



# Using ArcGIS® Schematics for Inside Plant Representation

*An ESRI® Technical Paper • June 2004*

Copyright © 2004 ESRI  
All rights reserved.  
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

The information contained in this document is subject to change without notice.

#### **U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS**

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

ESRI, the ESRI globe logo, ArcGIS, ArcMap, ArcInfo, www.esri.com, and @esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

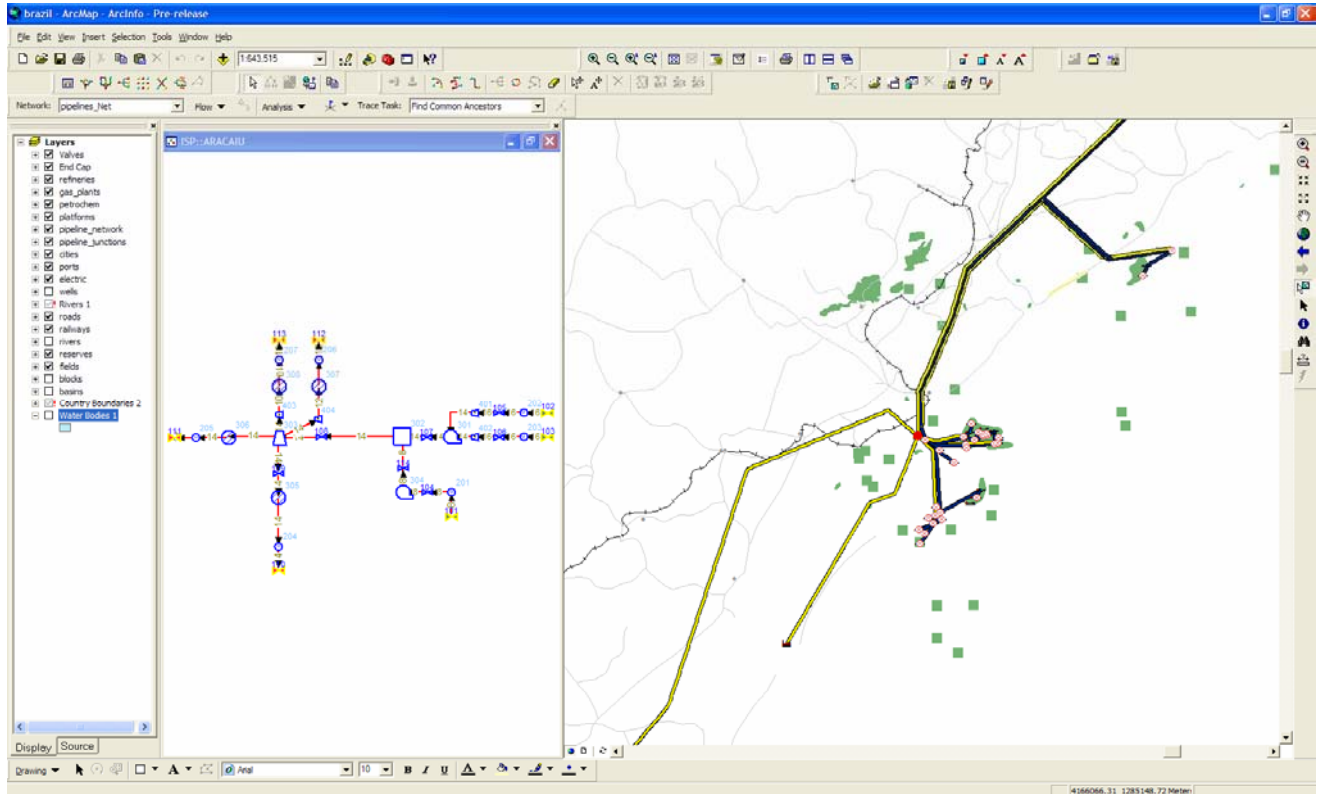
# Using ArcGIS Schematics for Inside Plant Representation

## An ESRI White Paper

<b>Contents</b>	<b>Page</b>
What Is an Inside Plant? .....	1
How Do I Create an Inside Plant Application Using ArcGIS Schematics? .....	1
Starting Questions .....	2
Items to Create in ArcGIS Schematics .....	2
Example .....	2
Finishing Touches .....	11
Final Testing .....	15



# Using ArcGIS Schematics for Inside Plant Representation



*Overview of an Inside Plant*

## What Is an Inside Plant?

"Inside plant" is a term that has different meanings depending on the industry involved. In the electric industry it will most likely be the internal workings of a substation; in the oil industry it may be the inside of a gas plant or refinery; in the telecommunications industry, it may be the internal workings of a switch cabinet. Basically, any data that has to/from node connectivity but is not normally viewable in a geographic information system (GIS) is a good candidate for using the concept of inside plant.

## How Do I Create an Inside Plant Application Using ArcGIS Schematics?

Each inside plant application will vary depending on how detailed it is or how many data sources are involved. The following example shows adding inside plant representations to a data set from Brazil that includes gas plants and refineries.

**Starting Questions**

Questions that need to be answered before getting started are

1. Where is the data about the items in the inside plant, the internals?
2. Is there to/from node connectivity in the data, or is there a way to calculate connectivity from attributes?
3. What items does a user want to see on the inside plant diagrams?
4. Is interaction desired from ESRI® ArcMap™ or only from ArcGIS® Schematics?

**Items to Create in ArcGIS Schematics**

The things that will need to be created to represent the inside plant diagram with ArcGIS Schematics are

1. A new data source to connect to the external data
2. A new document type representing the inside plant
3. Node and link graphic types to represent the items that will be displayed in the diagram

***Example***

In this example, a Microsoft® Access database called ISP.mdb is being used. As can be seen from the following screenshot, all the tables are related to the PLANTS Table.

