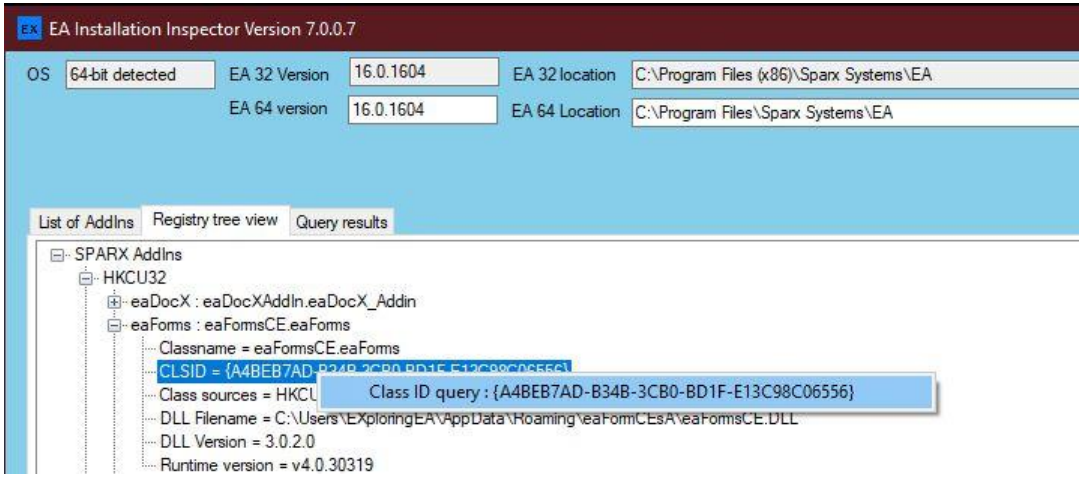


This means that if there are entries for an AddIn which for example has been installed both for 32-bit and 64-bit use then it can be seen – See E002 which has been installed for 32-bit and 64-bit use. Also, worth noting that the Class ID's are the same although but they refer to different dll's in different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

REGISTRY TREE VIEW QUERIES

The user can select the context menu for some items:

