Honeywell

Honeywell Smart Printing

Software Developer Kit

User Guide

Disclaimer

Honeywell International Inc. (“HII”) reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult HII to determine whether any such changes have been made. The information in this publication does not represent a commitment on the part of HII.

HII shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material. HII disclaims all responsibility for the selection and use of software and/or hardware to achieve intended results.

This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of HII.

Web Address: www.sps.honeywell.com

Other product names or marks mentioned in this document may be trademarks or registered trademarks of other companies and are the property of their respective owners.

For patent information, refer to www.hsmpats.com.

TABLE OF CONTENTS

Customer Support

Technical Assistance

To search our knowledge base for a solution or to log in to the Technical Support portal and report a problem, go to www.sps.honeywell.com and select **Support** at the top of the page.

For our latest contact information, see www.honeywellaidc.com/locations.

Product Service and Repair

Honeywell International Inc. provides service for all of its products through service centers throughout the world. To obtain warranty or non-warranty service, return your product to Honeywell (postage paid) with a copy of the dated purchase record. To learn more, go to www.sps.honeywell.com and select **Services** at the top of the page..

Limited Warranty

For warranty information, go to www.sps.honeywell.com and click **Support** > **Warranties**.

CHAPTER

1 SMART PRINTING RESOURCE KIT

This Developer Guide describes best practices for using the Honeywell Smart Printing Resource Kit to develop applications for your Honeywell printers and peripherals.

Before you work with the Smart Printing Resource Kit, you should be familiar with:

- general programming techniques for C# .NET 4.0
- your Honeywell printers and peripherals
- development tools such as Microsoft Visual Studio
- concepts for software design.

Note: *There are several references to using FTP and Telnet in this document. By default, these services are disabled in the printer and need to be enabled from your Printer Menu: System Settings - Manage Services.*

What Is the Smart Printing Resource Kit?

The Smart Printing Resource Kit provides the basic tools you use to develop applications for Honeywell printers and peripherals, and includes documentation, tutorials and sample code.

The Smart Printing Resource Kit covers these functional areas:

- Communication: Managing communication with the printer using TCP/IP over Ethernet or 802.11, Bluetooth, industrial interface hardware, serial, or USB host.
- Configuration: Managing printer settings including alerts, communications, printing, system, and network settings.
- Drawing: Rendering text, barcodes, shapes, and images for printing.
- Print Control: Managing the printing mechanism, providing features such as print feed, form feed, and test feed.
- Status: Query and subscribe to updates of printer system status, such as “printhead lifted” and “low media” events.

- User Interface: Managing the printer user interface by controlling the display and LED states, and reading key or touch press events.
- RFID Programming: Read, write, kill or protect RFID inlays.

Supported Platforms

The Smart Printing Resource Kit supports the following Honeywell printers.

- PM43/23 series
- PC43/23 series
- PD43 series
- PX940 series
- PM45 series
- PX4ie and PX6ie
- PX240/PD45 series

The Smart Printing runtime library in printer firmware versions from 10.05 to 10.15 is compatible with Microsoft .NET Framework 2.0. The runtime library in firmware version 10.16 and later release is compatible with Microsoft .NET Framework 4.x.

Note: *Despite the small size of their displays, the printers support the full .NET Framework, not just the .NET Compact Framework.*

Some functionality depends on the options installed in your Honeywell printer or its peripherals. For more information, see the Smart Printing API Reference or the User Manual for your specific printer at www.sps.honeywell.com.

Development Tool Requirements

You need these development tools and resources to use the Smart Printing Resource Kit:

- Visual Studio 2015 or later, or any CLR-compliant development tool
- Runtime Files:
    - Microsoft .NET 4.0 or later Framework
    - Honeywell Smart Printing .NET Class Library (included with the Resource Kit)

Install the Smart Printing Resource Kit

You download and install the Smart Printing Resource Kit on your development PC. You do not need to install any Resource Kit components on the printer, as the printer firmware includes support for Smart Printing applications.

1. Go to www.sps.honeywell.com and choose **Products > Software and Tools > Developer Library > Developer Library**.
2. Select Smart Printing RK and click **Downloads** tab and follow the instructions to download the Zip file. You need to login to download software.
3. On your PC, create the directory C:\Honeywell.
4. Extract the downloaded zip file into C:\Honeywell. The Resource Kit is installed in C:\Honeywell\SmartPrintingRK.