J-9248

ID	TYPE	ENABLED	MANUFACTURER	PLACING_DATE	MAINT_DATE	NUMBER	FCID	FOID	PLANT_NUMBER
8	Main In	0		1/14/2004		101		1	101
9	Main In	0		1/14/2004		102		1	101
10	Main In	0		1/14/2004		103		1	101
11	Secondary	0		1/14/2004		104		1	101
12	Secondary	0		1/14/2004		105		1	101
13	Secondary	0		1/14/2004		106		1	101
14	Secondary	0		1/14/2004		107		1	101
15	Secondary	0		1/14/2004		108		1	101
16	Secondary	0		1/14/2004		109		1	101
17	Main Out	0		1/14/2004		110		1	101
18	Main Out	0		1/14/2004		111		1	101
19	Main Out	0		1/14/2004		112		1	101
20	Main Out	0		1/14/2004		113		1	101
* er)		0				0	0	0	0

ID	TYPE	ENABLED	MANUFACTURER	PLACING_DATE	MAINT_DATE	NUMBER	FCID	FOID	PLANT_NUMBER
8	Meter	0		1/14/2004		201		1	101
9	Meter	0		1/14/2004		202		1	101
10	Meter	0		1/14/2004		203		1	101
11	Meter	0		1/14/2004		204		1	101
12	Meter	0		1/14/2004		205		1	101
13	Meter	0		1/14/2004		206		1	101
14	Meter	0		1/14/2004		207		1	101
15	Pump Type A125	0		1/14/2004		301		1	101
16	Tank Type TS25	0		1/14/2004		302		1	101
17	Type US56	0		1/14/2004		303		1	101
* er)		0				0	0	0	0

ID	NAME	TYPE	CAPACITY	NUMBER
1	ARACAIU Gas Plant		0	101
2	CUBATAO Refinery		0	102
* er)			0	0

ID	DIAM	MAT	MAN	PLAC	MAIN	FROM_NODE_NUM	TO_NODE_NUM	FCID	FOID	PLANT_NUMBER
18	16					103	203			101
20	16					203	106		1	101
21	16					106	301		1	101
22	16					102	202		1	101
23	16					202	105		1	101
24	16					105	301		1	101
25	8					101	201		1	101
26	8					201	104		1	101
27	8					104	301		1	101
28	14					301	107		1	101
29	14					107	302		1	101
30	14					302	108		1	101
31	14					108	303		1	101
32	14					303	204		1	101
33	14					204	110		1	101
34	14					303	205		1	101
35	14					205	111		1	101
36	12					303	206		1	101
37	12					206	112		1	101
38	10					303	207		1	101
39	10					207	113		1	101
* er)						0	0	0	0	0

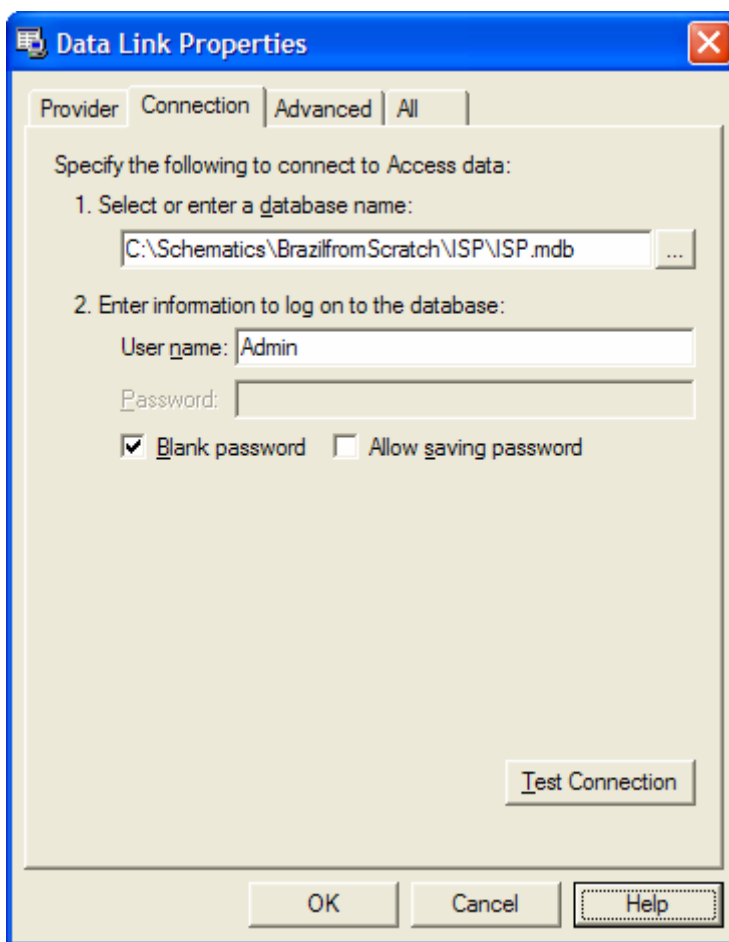
The example has four tables. There is an EQUIPMENTS Table that holds all the internal pieces—equipment is differentiated by the TYPE field. There is a VALVES Table that keeps track of the valves—valve types are differentiated by a TYPE field. There is a PIPES Table for the links between all the equipment pieces, and a PLANTS Table that is the equivalent of the gas\_plant or refineries features that are seen in ArcMap. These are the keys to both the outside world (ArcMap) and the inside world (inside plant).

All four questions from the Starting Questions section above have been answered. A database contains all the items for the internal representations. The PIPES Table has two columns, From\_Node\_Num and To\_Node\_Num, that define the to/from connectivity for the internals. Based on the three tables (not including the PLANTS Table) and the TYPE field on the EQUIPMENTS and VALVES Tables, everything to be shown on the inside plant diagram is defined. The user can have interaction with ArcMap because the items in the PLANTS Table have a geographic representation. The ArcGIS Schematics Designer application can now be used to define the parts and pieces to create the inside plant diagrams.