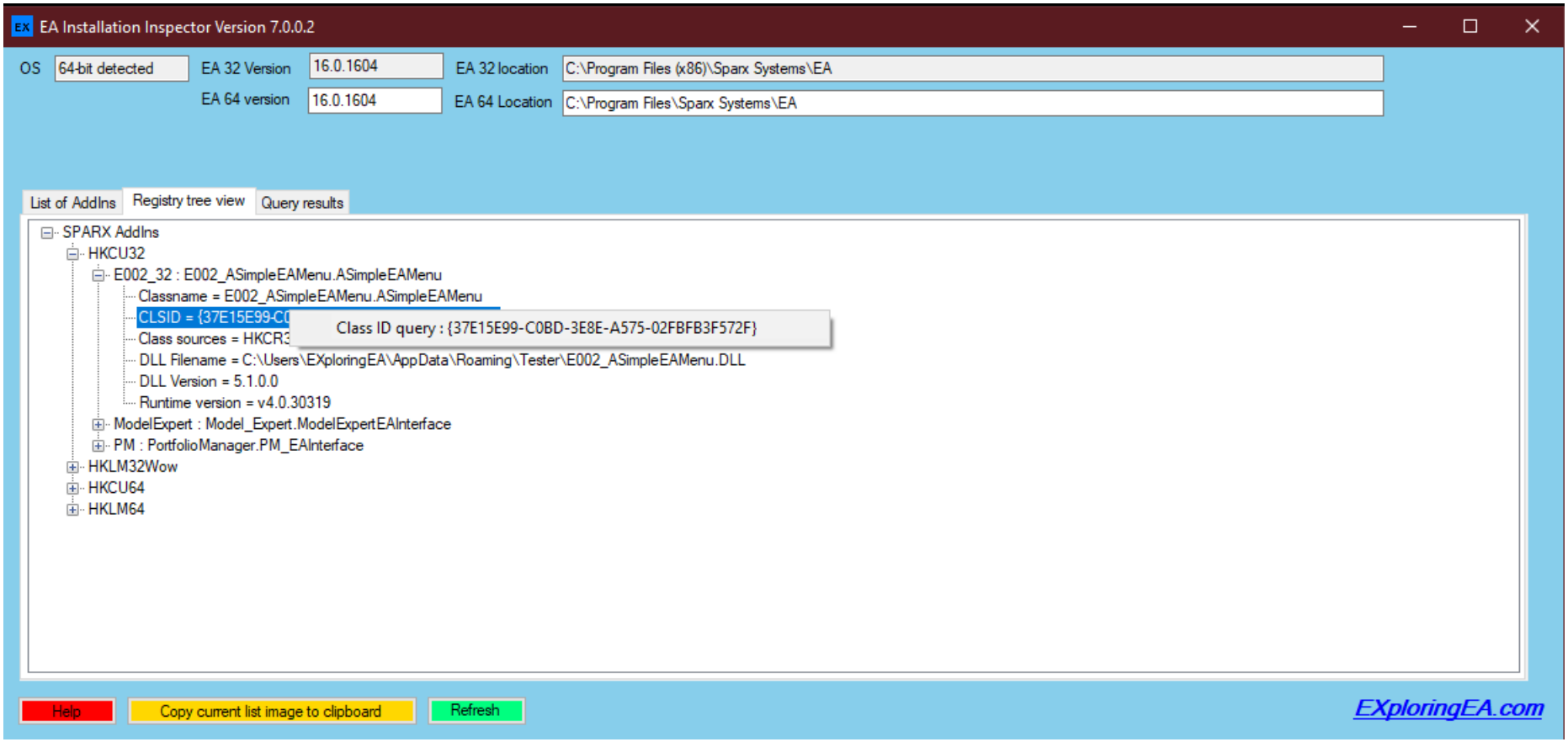
- CLSID
- Classname
- Filename
- ProgID



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.



FILENAME 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.

