B-KH

EntwicklerCamp '06 Lotus Notes

Notes Development – a walk on the wild side.

Bill Buchan

hadsl.com

Agenda

- Lotuscript and Java
- The Notes Object Model
- Visual Basic and the COM interface
- Diving Deeper – why ?
- Notes C++ interface
- Notes C-API interface
- LSX interface
- Web Services
- Questions

Lotuscript and Java

- Lotuscript
  - Lotuscript has been in Notes since v4.0
  - Major language change in v4.6
  - Language is basically unchanged since v4.6
- Java
  - Been in Notes since v4.6
- BOTH
  - access Domino via the Notes Object Model
    - This has been constantly enhanced with each new release.
  - are interpreted p-compiled languages
  - are platform independent.
  - Are distributed within Lotus Notes applications, replicated.

The Notes Object Model

- A set of classes which allow access to notes capability
- Available in front end and back end code
- Reliable, robust, performant
- Written as a set of C++ classes, calling low-level Notes C-API.

Visual Basic and the COM interface

- A good combination
  - Simplicity of Lotuscript
  - GUI capability of Visual Basic
- Deployment issues
  - Has to be deployed outside the notes database – no replication
  - Has to be able to find the local copy of Notes

Diving Deeper – Why ?

- You may wish to interface Notes with another system
  - Example – the LSX Data suite.
- You may wish to write a server add-in task.
- You may wish to expose Notes functionality that is not available via the Notes Object model.
- Use the language (or languages!) that gives you the best fit for your business problem.

Diving Deeper – Why ?

- Performance may not be improved
  - Still using Domino forms and views at the end of the day
  - A small percentage performance increase
- Deployment issues
  - You will have to deploy this application using more traditional routes – such as an installer, shared drive, etc.
  - Lose some of the convenience of the Notes application platform

Diving Deeper – Why ?

- Traditionally API is used for:
  - Server add-in tasks
    - You can “monitor” databases and processes for changes
      - Example: Virus scanners
    - You can write mail routers
      - Example: Interface mail with a bespoke mail system
      - Less popular now. Most systems understand SMTP
    - You can perform integration work which does not impact the users
      - Example: Blackberry enterprise server
    - You can interface with other systems
      - Example: LSX DO, SAP, etc.

Diving Deeper – Why ?

- Is it easier?
  - Absolutely not.
  - Expect to spend between 4-10 times MORE on API-based programs
  - Debugging is “difficult”
  - No safety net as you get closer to the metal. Some errors will take the server down.
  - Lots and Lots of testing.
    - Functional testing
    - Stability testing
    - Load testing

Diving Deeper – Why ?

- Remember
  - You are there to solve business problems in an economic manner for internal or external customers.
  - Use the best features of different levels of languages to
    - minimise work effort
    - Increase flexibility, code reuse. After all, the process may change!

Notes C++ Interface

- Gives an easy, structured way to access low-level capability.
- Is platform dependant.
- Creates an Executable, which you then have to distribute.
- Wraps complex low-level objects in classes, and deals with memory management.
- You can extend this with Notes C-API calls
  - but you have to deal with your own memory management

Notes C++ Interface - example

```
#include <stdafx.h>
// The Notes C++ API header file.
#include <LNCPPAPI.H>
// The following 2 lines are required to use any of the standard C functions such as cout.
#include <iostream>
using namespace std; // VS2003 uses namespaces so add this line. You could prefix cout with std::cout instead.

int main(int argc, char *argv[])
{
    char errorBuf[LNERROR_MESSAGE_LENGTH];
    LNNotesSession session;
    LNDatabase nab;
    LNSetThrowAllErrors( TRUE ); // get the API to trow catchable errors rather than return a status code from
    try
    {
        session.Init();
        session.GetDatabase("names.nsf", &nab, "server1/organisation");
        nab.Open();
        cout << "Opened database names.nsf - it's title is " << nab.GetTitle() << endl;
    }

    catch (LNSTATUS error )
    {
        LNGetErrorMessage( error, errorBuf);
        cout << "Error: " << errorBuf << endl;
    }
    nab.Close();
    session.Term();
    return 0;
}
```

Notes C-API interface

- Gives access to some of the same low-level Notes C-API functions that Iris use – 700+ functions.
- Is platform dependant.
- Creates an Executable, which you then have to distribute.
- YOU are responsible for memory management.
- “Running with scissors”.

