



SmartPlant P&ID Automation

An Enabler for Process Solutions

Objectives

- Understand Unified Modeling Language (UML)
- Understand Data Model of SPPID
- Understand Fundamentals of Logical Model Automation (LLAMA)
- Understand Placement Automation (PLAICE)
- Understand PID Automation (PIDAuto)
- Understand and modify Calculation/Validation interface of SPPID
- Understand and modify Import function of SPPID
- Understand the LMAutomationUtil for From/To



Skill Requirements

- Introductory knowledge of SPPID
- Concept of relational database
- Intermediate level of Visual Basic



What is Automation?

- *Automation is the process of using objects that belong to another application to control that application and/or use its functionality.*



The Automation Layer

- *Technology*
- *Structure*
- *Implementation*
- *Capabilities*
- *Limitations*
- *Version Compatibility*



The Technology

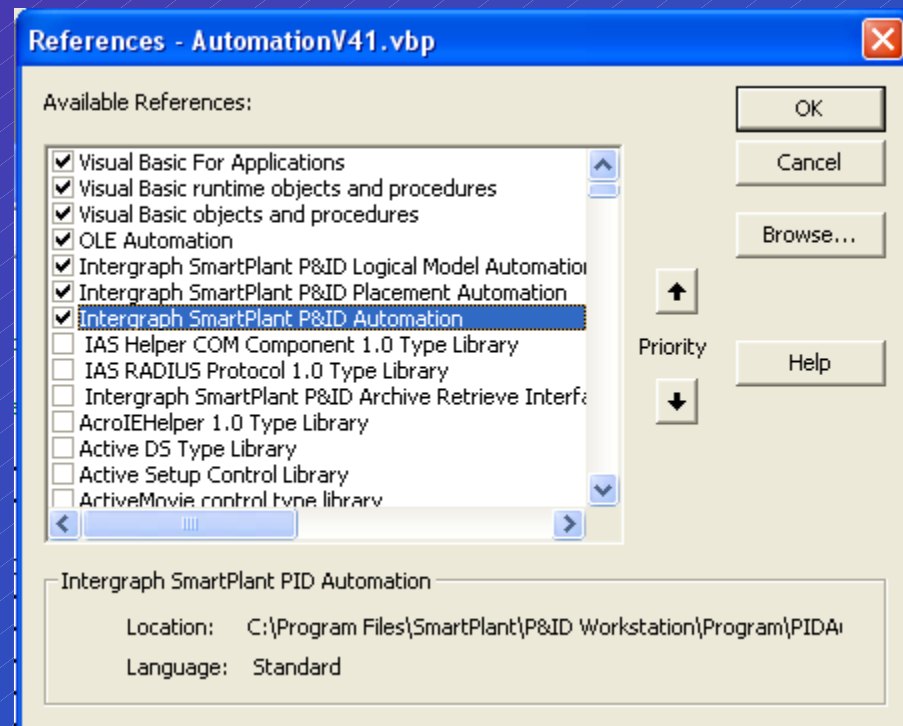
- *COM Specifications*
- *OLE Automation*
- *Standard Development Tools – Visual Basic*



The Structure

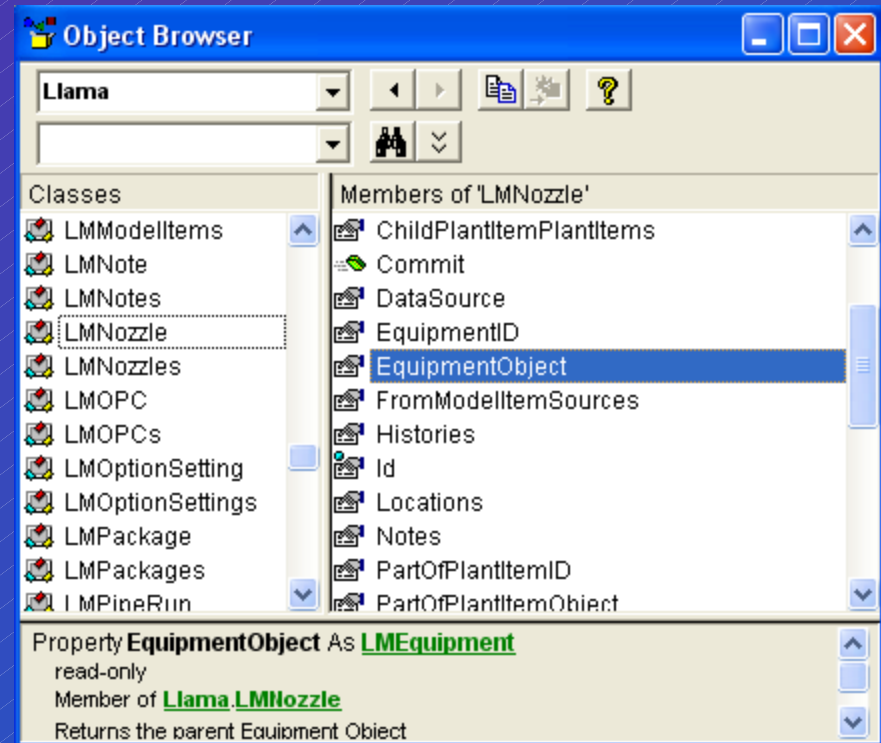
Automation Libraries

- *Can be referenced into any COM compliant development environment*
- *Four libraries provided for external Automation:*
 - LLAMA
 - PLAICE
 - PIDAutomation
 - LMAutomationUtil



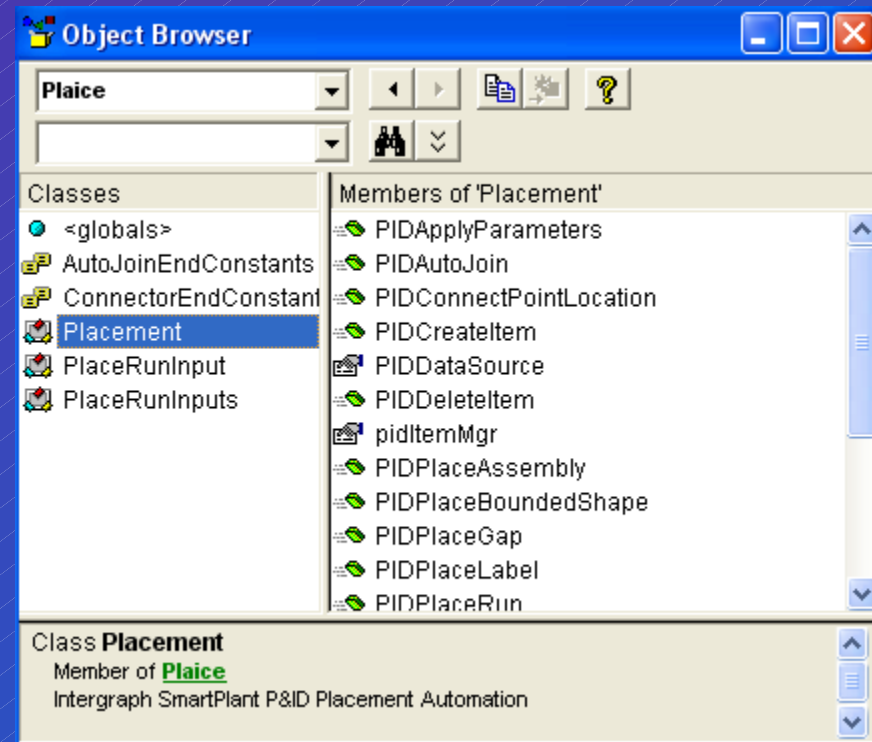
Logical Model Automation

- *A Class library*
- *Classes represent the data model*
- *Methods reflect the properties and data connectivity*
- *Provides a logical structure to access and manipulate data*