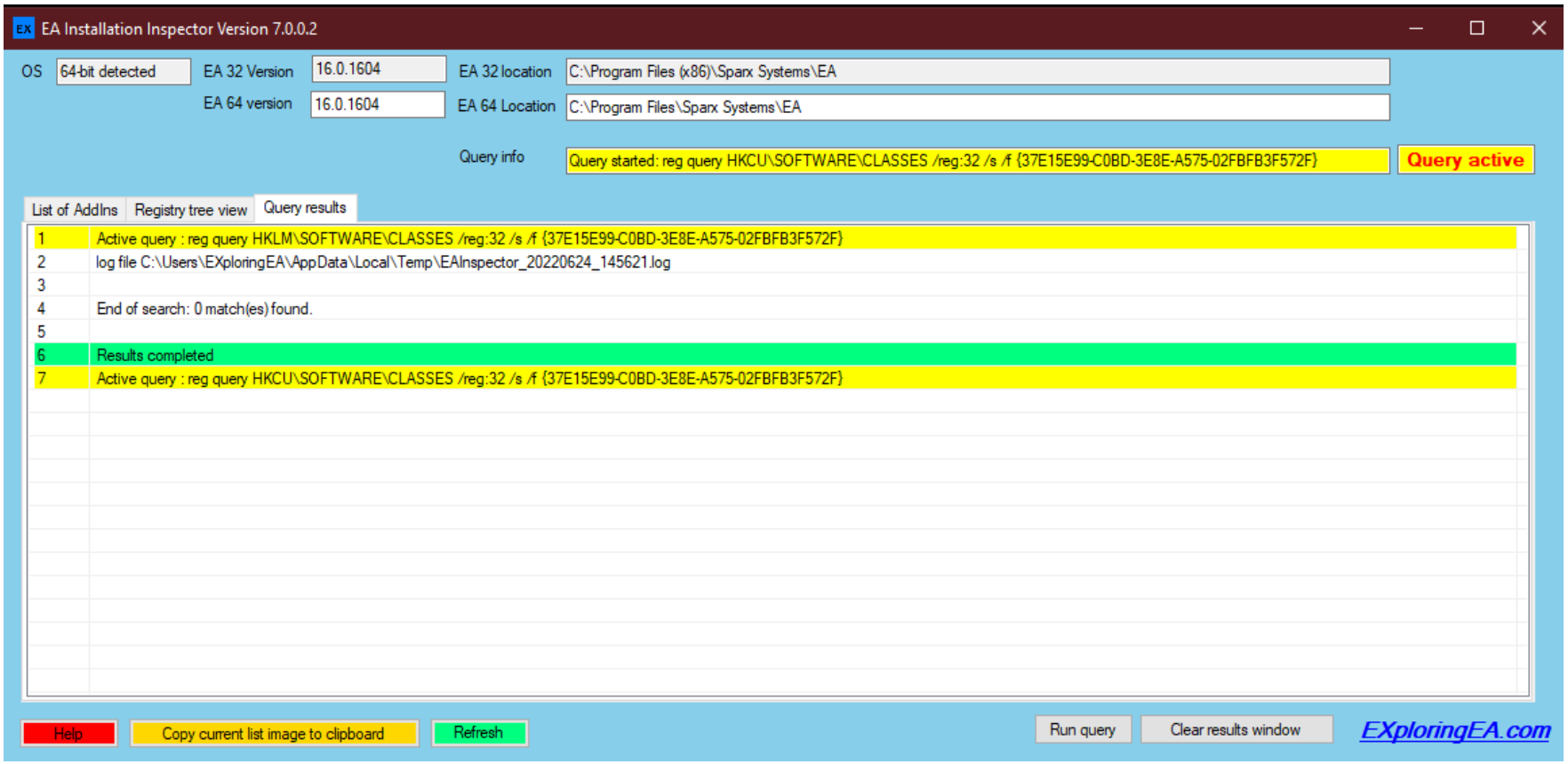
PROG ID ENTRIES

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**

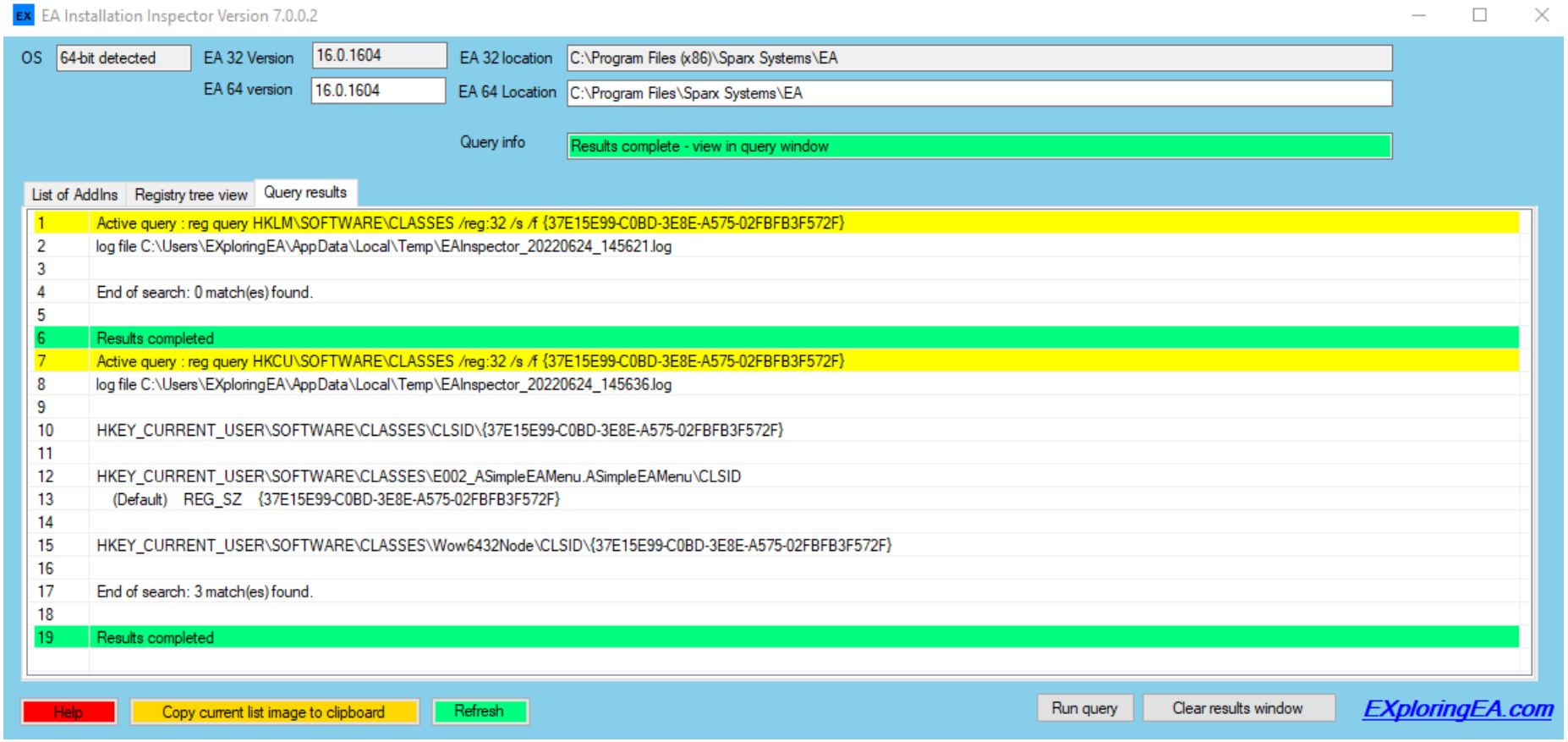
QUERY RESULTS TAB

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. A **Query active** label to the right of the query info box is presented and when all queries are completed then the **query info** box background turns green, and the **query active** label disappears. Remember that this can take some time but you perform other operations in parallel.



Below is a screenshot of all queries complete - also note the name of each log file if you want to review the results for a specific query.

