

Executive Summary

Now, more than ever before, technology drives productivity in the nonprofit workplace. Hardware becomes less expensive and more powerful every day. Software enhancements offer new features to efficiently automate administrative tasks. And the Internet is an amazing medium for communication and commerce.

Blackbaud embraces these technological advances and seeks to harness them in innovative ways that bolster your organization's productivity. Our goals for version 7 of **The Raiser's Edge®** included designing a product that is:

- ◆ Easy to learn and use
- ◆ Full of productivity-enhancing features
- ◆ Web-enabled
- ◆ Customizable

This technical paper discusses several key technologies in version 7 of **The Raiser's Edge** that helped us reach these goals for the product and — ultimately — help you, our client, reach your goals.

Advanced Technology in The Raiser's Edge® 7

Microsoft Visual Studio

The Raiser's Edge was developed using Microsoft's Visual Studio. Visual Studio includes a complete set of tools for creating software for the Microsoft Windows operating system and the World Wide Web. A key component of Visual Studio is Visual Basic, the most popular programming environment in the world — used by more than 3.2 million professional developers. Visual Studio also includes Java programming tools for creating applications to run on the Web.

The World Wide Web provides tremendous opportunities for nonprofit organizations to work more effectively, both in raising funds and fostering their missions. Blackbaud recognizes the power of the Web as a business communications tool and has designed **The Raiser's Edge** from the ground up to support Web access. By using the Web interface to **The Raiser's Edge**, clients can access fundraising data, features and reports over the Internet or an intranet.

Regardless whether **The Raiser's Edge** is accessed through Windows or the Web, all users work from the same database — the same security settings and passwords apply, and data stays consistent. The Web and Windows screens are similar in structure and appearance, reducing the need to re-learn the product when changing platforms.

We used advanced technology to make The Raiser's Edge ideal for Web access. By integrating a blend of Dynamic HTML, VBScript, JavaScript and Active Server Pages, COM, .NET and the multi-tier architecture, we provide a fast, reliable and secure method of Web access.

Component Object Model (COM)

The Component Object Model (COM) is a software architecture that facilitates creating and extending software functionality. In COM, an object is a piece of compiled code within a software system that represents a high-level abstraction of data elements and provides some service to

Contents

Executive Summary	1
Microsoft Visual Studio	1
Component Object Model (COM) ..	2
Component Object Model in The	
Raiser's Edge	2
.NET Interoperability	3
Visual Basic for Applications	3
Benefits of VBA in The Raiser's Edge	3
Open Application Programming	
Interface (API)	4
RODBA	5
RE:VBA vs. RE:API vs. RODBA	5
Multi-tier Architecture	6
Presentation Logic	6
Business Logic	6
Database	6
Database Options for The Raiser's	
Edge	7
Crystal Reports	7
Conclusion	7
About Blackbaud	8

other items in the system. Essentially, this “object-oriented” architecture breaks down complex coding into smaller usable components. These components serve as the building blocks of software programs.

COM was created because traditional software development required that application executables be compiled and linked with their dependencies. This made updating software programs inefficient because every time a developer wanted to add new capabilities, he or she needed to modify and recompile the primary executable to support the dependencies.

Consider this real-world analogy: What if you had to disassemble and rebuild the entire lamp every time you needed to change a light bulb? Owning and maintaining a lamp would become cost-prohibitive, and demand for lamps would decline. Consumers, home furnishing stores and manufacturers would suffer. But fortunately, we are enlightened enough to use bulbs that are separate from, but work with, lamps. Further, we have the luxury of choosing the light bulb we want for our lamp — soft white or bright, 60-watt or 100-watt, etc. — without worrying about the bulb’s compatibility with our lamp because table lamps have a standard socket size, and bulbs conform to that standard.

Like lamps and light bulbs, COM was developed so that software developers could plug new features into existing applications without having to rebuild the existing application — a more efficient and elegant solution.

Component Object Model in The Raiser’s Edge®

The Raiser’s Edge has been designed from the ground up using COM to provide easy customization and extensibility. A major goal for **The Raiser’s Edge** was for it to be programmable and accessible enough so that even the casual programmer could make some very powerful enhancements to the system. With this goal in mind, the entire system was built internally using the COM standard.

For example, there is a Gift object, which is used to access all of the underlying data stored with a gift. The object can be used to add, edit or even delete a gift. To modify a gift, the programmer need only set a “property” on the object. Properties are attributes that determine the details or behavior of objects. A Gift object, for example, has a Constituent property that reflects the constituent that donated the gift. Note that in **The Raiser’s Edge**, objects respect and check all validity rules, required field information and business rules that have been set up in the system. The objects are designed to be a bulletproof way of updating data in **The Raiser’s Edge**. Data cannot be corrupted via the misuse of a data object.

For example, if someone customizing **The Raiser’s Edge** were to issue the following statement:
`Gift.Fields(Date) = “12/13/67abc”`
the result would be an “Invalid Date” error message because the system wouldn’t recognize the “abc.”

The above line of code illustrates how a Visual Basic programmer would modify the date on a gift record. Note that the programmer does not need to understand anything about the database layout of **The Raiser’s Edge**®. The objects are “smart,” and they know how to insert and retrieve data from the system. This layer of abstraction is important because it provides our customers

protection if any structural changes are made to the database at a later time. Developers “talk” to the database through the objects — not directly — and absolutely no knowledge of Structured Query Language (SQL) is required.

In addition to the data elements, data entry forms, parameter screens and search windows are also created as objects. For example, a developer can write his or her own custom program, but leverage our search screen and constituent form. This component-based design is pivotal to **The Raiser’s Edge**. Reports, mailing functions and queries are also objects within the COM architecture.