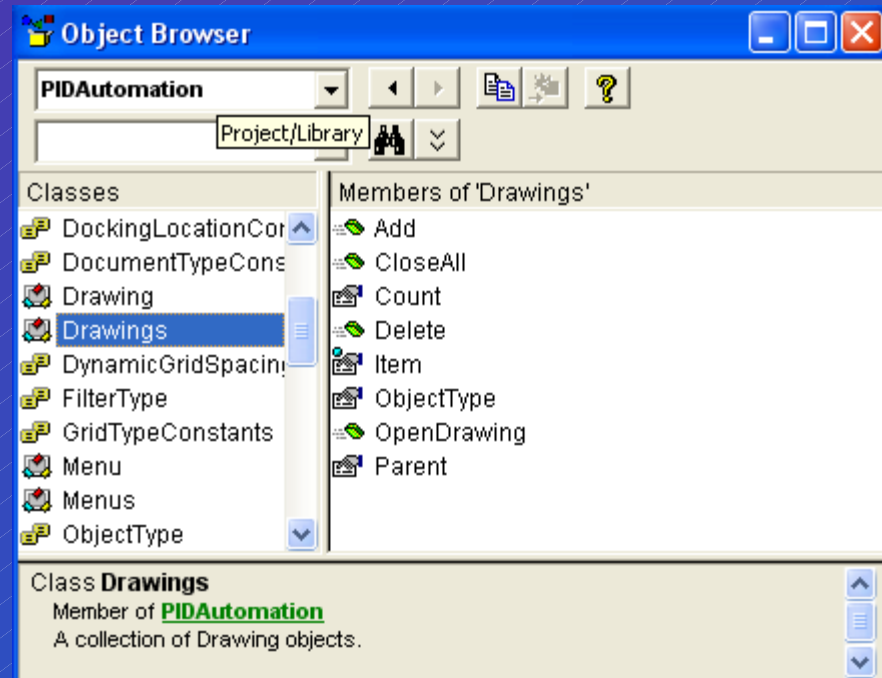
Placement Automation

- *A library of methods*
- *Place items on a drawing*
- *Interface with the Logical Model Automation (Llama) for data manipulation*



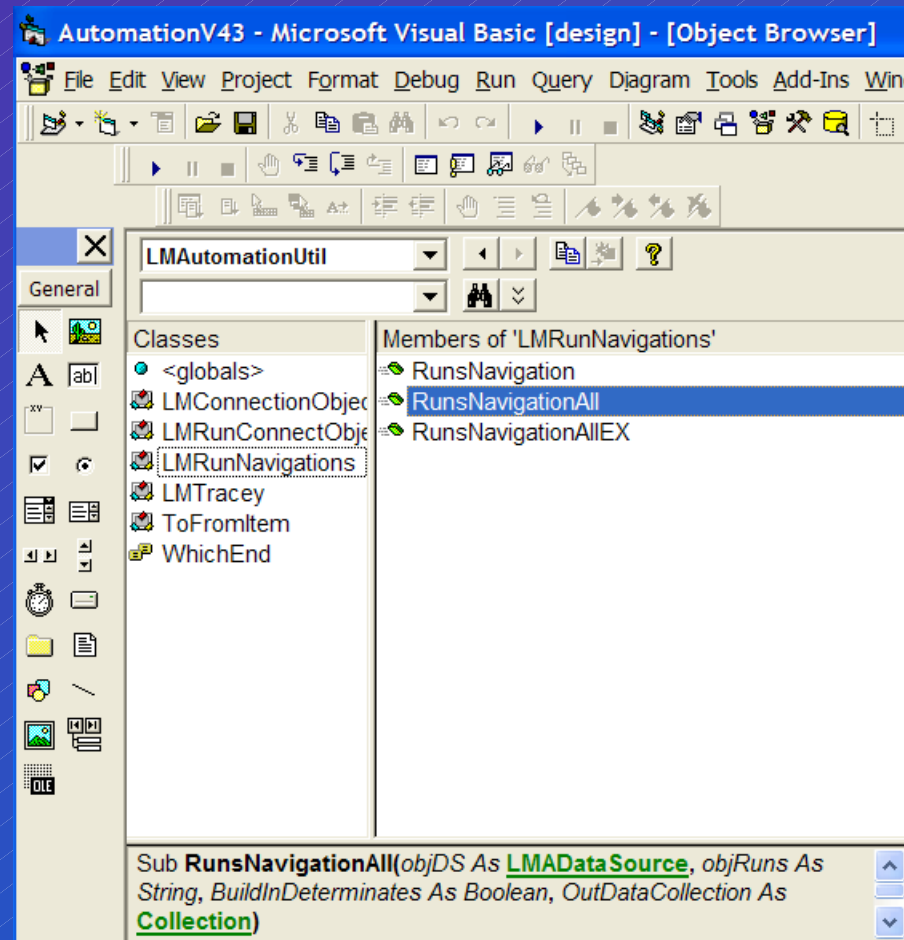
PIDAutomation

- *A Class library*
- *Only utilize three methods*
 - *Drawings.Add*
 - *Drawing.OpenDrawing*
 - *Drawing.CloseDrawing*



LMAutomationUtil

- *A Class library*
- *Only utilize two methods*
 - *RunsNavigation*
 - *RunsNavigationAll*