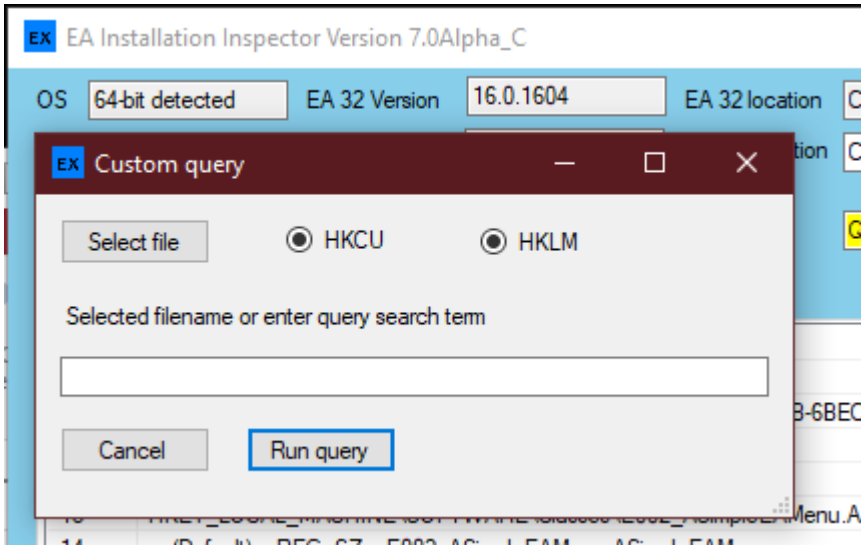


Note: Unless the user specifically clears the results, they are appended to the page so there may be a need to scroll down.

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

In addition, to queries initiated through the registry tree context menu there are 2 other options available.

1. Initiate a query based on the results of a previous query using the context menu for relevant items in the query results window
2. Press the “Run query button” to query within classes areas of the registry – typically a classname or a DLL filename



