

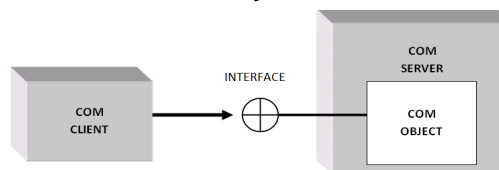
Labo Windows

Wat wordt van jou verwacht ?

- elke week 1-2 nieuwe topics
- labo is verplicht te volgen
- elk labo **verplichte** voorbereiding
- de MSDN manual gebruiken
- zoek zelf de oplossing
- controleer met de antwoorden

COM programmatie

Wat is Common Object Model ?



- COM server stelt haar functionaliteit ter beschikking via COM klassen
- COM cliënt gebruikt de COM objecten door de interface aan te spreken

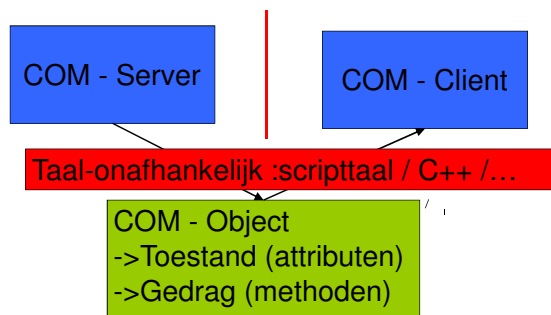
Labo Windows

Evaluatie

- 1^e test (WMI+Excel) op dinsdag 12/11
- inhaaltest in januari (tijdens examens)
- 2^e test (Ldap + Powershell)
- op theorie-examen

tweede zit : WMI+Excel / Ldap+Powershell
2 extra testen voor ½ labopunten(per deel)

COM server - cliënt



Labo Windows

Wat ?

- automatiseren van taken
- WMI : events definiëren, afhandelen
- user-beheer

Hoe ?

- COM-objecten aanspreken
- ActiveX Scripting -> **perlscript** (vbscript)
- Inleiding tot Powershell

COM programmatie

Wat is goede COM server?

- toepassing/operating system stelt haar functionaliteit ter beschikking
- klassen en standaard interfaces
- met goede documentatie

Documentatie in MSDN Library :
vbn in vbscript, C++

COM Server

- wordt geregistreerd bij installatie in HKEY_CLASSES_ROOT hive
- elke COM component (deel van de functionaliteit) heeft eigen ProgId
- en is onafhankelijk geïmplementeerd en geregistreerd.

regedit

COM Client

- aanspreken van toepassing met COM-objecten

COM - Client

binding

COM - Object

->Toestand (attributen)
->Gedrag (methoden)

Zoek op in MSDN Library

Object hiërarchieën

COM model = meerdere COM klassen die samen de COM-server implementeren

- nieuwe versie voegt functionaliteit toe

COM - Server

Objecten met :

->attributen
->methoden

Documentatie in MSDN Library :
vbn in vbscript,C++

Binding:initialiseren van een COM-object

- early binding: tijdens **compilatie** van COM cliënt
 - + zeer snelle uitvoering
 - + syntax checking van argumenten mogelijk
 - vrij complex (verwijzen naar CLSID en interfaces)
- late binding: tijdens **uitvoering** van COM cliënt
 - + enkel op basis van ProgId
 - trager
 - geen controle op argumenten bij aanroep methode
 - IDispatch interface noodzakelijk (automation objecten)
- duale interface:
 - IDispatch en rechtstreeks toegankelijk

Welke COM-servers?

- CDO (Collaboration Data Objects) : email
- Office -> Excel: eenvoudig rekenblad
- WMI: Windows Management Instrumentation
alle processen, instellingen,WQL,Events
- Active Directory (LDAP)
Users, groepen,...