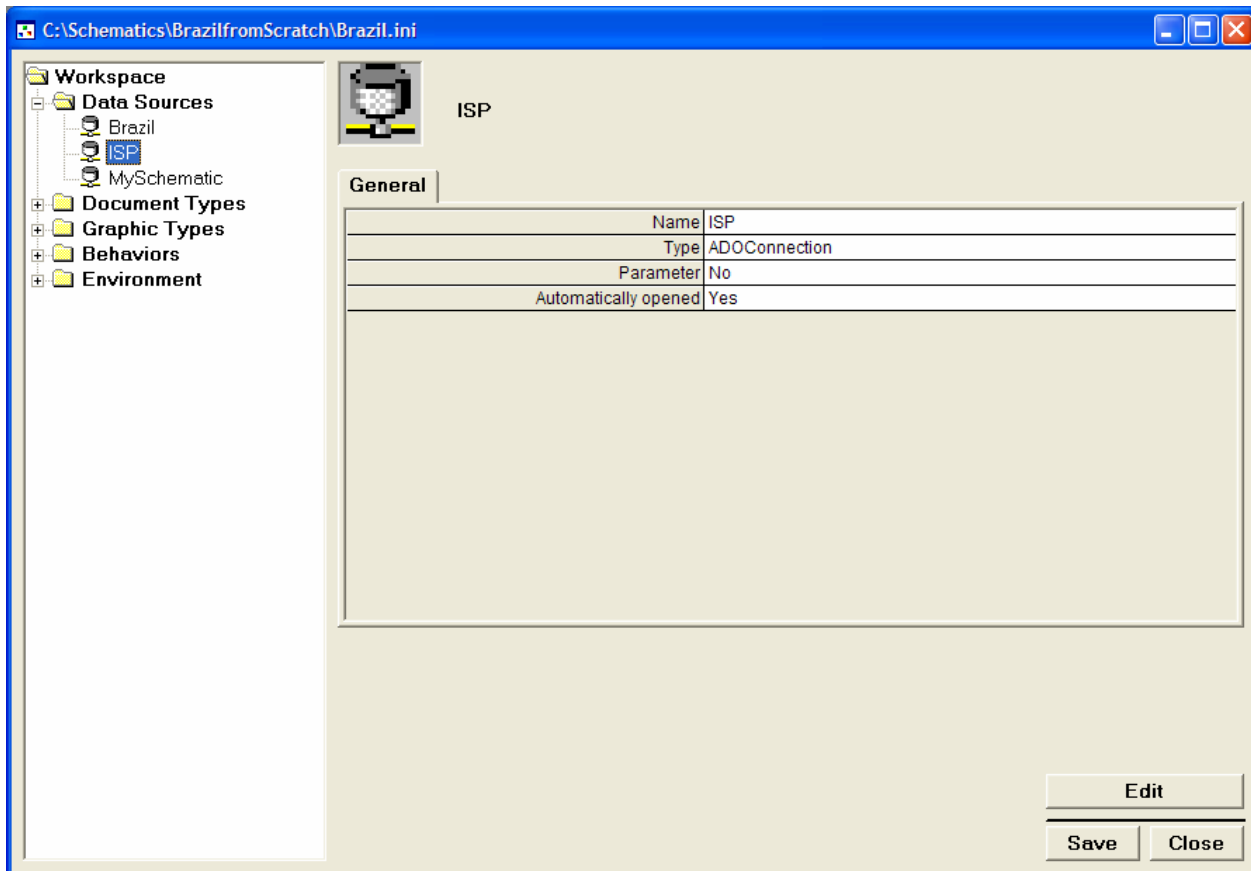
Start by either creating a new schematics workspace or using an existing one. This example does not go through the steps to create the workspace. Note: A workspace can be created using a prebuilt template that comes with ArcGIS Schematics. This allows the user to "get up and running," creating diagrams from an ArcMap selection within a few minutes. This example was started using the template. In the example you will see other items as this template has been expanded for use on inside plant items as well as outside plant items.

Once the workspace is up and running, the first step (from the Items to Create in ArcGIS Schematics section above) is to create a new data source. The data source needs to point to the database where the inside plant tables reside. Note that this could be in many databases. If that is the case, you will need to create one data source for each database. Creating a data source is very straightforward. ArcGIS Schematics uses Microsoft's Data Link wizard to create a connection. Depending on the database used, you will configure the Provider and Connection tabs accordingly, then test the connection to make sure it is running.



