

Device Discovery and Access

Overview

The Java Embedded Server provides support for automatically discovering and accessing devices. It contains a device manager designed to comply with the OSGi specification, as well as a sample implementation of the `DriverLocator` service. The device manager discovers new devices by listening for registration of `Device` services with the framework registry. Once a device is registered, the device manager uses the `DriverLocator` service to find and download the drivers necessary to access the various representations of the device. The JES implementation of the `DriverLocator` service works in conjunction with a web server where it is assumed that all drivers are present. Typically, device manufacturers and gateway operators provide the necessary driver and device services. A sample implementation has been provided to demonstrate how device discovery and access works.

How Device Discovery and Access Works

To demonstrate device discovery and access, we describe how a JES gateway might discover and access a Hewlet Packard Laserjet printer with a USB connector. For this example, we assume a gateway manufacturer has provided a `USBDriver` service with the JES gateway.

- 1. The `USBDriver` service listens for any device that attaches to a USB port. When the printer is hooked up, it communicates some of its basic properties. The driver discovers the printer and registers a `USBDevice` service with those properties. The `USBDevice` service represents the printer as a generic USB device, with no methods specific to printers.**

- 2. The device manager listens for registration of any services that implement the Device interface. When the `USBDevice` service is registered, the device manager calls the `findDrivers` method on all registered `DriverLocator` services, passing the properties registered with the `USBDevice` service to the `DriverLocator`.**
- 3. The `DriverLocator` service queries a web server for drivers that may be appropriate for the USB device. The `findDrivers` method returns zero or more `DRIVER_ID` values. The device manager tells the `DriverLocator` to download the bundles. The bundles are downloaded, installed, and started.**
- 4. The device manager calls `match` on each newly downloaded driver and then calls `attach` on the driver that matches best, the `HPPrinterDriver`. The `HPPrinterDriver.attach` method creates a dependency between the driver and the `USBDevice` service. The chosen driver then registers a new `HPPrinterDevice` service with methods for accessing printer functionality.**

The process for finding, downloading, matching and attaching drivers and registering device services (step 2, 3 and 4 in this sequence) continues until no more refined drivers can be found.

FIGURE 4-1 Detecting a Printer



