

iManager 2.0 Notes

4/28/2003

Introduction:	2
How to use this document:	2
What you need:	2
Required coding background:	2
Creating a custom gateway configuration snap-in	3
Modifying the snap-in template files for your gateway	3
Directories	4
Update the environment with code changes	5
Test your new snap-in.....	6
Adding a picture	7
Adding another web page to the flow of the user interface	7
Submitting the snap-in to Novell	11
Recommendations for learning more about iManager	11
iManager snap-in template functionality compared to NWAdmin functionality for Printer Agents	11

Introduction:

This document will help third party developers write their Gateway Configuration Snap-in for iManager 2. This document will show you:

- What is already provided by iManager.
- What the snap-in template does for you.
- How to create a Printer Agent using iManager.
- How to write code so that your printer agents will have the load command for your gateway.
- How to add a graphic to your custom snap-in.
- How to prompt the user for input, and how to use that input to create a gateway load string.
- How to add more user interface windows to the template, and use the data supplied by each user interface in your Java code.

How to use this document:

You should use this document as an addendum to the iManager SDK documentation. Read this document up to the "Clarifications to the SDK documentation" section to understand what you will need and what iManager already provides. Then read Chapters 1 and 2 in the iManager SDK documentation and follow along with this document for additions, clarification or corrections.

What you need:

1. Install a NetWare 6.5 server using the custom iPrint installation.
2. Windows 98 workstation with an initial environment setting of 4096 (i.e. default will not work).
OR
Preferably a Windows 2000 workstation with the latest service packs.
3. Internet Explorer 6 or better (i.e. not NetScape) on your development workstation.
4. Install the current Novell client (v4.81 or v4.83 for Windows 2000, or v3.31 to v3.33 for Windows 98). You must install the client with the NDPS option.
5. Sun's JDK 1.4 or later on your development workstation.
6. iManager Developer Kit documentation (www.developer.com) and also distributed with this document. The iManager SDK documentation is found in the imgr_enu.pdf file.
7. Extract the iManager SDK (imgrsdk.zip) (www.developer.com) on your development workstation to C:\Novell\NDK so that these instructions will match your environment.
8. Extract the iManager Print Developer Kit (iManagerPrintDeveloperKit.zip) to C:\Novell\NDK\imgrsdk so that these instructions will match your environment.
9. Map drive G: on your development workstation to the server you wish to manage.

Required coding background:

1. Java
2. JavaScript
3. HTML
4. XML
5. NDPS architecture

Creating a custom gateway configuration snap-in

*These instructions use “DS” (Developer Support) as the example company that is creating the snap-in. You may replace “DS” with the name of your company in these instructions and have them work for you. Changes from the sample to DS are in bold. Note that Java and JavaScript are case sensitive.

Modifying the snap-in template files for your gateway

1. Use the following files as templates:
 - C:\Novell\ndk\imgrsdk\iPrintGateway\MAKEBAT.bat
 - C:\Novell\ndk\imgrsdk\iPrintGateway\jsp\GatewayTemplate.jsp
 - C:\Novell\ndk\imgrsdk\iPrintGateway\plugins\CompanyXYZ.xml
 - C:\Novell\ndk\imgrsdk\iPrintGateway\resources\CompanyXYZResources.properties
 - C:\Novell\ndk\imgrsdk\iPrintGateway\com\novell\manage\iPrint\plugins\GatewayTemplate.java
2. Alter the GatewayTemplate.java file.
 - Change the name of the GatewayTemplate.java file to something else (i.e. **DS** GatewayTemplate.java)
 - Change the name of the class to match the name of the file (i.e. public class **DS** GatewayTemplate extends Task)
 - In the public boolean execute method, change the resultStrings line in the catch statement to match your gateway (i.e. ...getLocalizedString(“**DS**Resources”, “**DS**.GatewayConfiguration);
 - In the populateGatewayLoadString method, change the string in the attrSet.put command to match what your gateway requires. For example:

```
attrSet.put(Oid.NDPS_ATT_PDS_EXEC_AND_PARAMS.getString(), "NDPSGW  
PORT=LPR HOSTADDRESS=" + printerURL);
```

Note that this string is different than what is in the Template. This string is more current for the Novell gateway.
 - In the populateGatewayLoadString method, change the resultString in the catch statement to match your .properties file (i.e. **DS**Resources) and to match your class (i.e. **DS**.GatewayConfigurationFailure).
3. Alter the CompanyXYZResources.properties file.
 - Change the name of the file to something else (i.e. **DS**Resources.properties)
 - The file contains three lines. Each of the lines starts with CompanyXYZ. Change CompanyXYZ on each of the lines to the name of your company (i.e. **DS**).
 - On the third line, change the string to match your company (i.e. Configure **DS** gateway for printer).