Notes C-API interface - demo

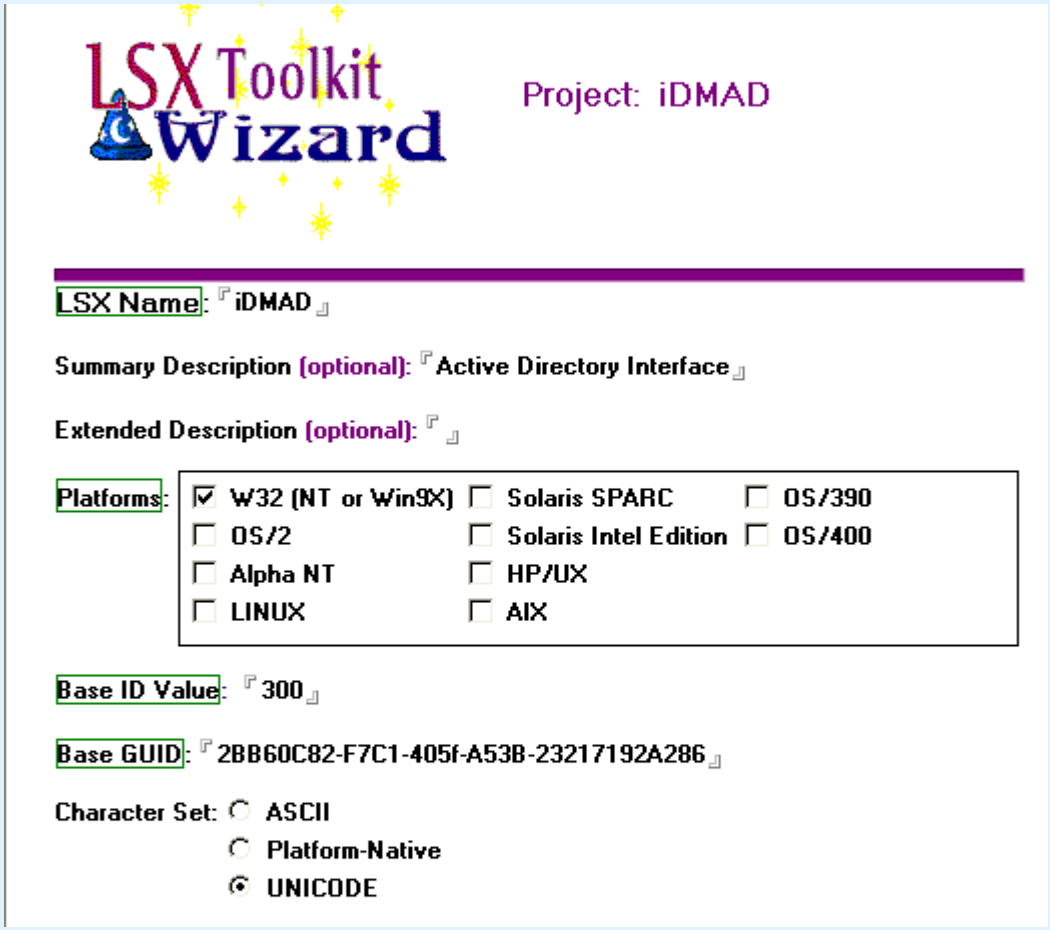
•

LSX interface

- Is platform dependant.
- Creates an Executable, which you then have to distribute.
- Wraps complex low-level objects in classes, and deals with memory management.
- YOU have to deal with memory management if you use Notes C-API calls within your LSX.
- Interfaces with Lotuscript and Java.

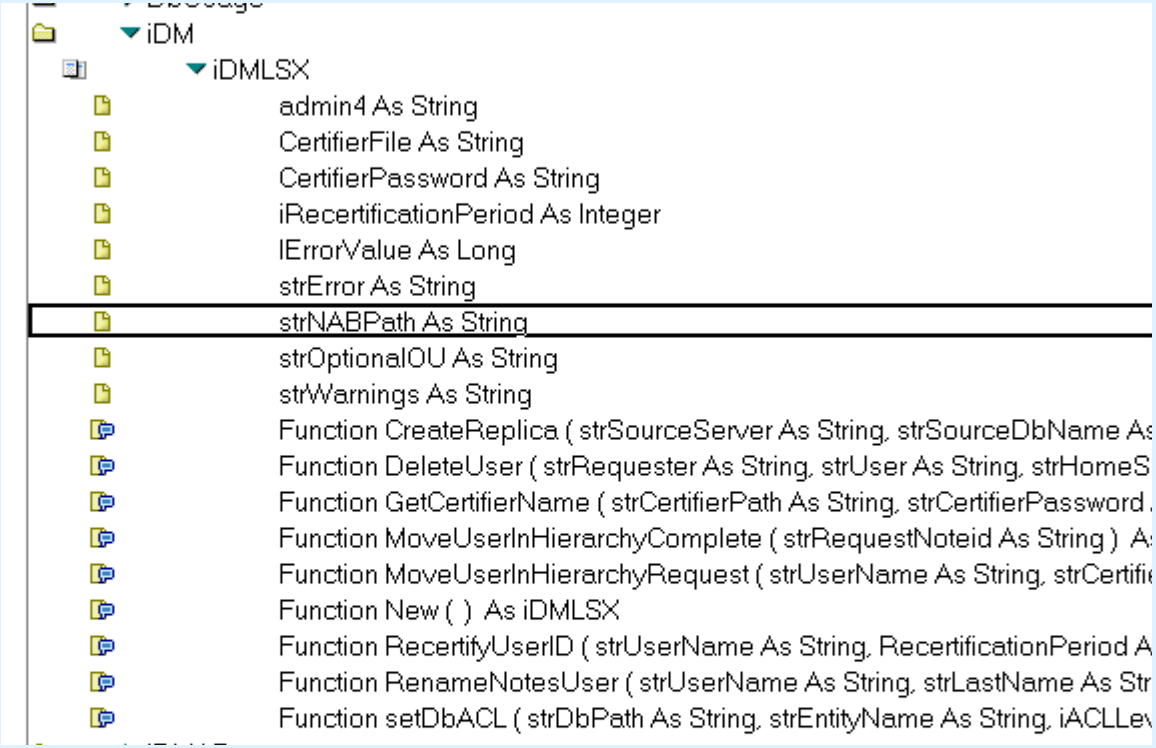
LSX interface – How ?

