Executive Summary

There currently is no process for tracking issues on my home network which includes laptops, desktops, tablets, phones, and other devices which connect through a wireless IP router. The term used commonly today is Internet of Things (IoT). The following sections of this report detail a solution of an issue tracker tool application to help document issues and an ASP.NET web site to help test connectivity to devices.

Problem Statement

Multiple users on my home network communicate issues to me (the ‘IT person’) about problems they are currently having with their devices. Currently the issues are noted but not documented. Issues can easily be forgotten and not resolved. There is also no centralized way to test connections to devices (i.e. ping).

Problem Resolution Plan

The problem was separated into two major tasks:

* Implemention of a C# Windows Form application for performing the features of an issue tracker. Issues are stored in a SQL Server Express LocalDB mdf file. Issues can be added and deleted. The tool includes a feature for displaying a list of issues sorted by user that can be printed. The application is called the ‘Internet of Things Issue Tracker’.
* Implemention of an ASP.NET web site which allows simple testing of a device by pinging a device IP address. The user can select the device in a drop-down control and hit a button to perform the test. A database is queried in order to populate the list of devices on the network. C# code was implemented in separate \*.cs files for easier maintenance.

Breakdown of Development Time / Costs

1. Implementation of C# Windows Form Application:

* SQL Database population – 4 hours
* C# Windows Form Issue Tracker feature using DataSource Wizard dataset – 16 hours
* C# Windows Form Report Viewer – 8 hours

1. Implementation of ASP.NET website – 8 hours

Risk Assessment

A major risk involves the time constraints for testing on other target machines and different web browsers.

Areas for Improvement of Code

Issue Tracker:

* The code can be enhanced for specifying more friendly exception messages to the end-user.
* The auto-documentation feature of C# could be used on methods and classes.
* The use of DataSource wizard resulted in the presentation and data access layers being intertwined. Separate classes for data access could separate the logical aspects of the application.

Web Site:

* Code can be enhanced to give status when pinging device. Currently the user does not realize that they need to wait for the response.
* Currently a duplicate database is used in the WebSite project. It would be best to have both applications pointing to the same database.
* The web site only seems to work on my machine’s Visual Studio Professional environment. An attempt to publish to a hosted web site was accomplished:  
  <http://jimhar66-001-site1.etempurl.com/>

Everything seems to work OK on web site except for the ping command which may be due to firewall issues. It may be benefical to setup a web server on my local network using IIS.

Conclusion

The implementation of the issue tracker and web site will allow for better support of issues that end-users on my home network may encounter. The tools are scalable for additional features that may be needed in the future.