4. Alter the GatewayTemplate.jsp file.
 5. Change the name of the file to something else (i.e. DSGatewayTemplate.jsp)
 6. On the fifth line, change the c.stringTable("CompanyXYZResources") to match the properties file in the previous step (i.e. c.stringTable("DSResources"); %>).
 7. In the function checkInputParameters, change the string in the alert to match the name of your company (i.e. alert("<%= c.string("DS.InformationRequired") %>");).
 8. In the <form name... section, change the first {SET ... command to reference your string tables in the .properties file (i.e. c.set("TaskHeader.title", c.plus(c.string("DS.ConfigurationTitle"),). In the c.set("TaskHeader.helpFile" line, change the second parameter to point to an HTML file that you will create for help on this page. For example:

```
c.set("TaskHeader.helpFile", "iPrint/DSHelp.htm");
```

Place your help file in:

```
SYS:\tomcat\4\webapps\nps\portal\modules\iPrint\help\en
```

You may base your help file on:

```
\tomcat\4\webapps\nps\portal\modules\iPrint\help\en\GW_LPR.htm
```

Additional directories exist for other languages. You should populate those directories with help files.

Directories

"iPrint/javascripts/IPUtils.js" refers to Java Script files in the NDK:

```
C:\Novell\NDK\imgrsdk\tomcat\webapps\nps\portal\modules\iPrint\javascripts\IPUtils.js
```

"iPrint.SelectDrivers" refers to JSP files in the NDK:

```
C:\Novell\NDK\imgrsdk\tomcat\webapps\nps\portal\modules\iPrint\skins\default\devices\default\SelectDrivers.jsp
```

"dev.GenMsg" refers to JSP files in the NDK:

```
C:\Novell\NDK\imgrsdk\tomcat\webapps\nps\portal\modules\dev\skins\default\devices\default\GenMsg.jsp
```

"dir/NDPS_Printer.gif" refers to GIF files in the NDK:

```
C:\Novell\NDK\imgrsdk\tomcat\webapps\nps\portal\modules\dev\images\dir\NDPS_Printer.GIF
```

"dev/TaskHeader_inc.jsp" refers to JSP files in the NDK:

```
C:\Novell\NDK\imgrsdk\tomcat\webapps\nps\portal\modules\dev\skins\default\devices\default\TaskHeader_inc.jsp
```

"iPrint/DSHelp.htm" refers to HTML files on the server:

```
\tomcat\4\webapps\nps\portal\modules\iPrint\help\en\DSHelp.htm
```

9. Alter the CompanyXYZ.xml file.
 10. Change the name of the file to something else (i.e. DS.xml).
 11. Change the id in line 3 to match your company name (i.e. <id>ndps.gatecfg.**Developer Support**</id>).

Explanation: The string after the last dot (i.e. Developer Support) is the string that will appear on a drop down list of gateway types. You may enter strings containing spaces. Your id must be distinct from all other id's. Developer Support reserves the right to deny you an id in the case that it is already registered by someone else.

 - Change the resource properties file in line 11 to match the .properties file that you created in step 3.

```
<resource-properties-file>DSResources<resource-properties-file>
```
 - Change the class name to match the name of the class as defined in the .java file (i.e. <class-name> com.novell.imanage.iPrint.plugins..**DS GatewayTemplate**</class-name>).
 - Change the merge template to match the file name of the .jsp file of the previous step (i.e. <merge-template>ndps.gatecfg.**DS GatewayTemplate**</merge-template>).

12. Alter the MakeBat.bat file. This batch file compiles the Java code that you edited into a .class file. The .class file is put into a .jar file. Then the .jar, .jsp, .properties and .xml files are copied to the appropriate directories on the local workstation so that Tomcat can serve them up in your development environment. I am also showing you how to add the files to the appropriate directories on the server so that the Webserver can serve up your new snap-in.

13. Change the name of the file to something else (i.e. **DS.bat**).

14. Change the copy command:

```
copy resources\DSResources.properties lib\*.*
```

15. Change the line that starts with javac so that the .java file you created in step 2 is compiled (i.e.

```
javac -classpath % CLASS_PATH% -d lib  
com\novell\iManage\ndps\plugins\DSGatewayTemplate.java)
```

Note that if you have several versions of the Java SDK installed, make sure you are using the javac.exe for JDK 1.4 or better.