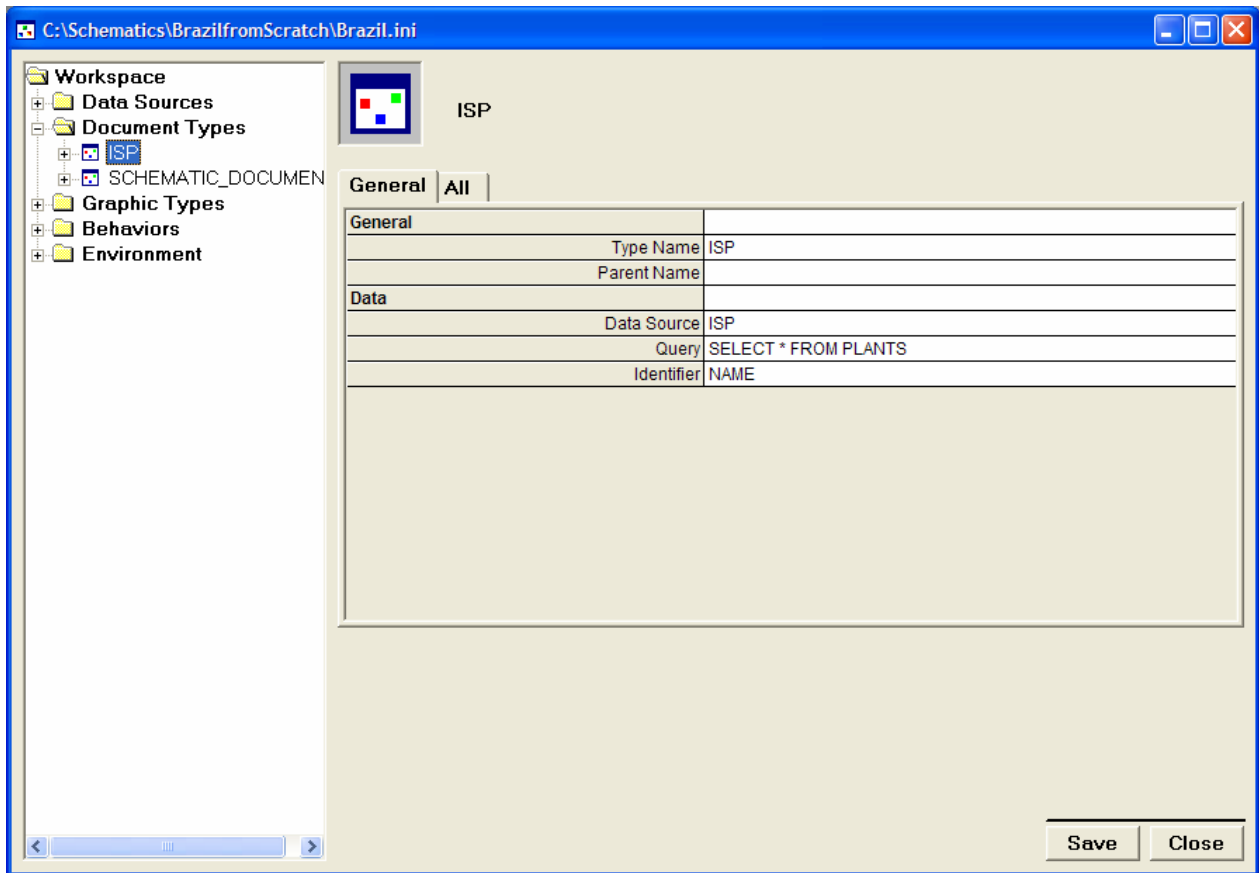


Once the connection is successful, you should have a data source available inside the Designer application. In this example the name is ISP.

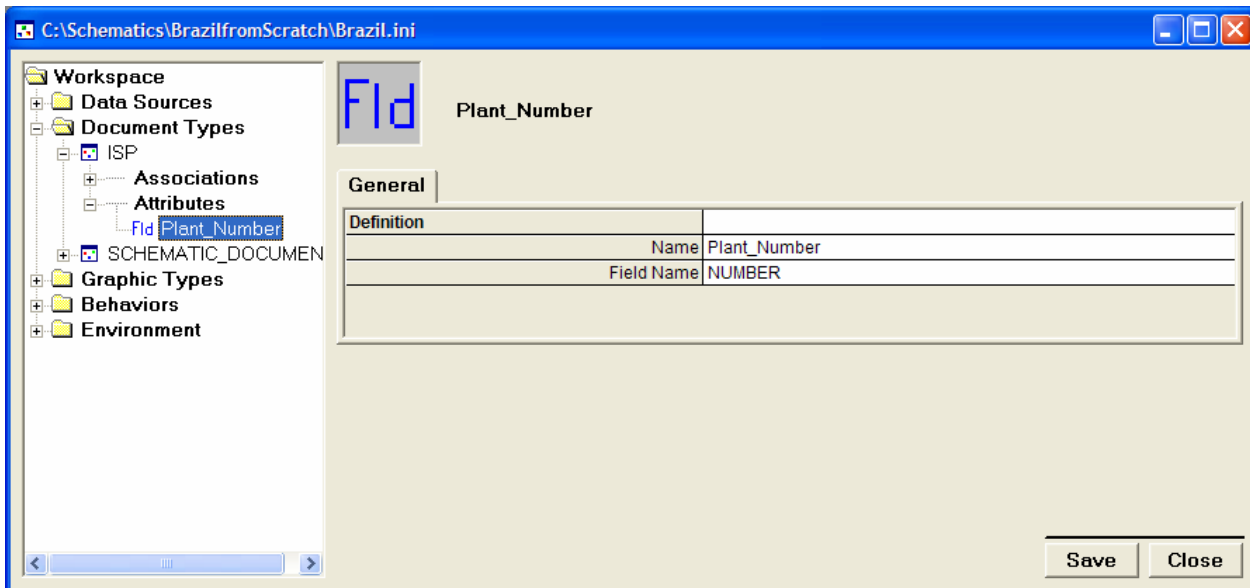


Step 2 from Items to Create in ArcGIS Schematics is to create a document type. Open the workspace using the Designer application and create a new document type for inside plant diagrams. In this example it is named ISP.

J-9248



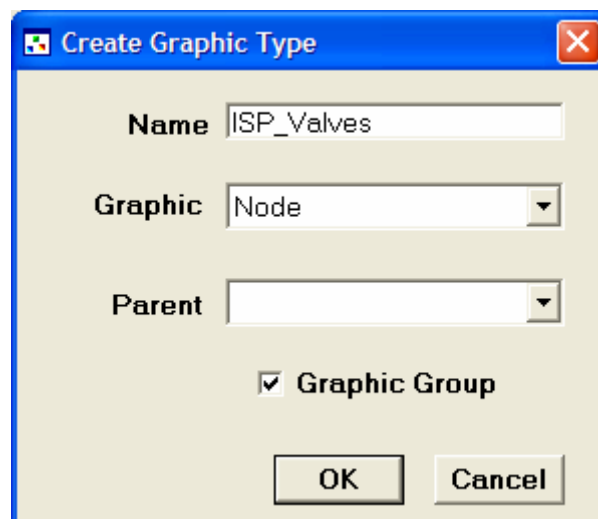
To complete the definition, you will need to set the data source to the one created in Step 1, then you will have to define a query that will return the records a user could choose from to view the internals for the selected document. `SELECT * FROM PLANTS` is used in this example. Now set the Identifier, which in this example is `NAME`, a unique field in the database. If a user wanted to open an ISP document, they would be given a list of plant names to choose from. Now create an attribute that will get the `Plant_Number` for the selected plant. That number will be needed by the graphic types to return just the internal objects that are related to the selected plant.



