

## Application of Fusioncharts in Web System

ZHANG Ting

Foundation Department,  
PLA University of Foreign Languages  
Luoyang, China  
e-mail: candy1984315@163.com

**Abstract**—This This paper mainly introduces Flash report components Fusioncharts of functions and basic usage, emphatically describes how Java Web development system provided by the use of dynamic data-driven Fusioncharts icon-based, and through examples to achieve Fusioncharts Web system.

**Keywords**- Fusioncharts, dynamic icon, Web System

### I. FOREWORD

Internet has been deeply in people's lives-more and more enterprises, organizations and institutions have formed or are setting up their own internal networks (intranet), and database application development from the traditional C / S (Client / Server) architecture to the Internet / Intranet-based B / S (Browser / Server) architecture changes. Statements, as an effective means of information organization and analysis, and important present form, are an important part of enterprise management information system. How to conveniently, fast and accurately, generate a variety of statistical reports is a subject worth studying.

With Flash technology-based dynamic charts component of FusionCharts, you can create a simple, interactive and compelling dynamic charts. This article first briefly describes the features and the use of Fusioncharts. Then a staff Task statistical system as an example complies the application of the Web system.

### II. INTRODUCTION

#### A. Introduction of FusionCharts

FusionCharts is a Flash chart components, it can be used to make the data animation chart, the animation produced by the Adobe Flash 8 (formerly Macromedia Flash). FusionCharts can be used for any web scripting language similar to HTML, .NET, ASP, JSP, PHP, ColdFusion, etc., to provide interactive and powerful charts, you do not need to know any flash programming knowledge, and you just need to know what programming language you use<sup>[1]</sup>. FusionCharts uses XML as its data interface, the full use of fluid beauty of Flash can create compact, interactive and visually arrest chart.

#### B. Features of FusionCharts

FusionCharts has the following characteristics:<sup>[2]</sup>

- Not requiring Active-X or extended controls.
- Adding life like aesthetic effect to the site.

- Reducing server load.
- Being compatible with a variety of scripting languages.
- Changing the dynamic database of client.
- Appending other features in the graph.

#### C. Basic Usage of FusionCharts

Use FusionCharts Free to create a simple pie 3D chart to display a summary of the monthly sales.(See Fig. 1)

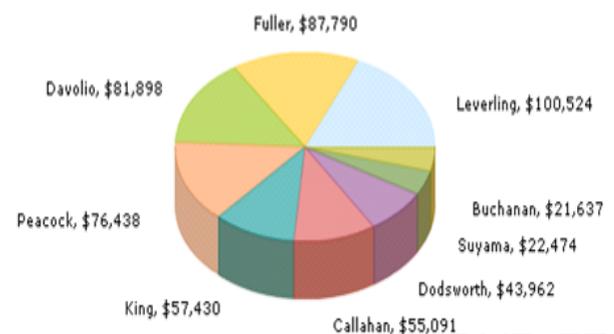


Figure 1. Fusioncharts Free-Pie 3D Chart

To use FusionCharts Free to create graphics, do the following three things:

First, choose what you want to create the type of graph corresponding SWF file. All of the SWF file in FusionCharts Free> Charts folder.

Second, the XML data files. FusionCharts Free only accepts data in XML-based format. Therefore, the data needs to be converted to XML format.

Third, the HTML files. This HTML file will contain the code used to embed graphics.

### III. EXAMPLE OF WEB SYSTEMS APPLICATION

Fusioncharts is based on XML data interface and generates charts<sup>[3]</sup>. It provides two XML forms: one is that the XML file directly provides data that is written in advance. The example above is based on this approach. Another is based on a database to dynamically generate XML. Taking a staff Task statistical system as an example complies this second method.

System will combine Fusioncharts with Extjs and use Extjs Ajax, and dynamic data extracted from the database

will be transmitted into XML format to Fusioncharts interface showing a brilliant flash graphical report.

When the user clicks the Statistics button on the page, the request is submitted after verification judgment on the legitimacy of the user input via JavaScript, backstage Servlet or Action summary of the data in the database will retrieve the required conditions, get the results and respond to the output in accordance with the FusionCharts control XML format required, then get the XML content to serve FusionCharts control through AJAX technology. What is particularly worth mentioning is that the document production attached is very specific and comprehensive. It is very convenient to realize the chart via documentation provided.<sup>[4]</sup>

#### A. Defined various of XML elements in Background

In the Fusioncharts Free the system uses the ScrollCombi2D.swf, and each element contained in the xml file is the dynamic data-driven icon, should be defined in Java in the background.

Java xml file operation is a very mature technology, and we use the xml comments in the JDK (Annotations) to define each element node in the xml file, what is needed to use are the XmlAttribute, XmlRootElement and XmlElement, three annotation types of javax.xml.bind.annotation package. Using these three types, we can define FusionChart class, FusionChartCategories class, FusionChartCategory class, FusionChartData class and FusionChartDataset. The XmlAttribute is mainly responsible for the properties of elements in the xml file, and the XmlElement is mainly responsible for the assignment of elements in the xml file, that is, the value got from the background is assigned to the element.