16. Change the line that starts with jar so that it creates a .jar file for your snap-in (i.e. jar cvf **DS.jar** com **DSResources.properties**)

Note that if you have several versions of the Java SDK installed, make sure you are using the jar.exe for JDK 1.4 or better.

17. Change the copy commands for your files:

```
copy lib\DS.jar ..\tomcat\webapps\nps\web-inf\lib\*.*.jar
```

```
copy jsp\DSGatewayTemplate.jsp
```

```
..\tomcat\webapps\nps\portal\modules\iPrint\skins\default\devices\default\gatecfg\*.*.jsp
```

```
copy plugins\DS.xml ..\tomcat\webapps\nps\portal\modules\iPrint\plugins\gatecfg\*.*
```

18. Copy the copy commands to install your new snap-in on a server (i.e. drive G:):

```
copy C:\Novell\ndk\imgrsdk\iPrintGateway\lib\DS.jar  
G:\tomcat\4\webapps\nps\WEB-INF\lib
```

```
copy C:\Novell\ndk\imgrsdk\iPrintGateway\jsp\DSGatewayTemplate.jsp  
G:\tomcat\4\webapps\nps\portal\modules\iPrint\skins\default\devices\default\gatecfg
```

```
copy C:\Novell\ndk\imgrsdk\iPrintGateway\plugins\DS.xml  
G:\tomcat\4\webapps\nps\portal\modules\iPrint\plugins\gatecfg
```

Update the environment with code changes

1. Update your environment with your changes.

19. Map drive G: to the root of volume SYS on your server. This is the drive mapping that you identified in the batch file.

- Open a Command Prompt window and run the new batch file that you just created (i.e. **DS.bat**).
- Check that each line in the batch file executed without failure.

2. Restart the server

- Type "Restart Server"
- Load the broker and manager.

Note: Modifying only the .JSP files does not require restarting the file server in order to see the changes. Modifying the .java or .properties files requires restarting the file server.

Test your new snap-in

1. Open your browser to the following URL: `Https://<IP address of your server>:2200`
2. Enter the authentication information
3. On the left navigation bar, click on "Open Novell iManager 2.0"
4. On the left navigation bar, click on the plus icon to expand the iPrint Management heading.
5. Click "Create Printer."
6. Enter a Printer Agent name.
7. Click on the magnifying glass next to the Manager field, and browse for your manager.
8. Pick your gateway type from the drop down list and click next .
If your snap-in does not appear in the list, verify that the batch file is compiling the java file without errors, and that all of the files are being copied to the correct directory.
9. Verify that your interface displays the appropriate strings.
If they do not, check your code to make sure that it is pointing to your .XML and .Properties files.
10. Follow the prompts to configure the PA and select the drivers.
11. Validate the gateway load string for the Printer Agent in the Manager .
 - a. Go to the NetWare 6.5 server and press Alt Esc and select the NDPS Manager.
 - b. The Printer Agent that you created should be in the Idle state, assuming that you have a printer that the PA can talk to.
 - c. Highlight your PA and press Enter.
 - d. Arrow down to Information, and press Enter.
 - e. Arrow down to Gateway Status and press Enter.
This is the gateway load string that your snap-in built and put into the MOD.
You may edit the load string by pressing Insert then Enter. Debug the load string in the .java file.

Adding fields

These steps will show how to add a field and prompt to the web interface, and bring that data into your java code.

1. Edit the .properties file to include the strings for the prompts that you want to give. For example, add the following line to the **DSResources.properties** file:
DS.PrinterType=Printer Type:
2. Edit the .jsp file to prompt for the data. (i.e. In the **DSGatewayTemplate.jsp** file enter the following lines after the `<!-- 3rd party input fields place below →` line:
`
<%= c.string("DS.PrinterType") %>`
`<input type=text name="CreatePrinterType" value="" size=35 maxlength=80>`
3. Edit the .java file to retrieve the parameters from the .jsp file.
 - Declare the variable (i.e. In the public class **DSGatewayTemplate** of the **DSGatewayTemplate.java** file add the following declaration: `String printerType;`
 - Read the data from the .jsp file into the .java file (i.e. Under the comment `//Retrieve parameter(s)` that are set in the **DSGatewayTemplate.jsp**, add the following line: `printerType = req.getParameter("CreatePrinterType");`).
 - Now the variable "printerType" can be used in your Java code (i.e. to build the gateway load string).

