

ASCOM Scripting

CEDIC 2015

Nicola Montecchiari

COM - What is it?

Component Object Model (COM) is a standard interface software components

Introduced by Microsoft in 1993

Used to enable inter-process communication and dynamic object creation

COM – Where is it used?

Because it is a Microsoft related software architecture, it is tightened with the Windows Operating System.

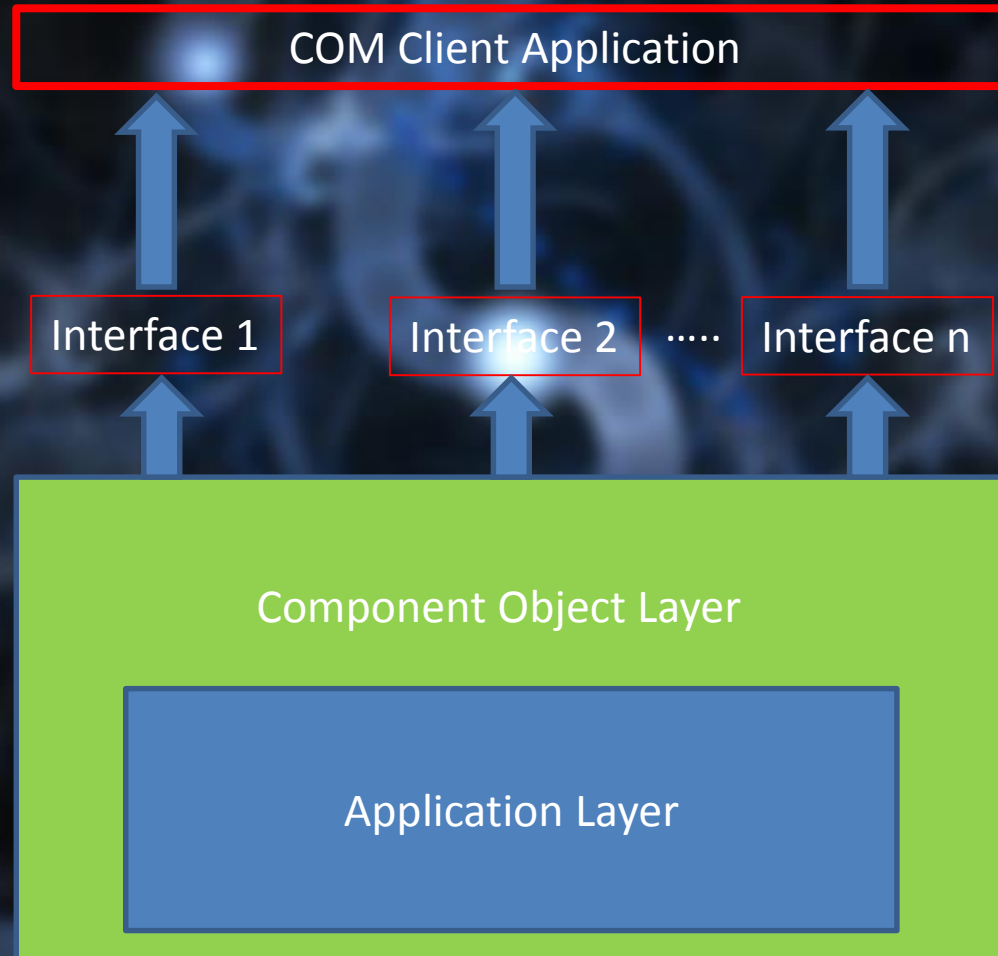
Any 32bit or 64bit Windows is suitable for using COM objects.

COM – Alternatives?

As of now there are no suitable and stable alternatives to the duality COM/Windows.

Linux users have the option of using the INDI library, but it does not offer the same functionality and OS integration as COM.

COM – A schema



COM – Advantages

Once a COM object is instantiated, many COM clients can communicate with the same object.

COM provides an abstraction layer, making simple to have an application running for different devices.

COM – Object Creation/Deletion

When a COM client creates the object, it means that a COM interface to an object is dynamically instantiated.

Only once all COM clients have disconnected from the COM interface, the object is deallocated.

COM – Object Usage

Once the COM client is connected to a COM object by an interface, it may interact with the object using the exported API by the COM server.

