Working with the Phidget Webservice

Introduction

The Phidget Web Service is a way to get and send data with any Phidget over the network, allowing any number of programs to use the Phidgets from another computer when distance or performance is a factor. It is available on each of the operating systems and languages that support Phidgets. This document will describe the Webservice concepts and usage in a general manner. For programming with Phidgets under a specific language, please see the getting started guides.

Environment and Libraries

First, we need to set up the proper environment and get the necessary files off the Phidgets web site. Visit the drivers and programming section at www.phidgets.com and get the latest:

- Phidget Framework for each system being used
- Additional libraries or packages for the language being used

You will need to install the Webservice (included with the Phidget Framework) on the system hosting the Phidgets. It is also important to keep the version of the framework installed on each networked computer consistent, as it's not guaranteed that different versions are compatible. We also recommend that you download the following reference materials from the Information section:

- API manual for your programming language
- Programming Manual

The API manual contains the specific calling conventions for your language. You can find a much more detailed discussion of the Webservice in the Programming Manual. You may want to have these manuals open while working through these instructions.

Multicast DNS and Zero Configuration Networking

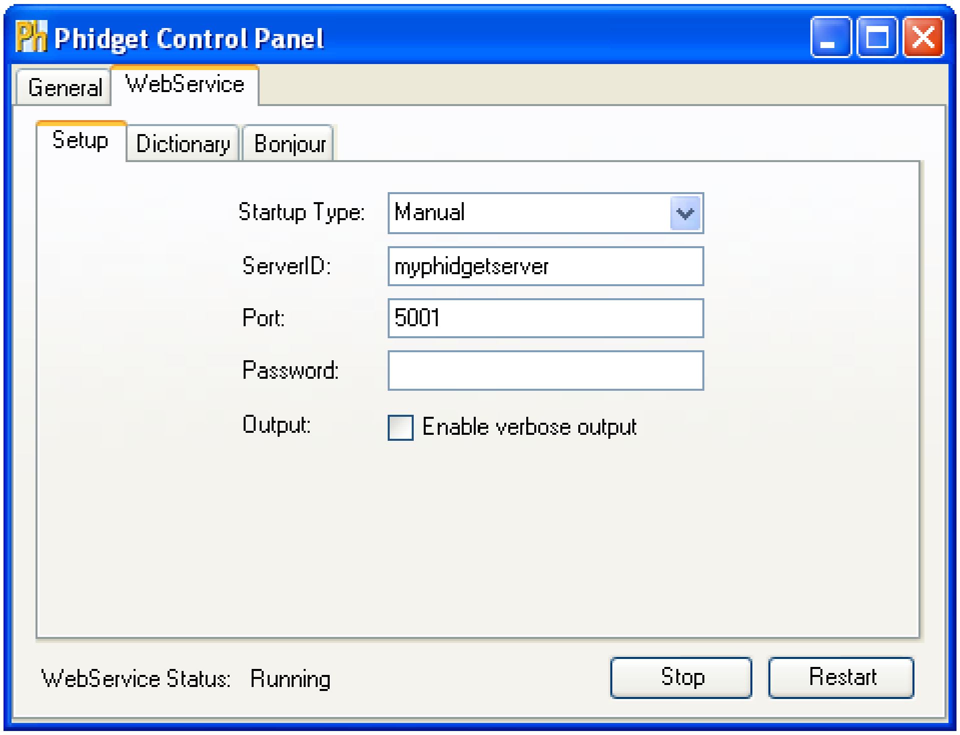
Phidgets optionally supports the use of mDNS, which allows Phidgets to be found and opened on the network by a ServerID instead of an IP address and port. When using ServerIDs, both the client and server will need to be running an implementation of zero configuration networking.

- Windows: Install Bonjour (http://support.apple.com/downloads/Bonjour_for_Windows)
- Linux: Use the Avahi package if not installed by default
- MacOSX: Bonjour is installed by default
- WinCE: This system generally does not support zeroconf

Configuring and Running the Webservice

The Phidget Webservice only needs to be running on a server that directly operates the Phidgets to broadcast their availability. A client application does not need to have the Webservice installed to be able to connect to a server and use its Phidgets.

In Windows, the Phidget Web Service can be configured and launched from the Phidget Control Panel through the WebService tab. Launching the Webservice is as simple as clicking the Start button. Similarly, the Webservice can be launched and configured from the Phidget Preference Pane in Mac OSX.