The core code of Fusioncharts class is as follows:

```
package com.ndsc.no7ss2.service.fusioncharts;

import java.util.ArrayList;
import java.util.List;

import javax.xml.bind.annotation.XmlAttribute;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "chart")
public class FusionChart
{
    @XmlAttribute(name = "caption")
    public String caption;
    @XmlAttribute(name = "lineThickness")
    public int lineThickness;
    @XmlAttribute(name = "showValues")
    public int showValues;
    @XmlAttribute(name = "areaOverColumns")
    public int areaOverColumns;
    @XmlAttribute(name = "formatNumberScale")
    public int formatNumberScale;
    @XmlAttribute(name = "useRoundEdges")
    public int useRoundEdges;
```

```
@XmlAttribute(name = "palette")
public int palette;
@XmlAttribute(name = "legendBorderAlpha")
public int legendBorderAlpha;
@XmlElement(name = "categories")
public FusionChartCategories categories;
@XmlElement(name = "dataset")
public List<FusionChartDataset> datasets;
@XmlElement(name = "trendLines")
public FusionChartTrendLines line;
```

```
public FusionChart( )
{
    categories = new
    FusionChartCategories();
    datasets = new
    ArrayList<FusionChartDataset>();
    line = new FusionChartTrendLines();
}
}
```

#### B. Data reception

Data from the database query is stored in the form of a List, which is needed to read from this List to the xml element corresponding to each data, assign, and finally get a FusionCharts which can read xml format file<sup>[5]</sup>.

From List to value, and assignment to previously defined individual elements is relatively easy, but the focus of this section and the difficulty is: how to package the value of the xml elements that has been got into a xml format file. Java provides an interface Marshaller, and the Marshaller class is responsible for the process of managing the serialized Java content tree back to XML data.

The first parameter of the overloaded Marshaller.marshall must be JAXB element calculated by JAXB Introspector.isElement; Otherwise, Marshaller.marshall must throw MarshalException, an instance coexistence of two kinds of mechanism for marshalling a JAXB element. One approach is to pack the instance as the JAXBElement value, and put the wrapper object as the first parameter to transmit to the Marshaller.marshall method. Another is a Java bound to the mode, you can simply use the @ XmlRootElement annotation to the class of the instance.

The core code is as follows:

```
String task_month;
StringWriter writer = new StringWriter( );
StreamResult result = new StreamResult(writer);

try
{
    JAXBContext context =
    JAXBContext.newInstance(FusionChart.class);
    Marshaller marshaller =
    context.createMarshaller( );
    marshaller.marshall(fusionChartObject, result);
}
catch (JAXBException e)
{
```

```

        logger.error(e.getMessage());
        return null;
    }
    task_month = writer.toString();
    task_month = task_month.replace("'", "");
    return task_month;
}

```

C. Foreground showing

The foreground uses the way of combing Extjs with Fusioncharts, which mainly use Extjs Ajax approach and the Ext Grid properties and methods. When the user clicks the Query button, the results are first presented via Grid form; When the user Double-clicks any line of the Grid, through Extjs Grid control "rowdblclick", it will pass the Parameter of this line to the back-end database, and query again with the pop-up window, and get columnar task Fusioncharts comparison chart. (See Fig. 2)

The core code is as follows:

```

var FusionChartView = function(tagId, XML)
{
    var MyChart = new FusionCharts(
        Content/test/ScrollCombi2D.swf",
        "SwatchQuerymyChartId", "1000", "400");
    MyChart.setDataXML(XML);
    MyChart.render("FusionChart");
    return MyChart;
}

```

```

Grid_Result.on("rowdblclick",function(grid, rowIndex, e)
{
    var Rec = Result_store.getAt(rowIndex);
    var OpIdParams = Rec.get("OP_ID");
    Params.OpId = OpIdParams;
    Ext.Ajax.request({
        url : ".../.../...",
        success : Ajax_Success,
        failure : Ajax_Failure,
        params : Params
    });
}

```

```

var FusionChart_Windows = new Ext.Window({
    title : 'cylindrical graphic',
    width : 1030,
    height : 500,
    layout : 'fit',
    renderTo : tagId,
    constrain : false,
    autoDestroy : true,
    html : '<div id="FusionChart"></div>',
    buttons : [{
        text : 'closed',
        handler : function()
        {
            FusionChart_Windows.close();
        }
    }
}

```

```

    }
    });
    FusionChart_Windows.show();
});
}
}

```

From the code can be seen, the Fusioncharts XML is based on the database dynamic.



Figure 2 Columnar chart of task comparison

IV. CONCLUSION

Fusioncharts presents practical and brilliant Flash graphics statements, but it not only provides functionality. We can also take advantage of the technology of Html Div to present multiple graphics statements in a page and we can use the Fusioncharts drilling technology to present more detailed data. In short, Fusioncharts is a good graphics-reporting tool. With the growing trend to the B / S architecture, it is playing an increasingly important role in statements field.

REFERENCES

- [1] FusionCharts Free Chinese Developer's Guide.
- [2] FusionCharts chart control Chinese version of the user manual.
- [3] Liu Genglong, Sui Wei aromatic. The FusionCharts chart under C / S mode design and realization [J]. Computer Knowledge and Technology. 2010.05:
- [4] Yang Kai. To use FusionCharts components to create dynamic web charts [J]. Microcomputer world. 2009.07:
- [5] Han Yibo, Song Li, CCP YUNNAN AJAX technology combines XML or JSON use [J] Computer Knowledge and Technology 2009, 5 (1) :101-103