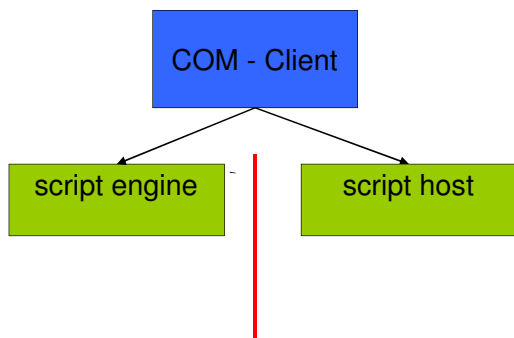
COM technologie met automation interface

manipuleren met **scripts** (geïnterpreteerde omgeving)

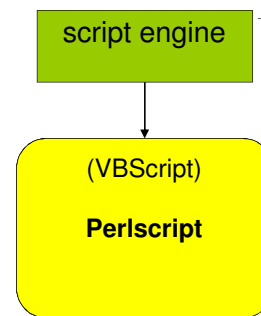
- trager
- minder goed beschermd tegen piraterij
(script encoding is beschikbaar)
- syntax fouten zijn moeilijk opspoorbaar

- + eenvoudiger, ook voor professionele programmeurs
- + vergt geen complexe ontwikkelingsomgeving
- + snelle oplossing voor kleine taken/problemen
- + combinatie van programmeertalen is mogelijk

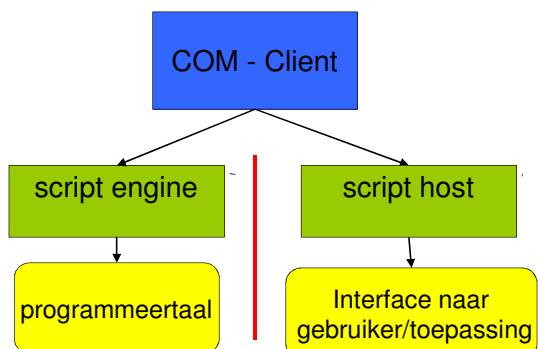
ActiveX Scripting architectuur



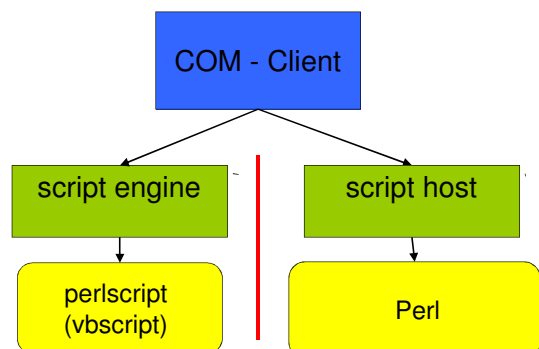
ActiveX Scripting architectuur



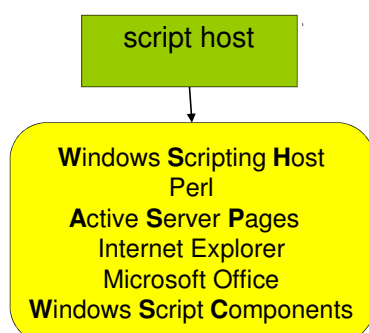
ActiveX Scripting architectuur



ActiveX Scripting architectuur



ActiveX Scripting architectuur



COM client

Script host : *Perl*
 argumenten: @ARGV
 Script engine : *perlscript*
 modules inladen: *use ...*
 OLE-module inladen: *use Win32::OLE ;*

Com-objecten initialiseren en gebruiken

Scripting Engine

- In MSDN: voorbeelden in **VBScript** (of C++)
- Omzetten naar **PerlScript**

Documentatie PerlScript :

- ActivePerl documentatie: voorbeelden in perl!
- Nota's van vorig jaar

COM Server

Meerdere COM componenten.