Startup Type: Manual

ServerID: myphidgetserver

Port: 5001

Password:

Output: □ Enable verbose output

WebService Status: Running

Under the WebService Setup tab, settings for the Startup Type, ServerID, Port, Password and logging are available. By default, the WebService uses port 5001. Alternatively, the Webservice can be launched by executing the PhidgetWebservice21 program from your installation directory and configured with command line parameters. Run with the -h switch to see a list of command line options

For Linux and WinCE, the Webservice must be launched through the command line. The Phidget Webservice is compiled with the Phidgets Linux Source, and in WinCE the Webservice is found in the default installation directory. Because WinCE does not support zeroconf, it does not support the use of ServerIDs.

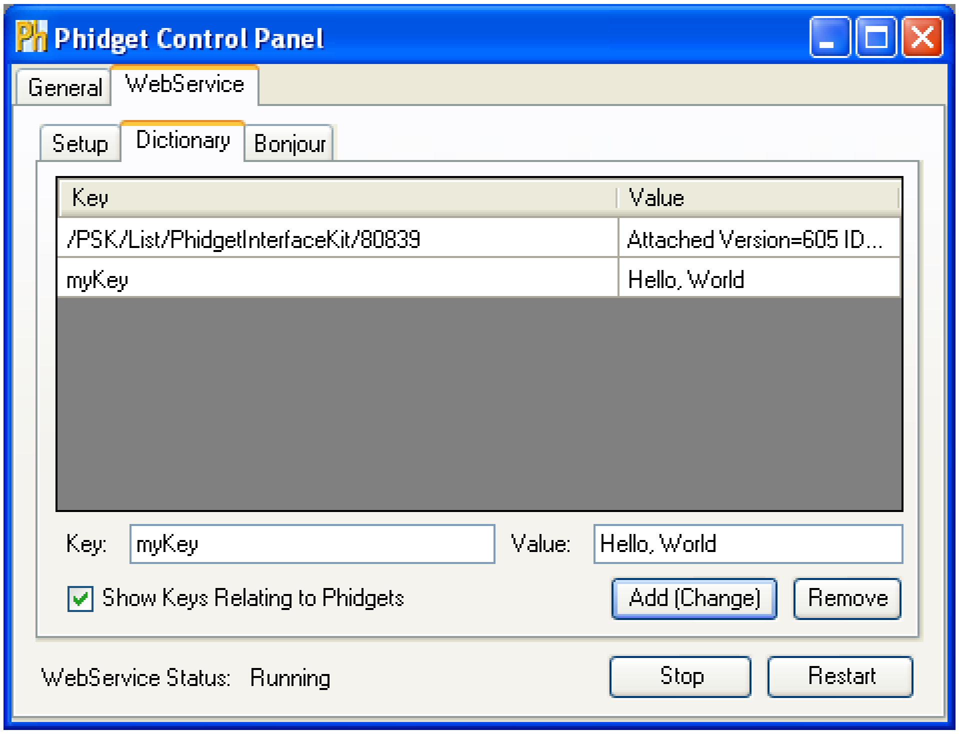
Programming with the Webservice

Once connected, Phidgets can be used in almost the exact same way over the Webservice as they are locally. There are some important changes to be aware of, which will be covered in this section.

Under the C library and extensions, you will need to use openRemote(phidget, serial, serverID, password) or openRemoteIP(phidget, serial, ipAddress, port, password) to connect to a remote server. Some languages (such as under .NET), overload the open() function instead. Also, if a serverID is not specified in openRemote, then the program will try to connect to any available server. Additional functions for getting the server information of a Phidget are also made available over the Webservice. Please see the Programming and API manuals for a complete list, and the API manual for specific calling conventions.

You should also be aware that the Webservice will allow any number of programs to connect to the server and manipulate the Phidget. However, when a program connects remotely you will no longer be able to open the device locally. A connection can still be made to itself through the Webservice using the localhost address. Also, when using the Webservice, you should account for latency in the propagation of information and commands.

The Webservice is managed behind the scenes through the use of the Phidget Dictionary.