Once the graphic types have been successfully created, they can be associated with the document type; but for now you are done with the document type.

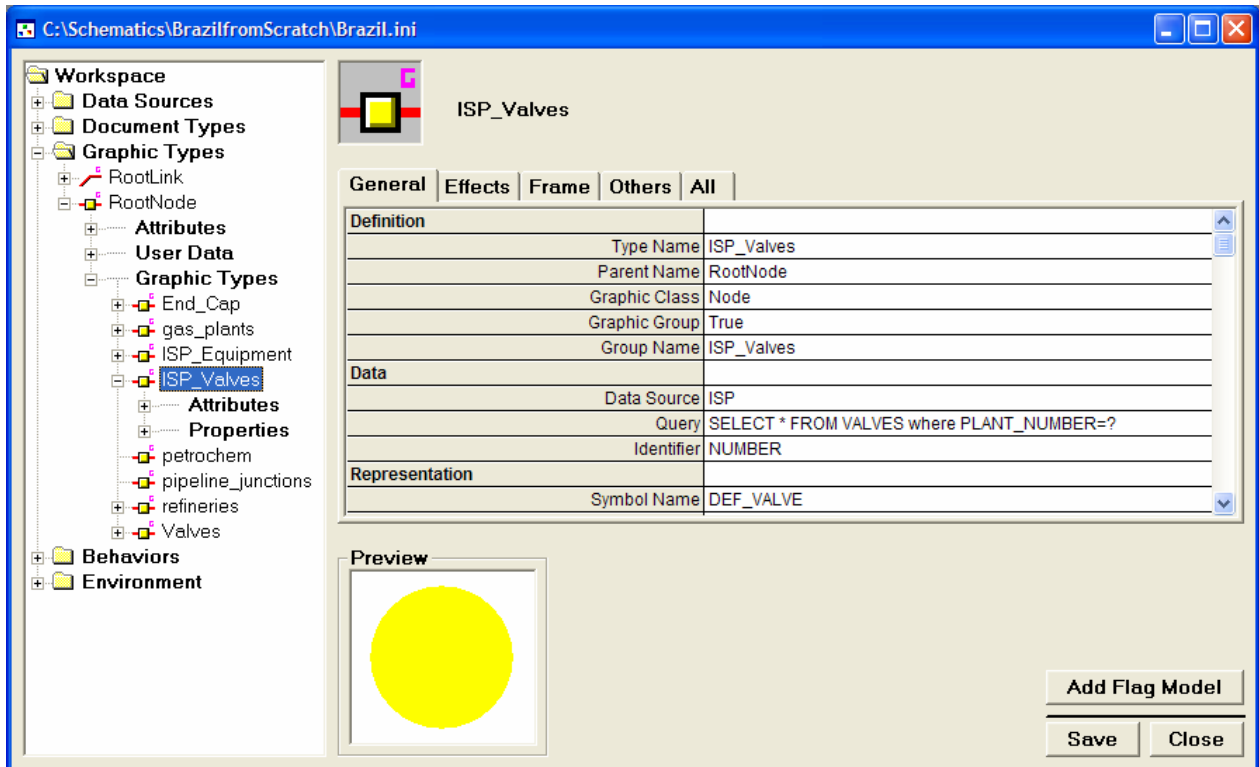
Step 3 from Items to Create in ArcGIS Schematics above is to create the graphic types. Here you will need to create a graphic type for each of the object types that you want to see on the diagram. In this example, you will be shown how to create one node graphic type and one link graphic type. You would need to repeat the steps for all needed graphic types.

For the node graphic type, right-click the Graphic Types tree view entry and select create. Give the new graphic type a name and select node from the graphic drop-down list. Click OK and you will see your new graphic type in the tree view.