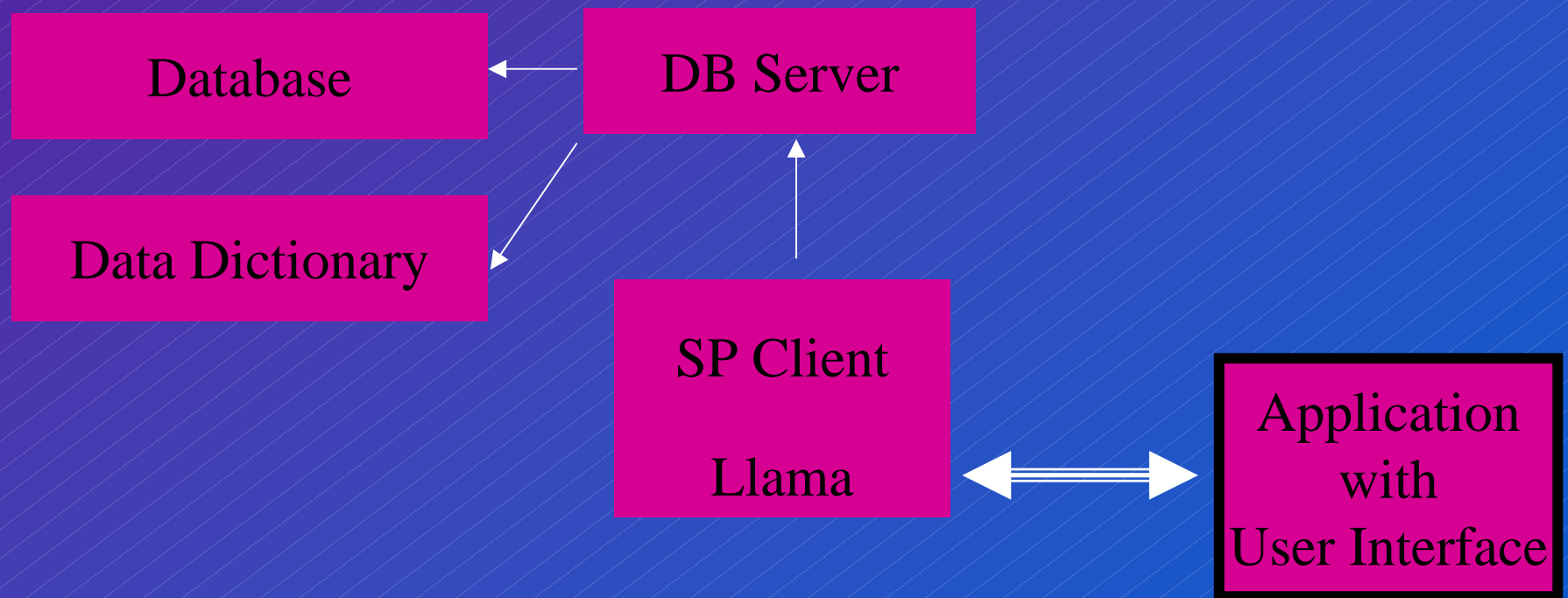


Implementation

Implementation of Automation

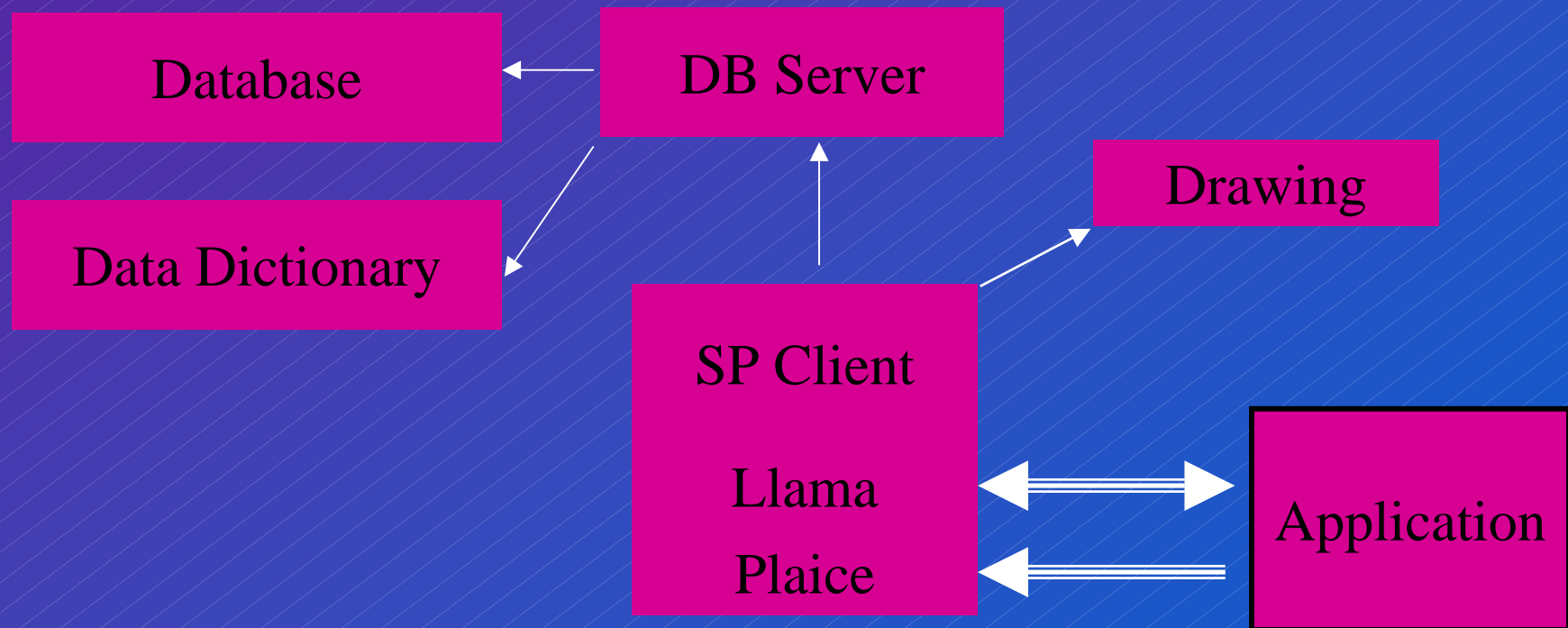
- *Stand-Alone Applications*
- *ActiveX components integrated into the SmartPlant Modeler*