Key	Value
/PSK/List/PhidgetInterfaceKit/80839	Attached Version=605 ID...
myKey	Hello, World

Key: myKey Value: Hello, World

☑ Show Keys Relating to Phidgets

WebService Status: Running

Information from Phidgets are stored in key-value pairs prefixed with /PSK/ or /PCK/, and remote function calls use these entries. Additional user defined information can be passed between programs through entries in the dictionary if necessary.

Testing the Phidget Webservice with Provided Examples

The Phidget Webservice can be tested with samples provided by Phidgets. In Windows and MacOSX, Phidgets over the Webservice are listed in the Control Panel under the Bonjour tab. Double clicking the an item in the list will bring up an example program for that Phidget.

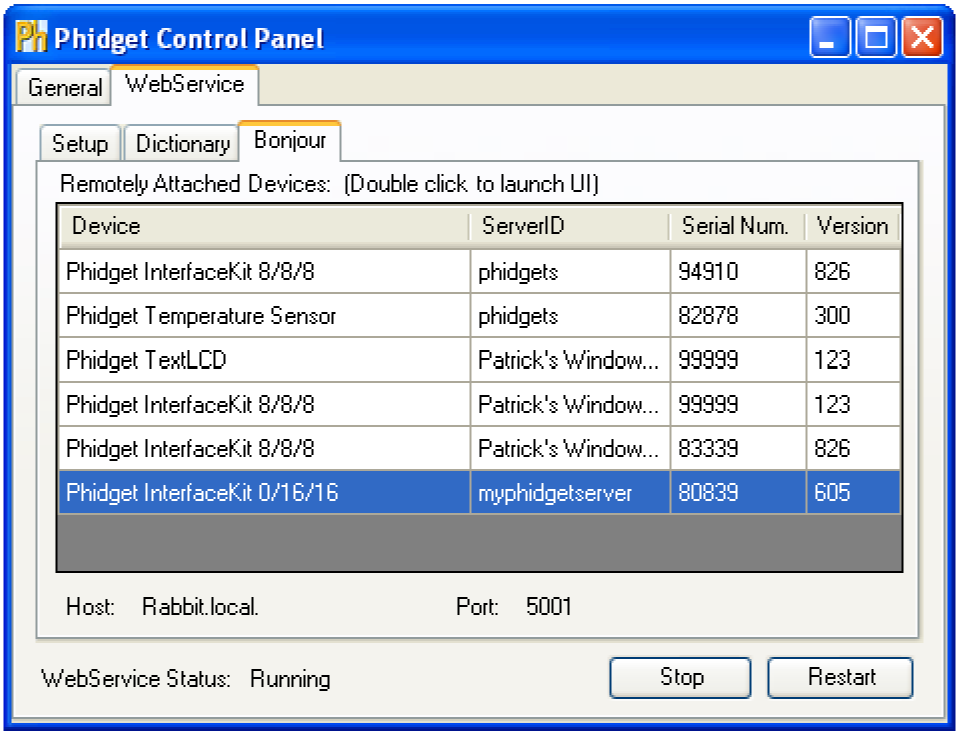
In addition, you can also look at and play with the code from the C# examples to see how it connects. Each full example calls an openCmdLine function that uses the appropriate open based on the parameters given.

Try it out yourself:

```
Usage: appname [Flags...]
Flags: -n Serial Number        Serial Number, omit for any serial
       -r                      Open remotely
       -s serverID             ServerID, omit for any server
       -i ipAddress:port       IP Address and port. Port is optional, defaults to 5001
       -p password             Password, omit for no password
```

```
Examples:   appname -n 50098
            appname -r
            appname -s myphidgetserver
            appname -n 45670 -i 127:0.0.1:5001 -p paswrd
```

The C# examples can be found at http://www.phidgets.com/downloads.php?example_id=10.



Remotely Attached Devices: (Double click to launch UI)

Device	ServerID	Serial Num.	Version
Phidget InterfaceKit 8/8/8	phidgets	94910	826
Phidget Temperature Sensor	phidgets	82878	300
Phidget TextLCD	Patrick's Window...	99999	123
Phidget InterfaceKit 8/8/8	Patrick's Window...	99999	123
Phidget InterfaceKit 8/8/8	Patrick's Window...	83339	826
Phidget InterfaceKit 0/16/16	myphidgetserver	80839	605

Host: Rabbit.local. Port: 5001

WebService Status: Running