COM – Methods and Properties

A method is basically a function that takes one or more arguments for input and returns an output. The output might be a value or an action performed.

A property is an entity that holds a value. It may be read-only or read/write: when it is read/write it may also perform an action like a method.

COM – Sync/Async Methods

A method is synchronous when the flow of the algorithm is stopped until the method is completed

A method is asynchronous when the script flow is not stopped to wait for the method completion.

ASCOM

Acronym for

Astronomical Component Object Model

A set of interfaces designed by software developers and amateur astronomers in order to make life easier to manage different astronomical devices.

ASCOM - Devices

Managed Devices include

- CCD Cameras
- Mounts
- Focusers
- Domes
- Weather stations

ASCOM - Interfaces

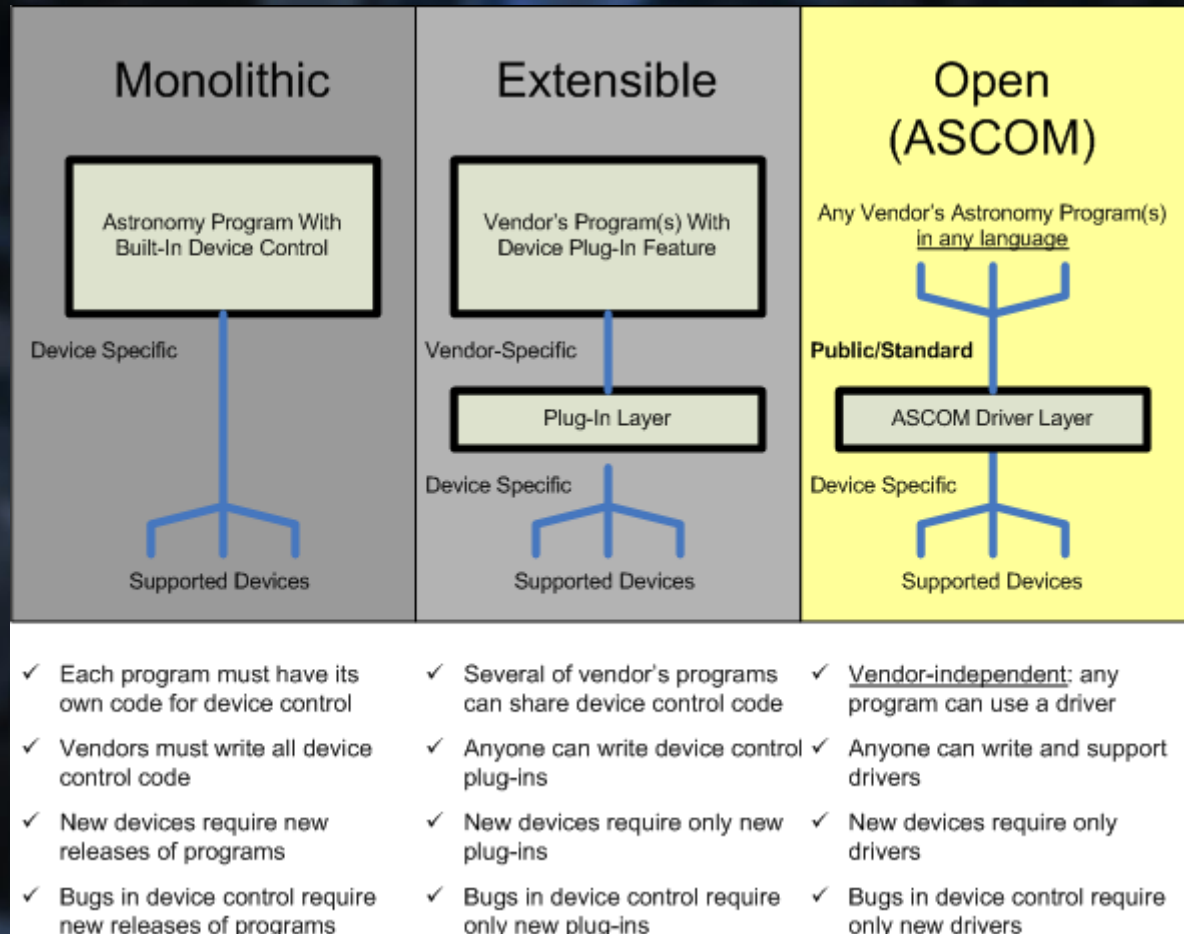
For each device there are one or more interfaces which expose methods and properties specifically thought for that device.