Standalone Application

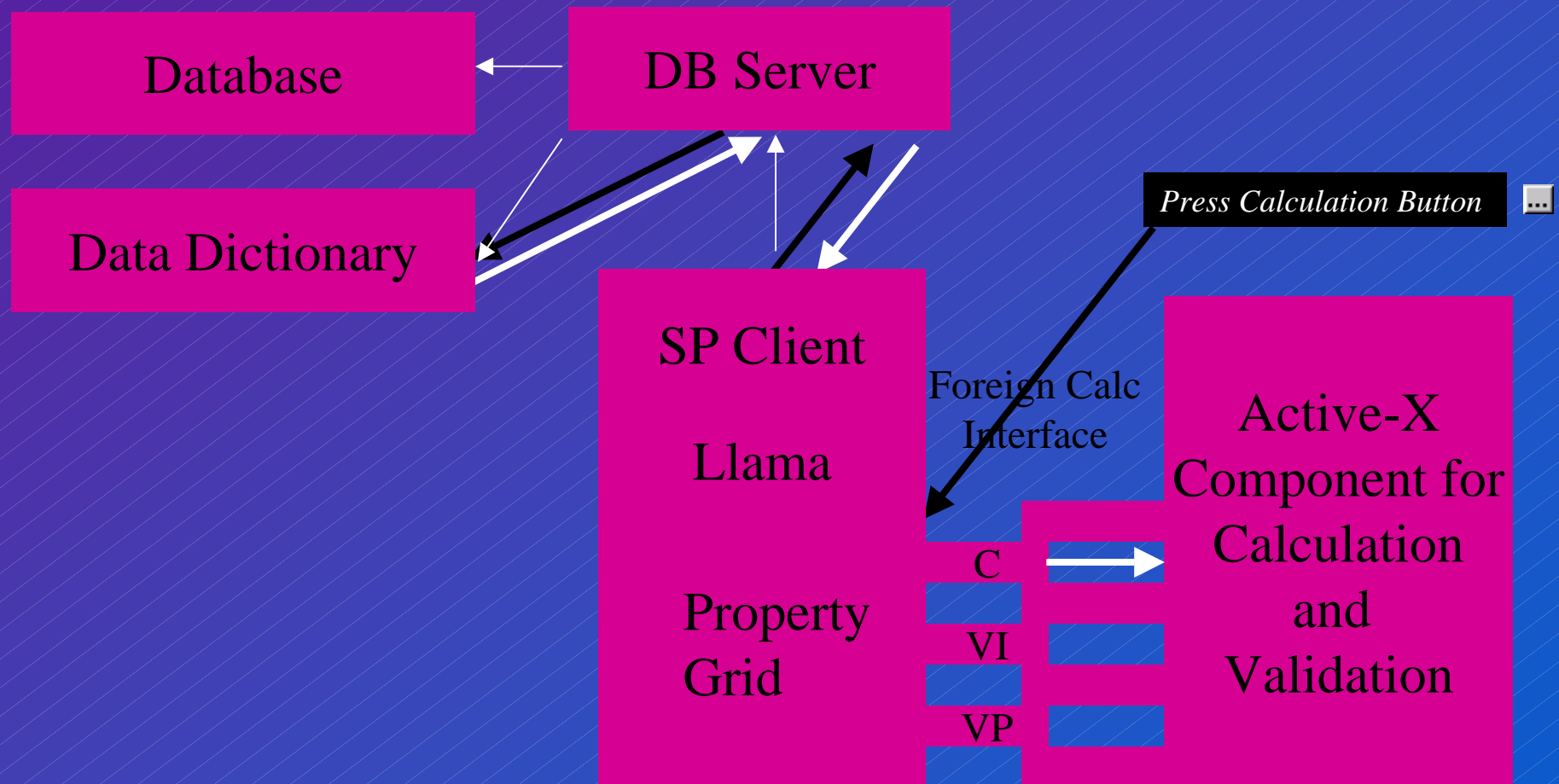


Standalone Application

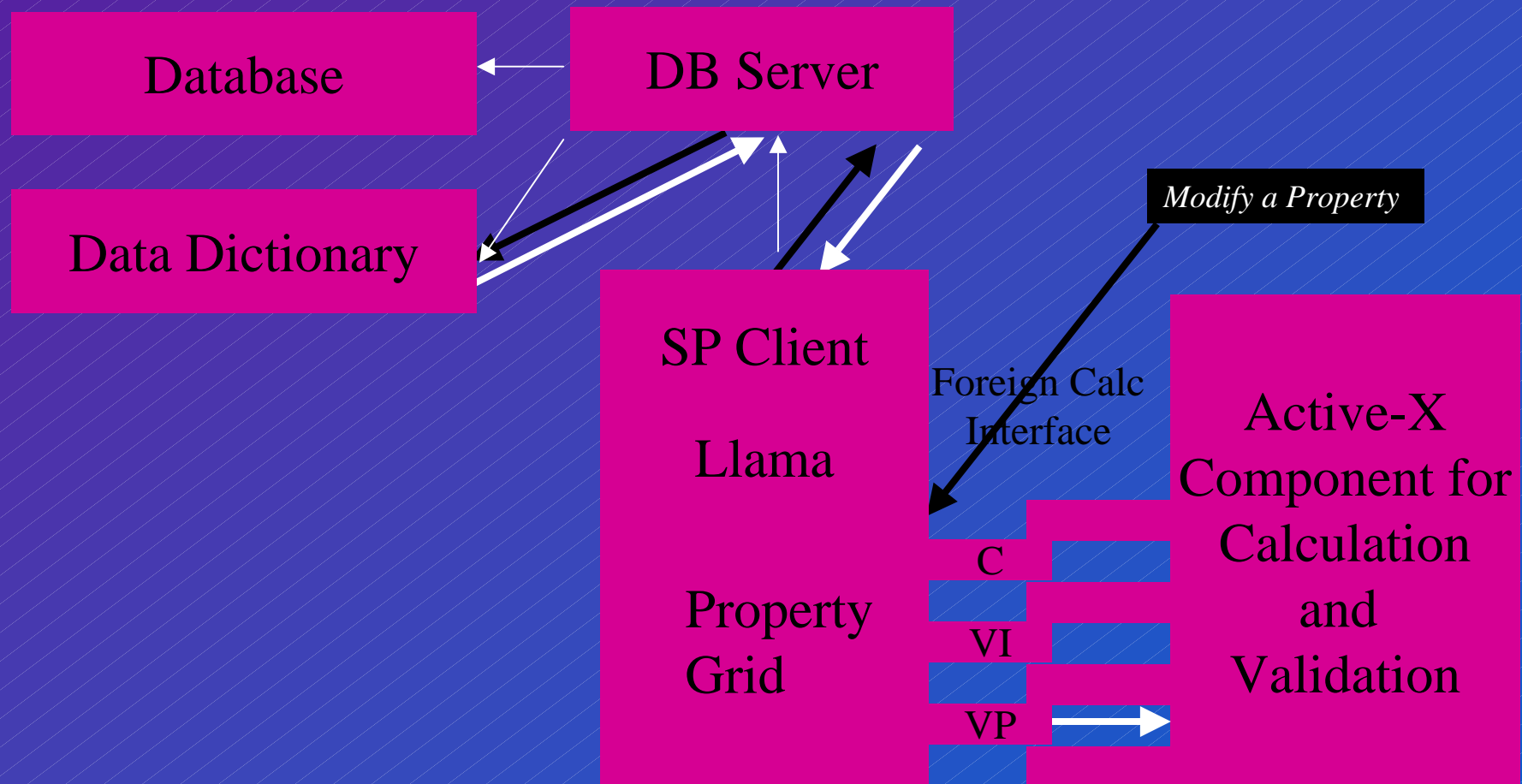
With Placement Automation



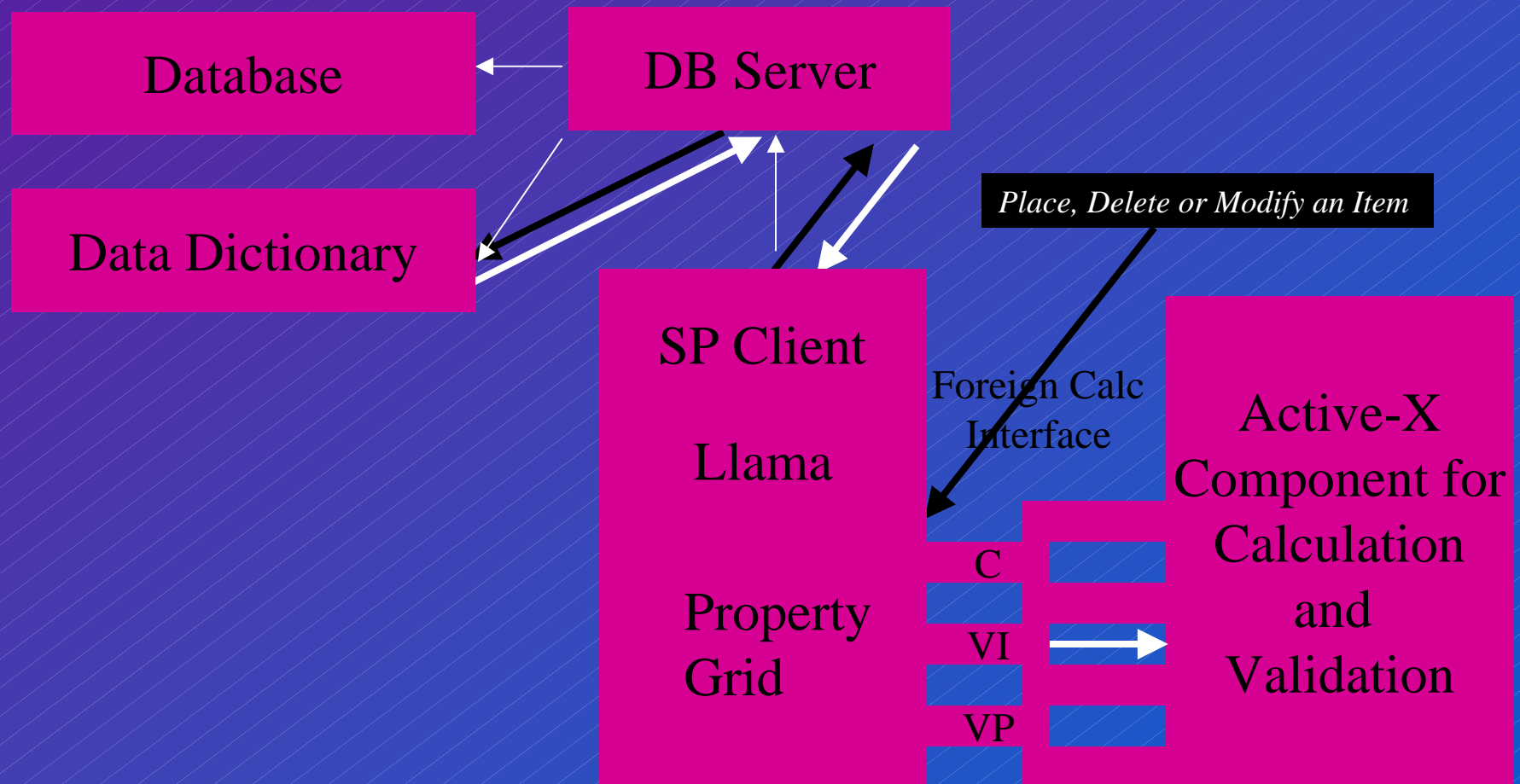
Integration - Calculation



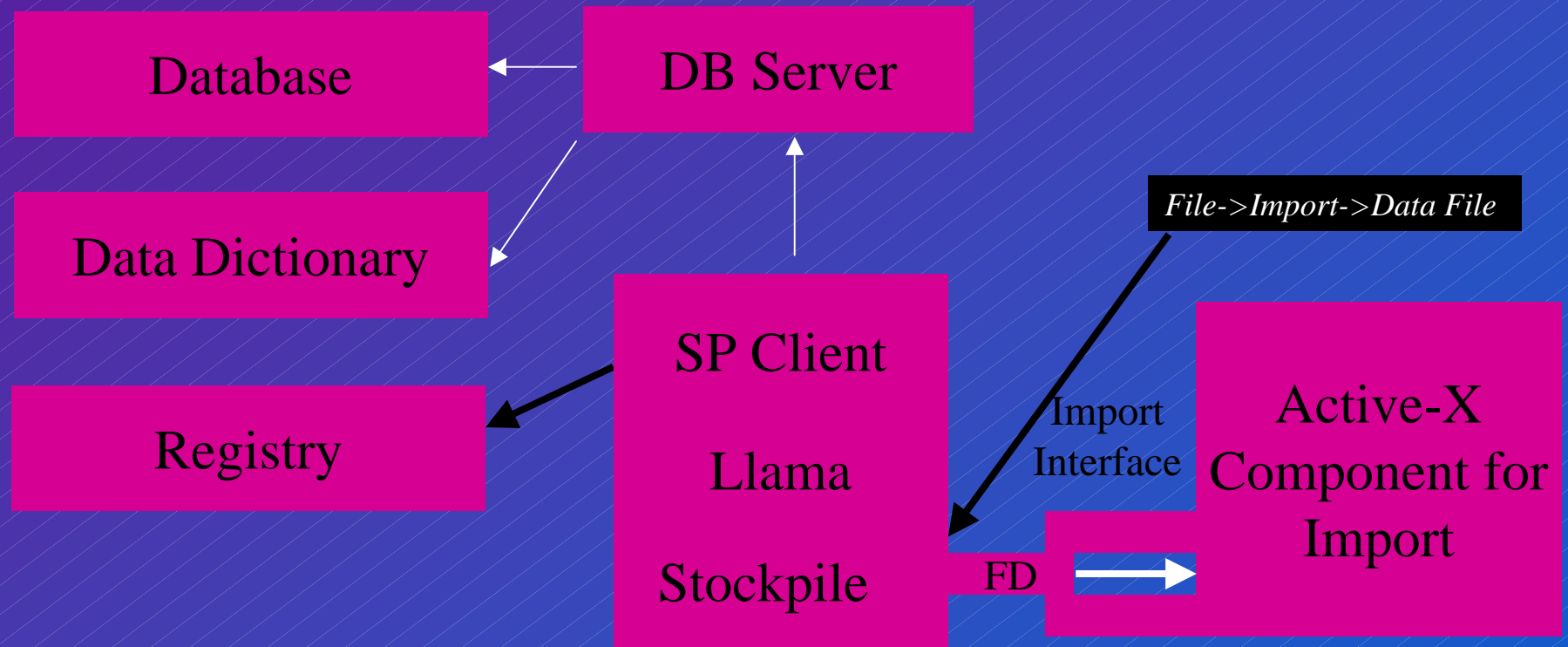
Integration - Property Validation



Integration - Item Validation



Integration - Import of Foreign Data





Capabilities



Power of Automation

- *Application Development (through standalone applications)*
- *Customization (through integration)*
- *Feature Enhancement (through integration)*



Power of Automation

- *Application Development (through standalone applications)*
- *Customization (through integration)*
- *Feature Enhancement (through integration)*



Application Development

Examples for application development:

- *Standalone application to access the database and generate a data report*
- *Standalone application to access the database and update property value*



Power of Automation

- *Application Development (through standalone applications)*
- *Customization (through integration)*
- *Feature Enhancement (through integration)*