Note: *If you choose to install the Resource Kit in a different path, you must manually update library and utility references in the provided example projects.*

Resource Kit Directory Structure and Contents

After you download and install the Resource Kit on your development PC, you can find these components in the directories listed.

Smart Printing Resource Kit Directories and Content Descriptions

Path	Description
Dll\HoneywellPrinter.dll	.NET library providing the printer API in the namespace Honeywell.Printer.
Dll\Mono.Data.Sqlite.dll	.NET library providing SQLite3 database connection.
Dll\Mono.Security.dll	.NET library providing security and cryptography functionalities.
Docs\EULA.pdf	End User License Agreement.
Docs\HoneywellPrinter.chm	Smart Printing Resource Kit API Reference
Docs\Mono.Data.Sqlite.chm	Microsoft Compiled HTML Help file describing the Mono.Data.Sqlite bindings API.
Docs\SQLite3.chm	Microsoft Compiled HTML Help file describing the SQLite3 database.
Docs\Honeywell Smart Printing User Guide.pdf	Smart Printing Developer Guide document.
Examples\Code	Directory containing sample C# applications and code snippets. These are the same samples provided in the Smart Printing API Reference.
Examples\Projects	Directory containing sample C# Visual Studio projects.
Utils\FtpPut.exe	Minimal FTP client utility used to transfer files to the printer.

Note: *You do not need to copy any .dll files to the printer. The .dll files are already present in the printer files system under **/usr/lib/mono** and **/usr/lib**, and are used by the printer mono run-time.*

CHAPTER

2 DEVELOP C# PRINTER APPLICATIONS

How to Develop C# Printer Applications?

This section includes general information on using the Smart Printing Resource Kit to develop applications.

Connect the Honeywell Printer to Your PC

During development, the easiest way to deploy applications to the Honeywell printer is through a network connection (TCP/IP over Ethernet or 802.11) between the printer and your development PC. For more information, see the user manual for your printer.

Build and Run a Sample Program

An easy way to verify that the Smart Printing Resource Kit is properly installed and functional is to build and run one of the sample programs.

1. Go to C:\Honeywell\SmartPrintingRK\Examples\Projects\HelloWorld.
2. Double-click HelloWorld.sln to open it in Visual Studio.
3. In the solution explorer, double click **Properties**.
4. Click **Build Events**.
5. In the Post-build event command line, update the IP address to your printer IP address, as seen in this example:
   … ftp://10.203.33.124/ … ? … ftp://<your-printers-ipaddress>/…
6. Press **Build Solution** to build the project and deploy the application to your printer.

Run the Sample Program from the Printer front Panel

1. Open the printer menu
2. Select **Programs**

3. Select **C# Programs**
4. Select **HellowWorld.exe**

Run the Sample Program from the Telnet

1. Open the telnet connection to the printer IP address and port 23
2. Enter the username **user** when prompted. This username has no default password.
3. Run the application by typing: **mono apps/HelloWorld.exe**

Create Your Own Program

This section describes how to create a project in Visual Studio targeted for Honeywell Printers.

1. In Visual Studio, select **File > New > Project**. The New Project dialog box appears.
2. In the **Project types** list, select **Visual C# > Windows**.
3. In the **Templates** list, select **Console Application**.
4. In the **.NET Frameworks** list, select **.NET Framework 4.0**.
5. Enter a name for the project and click **OK**. The project is created.
6. In the **Solution Explorer**, double click **Properties**.
7. Click **Build Events**.
8. In the **Post-build event command line** field, add the following code all on one line and replace the IP address with the IP address of your printer.

   ```
   c:\Honeywell\SmartPrintingRK\Utils\FtpPut.exe$(TargetFileName) ftp://
   10.203.33.124/apps/$(TargetFileName)user pass
   ```

   When you are ready to compile the application, press **Build Solution** to compile and automatically download the application to the printer.

Access Smart Printing Resource Kit Components in your Application

This section describes how to access Smart Printing Resource Kit components in your application. When you install the Smart Printing Resource Kit, the components are not automatically added to Visual Studio, so you need to manually add the assemblies.