Adding a picture

You may wish to add a company logo, or a picture of your product. This is strictly HTML coding, but here is an example of putting a picture of a printer:

- Bring up the .jsp file in an editor.
- Change the following line to point to the .GIF you desire:
 `c.set("TaskHeader.iconUrl", "dir/NDPS_Printer.gif");`

The directory that contains the GIFs is: `YS:\tomcat\4\webapps\nps\portal\modules\dev\images\dir`
The rest of the page is a form, which does not support displaying images.

Adding another web page to the flow of the user interface

This example will open a new window after the window that prompts for the printer's URL. The new window will prompt the user with a drop down list for the model of the printer. Once the user picks a model and clicks Next on the new window, the sequence of user interface windows will continue as before. The printer model will then be passed back to the .java file and used as an argument in the command that populates the gateway load string.

1. Create a new JavaScript file.
 - Open the **DS GatewayTemplate.jsp**, and save it as a new file (i.e. **DS GatewayTemplate1.jsp**).
2. Make the following edits to the original file (**DS GatewayTemplate.jsp**).

- Find the line that reads:
 `<input type=hidden name="merge" value="iPrint.SelectDrivers">`

Change the value parameter to:

```
<input type=hidden name="merge" value="iPrint.DSGatewayTemplate1">
```

The line above identifies which is the next JavaScript file to execute after this one. The parameter "merge" is the parameter that defines the next step if the current step is successful. You may also add another line like this one to define which JavaScript file to execute in the event that this step is not successful; instead of the "merge" parameter, use the "error" parameter.

The value parameter identifies the name of the file to load. The complete path to the file is:
`SYS:\tomcat\4\webapps\nps\portal\modules\iPrint\skins\default\devices\default`

Notice that part of the path is missing from the parameter

Also notice that the parameter replaces the back slashes with dots, and does not include the extension (i.e. .jsp) in the file name.

Finally, notice that the name of the file is the new JavaScript file that you created in Step 1.

- Find the line that reads:
 `<input type=hidden name="nextState" value="ConfigureGateway">`

Change the value parameter to some state name that you define, like:

```
<input type=hidden name=" nextState " value="iPrint.ConfigureGatewayState1">
```

This gateway configuration snap-in is based on a state machine design. You are allowed to define your own states. The java code that you write can execute code based on the state that you designate in the JSP code. In this case we are getting the IP address of the printer from the user. The code in the .java file will get the data from the CreatePrinterIP variable when in the ConfigureGatewayState1 state.

- Save the file.

3. Make the following edits to the new JavaScript file (**DS GatewayTemplate1.jsp**).
 - In the function `checkInputParameters()`, change all occurrences of `"CreatePrinterURL"` to `"Model"`. `CreatePrinterURL` is a variable defined in the `<input...>` tag toward the bottom of the file. We are changing the variable to `Model` because we are going to prompt the user for a printer model.
 - In the function `setFocusToFormElement()`, change `"CreatePrinterURL"` to `"Model"`.
 - Find the line that reads:

```
<input type=hidden name="nextState" value="ConfigureGateway">
```

Change the value parameter to some state name that you define, like:
<input type=hidden name="merge" value="ConfigureGatewayState2">
 - Alter the `c.set("TaskHeader.title" ... command` to the following:
`c.set("TaskHeader.title", c.plus(c.string("DS.ConfigureModel"), c.var("CreatePrinterDN")));`

Explanation: This will change the title of the window to something appropriate. `DS.ConfigureModel` references a string that is defined in the `DSResources.properties` file. We will add this definition later.
 - Go to the section of code after the line: `<!-- 3rd party input fields place below -->`
 - Alter the line that starts with `<%= c.string` to the following:
`<%= c.string("DS.Model") %>`

Explanation: This will change the prompt above the drop down list to something appropriate. `DS.Model` references a string that is defined in the `DSResources.properties` file. We will add this definition later.
 - Remove the line:
`<input type=text name="CreatePrinterURL" value="" size=35 maxlength=80>`

Explanation: The printer's URL was already prompted for in the previous JSP. We want to prompt for the model as shown in the next step.
 - Add the following lines to create a drop down list. I will leave it to you to read a good JavaScript book on how to design your own user interface.

```
<select name="createPrinterModel">
  <option value="Model1">Model 1
  <option value="Model2">Model 2
  <option value="Model3">Model 3
  <option value="Model4">Model 4
</select>
```
 - Save the file.
4. Edit the `DSResources.properties` file. This is the same file you have been using.
 - Add the following lines
`DS.Model=Choose the printer model`
`DS.ConfigureModel=Configure Developer Support printer model`
 - Save the file.