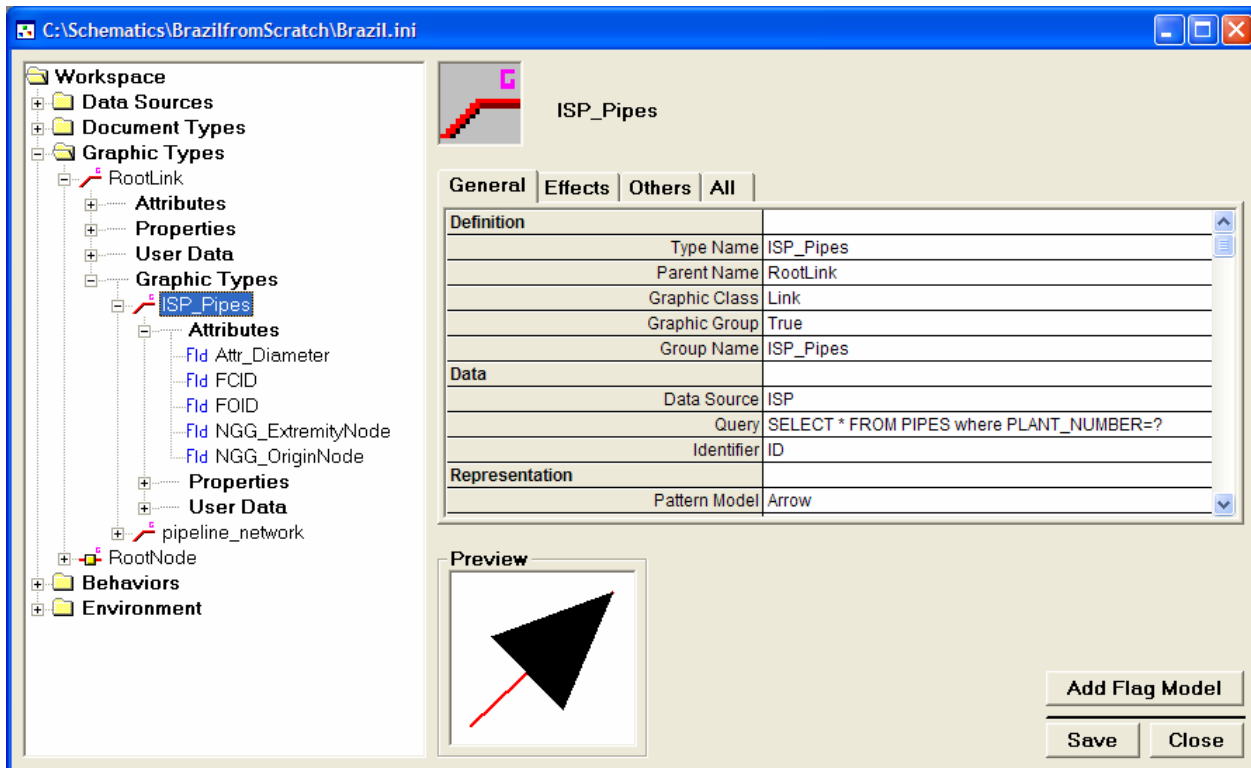
J-9248

To complete the graphic type, you will need to set the data source to the ISP data source that was created. Then you will need to define the query, identifier, and a symbol to use.



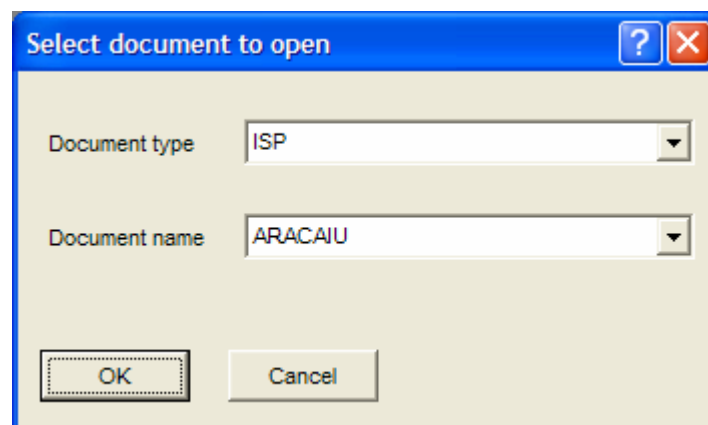
Notice that the query ends with a question mark. In the query creation screen it has been identified that the PLANT\_NUMBER parameter will come from the attribute that was created for the document type, Plant\_Number. You can further define the ISP\_Valves graphic type by defining filters for the different types of valves that are stored in the TYPE field of the VALVES Table.

Create a link graphic type the same as the node type, except Link must be selected in the Create Graphic Type dialog box and you need to define two more attributes.



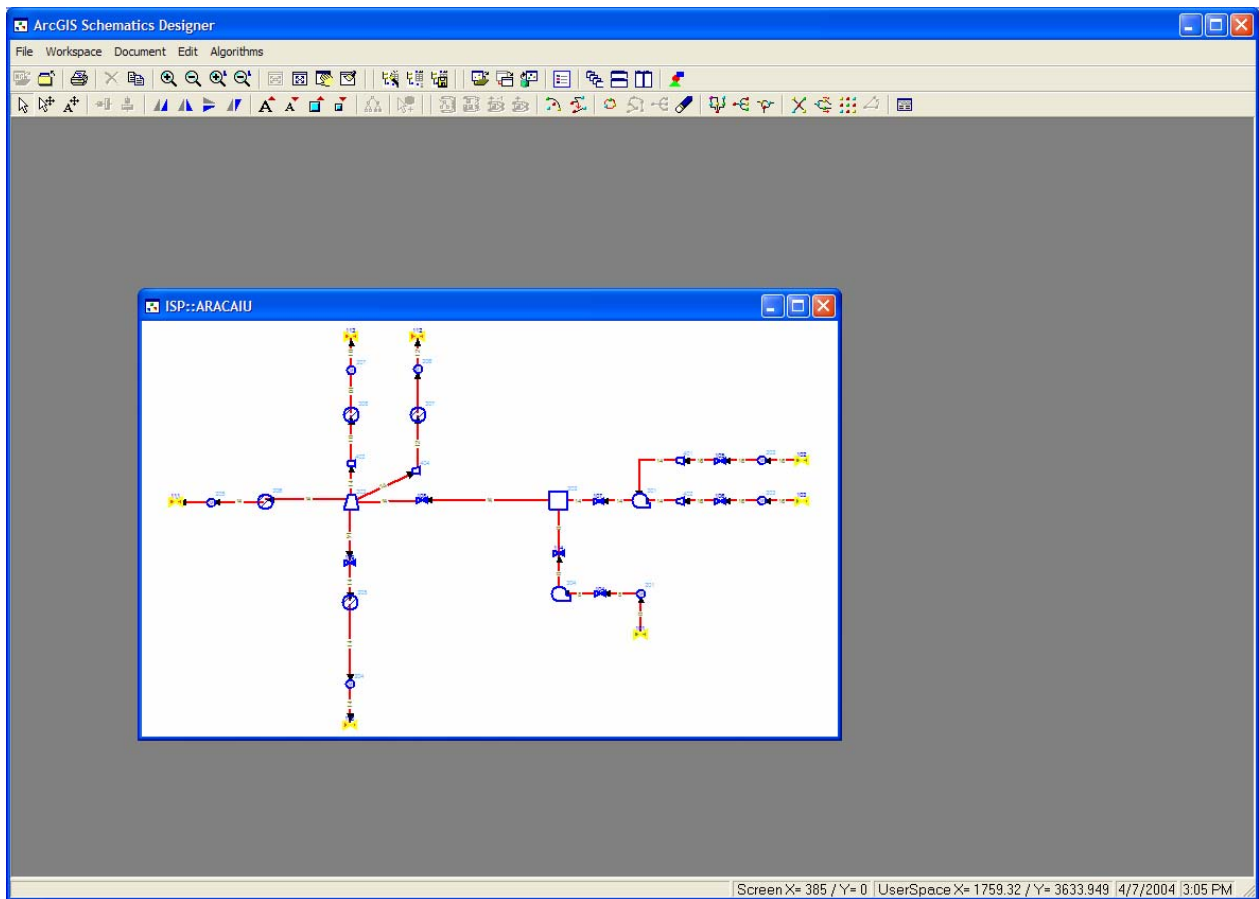
The mandatory attributes to create are NGG\_ExtremityNode and NGG\_OriginNode. These are set to the To\_Node\_Num and From\_Node\_Num fields in the PIPES Table.