1. In the Solution Explorer, right-click References and select Add Reference from the pop-up menu.
2. Click the Browse tab and go to C:\Honeywell\SmartPrinting\Dll

3. Select the assembly or assemblies your application intends to use (typically HoneywellPrinter.dll)
4. Click OK
5. In the Source Code Editor, add this line to the other “uses” statements at the beginning of the file:

   ```
   using Honeywell.Printer;
   ```

Note: *When you deploy your application from Visual Studio using the Post-Build event, the assemblies on the host PC are not copied to the printer. The printer resident assemblies are used instead.*

CHAPTER

3 DEVELOP AND DEPLOY APPLICATIONS

How to Develop Database Applications?

Honeywell printers come with the SQLite3 database engine. SQLite is a relational database management system supporting most of the SQL-92 standard. Each database is contained in a single file.

The assembly Mono.Data.Sqlite provides support for accessing SQLite3 databases from a C# application. Mono.Data.Sqlite provides an ADO.NET data provider, fully compliant with the ADO.NET 4.0 API interface.

This Resource Kit provides API reference manuals in .chm format for SQLite3 and Mono.Data.Sqlite in the directory C:\Honeywell\SmartPrintingRK\Docs.

The Smart Printing Resource Kit also provides an example application under C:\Honeywell\SmartPrintingRK\Examples\Code\MonoDataSqlite.cs.

Note: *The namespace System.Data.SQLite described in the SQLite3 documentation is not supported. Only Mono.Data.Sqlite is supported.*

The Mono.Data.Sqlite API documentation is missing descriptions of certain parameters and return values. Refer to other ADO.NET 4.0 provider API documentation (such as System.Data.SQLite) for more complete information.

How to Deploy Your Application?

The best way to deploy your application to the Honeywell printer depends on the number of printers involved.

Honeywell recommends that you install applications in the /home/user/apps directory on the printer. The procedures in this section assume you are using this path, but you can use your own directory structure if desired. In this case, be sure to replace /home/user/apps with the correct path for your application.

Deploying Applications with a USB Storage Device

If you only have a few Honeywell printers that use your application, you can deploy your applications to one printer at a time using a USB storage device. Start with an empty drive formatted as either FAT16 or FAT32, and with only one partition.

USB Storage Device Directory Structure

The directory structure is important as it affects how and where the files on the USB storage device are installed to the printer.

For your C# applications, there are two relevant directories. Files and directories placed in these directories will be copied over to the corresponding directories in the printer file system:

Directory Structure for USB Storage Devices

Path on USB Storage Device	Path in Printer File System	Directory Contains
/fonts	/home/user/fonts	User-downloaded fonts
/apps	/home/user/apps	C# apps and ancillary files
/images	/home/user/images	User-downloaded images

Note: *There are other directories in the printer file system that are used for Fingerprint applications or for host-based printing.*

Place Applications on the USB Storage Device

Honeywell recommends that you store all C# applications and related files in the /home/user/apps directory on the printer as follows:

/home/user/apps/<vendor-name>/<application-name>

Follow these steps in order to prepare the USB storage device for your application.

1. Create a directory names apps on the USB storage device (must be lower-case)
2. In the apps directory, create a directory named for the application vendor
3. In the vendor directory, create a directory named for the application (for example, HelloWorld)
4. Copy the .exe and .dll (and possibly other resources) from your project bin/ Release directory to the application-name directory on the USB storage device. Assemblies provided with the Smart Printing Resource Kit are already resident on the printer and should not be bundled with an application. For more information, see Printer-Resident Assemblies and Libraries.
5. (Optional) To make the application available in the printer UI Programs menu, you can either:
   - Place the executable .exe file directly in /home/user/apps/.

- Place the executable in a subdirectory of /home/user/apps, and create a symbolic link to the application by executing this command:

  ```
  ln -s /home/user/apps/<subdirectory-name>/<application-name>/
  application-executable> /home/user/apps/<application-executable>
  ```

Note: *The command is case-sensitive. All content should be on one line. Replace <strings> with the actual paths and names used.*

Deploy the Application

1. Turn off the printer
2. Insert USB storage device
3. Turn on the printer
4. When the printer Ready screen appears, or your application has started, you can remove the storage device from the printer
5. (Optional) To make the application automatically start up when the printer is turned on, create a start-system script as described in Automatically Start the Application on Printer Power Up.

How to Deploy Applications through FTP

Deploying applications through FTP is suitable for large quantities of network enabled printers, as the FTP operations can be scripted and automated. Each network-enabled Honeywell printer has a built-in FTP server.