5. Edit the DSGatewayTemplate.java file. This is the same file you have been using.

- Under the line:
public class DSGatewayTemplate extends Task

and under the line:
String printerURL;

Add the following variable declaration:

String model;

Explanation: This variable will receive the data that the user provided in the .jsp file. The variable name can be anything; it does not have to match the variable name in the .jsp file.

- Modify the first “else if” statement to check for the PrinterURL that the user input in the first JSP file. Do this by modifying the following line:

```
else if (nextState.equalsIgnoreCase("ConfigureGateway"))
```

And make it read as follows:

```
else if (nextState.equalsIgnoreCase("ConfigureGatewayState1"))
```

- Modify the return line that reads:
return populateGatewayLoadString(context, resultStrings);
so that it reads:
return (**true**);

- Copy the entire else if block and paste it immediate below. i.e. copy this block:

```
else if (nextState.equalsIgnoreCase("ConfigureGatewayState1"))  
{  
    try  
    {  
        PrinterURL = req.getParameter("CreatePrinterURL");  
        return (true);  
    }  
    catch(Exception e)  
    {  
        resultStrings.put("GeneralMsg.MsgTitle", eMFrameUtils.getLocalizedName("DSResources",  
"DS.GatewayConfigurationFailure", context));  
        resultStrings.put("GeneralMsg.MsgBody", e.toString());  
        resultStrings.put("GeneralMsg.Mode", "error");  
        return(false);  
    }  
}
```

- Modify the second “else if” statement to check for the printerModel that the user input in the second JSP file. Do this by modifying the following line:

```
else if (nextState.equalsIgnoreCase("ConfigureGatewayState1"))
```

And make it read as follows:

```
else if (nextState.equalsIgnoreCase("ConfigureGatewayState2"))
```

- Modify the try statement to read as follows:

```
try  
{  
    Model = req.getParameter("createPrinterModel");  
    return populateGatewayLoadString(context, resultStrings);  
}
```

Explanation: Model is retrieving the data set in the JSP code that set the state to ConfigureGatewayState2. Now that we have all of the data that we are looking for, we will build the gateway load string by calling populateGatewayLoadString.

- Edit the `attrSet.put` command in the `populateGatewayLoadString` function. Make the command read as follows:

```
attrSet.put(Oid.NDPS_ATT_PDS_EXEC_AND_PARAMS.getString(), "NDPSGW  
PORT=LPR HOSTADDRESS=" + PrinterURL + " PRINTERNAME=" + Model);
```

6. Edit the `DS.bat` file to copy both `.jsp` files to the appropriate locations. For example:

```
copy C:\Novell\ndk\imgrsdk\iPrintGateway\jsp\DSGatewayTemplate.jsp  
G:\tomcat\4\webapps\nps\portal\modules\iPrint\skins\default\devices\default\gatecfg
```

```
copy C:\Novell\ndk\imgrsdk\iPrintGateway\jsp\DSGatewayTemplate1.jsp  
G:\tomcat\4\webapps\nps\portal\modules\iPrint\skins\default\devices\default
```

Submitting the snap-in to Novell

1. The complete build environment for your snap-in (i.e. the complete \Novell\NDK\imgrsdk directory structure) in a zip file.
2. Put the following files in a zip file:
 - C:\Novell\ndk\imgrsdk\iPrintGateway\com\novell\imanager\iPrint\plugins\DSGatewayTemplate.java
 - C:\Novell\ndk\imgrsdk\iPrintGateway\resources\DSResources.properties
 - C:\Novell\ndk\imgrsdk\iPrintGateway\jsp\DSGatewayTemplate.jsp
 - C:\Novell\ndk\imgrsdk\iPrintGateway\plugins\DS.xml
 - C:\Novell\ndk\imgrsdk\iPrintGateway\DSMAKEBAT.bat
 - C:\Novell\ndk\imgrsdk\iPrintGateway\lib\DS.jar
 - SYS:\tomcat\4\webapps\nps\portal\modules\iPrint\help\en\DSHelp.htm
 - Any other .jsp, .properties, .xml, .java, .jar, .htm, .bat files that you may have used to develop with.
3. Send the zip file to your engineering contact at Novell.

Recommendations for learning more about iManager

1. Follow the sample code in the NDK for creating a Manager. This is the simplest example.
2. Follow the sample code in the NDK for creating a Printer Agent. This is the most difficult.

iManager snap-in template functionality compared to NWAdmin functionality for Printer Agents

4 Indicates it has been implemented in iManager.

Information in bold indicates where to find the NWAdmin functionality in iManager.