Example:

For CCD Cameras there is an interface called `Camera` which expose the method

```
Camera.SetCCDTemperature (Double)
```

ASCOM - Architecture

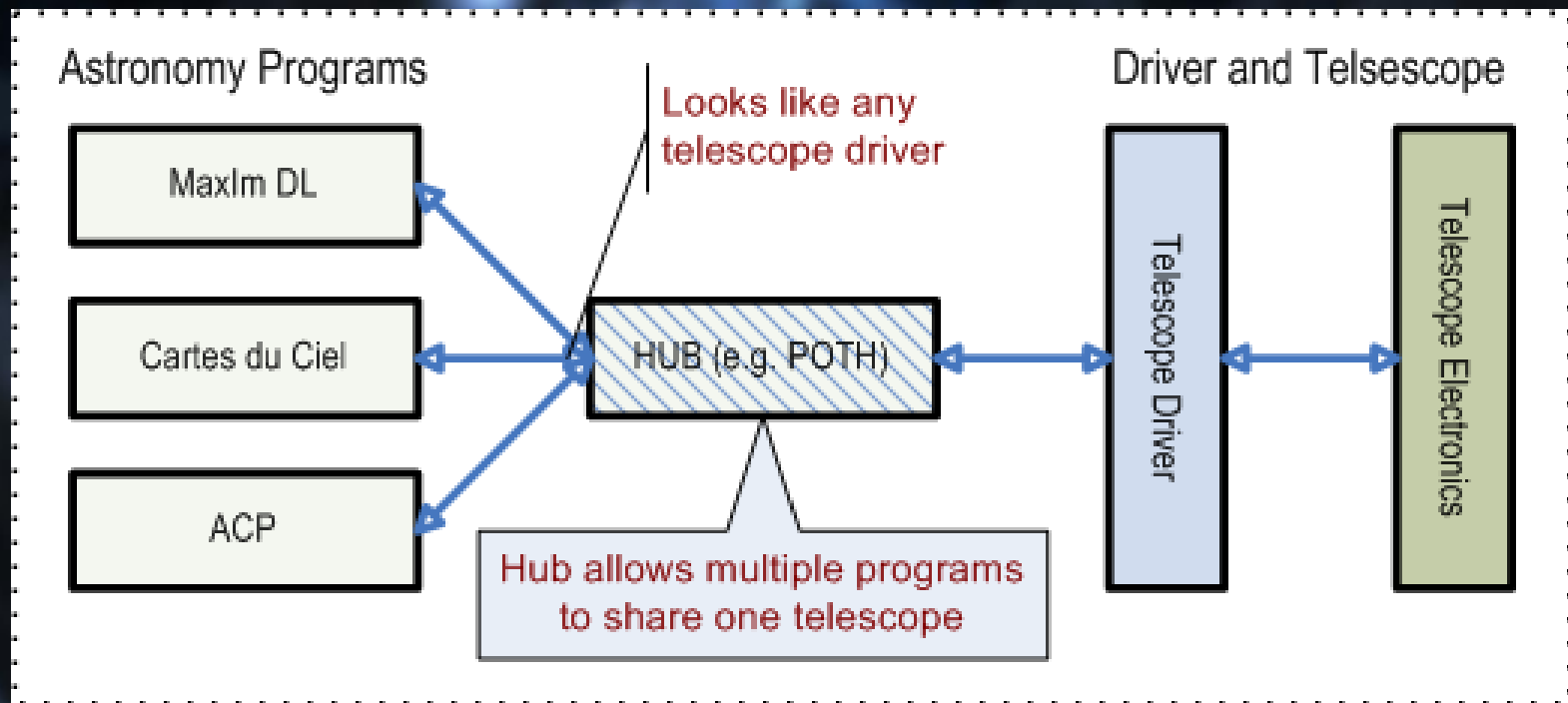


ASCOM – Pre-requisites

ASCOM requires that the .NET platform is correctly installed on the PC.

The latest version of ASCOM and the list of pre-requisites is available from the ASCOM website.