Honeywell recommends that you store all C# applications and related files in the /home/user/apps directory on the printer as follows:

```
/home/user/apps/<vendor-name>/<application-name>
```

Prepare the Application for FTP Deployment

Honeywell recommends that you store all C# applications and related files in the /home/user/apps directory on the printer as follows:

/home/user/apps/<vendor-name>/<application-name>

Follow this procedure to create a deployment folder on your development PC.

1. On your PC, create a directory named **apps** (must be lower case)
2. In the apps directory, create a directory named for the application vendor
3. In the vendor directory, create a directory named for the application (for example, HelloWorld)
4. Copy the .exe (and possibly other resources) from your project bin/Release directory to the application-name directory. Assemblies provided with the Smart Printing Resource Kit are already resident on the printer and should not

be bundled with an application. For more information, see Printer-Resident Assemblies and Libraries.

5. (Optional) To make the application available in the printer UI Programs menu, you can do one of the following:
   - Place the executable .exe file directly in /home/user/apps/
   - Place the executable .exe file in a subdirectory of /home/user/apps, and create a symbolic link to the application by executing this command:

     ```
     In -s /homeuser/apps/<vendor-name>/<application-name>/<application-
     executable> /home/user/apps/<application-executable>
     ```

Note: *The command is case-sensitive. All content should be on one line. Replace the <strings> with the actual paths and names used.*

Deploy the Application

1. Transfer the content of our local apps directory to the /home/apps directory on the printer.
2. Restart the printer (remotely via telnet, http or snmp).
3. (Optional) To make the application automatically start up when the printer is turned on, create a start-system script as described in Automatically Start the Application on Printer Power Up.

CHAPTER

4 START APPLICATION

How to Start the Application?

There are multiple ways to start your application on a Honeywell printer:

- You can set up the printer to start the application when the printer is turned on.
- You can start the application from the printer front panel.
- You can start the application from a shell command prompt.
- You can run the application as a utility from a Fingerprint application.

Each of these methods is described in the following sections.

Note: *You cannot run an application that uses the Honeywell.Printer classes on a host PC.*

Automatically Start the Application on Printer Power Up

To automatically start an application when the printer is turned on, create a startsystem script and place the script in the printer file-system directory /home/user/apps.

The script file contains the commands to execute when the printer is turned on. Each command must be a single line.

1. Using a text editor on your PC, create a script file names “start system” (all lower-case and no extension)
2. To start the application when the printer is turned on, include this line in the file:

   ```
   mono /home/user/apps/<application-executable-path>
   ```

   where application-executable-path is the path to the application. The command is case-sensitive and all content should be on one line. The file must use Unix/Linux-style line ends containing only the LF character.

Note: *To start multiple applications when the printer is turned on, include multiple command lines in the script file. To start all application at the same time (parallel operation), add a "&" character to the end of each line. Without the "&" character, each application starts after the previous application ends.*

3. Copy the start-system file to the /home/user/apps directory on the printer. If you copy the file to the /apps folder on a USB storage device or to an FTP deployment folder, the file is automatically copied to the printer when the application is installed.

Start the Application from the Printer Front Panel

For printers with an LCD user interface, you can start an application from the printer front panel.

1. In the printer menu system, select Programs > C# Programs.
2. Select the application from the list. For information on using the printer menu system and display, see the printer user manual.

Note: *Either the application or a symbolic link to the application must reside in the /home/user/apps directory.*

Start the Application from Shell Command Prompt

1. Open a telnet connection to the printer IP address on port 23.
2. Type the username user when prompted. This username has no default password.
3. Type this command:

```
mono /home/user/apps/<application-name>
```

Start the Application as a Utility from a Fingerprint Application

You can execute an application from a Fingerprint application if you need to perform complex data processing or access a database. All other input and output controls, user interface changes, and print handling is still managed from the Fingerprint application.

Note: *The C# application being run should not use the Honeywell.Printer.Communication.* or Honeywell.Printer.UI.* classes.*

The Fingerprint syntax to start an application is:

```
RUN "mono /home/user/apps/<application-name>.exe"
```

Interrupt Application Execution

During development, you may need to forcefully terminate a running C# application. Connect to the printer through a remote terminal connection and issue this command:

```
kill -s 9 ‘pidof mono‘
```

Note: *You must use the grave accent character ‘ (ASCII 96) and not a standard single quote character ‘ (ASCII 39) to enclose the pidof mono command. On U.S. keyboards, the grave accent character key is located to the left of the 1 key.*

If you started the application from a terminal window and the window is still responsive, you can send an interrupt signal to the application by pressing **Ctrl-C**.