- Create a framework using the LSX Toolkit



LSX interface – How ?

- Define Methods and properties



LSX interface – How ?

- Generate the code
- All code is created in the c:\lsx\src\<project> directory
- Stub functions created
  - Oh. God. What does this look like ?
  - Can I change definitions afterwards ?
    - Yes.
- “make” files created for all platforms.

LSX interface – Example Code

```
// ---------------------------------------------------------------------------
LSSLONG firmADUserClass::userAccountPasswordUnique(LSSSHORT PasswordsAreUnique)
{
   //{{LSX_AUTHOR_CODE_Method_userAccountPasswordUnique

    if (PasswordsAreUnique)
    {
        m_strDebugString = L"userAccountPasswordUnique: Forcing Unique Passwords";
        m_HR = pUser->put_RequireUniquePassword(true);
    }
    else
    {
        m_strDebugString = L"userAccountPasswordUnique: Allowing non-unique passwords";
        m_HR = pUser->put_RequireUniquePassword(false);
    }

    if (SUCCEEDED(m_HR))
    {
        m_strDebugString = L" - Succeeded";
        m_HR = pUser->SetInfo();
        pRootDSE = pUser;

    }
    else
    {
        m_strDebugString = L" - Failed";
    }

    pUser->GetInfo();

    WorkOutADSIError();

    return m_HR;

   //}}
```

LSX interface – Example Code

t Library - IBM Lotus Domino Designer

quest Processor v2.1... | class:IMRequestADUserResetP...

Objects | Reference

Domino: Classes

- Button
- Field
- Firmaddirectoryclass
- Firmadgroupclass
- Firmadobjectclass
- Firmaduserclass
  - Properties
  - Methods
    - Getattribute(Strattributename As String,
    - Listattributes() As Variant
    - Listpossibleattributes() As Variant
    - New(Strusername As String, Strpasswo
    - Reloadobject() As Long
    - Removeattribute(Strattributename As St
    - Setattributelong(Strattributename As Str
    - Setattributestring(Strattributename As S
    - Setattributestringarray(Strattributename
    - Useraccountenabledisable(Byval Enabl
    - Useraccountpasswordautoexpire(Byval
    - Useraccountpasswordchange(Byval Str
    - Useraccountpasswordexpiry(Byval Neve
    - Useraccountpasswordrequired(Byval Pa
    - Useraccountpasswordunique(Byval Pas
    - Useraccountpasswordusercanchange(E
    - Useraccountunlock() As Long
    - Workoutadsierror() As String

class:IMRequestADUserResetPasswordClass (Script Library) : (Options)

```
Option Public
Option Declare
Use "class:IMBaseADUserRequestClass"
Use "class:IMSupportCheckDominoVersionClass"
Uselsx "?firmAD.dll"
```

Lotuscript C-API calls

- Is platform dependant.
- Is contained within the notes database and can be replicated
- You have to deal with memory management.
- Allows calls to low-level Notes functionality from high-level languages.
- No Mercy. Expect RBOD.
- http://www.ls2capi.com

Web Services

- The way forward.
- A http-like protocol (in most cases) moving XML wrappered data around.
- Solves a different set of problems.

Questions

- How long does it take to write a good Notes C-API Program ?
  - Actually, the documentation around the C-API is pretty good and has lots of example code.
  - If you are a good C Programmer, it's a fairly simple interface.
  - Just remember to watch the handle allocation and deallocation!

Thank you!

- Bill Buchan
  - Personal blog at http://www.billbuchan.com
- Hadsl
  - Company website at http://www.hadsl.com