Lightning Talk

Use of CA Plex Model API

Sergio Mendes
Lusodata

PLEX2E.COM COMMUNITY
RACE TO THE FUTURE 2023

cm FIRST
Rethink Modernization

AMI SOLUTIONS

BROADCOM SOFTWARE

A·B·A·C·U·S SOLUTIONS
BUSINESS UNIT OF FRESCHE SOLUTIONS

REMAIN SOFTWARE

LUSODATA

HAWKBRIDGE

Client Infrastructure

Client / Server:

- IBM i (AS400) Servers
- Windows PC Clients

Software Updates Distribution (DevOps)

Development:

- Portugal

Tests/QA/Production Environments:

- Portugal
- Greece
- Egypt
- Saudi Arabia
- Morocco
- Spain

LUSODATA PDM Company

Object Collection and Distribution

Server Objects:

- Collected and distributed manually
- Manually updated by SysAdmin

Client Objects:

- DLLs / PNLs / EXEs
- How to collect the correct objects in a controlled way ?
- Deployed objects would have to be fetched by a “check for updates” mechanism

Know what objects to collect

Plex Lists:

– Are the main repository for logical object collection at development time

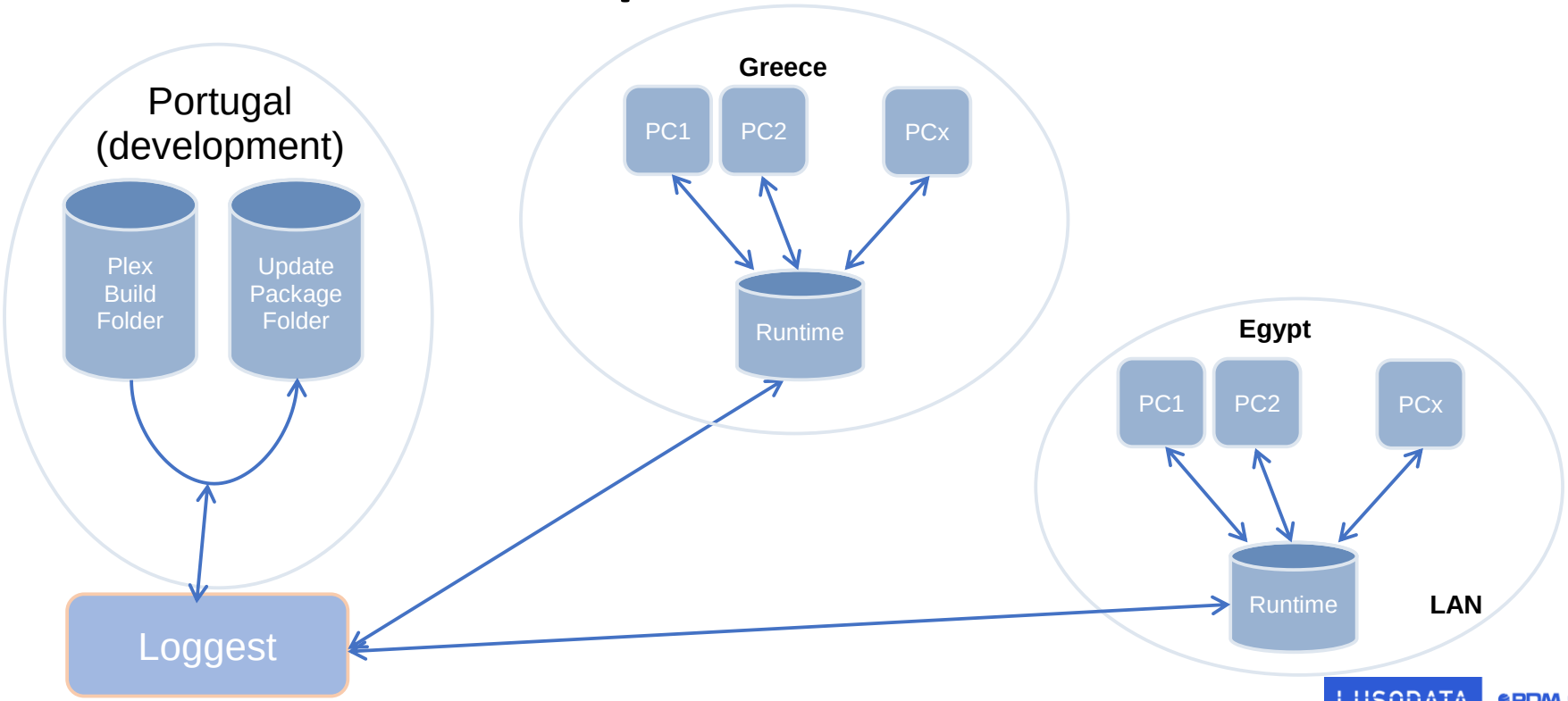
– Functions in the list are compiled, analyzed and their DLLs/PNLs stored inside “Update Packages”.