About Application Priority

The system scheduling priority of an application is determined by its nice value. The valid range for a nice value is -20 (highest priority) to 19 (lowest priority), with the default nice value as zero.

C# applications started on the printer are assigned a default nice value of zero.

Application priority may be changed (only lowered) using the utility nice, which takes nice-offset as an argument. For example, to start a sample application with nice value 10 (10 = 0 + 10):

```
nice -n 10 mono /home/user/apps/HelloWorld.exe
```

Printer-Resident Assemblies and Libraries

This section provides a list and description of the assemblies bundled with the standard printer firmware.

Standard Libraries

Library Name	Description
mscorlib.dll	Core run-time functionality
System.dll	Core run-time functionality
System.Configuration.dll	Configuration data functionality
System.Core.dll	Core run-time functionality
System.Data.dll	Generic data connection functionality
System.Transaction.dll	Transactional functionality
System.Xml.dll	XML functionality (that is, serialization)

Mono Libraries

Library Name	Description
Mono.Data.Sqlite.dll	SQLite 3 bindings
Mono.Security.dll	Security components required by debugging server Mono Tools

Honeywell-Specific Libraries

Library Name	Description
HoneywellPrinter.dll	Printer specific functionality

How to Add Assemblies or Libraries

If your application requires additional assemblies (standard or non-standard), you should deploy them together with the application. You can get standard assemblies from the open-source Mono package.

Note: *Additional assemblies should target .NET Framework 4.0.*

CHAPTER

5 SERIALIZE LABEL FORMAT

How to use XML Serialization for Stored Label Formats

Because the Smart Printing API provides access to directly read or modify the list of objects attached to a label, you can serialize your label formats.

The Smart Printing Resource Kit provides a sample application illustrating how to serialize label formats. The sample is located at:

c:\Honeywell\SmartPrintingRK\Examples\Code\XmlSerialization.cs

Populate an Imported Label Format with Variable Data

There is no built-in method to set variable data fields in the Drawing objects, as XmlSerialization is a generic feature. Honeywell recommends that you use an iterator to go through all the objects in the Drawing.DrawingObjects list and search and replace data in the Data property.

If the XML format is being generated by a C# application, you may choose to set the property Name for each object, which allows you to identify an object in the DrawingObjects list by a textual identifier of your choice.

Sample Code: Export Label Objects in a Drawing

```
private static void ExportXml(Drawing drawing, string filename)
{
 // Serialize the label format (objects) to XML
 XmlSerializer xmlSerializer =
 new XmlSerializer(typeof(List<Drawing.Base>));
 TextWriter textWriter = new StreamWriter(filename);
 xmlSerializer.Serialize(textWriter, drawing.DrawingObjects);
 textWriter.Close();
}
```

Sample Code: Import Label Objects to a Drawing

```
private static void ImportXml(Drawing drawing, string filename)
{
 // Deserialize/import label format from XML
 XmlSerializer xmlDeserializer =
 new XmlSerializer(typeof(List<Drawing.Base>));
 TextReader textReader = new StreamReader(filename);
 drawing.DrawingObjects =
 (List<Drawing.Base>)xmlDeserializer.Deserialize(textReader);
 textReader.Close();
}
```

CHAPTER

6 GENERAL GUIDELINES AND BEST PRACTICES

Because the printer is an embedded system that does not support Microsoft .NET Windows.Forms, you need to take certain considerations into account when creating your application. This section lists guidelines for developing your printer application, and includes suggestions for best practices.

Limitations on Class Instantiation

Classes tied to physical printer components may be instantiated once per component per application, as shown in the next table. A second instantiation of these components in an application raises an exception, or results in undefined or unstable behavior.