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

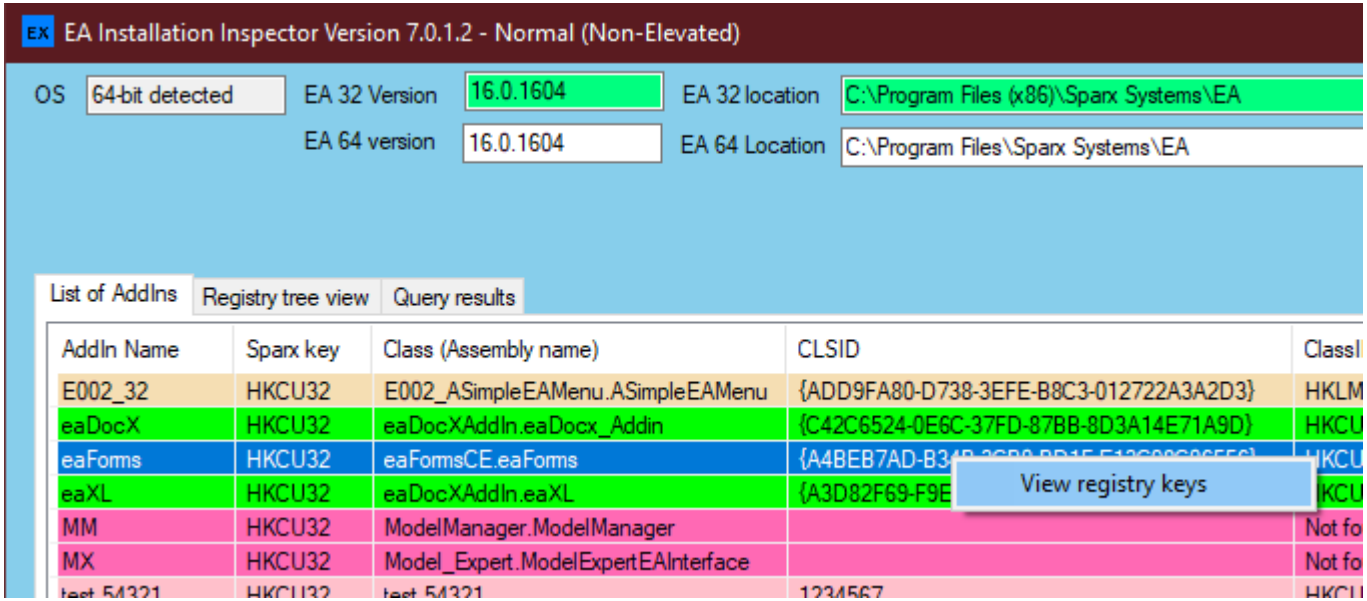
OTHER BUTTONS

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

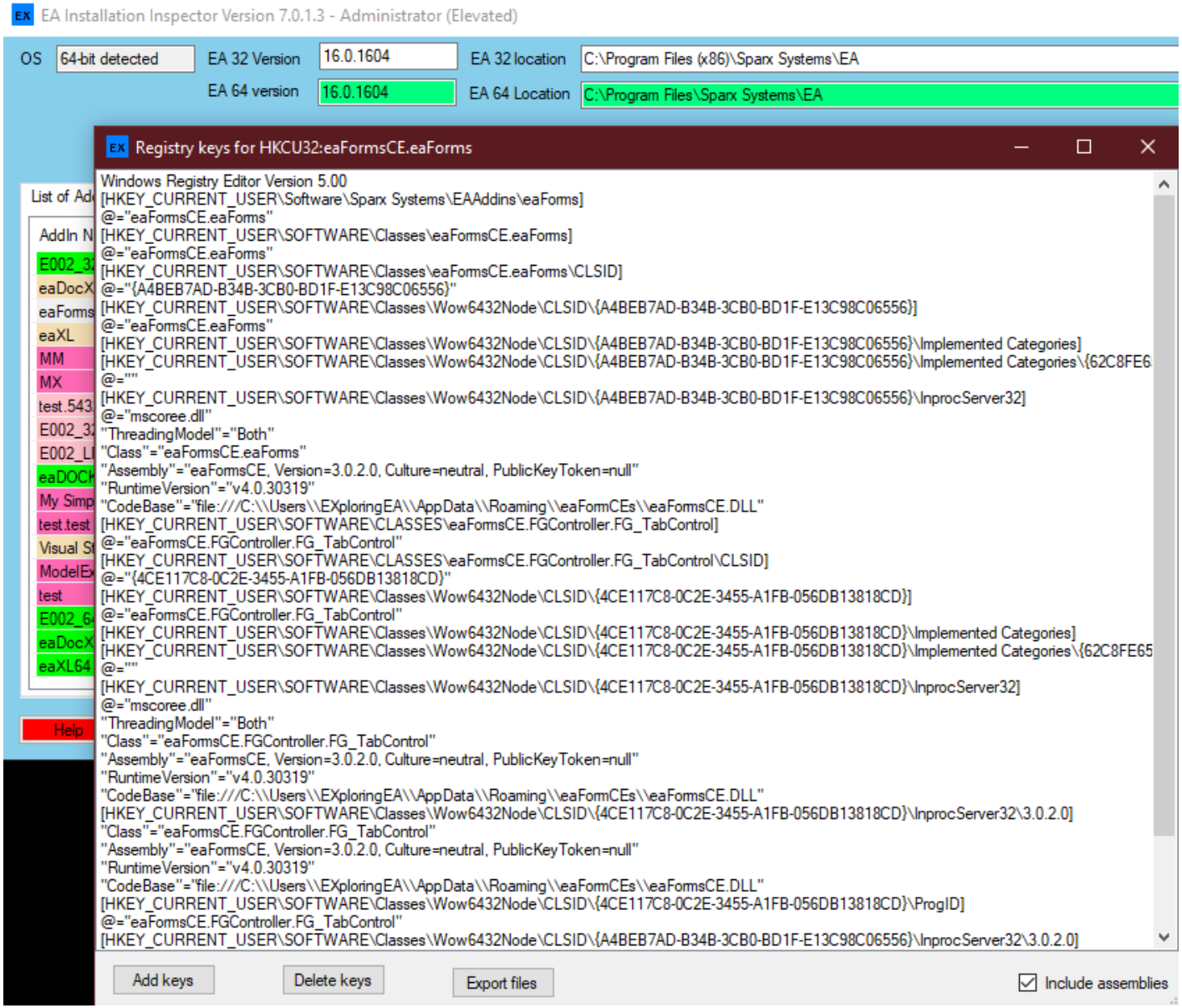
REGISTRY KEYS DIALOG

This dialog presents the registry entries, in reg file format, which can be used to add or delete the registry keys for the currently selected AddIn.



In the list of AddIns, select the line for the required AddIn, and using the right-click context menu select “view registry keys”. A dialog is presented which initially displays the registry keys for the selected AddIn. For an installed AddIn the following keys would be present:

- Sparx keys
- Class keys
- CLSID keys



Buttons at the bottom of the dialog perform the following functions:

- Add – Display keys and values that would be needed to support the AddIn
- Delete – Display key definitions to delete the AddIn keys
- Export – Creates ".reg" files for both Add and Delete the keys. The files are exported in the users "Documents" folders with filenames derived from the Hive & ClassName as well as including a time stamp.
- Include Assemblies – additional exposed types in the same namespace as the AddIn are included – changing the checked status will update the contents

This export option allows the user to create files that can be used with RegEdit to modify the registry.

- Run RegEdit (in an Administrator command window) using the required file to delete the keys (or add keys) - which may provide the functionality you need.

For example:

"regedit DeleteRegFile_HKCU64_E002_ASimpleEAMenu.ASimpleEAMenu20221210_094438.reg" – deletes the keys

"regedit AddRegfile_HKLM64_E002_ASimpleEAMenu.ASimpleEAMenu_20221210_095911.reg" – adds the keys

PLEASE ENSURE YOUR FULLY UNDERSTAND WHAT YOU ARE DOING AS USING THIS FEATURE CAN MAKE CHANGES TO YOUR REGISTRY – MAKE A BACKUP EVEN IF YOU ARE AN EXPERT!!!!