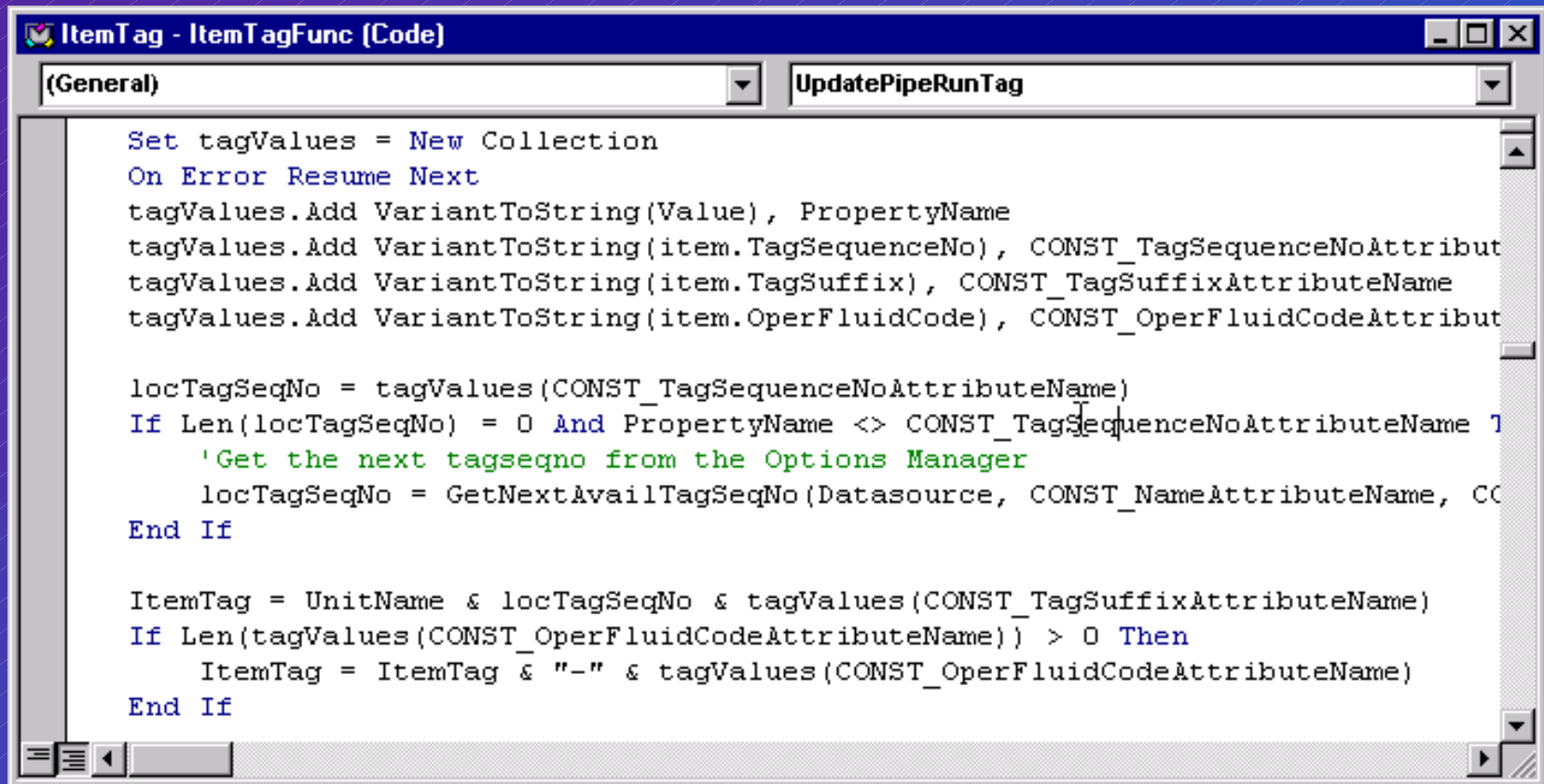


Customization

Source code provided to user for:

- Item Tag validation*
- Populating the Unit Code*
- PlantItem validation*
- Import*
- Drawing copy/paste*

Customizing Item Tag Validation



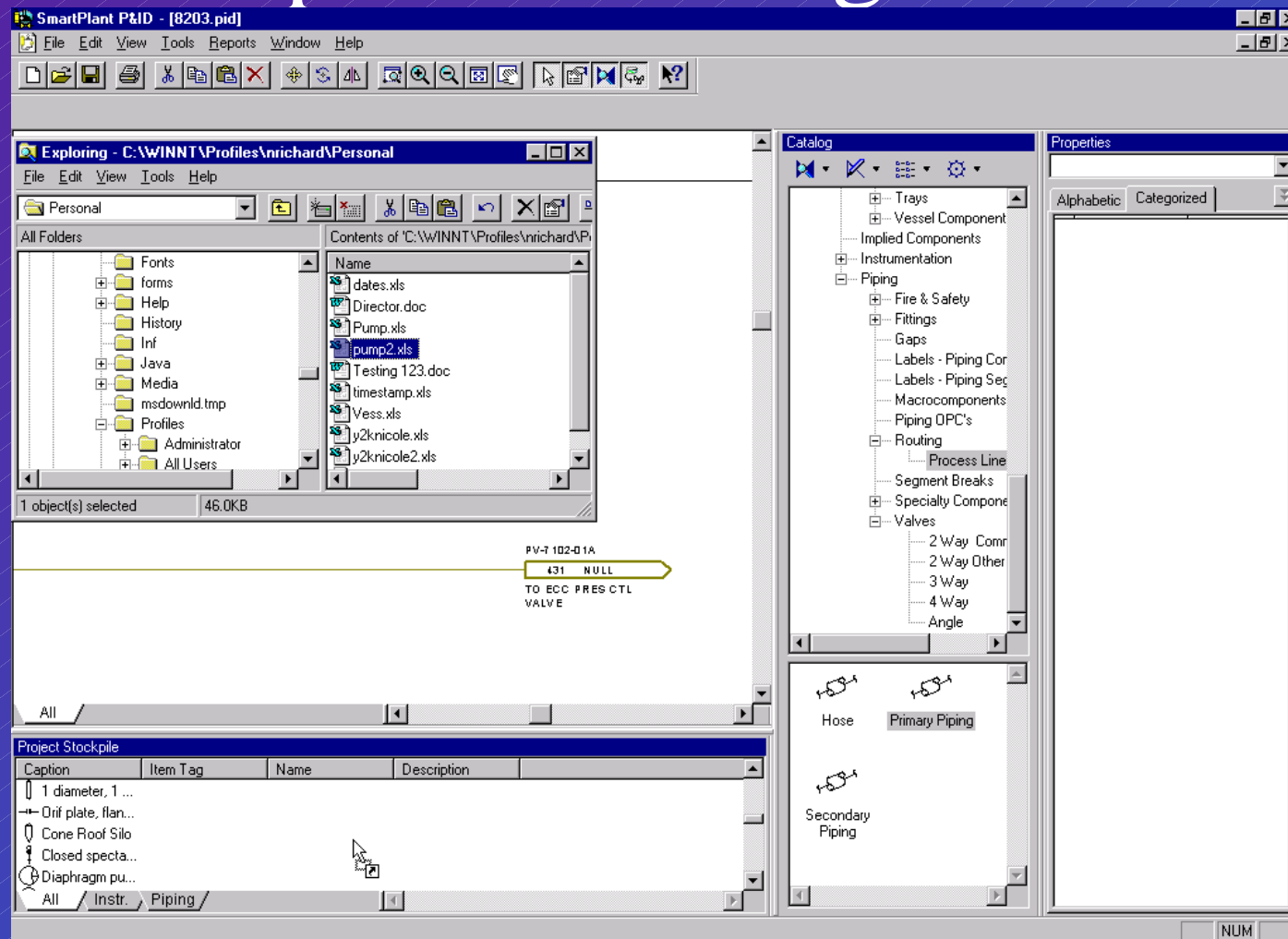
```
ItemTag - ItemTagFunc [Code]
(General) UpdatePipeRunTag

Set tagValues = New Collection
On Error Resume Next
tagValues.Add VariantToString(Value), PropertyName
tagValues.Add VariantToString(item.TagSequenceNo), CONST_TagSequenceNoAttribut
tagValues.Add VariantToString(item.TagSuffix), CONST_TagSuffixAttributeName
tagValues.Add VariantToString(item.OperFluidCode), CONST_OperFluidCodeAttribut

locTagSeqNo = tagValues(CONST_TagSequenceNoAttributeName)
If Len(locTagSeqNo) = 0 And PropertyName <> CONST_TagSequenceNoAttributeName Then
    'Get the next tagseqno from the Options Manager
    locTagSeqNo = GetNextAvailTagSeqNo(Datasource, CONST_NameAttributeName, CC
End If

ItemTag = UnitName & locTagSeqNo & tagValues(CONST_TagSuffixAttributeName)
If Len(tagValues(CONST_OperFluidCodeAttributeName)) > 0 Then
    ItemTag = ItemTag & "-" & tagValues(CONST_OperFluidCodeAttributeName)
End If
```