Maximum Instance of a Class

Class	Maximum Instances
Communication.BluetoothListener	One
Communication.IndustrialInterface	One
Communication.SerialPort	One per physical serial port
Communication.TcpListener	One per IP port
Communication.USBHost	One per physical USB port
Drawing	One
Print Control	One
UI.Canvas	One
UI.Keypad	One
UI.LED	One

Object Lifetime Management

Although common practice in C# applications is to create objects when you need them, and then let the garbage collector clean them up after you no longer need them, some Honeywell.Printer classes should be handled differently. Honeywell recommends that you instantiate the following classes at application startup, and explicitly dispose them as the application exits:

- Drawing
- PrintControl
- UI.Canvas
- UI.Keypad
- UI.LED

Multi-Threading in UI Applications

The UI functionalities provided through the C# API are single-threaded. You should only enter the UI main-loop from the main thread, and you should only modify UI objects from the main thread.

In practice this means that a multi-threaded application must communicate any requested UI changes to the main-thread, preferably through a synchronized (thread-safe) queue. A timer event may be set up to check the synchronized queue periodically.

How to Use UI Resources?

The printer firmware contains UI resource images in .png format, used by the standard firmware. Your C# application may use these images.

For best results, Honeywell recommends that you download these resources from the printer and bundle them with the C# application instead of directly referencing them. By downloading the resources, changes to the UI resources in future firmware releases will not affect your application.

To retrieve the UI resources, connect to the printer through FTP and download this directory with its subfolders:

/usr/share/ui/images

How to Use Third-Party Libraries?

Third-party C# libraries or applications may be installed on the printer along with user created application. The printer system does not impose any restrictions, but Honeywell does not guarantee that all third-party libraries or applications will work with your printer.

How to Design Label Formats?

To maintain label formats for your C# application, the best practice is to create XML serializations of your formats. By separating the label format representations from the C# application, you can make changes to the formats without recompiling the application.

The sample applications for the Drawing classes illustrate how to create different types of label objects. The API Reference Manual includes additional information on object positioning, alignment, and other design parameters.

Configure the Printer

The C# API provides the ability to configure printer settings. Networking and network services configuration parameters require authentication (as itadmin) before you can change printer settings. This sample code illustrates switching to user itadmin temporarily to set an IPv6 IP assignment:

```
// Init
Security security = new Security();
Configuration configuration = new Configuration();
// Switch to itadmin
security.SetUser("itadmin", "pass");
// Update IPv6 IP assignment method
configuration.SetParameter(
"Communications,Ethernet,IPv6,IP Assignment Method", "DHCP");
// Switch back to user
security.SetUser("user", "");
// Cleanup
configuration.Dispose();
security.Dispose();}
```

About Printer Command Language Modes

The printers can run either with or without a command language parser listening. Set the command language mode from menu/Settings/System Settings/General:

- Smart Printing: The printer disables the Fingerprint language and instead expects your C# application to handle everything.
- Other languages: The selected command language handles the printer.

If you start a C# application while Fingerprint is the selected language, general firmware features are overridden based on which classes the C# application instantiates, as listed in the next table.

Fingerprint Feature Behavior Overridden by C# Applications

Class	Behaviors
Communication	Communication I/O (Telnet port 9100, Serial, Parallel, USB Host, USB Device, Bluetooth, XML printing, PrintSet support) is released by Fingerprint and can be controlled by the C# application.

Class	Behaviors
Print Control	Print engine control is released by Fingerprint and can be controlled by the C# application.
UI	User interface elements (LCD, LED, Buttons) are released by Fingerprint and can be controlled by the C# application.

Certain firmware functionalities are always available and active, independent of command language selection or whether a C# application is running:

- Web page service
- SNMP service & email events
- Avalanche
- SmartSystems

Considerations when Starting C# Applications from Fingerprint Applications

To start your C# application from inside a Fingerprint application, you can use the command:

RUN "mono /home/user/apps/MyApplication.exe"

However, if you plan to start your application using this command, your application may not use the PrintControl, Communication and UI classes, as they conflict with the Fingerprint runtime environment.

Printer Platform and File System Structure

The printer platform is a Linux-based system. The printer file system structure relevant to application developers is described in this table.