The last step is to associate all graphic types to the document type, then you can test your application. Right-click the document type and select Create Association from the popup menu. You will be presented with a drop-down list of all the graphic types that have been defined. Select the one you want and click OK. Repeat until all your graphic types have been associated. To test, click the Open Document Form button on the Designer application.



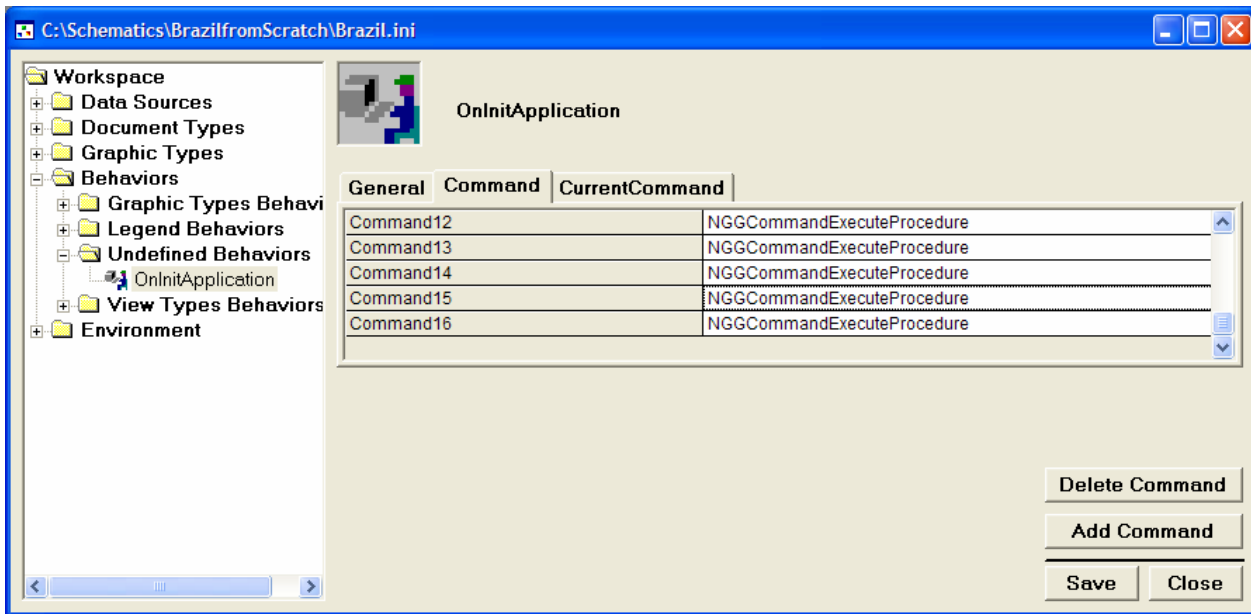
J-9248

You should see ISP in the Document type field and a list of the plant names in the Document name field. Select a plant and click OK. If all the steps worked, you should have your new diagram. Note that the first time a diagram is opened, the items will be placed on a grid layout. After the user moves items around, or uses a predefined layout algorithm, and saves the diagram, it will then be opened as it was left from there on. If you want to have a custom algorithm applied or special formatting, you can do that by creating a custom DLL procedure that is executed when a document of a certain type is opened.