ASCOM – Mount Functional View



ASCOM – Installation

Package .NET framework 3.5.1 from Microsoft
Web Site: <http://www.microsoft.com/en-us/download/details.aspx?id=22>

ASCOM platform from the ASCOM web site
(currently, the latest version is 6.1 SP1):
<http://download.ascom-standards.org/ASCOMPlatform61SP1.exe>

ASCOM – Goodies

The ASCOM platform provides a set of tools designed to help developers with Debugging tools.

- Interface specifications
 - A set of documents regarding the methods/properties of a certain interface
- Simulators (Mount Simulator, Camera Simulator, etc.)
- Pipe connection watcher
 - Tool to monitor the communication between the ASCOM client and the Interfaces

ASCOM – Getting Support

- ASCOM-Talk mailing list
 - <https://groups.yahoo.com/neo/groups/ASCOM-Talk/info>
- Facebook Page
 - <https://www.facebook.com/groups/134421113247694/?fref=ts>

AutoIT

AutoIT is a freeware automation language for Microsoft Windows.

In its earliest release, the software was primarily intended to create automation scripts but has since grown to include enhancements in both programming language design and overall functionality.

AutoIT – Main Features

- **Supports COM objects!**
- **Simulate mouse movements.**
- Create GUI interfaces, including message and input boxes.
- Scripting language with BASIC-like structure for Windows Desktop Environment.
- Add-on libraries and modules for specific applications.
- Supports TCP and UDP protocols.
- Manipulate windows and processes.
- Automate sending user input and keystrokes to applications, as well as to individual controls within an application.

Other Languages?

Yes, possible, provided they support the
Windows COM

Examples:

Visual Basic, Visual C++, Python

AutoIT – The basics

- `$iNumber1 = 0.1`
- `$fBoolean1 = True`
- `$sString1 = «this is a string»`

Variables may be declared Local or Global, depending on the scope.

AutoIT – Arrays

```
$aArray[0] = "A "
```

```
$aArray[0][0] = "Upper-Left"
```

Values in the same array may be of different type

AutoIT – Conditional Statements

If...Then...Else

Select...Case

Switch...Case

AutoIT – if then else

```
If $iNumber > 0 Then
    MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber
was positive!")
ElseIf $iNumber < 0 Then
    MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber
was negative!")
Else
    MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber
was zero.")
EndIf
```

AutoIT – select

```
Select
    Case $iNumber > 1 And $iNumber <= 10
        MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 1")

    Case $iNumber > 10 And $iNumber <= 20
        MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 10")

    Case $iNumber > 20 And $iNumber <= 30
        MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 20")

    Case $iNumber > 30 And $iNumber <= 40
        MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 30")

    Case $iNumber > 40
        MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 40")
EndSelect
```

AutoIT – switch

```
Switch Int($iNumber)
  Case 1 To 10
    MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 1")

  Case 11 To 20
    MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 10")

  Case 21 To 30
    MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 20")

  Case 31 To 40
    MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 30")

  Case Else
    MsgBox($MB_SYSTEMMODAL, "Example", "$iNumber was
greater than 40 or less or equal to 0")

EndSwitch
```


AutoIT – Loop Statements

For...Next

While...WEnd

Do...Until

For...In...Next

AutoIT – for-next

```
For $i = 5 To 1 Step -1  
    MsgBox($MB_SYSTEMMODAL, "", "Count down!" & @CRLF & $i)  
Next
```

AutoIT – While

```
While $i <= 10
    MsgBox($MB_SYSTEMMODAL, "", "Value of $i is: " & $i)
    $i = $i + 1
WEnd
```

AutoIT – Do/Until

```
Do
    MsgBox($MB_SYSTEMMODAL, "", "The value of $i is: " & $i)
    $i = $i + 1
Until $i = 10
```


AutoIT – For..In

```
Local $aArray[4]
$aArray[0] = "a"
$aArray[1] = 0
$aArray[2] = 1.3434
$aArray[3] = "test"
```

```
Local $sString = ""
```

```
For $vElement In $aArray
```

```
    $sString = $sString & $vElement & @CRLF
```

```
Next
```

AutoIT – Functions

Example

```
Func MyDouble($iValue)  
    $iValue = $iValue * 2  
    Return $iValue  
EndFunc
```

AutoIT – Directives

Example

```
#include "[path\]filename"  
#include <filename>
```



Questions?