Solution:

- Make use of “CA Plex Model API” to process those lists

- Develop a Plex App, called: “LogGest”

Software Updates Distribution



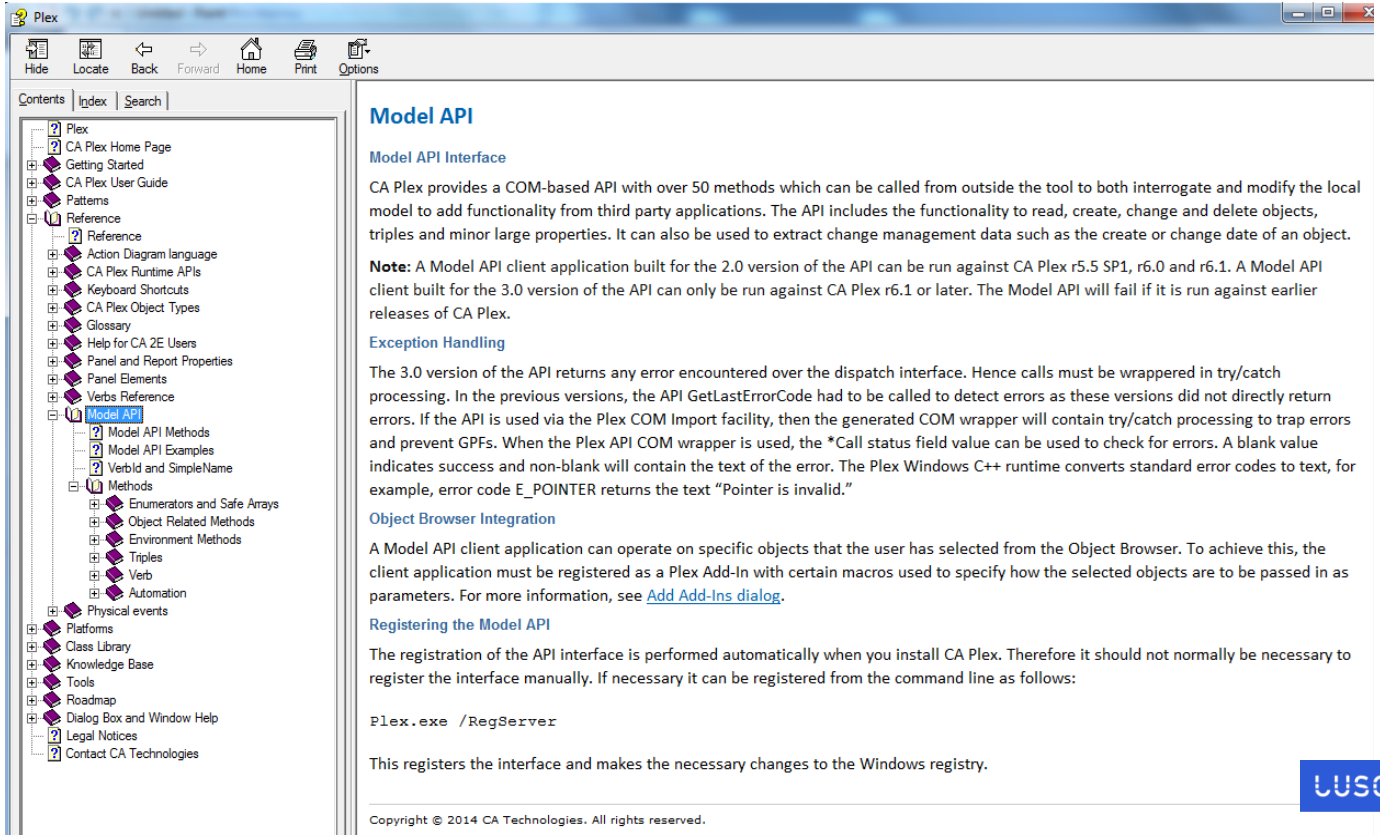
The Model API (using C++)