A tool I have used to take registry snapshots and look for changes in the registry is https://www.nirsoft.net/utils/registry_changes_view.html

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 which meant that an approach that 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. It is worth noting that although there can 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 know the bitness of the running app and will select the relevant DLL for the correct location.

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 differences 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 tree view and you are looking for problems associated with misplaced keys use can be made of the Query function to look for stray keys!

ADDIN REQUIREMENTS

An AddIn is a class library (DLL) which must provide an API as defined by Sparx so that once loaded EA can make a request to access the class and interact with the AddIn.

OVERVIEW OF LOADING AN ADDIN

An Addin consists of a class library which provides a defined API that is defined by Sparx. It may be that the AddIn class library uses other library and configuration files that are supplied- the information about these is outside the scope of the Installation Inspector.

- SPARX (ADDIN) KEYS:
    - EA identifies the presence of an AddIn through registry key(s), normally set by the installer, which name the AddIn and the related classname of the DLL
- CLASSNAME KEYS:
    - Windows looks for a key in the registry HIVE HKCR matching the classname to get the value of its CLASSID
- CLASSID KEYS:
    - Windows uses the CLASSID together with information relating to the calling app (32/64?) to find the class library entry
        - Initially for the current user
        - If no current user entry found, then the local machine
    - If an appropriate CLASSID entry is found details of the DLL file to load should be present and can be loaded

If any of the information is missing, then the DLL will not load. Typical errors include:

- Class has not been registered, hence so no CLASSID entry - 80040154
- Invalid class – 800401F3
- File does not exist at defined location – 800401F8
- Error in DLL – 800401F9

The following sections provide some more detail on these keys

ADDIN'S

EA identifies AddIns through inspection of defined registry keys. It is important to note that there are differences between 32-bit and 64-bit operating systems.

1. On 32-bit systems you cannot run 64-bit EA

2. On 64-bit systems you can run both 32-bit and 64-bit EA – and Sparx use different AddIn keys to indicate whether it is a 32-bit or 64-bit addin

The abbreviation in () is the value that is used within the EA Installation Inspector.

SPARX ADDIN KEYS FOR 32-BIT EA

When

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

OR if running on a 64-bit operating system

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

SPARX ADDIN KEYS FOR 64-BIT EA

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

An Addin key will have:

- Name – this name is used to identify the Addin as shown in the "Manage Add-Ins dialog"
- Default value - the AddIn classname – which will be the class that provides the interface to EA i.e., menus etc.

CLASS IDENTITY

EA will attempt to loads the class for a specific Add-In as defined in the Sparx Add-In registry entries (above). The class must be registered so that windows can find the class library (DLL) to load – this is normally done by the installer or registering the keys manually,

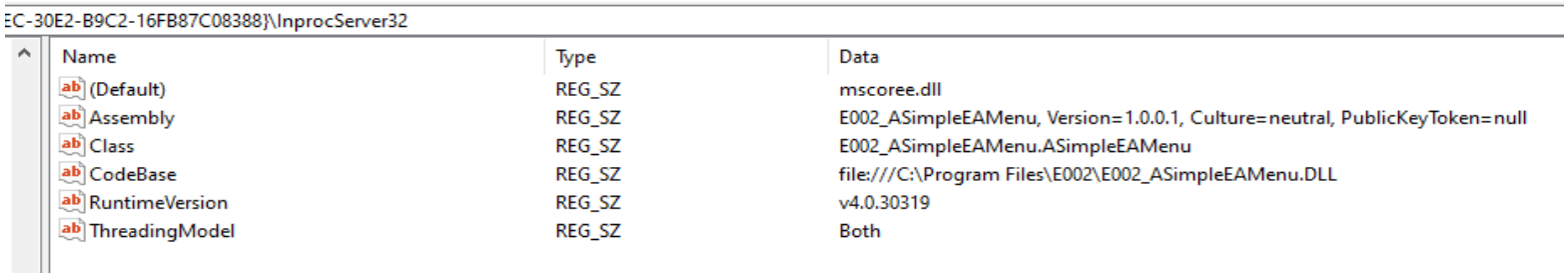
As an Add-In developer you may wish to inspect the information that windows has about your Add-In(s) and this is the main purpose of the installation Inspector. It can also be used to check the installation to verify the class library (DLL file) that is being used as well as look for anomalies, this information includes:

1. The Sparx AddIn registry entries 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)

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

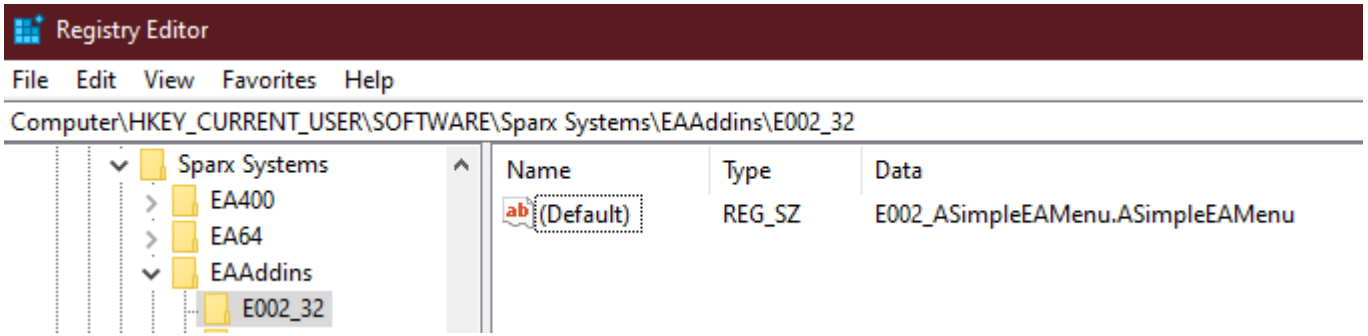
32-BIT REGISTRY KEYS

- 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

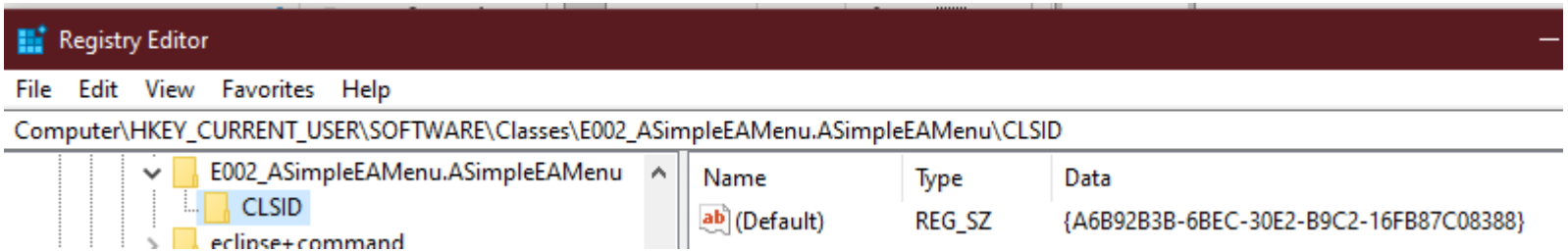


EXAMPLE E002_32

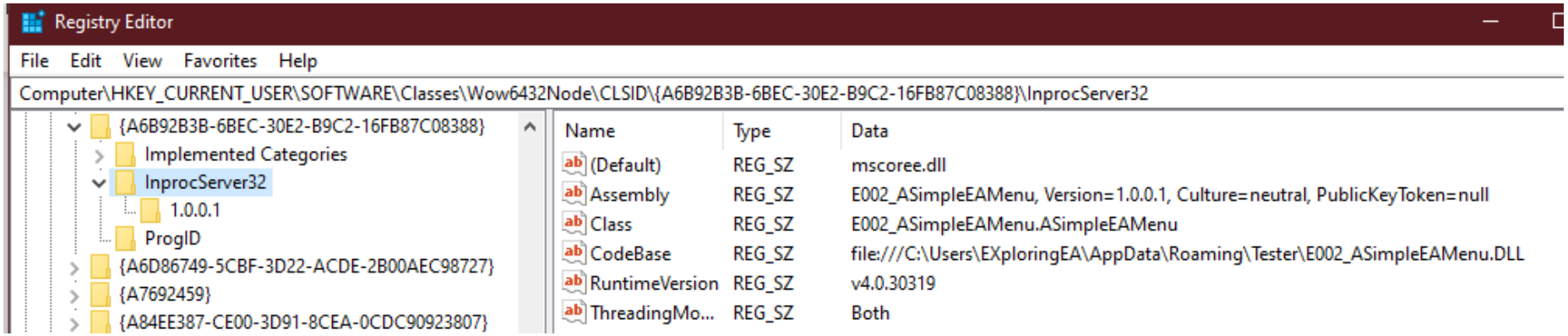
Sparx entry for AddIn **E002_32**, whose class name is **E002_ASimpleEAMenu.ASimpleEAMenu**



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)

