Paper 49-28

Developing Data-Driven Applications Using JDBC and Java Servlet/JSP Technologies

Chad Ferguson, SAS Institute Inc., Cary, NC
Sandra Brey, SAS Institute Inc., Cary, NC

ABSTRACT

Many enhancements have been made to the SAS Java component library available with Version 3 of AppDev Studio™ (ADS). Several of the improvements include the development of new TransformationBeans™ and JDBC™ adapters for building custom table and form-based, data-driven applications. This paper will provide server-side Java examples that you can use to build Web applications which will allow the end user to:

- edit data via a table or a custom form
- sort the data
- navigate through the data using a flexible NavigationBar component
- filter the data
- add computed items
- export the data to Excel
- and much more.

A greatly expanded version of this paper will be available on the web by SUGI. It will contain a more detailed discussion of all the topics listed above, including actual code examples. You can look for the final version of the paper on the AppDev Studio Developer's Site (http://www.sas.com/rnd/appdev/).

INTRODUCTION

Due to the increased need for Web-based applications, Version 3 of ADS includes enhanced support for developing data-driven Web applications using Java Servlet™ and Java Server Pages™ (JSP) technology. The new release includes new component viewers that can access relational data sources via JDBC™ technology. Using WebAF™, AppDev Studio's interactive Java development environment, you can easily create a Java Servlet or JSP project by dragging and dropping the desired components onto the servlet or JSP. The servlet or custom tag code will be generated automatically for you.

This paper will focus on the powerful TableView component and how it is used to display data from JDBC data sources. Other components in the SAS Java component library will be described as well, such as the DataBeanForm, JDBC adapters, and various data selectors.

The TableView is a composite component; it is made up of several sub-components which assist in the manipulation of the data. The NavigationBar component, for example, assists the TableView in scrolling through the rows or columns of data. The TableView allows you to set the number of rows and/or columns that will be visible on each page. Using the visible controls displayed by the NavigationBar, the end user can scroll forward and backwards through the pages of data. Another component, the RelationalMenuBar, gives you an area in the table to display any desired data selectors. The data selectors work in conjunction with the TableView to allow functionality such as filtering the data, adding computed items, sorting the data and exporting the data to Excel. The TableView also allows you to set cell renderers on a per cell basis. This gives you a way to customize the TableView and display your data any way that you like.

An exciting new feature in the TableView for Version 3 is editing. This allows you to build web-based servlet or jsp applications which enable the end user to edit data via HTML. The data may be edited one row at a time or all at once (like a spreadsheet). When editing, you can set a cell editor which give you more control over valid value. For example, you could use a ChoiceBox as your cell editor and the ChoiceBox would contain a list of valid values to choose from while editing a particular cell. Or you could use a TextEntry as your cell editor and then you could set a validator to maintain the valid values during the data entry process.

The TableView's model must implement Java's Swing TableModel interface. The SAS component library provides multiple adapters for the TableView that allow it to connect to different data sources. One of these adapters is the JDBCToTableModelAdapter, which allows the TableView to take advantage of the JDBC technology and communicate with a relational database or data source.

If you are interested in building Web applications for data entry or editing, you will also want to take a look at the DataBeanForm component. This component is also new to Version 3 and allows the end user to view and edit data from a JDBC data source as well. Unlike the TableView component which presents data in a spreadsheet view, the DataBeanForm is used to build form based Web applications that display or edit one row of data at a time.

The DataBeanForm also takes advantage of the NavigationBar component's functionality, allowing the end user to scroll through the data row by row. Using the GoTo field which is part of the NavigationBar, the user can also enter the row number they want to scroll directly to – a short cut which quickly gets them to the right place in their data.

The model for the DataBeanForm is a databean generated by WebAF's DataBean Wizard. The DataBean Wizard creates a JavaBean which models a row in your data source and provides properties for selected columns in the data source. The databean's properties are then set on various form elements within the DataBeanForm.

The JDBC adapters that are new with Version 3 of ADS allow JDBC data sources to be models for a variety of viewers requiring Java's Swing models. In addition to the TableView, the JDBCToTableModelAdapter can also be used by many of the new chart components in ADS. Other JDBC adapters available in ADS are:

- JDBCToTreeModelAdapter for viewers requiring a Java Swing TreeModel
- JDBCToListModelAdapter for viewers requiring a Java Swing ListModel
- JDBCToComboBoxModelAdapter for viewers requiring a Java Swing ListModel.

Also new in Version 3 of ADS is a set of BusinessQuery adapters. These adapters use an Intelligent Query technology which provides a business view of the data where the end user is unaware of where physical data is located. Similar to the JDBC adapters, these will allow you to use Intelligent Query data sources as models for a variety of viewers. The TableView, for example, will use the BusinessQueryToTableModelAdapter.

CONCLUSION

AppDev Studio gives application developers a simple way to build data-driven Web applications that utilizes JDBC technologies. Using a combination of the new TransformationBeans available in Version 3 – tables, charts, treeviews, listboxes, comboboxes and more – combined with the new JDBC adapters, you can easily build sophisticated data-driven applications using server-side Java.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Chad Ferguson
SAS Institute Inc.
Work Phone: (919) 531-6127
Email: Chad.Ferguson@sas.com

Sandra Brey
SAS Institute Inc.
Work Phone: (919) 531-2544
Email: Sandra.Brey@sas.com