

Using Dundas Chart for .NET within an ASP.NET web service for server-side production of charts and graphs for inclusion in WordML documents

The Jackson Organization is a full-service healthcare market research firm that provides high-response custom surveys, comprehensive reporting and consulting. We offer custom research for patient, medical staff, referring physicians, employee satisfaction and engagement, and consumer research. This research is conducted by phone, web and mail and is delivered in paper reports as well as through our online reporting system, StatData®.

While the trend in the survey research industry is toward web-based reporting, many of our customers continue to receive Microsoft Word-based reports. The Jackson Organization produces thousands of reports in Word format each year, and our team was tasked with automating the production of charts and graphs for these reports.

The Problem

We have created a .NET web service that produces WordML representations of our reports. WordML is an XML schema for representing Word documents introduced by Microsoft with Office 2003. The web service retrieves data for a report from an Oracle 9i database, performs any necessary calculations and produces an XML representation of the report. The web service then runs the XML document through an XSLT transformation to produce a WordML document that can be opened directly with Microsoft Word or converted to a binary Microsoft Word (.doc) file.

Microsoft Word's native MSGraph charts and graphs are represented in WordML as binary data, and there is not a way to generate this binary data on the server without automating MSGraph. MSGraph does not lend itself to server-side automation, so we researched third-party alternatives.

The Solution

Our team had had a positive experience using the



ASP version of Dundas Chart in our reporting web site, so we decided to download a trial of the ASP.NET version of Dundas Chart. The ASP.NET version of Dundas Chart was key to making this a purely server-based solution.

Dundas Chart's support for XML allows the web service to create a chart or graph by simply running our existing XML chart representation through an XSLT transformation. Using the Dundas API, the web service generates a binary representation of a chart or graph and inserts the binary data directly into the WordML document. Because the Dundas Chart API supports writing to memory streams, we can accomplish this task in-memory without having to manage files on the server.

The key features of the ASP.NET version of Dundas Chart that enabled this project are:

- Fully managed/easy integration with .NET Web Services
- Support for directly creating charts and graphs using XML
- Support for multiple binary file formats (PNG, BMP, etc.)
- Support for writing to .NET memory streams
- Ability to reuse chart and graph classes created for this project in future ASP.NET reporting web sites.

(cont'd page 2)

Using Dundas Chart for .NET within an ASP.NET web service for server-side production of charts and graphs for inclusion in WordML documents

Dundas Chart has drastically reduced our development time. Compared to the previous version of our software that rendered charts and graphs on the client using MSGraph, I estimate that we are able to create the presentation-layer code for a chart or graph in a third of the time.

Given our team's positive experience with the ASP version of Dundas Chart, I expected that Dundas Chart for ASP.NET would be a great solution for rendering charts and graphs in ASP.NET web applications. What I did not expect was how well the package works within an ASP.NET web service for server-side production of charts and graphs for inclusion in WordML documents.

Ben Whaley
Lead Developer
The Jackson Organization
<http://www.jacksonorganization.com>