Import of Foreign Data



Customizing Import Module

```
SPPIDImportImpl - ImportFromFile (Code)
(General) ExcelDrop

'Determine the base directory for symbols.
sBaseSymbolDir = GetBaseSymbolDir(LMADataSource)

'Look in the workbook to find out what type of report we're importing.
bFoundImportSheet = False
For Each objSheet In objWorkbook.Sheets 'Process all sheets in the workbook
    m_Pbar.PBMessage = LoadResString(135)
    m_Pbar.PBValue = m_Pbar.PBMax * 0.3
    If objSheet.Cells(5, 1) = "EQUIPMENT LIST" Then
        Set objImport = New ImportSPEquipmentList
    ElseIf objSheet.Cells(5, 1) = "PIPE RUN LIST" Then
        Set objImport = New ImportSPLineList
    ElseIf objSheet.Cells(5, 1) = "EQUIPMENT NOZZLE LIST" Then
        Set objImport = New ImportSPNozzleList
    ElseIf objSheet.Cells(2, 24) = "EQUIPMENT LIST FOR VESSELS (V)" Then
        Set objImport = New Import2yqadVessels
    ElseIf objSheet.Cells(2, 19) = "EQUIPMENT LIST FOR PUMPS (P)" Then
        Set objImport = New Import2yqadPumps
    ElseIf objSheet.Cells(2, 22) = "EQUIPMENT LIST FOR HEAT EXCHANGERS (E)" Then
        Set objImport = New Import2yqadExchangers
    ElseIf objSheet.Cells(1, 1) = "Report Criteria and Raw Data" Then
        Set objImport = New ImportSmartSketchEquip
    End If
```

```
SPPIDImportImpl - ImportSPLineList (Code)
(General) UpdateItemProperties

Private Sub UpdateItemProperties(objPipeRun As LMPipeRun, vPropertyValues As Variant)

    objPipeRun.ItemTag = vPropertyValues(ImportAttributes.PipeRunItemTag)
    objPipeRun.CoatingReqmts = vPropertyValues(ImportAttributes.CoatingReqmnt)
    objPipeRun.HTraceReqmt = vPropertyValues(ImportAttributes.HTraceReqmt)
    objPipeRun.HTraceMedium = vPropertyValues(ImportAttributes.HTraceMed)
    objPipeRun.HTraceMediumTemp = vPropertyValues(ImportAttributes.HTraceTemp)
    objPipeRun.NominalDiameter = vPropertyValues(ImportAttributes.Diameter)
    objPipeRun.PipingMaterialsClass = vPropertyValues(ImportAttributes.PipMatlClass)
    objPipeRun.InsulPurpose = vPropertyValues(ImportAttributes.InsulPurpose)
    objPipeRun.InsulThick = vPropertyValues(ImportAttributes.InsulThickness)
    objPipeRun.ScheduleOrThickness = vPropertyValues(ImportAttributes.SchThickness)
```

```
SPPIDImportImpl - ImportSPLineList (Code)
(General) GetDataFromExcel

For i = 8 To lLastRow
    sSymbolName = objExcelSheet.Cells(i, 18)
    sID = objExcelSheet.Cells(i, 19)
    sItemTag = objExcelSheet.Cells(i, 1)
    If (sSymbolName <> "" Or sID <> "" Or sItemTag <> "") Then
        vPropertyValues(ImportAttributes.PipeRunItemTag) = sItemTag
        vPropertyValues(ImportAttributes.PipMatlClass) = objExcelSheet.Cells(i, 2).Value
        vPropertyValues(ImportAttributes.NormOpPressure) = objExcelSheet.Cells(i, 4).Value
        vPropertyValues(ImportAttributes.NormOpTemperature) = objExcelSheet.Cells(i, 5).Value
        vPropertyValues(ImportAttributes.AltOpPressure) = objExcelSheet.Cells(i, 6).Value
        vPropertyValues(ImportAttributes.AltOpTemp) = objExcelSheet.Cells(i, 7).Value
        vPropertyValues(ImportAttributes.ConstStatus) = objExcelSheet.Cells(i, 9).Value
```