Elke COM component is

- container van meerdere COM-klassen
- logische deelfunctionaliteit

COM-component->geregistreerd als geheel

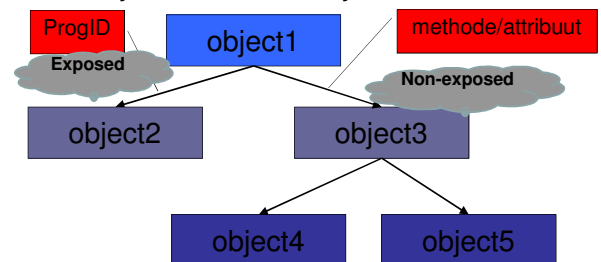
- belasting spreiden op de server

COM-Object in VBScript

- Declareer variabele/object:
`Dim obj` 'niet case-sensitive
- Initialiseer COM-object:
`set obj =`
- Attribuut opvragen:
`obj.name`
- Methode oproepen:
`obj.do(param1,param2)`

Object hiërarchieën

COM object kan nieuw object creëren



COM-Object in Perlscript

- Declareer variabele/object:
`my $obj; #case-sensitive`
- Initialiseer COM-object:
`$obj =`
- Attribuut opvragen:
`$obj->{name}`
- Methode oproepen:
`$obj->do(param1,param2)`



COM Server

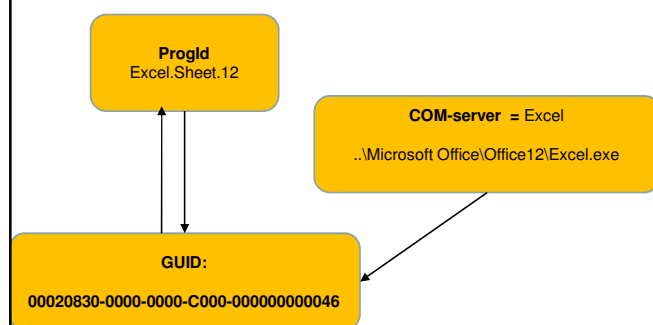
Elke COM component heeft eigen ProgId en is onafhankelijk geïmplementeerd en geregistreerd.

Dit kan op 2 manieren :

- .dll : dynamic link library
- .exe : executable

Registratie COM-component

het register (**regedit**) : **HKEY_CLASSES_ROOT**



COM-component <-> ProgId

CDO.Message -> ProgID is "CDO.Message"

- GUID (Clsid) -> zoek op in de CLSID-subtak
ProgId
InprocServer32 : cdosys.dll
geen typelib ??

Oleview:
Microsoft CDO for Windows 2000 Library

COM-component <-> ProgId

Excel -> ProgID is "Excel.Application"

-> zoek de klassen ivm *Excel*

Excel.Sheet

Excel.Application

- GUID (Clsid) -> zoek op in de CLSID-subtak
ProgId

LocalServer32 : excel.exe

Typelib beschikbaar

Oleview:
Microsoft Excel 12.0 Object Library

Labo Windows

Wat ?

- automatiseren van taken
- WMI : events definiëren, afhandelen
- user-beheer

Hoe ?

- Inleiding tot Powershell
- COM-objecten aanspreken
- ActiveX Scripting :
 - **Perl**
 - **perlscript** / vbscript (engine)

COM-Object initialiseren met ProgId

```
use Win32::OLE ; #inladen module
$excel= Win32::OLE->new("Excel.Application");
```

Excel start op (niet zichtbaar !)

- Task Manager
- indien een fout optreedt wordt Excel niet gesloten

Welke COM-servers?

- **CDO (Collaboration Data Objects)** : email
- **Office -> Excel**: eenvoudig rekenblad
- **WMI: Windows Management Instrumentation**
alle processen, instellingen, WQL, Events
- **Active Directory (LDAP)**
Users, groepen, ...

Labo Windows

Wat wordt van jou verwacht ?

- elke week 1-2 nieuwe topics
- labo is verplicht te volgen
- elk labo **verplichte** voorbereiding
- de MSDN manual gebruiken
- zoek zelf de oplossing
- controleer met de antwoorden