Pro SQL Server 2008 Analytics: Delivering Sales and Marketing Dashboards

Copyright © 2009 by Brian Paulen and Jeff Finken

ISBN-13 (pbk): 978-1-4302-1928-6

ISBN-13 (electronic): 978-1-4302-1929-3

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `http://www.springeronline.com`.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail `info@apress.com`, or visit `http://www.apress.com`.

The source code for this book is available to readers at `http://www.apress.com`. You will need to answer questions pertaining to this book in order to successfully download the code.

CHAPTER 1

Introduction
Let's Find Out What This Book Is All About!

This chapter introduces you to the theme of "rich clients." In the process, you will learn what a rich client is and how a rich client platform can help you. In addition, we will briefly touch on the main advantages and characteristics of the NetBeans Platform.

What Is a Rich Client?

In a client server architecture the term "rich client" is used for clients where the data processing occurs mainly on the client side. The client also provides the graphical user interface. Often rich clients are applications that are extendable via plugins and modules. In this way, rich clients are able to solve more than one problem. Rich clients can also potentially solve related problems, as well as those that are completely foreign to their original purpose.

Rich clients are typically developed on top of a framework. A framework offers a basic starting point on top of which the user can assemble logically related parts of the application, which are called modules. Ideally, unrelated solutions (such as those made available by different providers) can work together, so that all the modules appear to have been created as one whole. Software developers and providers can also bundle rich client distributions from distinct modules, with the aim to make these available to specific users.

Above and beyond all that, rich clients have the advantage that they are easy to distribute and update, such as via an automatic online update function within the client itself or through a mechanism that enables the rich client to start over the Internet (for example, via Java Web Start).

Here's an overview of the characteristics of a rich client:

- Flexible and modular application architecture
- Platform independence
- Adaptability to the end user
- Ability to work online as well as offline
- Simplified distribution to the end user
- Simplified updating of the client

What Is a Rich Client Platform?

A rich client platform is an application lifecycle environment, a basis for desktop applications. Most desktop applications have similar features, such as menus, toolbars, status bars, progress visualizations, data displays, customization settings, the saving and loading of user-specific data and configurations, splash screens, About boxes, internationalization, help systems, and so on. For these and other typical client application features, a rich client platform provides a framework with which the features can quickly and simply be put together.

The configurability and extensibility of an application take center stage in a framework of this kind. As a result, you can, for example, declaratively provide the menu entries of an application in a text file, after which the menu will be loaded automatically by the framework. This means that the source code becomes considerably more focused and manageable, and developers are able to concentrate on the actual business needs of the application, while the menu is maximally configurable.

The most important aspect of a rich client platform is its architecture. Applications based on rich client platforms are written in the form of modules, within which logically coherent parts of an application are isolated. A module is described declaratively and automatically loaded by the platform. As a result, there is no explicit binding necessary between the source code and the application. In this way, a relatively loosely coupled relationship is established between independently functioning modules, by means of which the dynamic extensibility of the application and the ability to swap its constituent parts are enormously simplified. In this way it is also very easy to assemble user- or domain-specific applications from individual modules.

A rich client platform also frees the developer from being concerned with tasks that have little to do with the application's business logic. At the end of the development cycle, you achieve a well-deserved and modern application architecture.

Advantages of a Rich Client Platform

Aside from the modularity offered by a rich client architecture, which simultaneously implies a high degree of robustness and end user value, the extensive development support it provides needs to be highlighted as well. These and other advantages of rich client platforms are briefly described here:

- Reduction in development time
- User interface consistency
- Updating
- Platform independence
- Reusability and reliability

We'll look at each in turn.

Reduction in Development Time

A rich client platform provides a multitude of APIs for desktop application development. For example, these can be used by developers to manage windows and menus or support the display of customization options. Through the reusability of many predefined components, developers are able to concentrate very closely on the business logic of the application in question.

User Interface Consistency

Usability of an application is always of crucial concern, in particular when the application is intended to be used by professionals of a particular domain. A rich client platform makes available a framework for the display of user interfaces, while taking particular care of its consistency, accessibility, and usability.

Updating

Using a rich client platform, it becomes possible to quickly and efficiently distribute new or updated modules to end users. As a result, not all the clients of an application need be informed by developers to switch to a new version. Updates can be distributed and installed in the form of modules, so distinct features can be developed and delivered by independently operating teams. The modular architecture of the application ensures that completed modules can be distributed without having to wait for other modules to be finalized.

Platform Independence

Rich client platforms are based on international standards and reusable components. As a result, Java applications based on them can be automatically deployed to multiple different systems, such as Windows or Linux, so long as an implementation of the Java Runtime Environment is available. Since the feature set and the applicability of applications keep changing, it is very important that they are developed in such a way that they are extendable and can be deployed to different target systems. All this is provided by a rich client platform, saving time and money. Applications based on rich client platforms do not require further libraries or components, other than the Java Runtime Environment.

Reusability and Reliability

Rich client platforms make a range of features and modules available, which can be used in the developer's own applications. If the module does not completely match the application's requirements, it is entirely possible to use it as a starting point, while extending it or changing it as needed. Since most platforms also make their source code available, it may also, in some cases, be worth considering changing or extending the platform itself. These factors imply a high degree of reliability and freedom.

Characteristics of the NetBeans Platform

The NetBeans Platform offers, aside from the generic advantages of a rich client platform, numerous frameworks and several further specifics that can be particularly useful to your applications. The important ones, which constitute the main characteristics of the NetBeans Platform, are outlined here:

- User interface framework
- Data editor
- Customization display
- Wizard framework
- Data systems

- Internationalization
- Help system

We'll look at each in turn.

User Interface Framework

Windows, menus, toolbars, and other components are made available by the platform. As a result, you focus on specific actions, which condense your code, making it better and less error-prone. The complete user interface offered by the NetBeans Platform is based 100% on AWT/Swing and can be extended with your own components.

Data Editor

The powerful NetBeans editor within the NetBeans IDE can be used by your own application. The tools and functionality of the editor can quickly and easily be extended and adapted to the purposes of the application.

Customization Display

A display of user- and application-specific settings is needed in every application. The NetBeans Platform makes a framework available, making it extremely simple to integrate your own options dialogs, letting the user save and restore settings in a way that is pleasing to the eye.

Wizard Framework

The NetBeans Platform offers simple tools to create extendable and user-friendly assistants, guiding the user through complex steps in the application.

Data Systems

In terms of the NetBeans Platform, data can be local or available via FTP, CVS, a database, or an XML file. By means of abstraction, data access by one module is transparent to all other modules. Actual data access itself is therefore not a concern, since it is dealt with by the NetBeans Platform's APIs.

Internationalization

The NetBeans Platform provides classes and methods enabling the internationalization of JavaHelp and other resources. You can easily store text constants in properties files. The NetBeans Platform also loads text constants and icons applicable to the current country and language settings.

Help System

By means of the standard JavaHelp system, the NetBeans Platform offers a central system for the integration and display of help topics to the end user. In addition, individual modules can contribute their own topics to the application's help system. On top of all that, the NetBeans Platform lets you provide context-sensitive help as well.

Summary

In this chapter, you learned the difference that a rich client can make. We discussed advantages a rich client brings to the table, including its modular architecture, made possible by a module system unique to rich client platforms. However, a rich client platform offers many other advantages and features. Among these, support for a consistent user interface and the update of applications with new features at runtime. Finally, we examined the most important characteristics of the NetBeans Platform.