Power of Automation

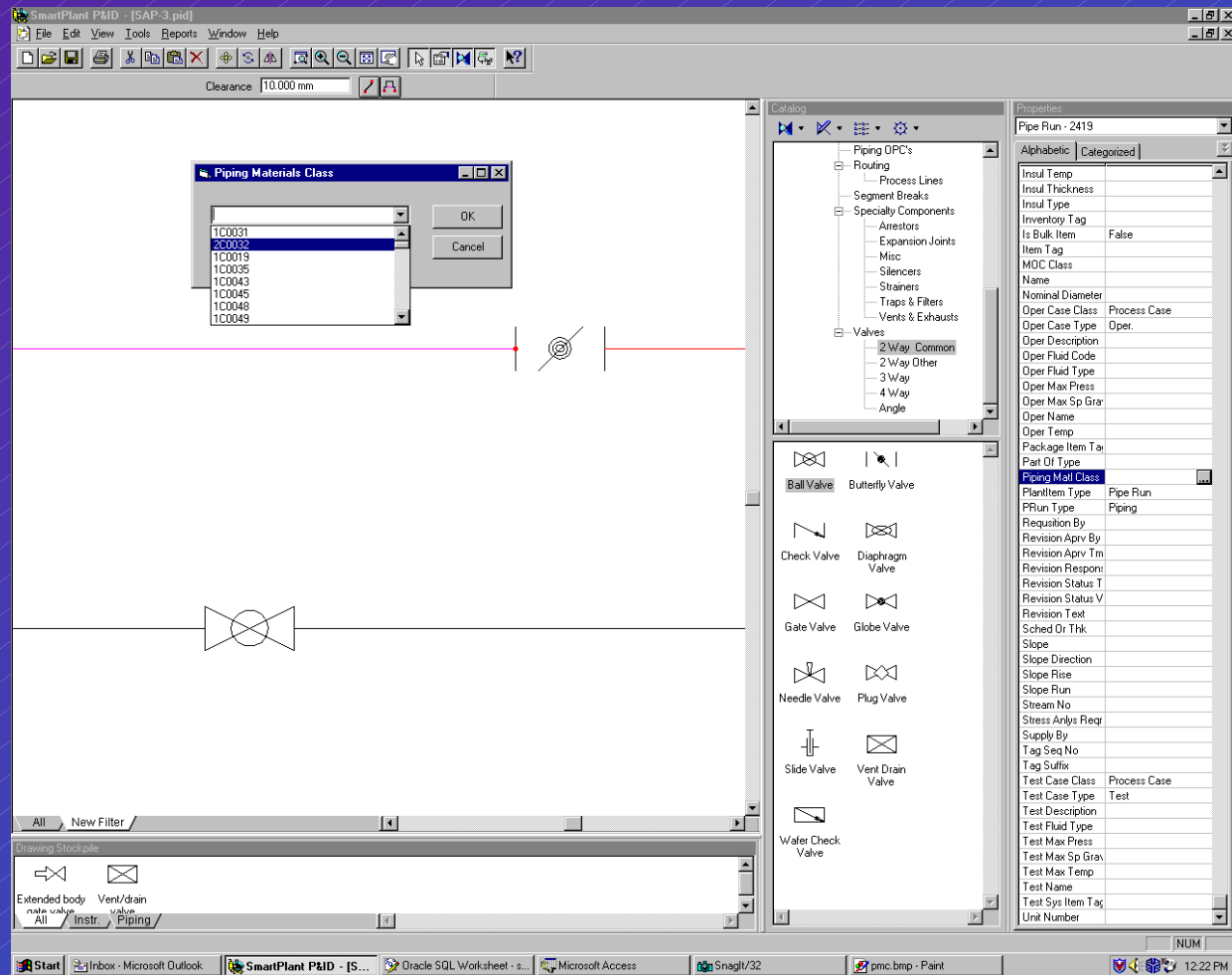
- *Application Development (through standalone applications)*
- *Customization (through integration)*
- *Feature Enhancement (through integration)*

Feature Enhancement

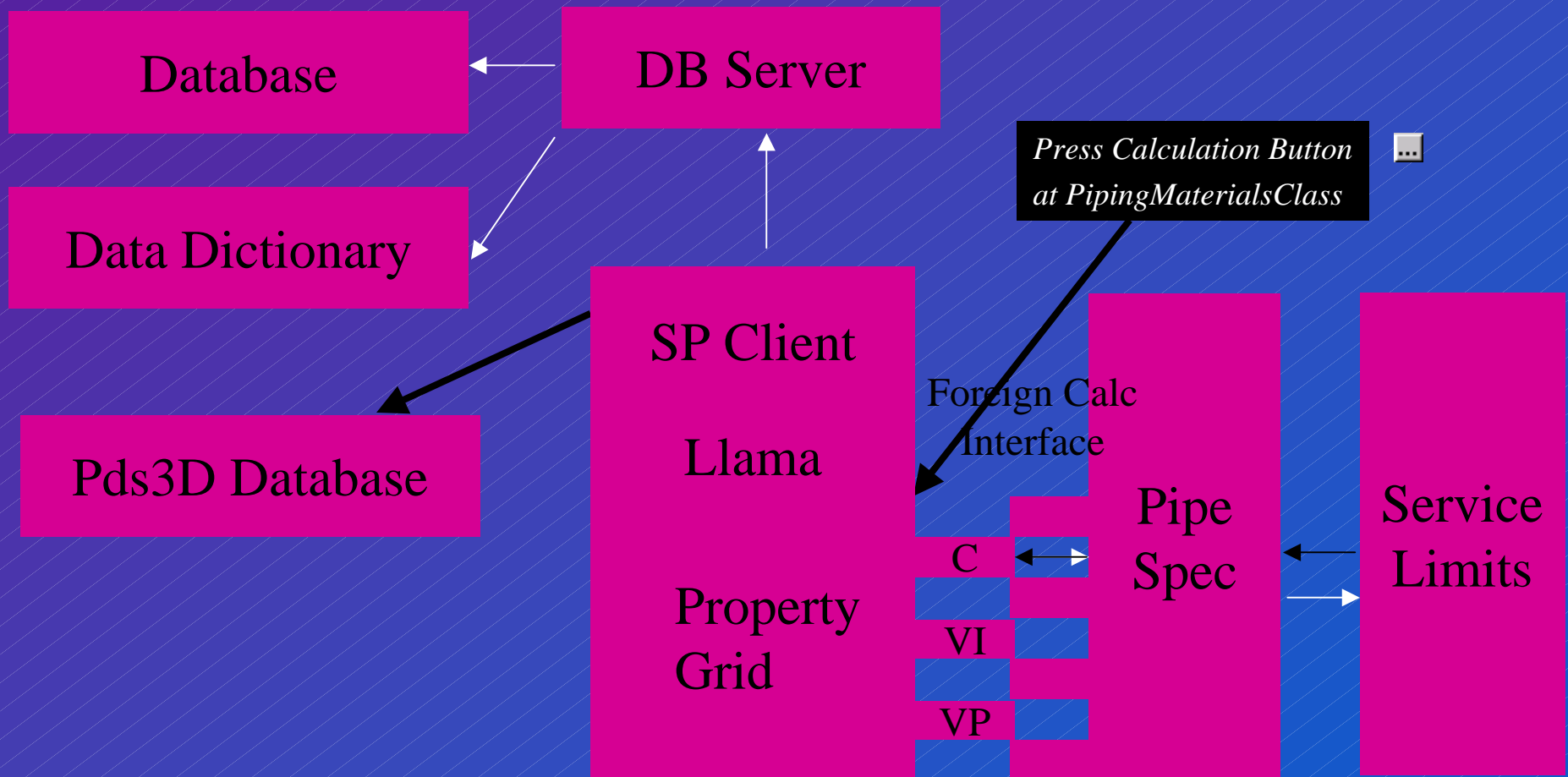
Add new features to SP P&ID



Piping Specifications



Piping Specification Mechanics



Possibilities

- *Retrieve data from DataBase*
- *Modify data in DataBase*
- *Reduce Redundant Data Input*
- *Trace the topology of pipelines*
- *Automate placement of related items*
- *Enhance reporting capability*
- *Integrated with other data source*



Limitations



Limitations

- Only has limited read access to Engineering Manager Compounds Data
- No function to manipulate the graphic objects



Version Compatibility



Version Compatibility

- Intergraph will try its best to keep Dlls binary compatible
- When a Dll's binary compatibility is broken, re-compile is needed
- Usually, all existing functions will keep working even re-compile is needed



At Your Service

Automation Consulting Service

- *Customization of ItemTag Validation*
- *Customization of Import Code*
- *Customization of Report, From/To macro*
- *New Enhancement*
- ...