Locations of functionality in iManager are in bold.

1. Identification (**Printer Control tab, Identification menu item**)
 - a. 4 DS Name
 - b. 4 Printer Agent name
 - c. 4 NDPS Manager
 - d. 4 Description
 - e. 4 Location
 - f. 4 Manufacturer
 - g. 4 Model
 - h. 4 Net Address
 - i. **N/A** Printer Control DLL Name (i.e. .PC2)
 2. Features (**Custom implementation by third party developer**).
 - a. Document formats supported
 - b. Duplex
 - c. Maximum resolution (dpi)
 - d. Mechanism
 - e. Maximum speed
 - f. Memory
 - g. Type (Laser)
 - h. Media Supported (Letter, A4, No. 10 envelope,...)
 - i. PDLs supported (PCL, PJI)
 - j. Input devices
 - k. Output devices
- 2a. Features advanced (These vary from vendor to vendor; HP has a lot more of these, Xerox has a lot)

- a. 4 IPP Security Configured (**Client Support tab, IPP Support menu item**)
 - b. 4 Last job completion time (**Printer Control tab, Printer Health monitor link**)
 - c. 4 Last job number (**Printer Control tab, Printer Health monitor link**)
 - d. Printer device id
 - e. 4 Printer model (**Printer Control tab, Printer Health monitor link**)
 - f. 4 Printer Manufacturer (**Printer Control tab, Printer Health monitor link**)
 - g. 4 Printer startup time (**Printer Control tab, Printer Health monitor link, Shutdown Startup History link**)
 - h. 4 Printer state (**Printer Control tab, Printer Health monitor link**)
 - i. 4 Previous printer state (**Printer Control tab, Printer Health monitor link**)
 - j. Port handler executable and parameters
 - k. 4 PDS executable and parameters (**Printer Control tab, Printer Health monitor link**)
 - l. 4 Printer state severity (**Printer Control tab, Printer Health monitor link**)
 - m. 4 Printer enabled (**Printer Control tab, Printer Health monitor link**)
- 2b. Support services
- a. 4 NDPS Manager (**Printer Control tab, Identification menu item**)
 - b. 4 Notification service (**Printer Control tab, Printer Health monitor link**)
 - c. 4 Resource Management Service (**Printer Control tab, Printer Health monitor link**)
3. Jobs
- a. 4 NDPS Job list (**scheduled, retained**) (**Printer Control tab, Printer Health monitor link, Job List link**)
 - b. Spooling Configuration
 - 4 Spooling location: volume
 - 4 Limit disk space: spooling, retained jobs
 - 4 Scheduling: FIFO, Print only current medium, Minimize media changes, Print smallest job first.
 - 4 Add service jobs from NetWare Queues (**Client Support tab, QMS Support menu item**)
4. 4 Pause Output / Resume Output (**Printer Control tab, Printer Control menu item**)
5. 4 Pause Input / Resume Input (**Printer Control tab, Printer Control menu item**)
6. Control
- a. 4 Reset (**Printer Control tab, Printer Control menu item, Shutdown; Restart**)
 - b. 4 Form Feed (**Client Support tab, LPR Support menu item**)
 - c. Mount Media
 - d. 4 Statistics: Most but not all (HP, Axis, Xerox) vendors use this. (**Printer Control tab, Printer Control menu item, Printer Health Monitor link**)
 - e. Set defaults
 - General (**Configuration tab, Defaults menu item**)
 - 4 Copies
 - 4 Max copies
 - 4 Priority
 - 4 Max Priority
 - 4 Banner
 - 4 Medium
 - Job Holds (**Configuration tab, Job Holds menu item**)
 - 4 Operator hold
 - 4 User hold
 - 4 Delay printing until
 - 4 Pause Printer on job start

Beta document

- Message
- 4 Pause Printer on job end
 - Message
 - 4 Retain Job no longer than (min, hour, day, year)
 - 4 Retain Job for (min, hour, day, year)
- Notification
 - Directed Pop-up notification (see NWAdmin for list)
 - Popup notification
- 4 Assign drivers (**Drivers tab**)
- 4 Status (**Printer Control tab, Printer Control menu item, Printer Health Monitor link**)

The location of this file is

*PRV-Banshee/Data:PrintGroup\Development\Snapins\iManager2 \ iManager2_0 NDPS Snap-in
Development Notes.doc*