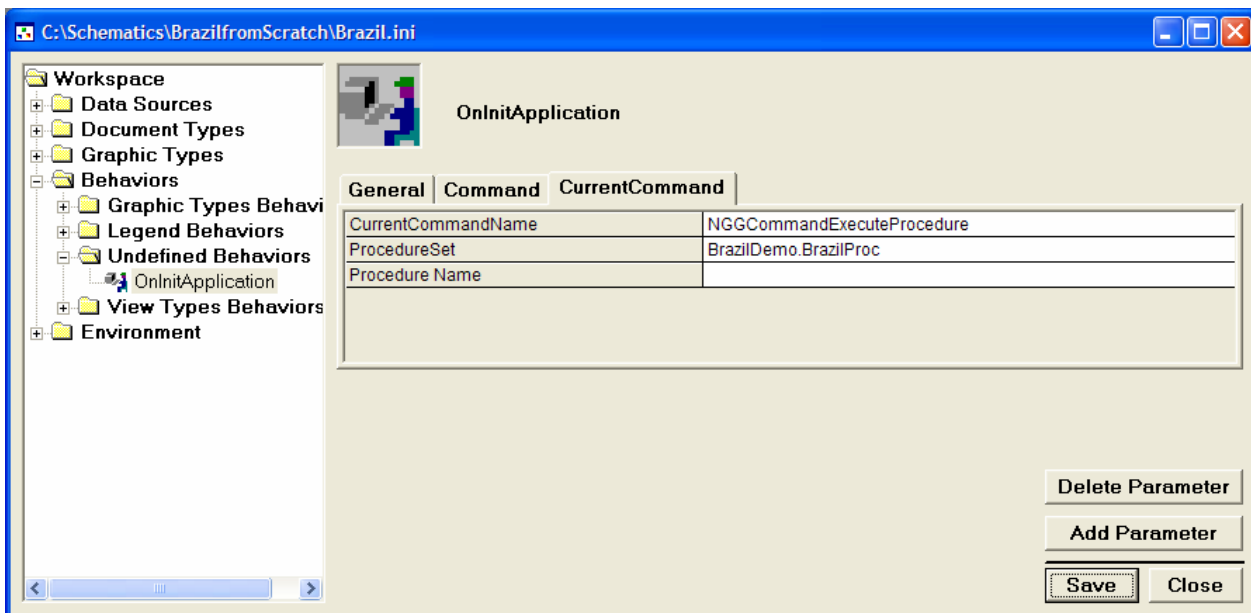


### ***Finishing Touches***

To finalize this example application, you need to create the diagram from a selection inside ArcMap. Create a simple DLL using Visual Basic® (VB) to handle the opening of a new document, then write Visual Basic for Applications (VBA) code in ArcMap to call the VB DLL. Any custom DLL that you want to create and add to a schematics application requires that the DLL is initialized by the schematics session. To do that, using the Designer, open the tree view to OnInitApplication, select the Command tab and click the Add Command button. Select NGGCommandExecuteProcedure from the drop-down list.



Select the CurrentCommand tab and fill in the details of your DLL. Assuming that you created a DLL named BrazilDemo and a class called BrazilProc, the application should look like the following:



Do not put in a Procedure Name at this point. ArcGIS Schematics automatically calls the initialize routine of the DLL to pass a handle to the current workspace. You can see the code sections below.



**Following is the VBA code:**

You need to paste the following code into the VBA environment of ArcMap. You can see that the code gets invoked when someone right-clicks in ArcMap. It checks to see if any of the selected features are refineries or gas plants. If they are, it calls a VB function, passing it the name of the plant. The function then creates the diagram.

```

Private Function MxDocument_OnContextMenu(ByVal x As Long, ByVal y As Long) As
Boolean
  Dim pMxDoc As IMxDocument 'Document object
  Dim pMap As IMap 'actual map
  Dim pEnumFeature As IEnumFeature
  Dim pEnumFeatureSetup As IEnumFeatureSetup
  Dim pFeature As IFeature
  Dim pDS As IDataset
  Dim p_app As IApplication 'Main application object

  Dim pFeatureName As String 'used to check feature class names
  Dim ngoWorkspace As NgWorkspace 'Main Schematics object
  Dim pMyClass As BrazilProc 'Custom VB class

  Set ngoWorkspace = NgSpatial.NgWorkspace 'get the current Schematics workspace

  Set p_app = Application 'get the current ArcMap application
  Set pMxDoc = p_app.Document 'get the current ArcMap document object
  Set pMap = pMxDoc.FocusMap 'get the current map in focus
  Set pEnumFeature = pMap.FeatureSelection 'get all currently selected features
  Set pEnumFeatureSetup = pEnumFeature
  pEnumFeatureSetup.AllFields = True
  Set pFeature = pEnumFeature.Next
  Do While (Not pFeature Is Nothing) 'loop through the features

    Set pDS = pFeature.Class
    mystr = Mid(pDS.Name, 9)
    If mystr = "refineries" Or mystr = "gas_plants" Then 'see if this is a refineries or
gas_plants object
      Dim myint As Integer
      myint = pFeature.Fields.FindField("NAME") 'find out which field is the Name
column

      For Each procset In ngoWorkspace.UserProcedureSets 'loop through
schematics procedures
        If procset.Name = "BrazilDemo.BrazilProc" Then 'find custom procedure
          Debug.Print procset.Name
          Set pMyClass = procset.Object 'instantiate it
          Exit For
        End If
      Next procset
    End While
  End Function

```

```

        If Not pMyClass Is Nothing Then
            pMyClass.LaunchISP pFeature.Value(myint) 'call the procedure
        End If

        End If
        Set pFeature = pEnumFeature.Next
    Loop
    MxDocument_OnContextMenu = False 'allow the right click menu to still work
End Function

```

**Following is the VB dll code:**

You will need to create a DLL and compile it with the same name for this to work. This is the code from the BrazilProc class of the BrazilDemo DLL.

Option Explicit

```

Private WithEvents m_ngWorkspace As NGOLib.NgWorkspace 'Main Schematics
Object
Private WithEvents m_ngAlgo As NGOALGOLib.NgAlgo 'Algorithm Object

```

```

Public Function Initialize(CurrentWorkspace As NGOLib.NgWorkspace)
    Dim lngCW As NgToolsComp.NgCommandsWrapper
    Dim lngUP As NGOLib.NgUserProcedureSet

```

```

    Set m_ngWorkspace = CurrentWorkspace

```

```

    For Each lngUP In CurrentWorkspace.UserProcedureSets 'Initialize all user
procedures
        If lngUP.Name = "NgToolsComp.NgCommandsWrapper" Then
            Set lngCW = lngUP.Object
            Exit For
        End If
    Next

```

```

    Set m_ngAlgo = lngCW.NgAlgo 'New NGOalolib.NgAlgo

```

```

End Function

```

**Public Function LaunchISP(ByVal AttrPlantName As String)**

```

    On Error GoTo errhandler

```

```

    Dim curDoctype As NgDocumentType 'create a document type object
    Set curDoctype = m_ngWorkspace.GetDocumentType("ISP") 'set it to a specific
document type

```

```

    curDoctype.OpenDocument (AttrPlantName) 'Open the document you want

```

```

Exit Function

```

J-9248

*errhandler:*  
*MsgBox Err.Description*  
*End Function*

## Final Testing

Once the VB DLL and VBA code are in place, the only thing left is to test. Open your MXD file and start the schematic session. Select one of the refineries or gas plants and right-click. This should start the VBA code, which calls VB, and give you an inside plant diagram.