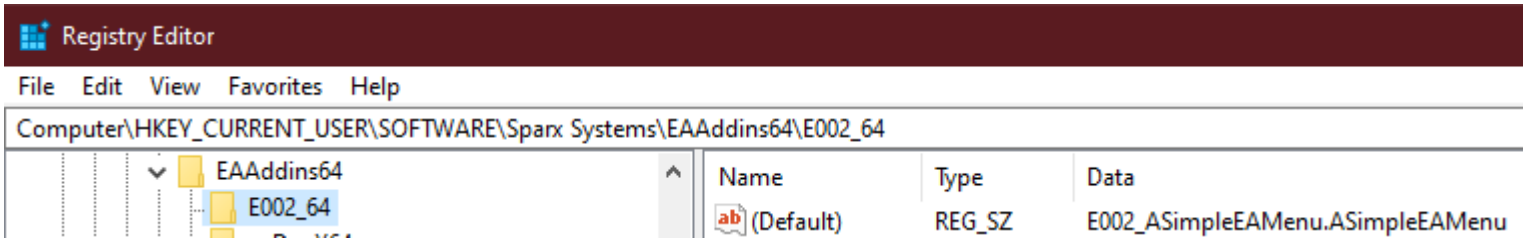


64-BIT EXAMPLE

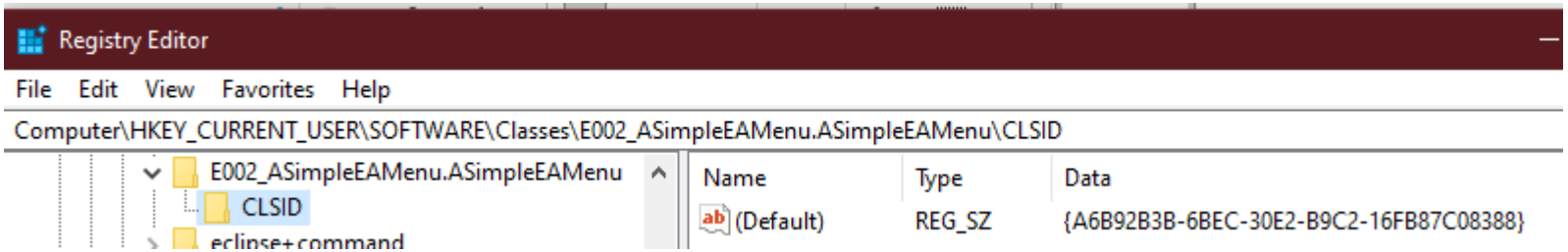
- Class name (MyAddIn.EAInterface) from Computer\HKEY_CURRENT_USER\SOFTWARE\Sparx Systems\EAAddins64\MyAddIn
- ClassID ({MyClassID}) from Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\MyAddIn.EAInterface\CLSID
- Information on DLL located in Computer\HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID\{MyClassID}
    - 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

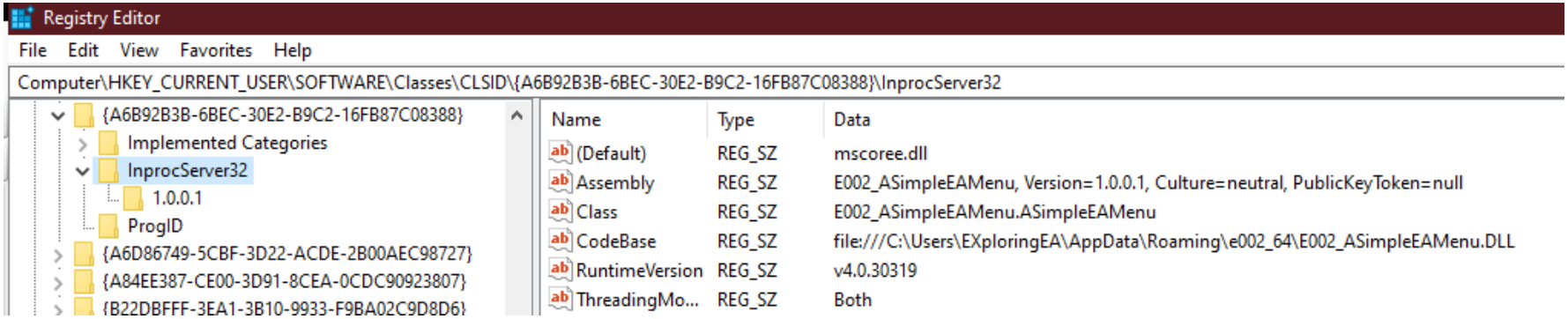
The same AddIn installed for 64-bit EA and hence as a 64-bit class library, where the Sparx entry is listed as a 64-bit Addin



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



Class Information for HKCU



REFERENCE - 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 running 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 its 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 is 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. Note the class entries in HKCR are a merge of HKCU and HKLM so the class ID may be found but NOT relate to the current entry. For example, checking for a 32-bit entry may get a CLSID which relates to a 64-bit AddIn.
3. Use the CLSID look for CLASSID entry in the expected HIVE
    a. 32-bit / 64-bit
    b. HKCU or HKLM
4. For the CLASSID get the assembly details for the AddIn from within HKCR:
    - Filename and location
    - File date stamp
    - Assembly version
    - DotNet runtime version

NOTE: Multiple versions of the AddIn may be registered and since the HKCR entry is a merge of HKCU and HKLM checks are started in the relevant location if not found checks are made (and indicated) if found elsewhere. For example,

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