Printer File System Paths and Descriptions

Path	Description
/dev/ttyS0	First serial port. Additional serial ports are ttyS1, ttyS2, and so on.
/dev/ttyUSB0	First USB Host port. Additional USB ports are ttyUSB1, ttyUSB2, and so on.
/home/admin	User files owned by the user “admin”.
/home/user/	User files owned by the user “user”.
/home/user/80211	User 802.11 files.
/home/user/apps	User C# applications.
/home/user/avalanche	User Avalanche files.
/home/user/certificates	User certificate files for 802.11 and IPsec.
/home/user/config	User configuration files for command language simulators.
/home/user/display	User display files for Fingerprint customization of LCD content.
/home/user/fonts	User-installed font files.
/home/user/forms	User label formats for command languages and simulators.
/home/user/images	User image files.
/home/user/keypad	User keypad mapping files.
/home/user/logs	User log files.
/home/user/profiles	User configuration profiles.
/home/user/scripts	User Fingerprint applications.
/home/user/webforms	User label formats for INprint web service.
/home/user/webpage	User web page customization files.
/media/sda1	First mounted USB storage device, first partition. Second partition is sda2, second device is sdbX.
/tmp	Temporary (non-volatile) files.

Note: *Linux file system paths are case-sensitive.*

Access the Printer Shell

You can access the printer shell to perform basic file operations and to start applications.

1. Open a telnet connection to the printer on port 23.
2. Log in using one of these credentials:

Print Shell Login Credentials

User	Password	Access Level and Default Directory
user	(none)	The default user. Most applications use this access level. Default directory after login is /home/user.

User	Password	Access Level and Default Directory
admin	pass	Administrator with rights to modify printing-related settings. Default directory after login is /home/user.
itadmin	pass	Administrator with rights to modify printing, network, service, and access control related settings. Default directory after login is /home/admin.

Printer Shell Commands

Command	Description
cat	Display content file
cd	Change the current or working directory
cp	Copy files and directories
date	Get the current time
ls	List contents of the current directory
mono	Start a C# application
mkdir	Create a directory
mv	Rename or move files and directories
ping	Send packets to the network host
pwd	Print the name of the current or working directory
su	Switch user
top	View process activity in real time
whoami	Get the current user

Note: *To view the online help for printer shell commands (when available), type <command-name> --help and then press **Enter**.*

About Line Breaks

Line-breaks in Linux-based systems are represented as LF (ASCII 10). This is different from Windows-based systems, which use CR (ASCII 13) + LF (ASCII 10) to represent line-breaks.

A C# application developed for both Windows and Linux systems may get the property value for Environment.NewLine, which returns “\n” on the printer and “\r\n” on Windows systems.

View the Printer Image Buffer

The current label buffer content (typically the last label printed) is stored in the printer.

1. Open a web browser
2. In the URL field, type:

   http://<printer IP address>/printer/label.png
3. Press **Enter**. The label appears in the browser.

Printer LCD Considerations

When selecting a font size for displaying text on the printer LCD screen, make sure you select a size that is easily readable. Certain smaller font sizes or color combinations may be difficult to read.

The LCD screen dimensions for the printers are:

- PMx3 printers: 240 pixels wide by 320 pixels high
- PC-series printers: 314 pixels wide by 234 pixels high
- PD43 series printers: 320 pixels wide by 240 pixels high
- PD45/PD45S: 320 pixels wide by 240 pixels high
- PX240/PX240S: 320 pixels wide by 240 pixels high
- PX940 andPM45 series printers: 240 pixels wide by 320 pixels high
- PX4ie/PX6ie series printers: 320 pixels wide by 240 pixels high

Printer Keypad Information

PM-series printers with a full numeric keypad generate duplicate key events for keys with double functions (such as for the numeral “8”, which is also “arrow up”). For a list of all key codes, see the Smart Printing API Reference.

The keypad options for the printers are:

- PMx3 printers: 1-button or full numeric keypad
- PC-series printers: 1-button or navigational keypad
- PX940 and PM45 series printers: Touch screen
- PD43/PD43c/PD45/PD4S/PX240/PX240S/PX4ie/PX6ie printers: Navigation keypads

About Accessory Support

Support for certain API classes depends on the accessories installed on the printer. For example, the Communication.IndustrialInterface class requires a physical Industrial Interface board in the printer, and only the PM-series printers support this accessory.

For more information on printer accessories, contact your local Honeywell sales representative.

Additional Resources

Visit Honeywell Knowledge Base at www.hsmcontactsupport.com to review technical information or to request technical support for your Honeywell products.

Visit www.sps.honeywell.com to download PDF versions of our current product user manuals.

Join the Honeywell AIDC Technical Support Community at support.honeywellaidc.com.

Honeywell
9680 Old Bailes Road
Fort Mill, SC 29707

www.sps.honeywell.com