Plex Model API: COM-based API with over 50 methods

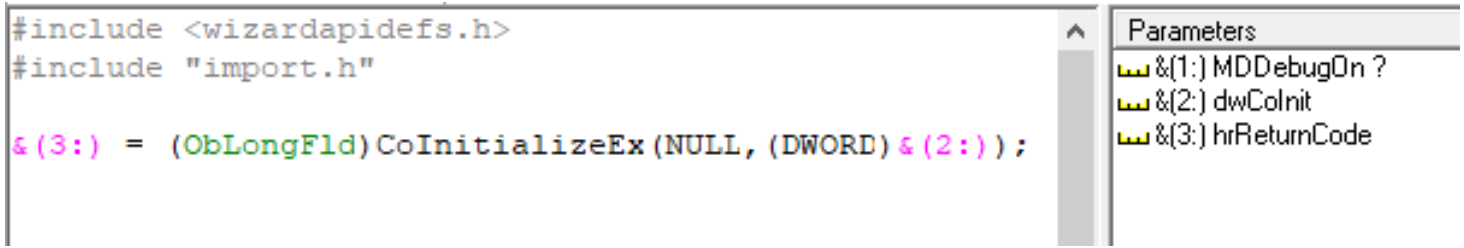
- Allows external applications to interrogate and even modify Local Models.
- Can be used to extract “Change Management” data
- Documented in Plex Help
- Used version 1.0 back in 2007

Current Version is 3.0, has better exception handling.

Plex Model API Documentation



Plex Model API – some source codes




```
&(2:) -> Concurrency Level (usually, COINIT_MULTITHREADED)
&(3:) -> Return Code (S_OK is good)
```

Plex Model API – get List Ptr

```
#include <wizardapidefs.h>
#include "import.h"
{

    int ret_val;
    long listobject = 0;

    try {
        IPlexAPIPtr p(__uuidof(PlexAPI));
        ret_val = p->FindObj((LPCTSTR)&(1:), _TypeList, &listobject);

        p = NULL;
    }
    catch (_com_error e) {
       AfxMessageBox(e.ErrorMessage());
    }


    &(2:) = (ObLongFld)listobject;
}
```

List Name (parameter 1)

List object pointer (used by other APIs)

Parameters

- &(1:) LG List Name
- &(2:) LG List Ptr

Plex Model API – get FNC name

```
#include <wizardapidefs.h>
#include "import.h"
#include <atlconv.h>

#ifndef av_USES_CONVERSION
#define av_USES_CONVERSION
USES_CONVERSION;
#endif
{
&(2:) = " ";


    int ret_val;

    try {
        IPlexAPIPtr p(__uuidof(PlexAPI));
        BSTR name = 0;
        ret_val = p->GetObjName((long)&(1:), &name);

        &(2:) = (LPCSTR)W2A(name);

        p = NULL;
    }
    catch (_com_error e) {
       AfxMessageBox(e.ErrorMessage());
    }

}
```

Function object pointer

Function's name

Parameters

&(1:) LG Object Ptr

&(2:) LG Object Name

Plex Model API – last step

Unitialize (free) COM

```
CoUninitialize();
```


LogGest DEMO

Contact

to fill in

sergio.mendes@lusodata.pt

www.lusodata.pt