An important point to note is that we provide the exact same objects for customization that we have used to build **The Raiser’s Edge**. The objects we provide for customization are not “dumbed down”; they are the ones in use in the actual product. This provides an amazing amount of power for our clients that customize **The Raiser’s Edge**.

.NET Interoperability

Since Blackbaud’s API is 100% COM-based, it can be leveraged directly from .NET native applications. You can start writing .NET desktop applications, ASP.NET web applications, and XML Web Services that leverage Blackbaud’s APIs today. Blackbaud has continued to enhance our API and plug-in model to ensure that .NET developers are fully supported in their Blackbaud-based development efforts. For example, Blackbaud’s “plug-in” functionality has been enhanced to support hosting and embedding native .NET-based assemblies. More on the plug-in technology will be discussed later in this document.

At Blackbaud, we are working with .NET in some very exciting ways: improving the “reach” of our integration to third-party applications with .NET and forming a well-defined, standards based bridge via technologies such as Web Services.

Visual Basic for Applications

Visual Basic for Applications (VBA) is the premier development technology for rapidly customizing applications, such as **The Raiser’s Edge**, and integrating them with existing data and systems, such as Microsoft Office programs — Word, Excel, Access and other applications.

VBA is built from the same source code base as Visual Basic (VB). Because of this shared code base, VBA leverages much of the advanced technology found in Visual Basic, and its language features are virtually identical to those found in Visual Basic. This makes it very easy for Visual Basic developers to reuse code in VBA. It also offers VBA developers the same integrated development environment found in developer editions of Visual Basic, including the same code editing, debugging, object browsing, project management and form creation features.

Benefits of VBA in The Raiser’s Edge

VBA in **The Raiser’s Edge** provides several specific benefits.

Meet unique business needs: You can tailor **The Raiser’s Edge** to exactly match the way your nonprofit organization does business. By making **The Raiser’s Edge** work the way your

organization does, you are able to realize specific benefits while at the same time benefiting from the economy of not having to write your fundraising software program from the ground up.

Tightly integrate The Raiser's Edge with your existing systems and data: With VBA, **The Raiser's Edge** is instantly capable of connecting to your other databases and sharing information with existing systems, including Microsoft Office applications. This eliminates redundant data entry and maximizes your existing software investment.

Empower your systems staff: If you have systems staff who are familiar with Visual Basic, you can create rich custom solutions because VBA uses the same powerful technology found in Visual Basic.

Develop solutions faster: By building solutions from customizable applications, developers can build more powerful solutions and write less code. Less code means more productive developers and faster development cycles.

Lower training costs: Custom solutions utilize applications that are familiar to end-users. Consequently, end users can utilize their experience with the customizable applications to get up to speed more quickly.

Realize a greater return on your existing software investment: You can extend the functionality of **The Raiser's Edge** to meet specific business needs. Extending the functionality of **The Raiser's Edge** makes it infinitely more valuable because you don't have to invest in additional software to meet additional needs. You write your own solution.

VBA in **The Raiser's Edge** can serve a wide range of functions, and it is exposed in various places throughout the system including Records, Query and Export. Clients can use VBA to enforce custom business rules before records are saved, show custom screens and forms, or synchronize **The Raiser's Edge** database changes with an internal system.

VBA can also integrate **The Raiser's Edge** with other applications. For example, the programs in Microsoft Office employ the same version of VBA as **The Raiser's Edge**, which opens up a world of possibilities for interfacing with these programs:

- ◆ When a new constituent is added to **The Raiser's Edge**, you could automatically add that constituent to your Outlook Contacts list.
- ◆ When a gift of \$1,000 or more is entered, you could trigger a reminder that the donor should receive a personal phone call and optionally send an e-mail to your Director of Development.
- ◆ Use a VBA macro to automate a mail merge from start to finish. The entire procedure can be set in motion by a pre-defined action.

Open Application Programming Interface (API)

The application programming interface (API) is an extremely powerful and lower-level programming interface to **The Raiser's Edge**. We have opened access to **The Raiser's Edge** API so that developers can create completely customized solutions that use **Raiser's Edge** features and data.

By using the API to create custom solutions, developers save time because they simply “borrow” programming that already exists within **The Raiser’s Edge** — no need to reinvent the wheel. Even with this access, **The Raiser’s Edge** remains “bulletproof.” Sensitive data stays secure, and it’s impossible to jeopardize the stability of the system.

The API allows organizations to write fully functional “standalone” programs that have full access to **The Raiser’s Edge** data and services. The API is ideal for organizations that might want to write their own “front-end” to **The Raiser’s Edge** or to create a customized system that integrates **The Raiser’s Edge** with some other specialized application. Developers who want to access **The Raiser’s Edge** API can use any development platform that can access COM objects, such as C#, VB.NET, Visual Basic, ASP.NET, ASP, VBScript, Java, JavaScript and C++.

The Raiser’s Edge API is based on the industry-standard Component Object Model (COM). No proprietary languages or extensions are required to leverage its power. This COM-based design enables **The Raiser’s Edge** API to seamlessly integrate with the programmability components of many desktop application productivity suites, such as Microsoft Office.

The possibilities for using **The Raiser’s Edge** API to link to other programs are endless. Some of the most common applications include:

- ◆ Web/Internet programs
- ◆ Microsoft Office productivity tools
- ◆ Human resources
- ◆ Patient tracking
- ◆ College billing and records
- ◆ Ticketing
- ◆ Inventory
- ◆ Childcare
- ◆ Child welfare
- ◆ Planned giving
- ◆ Housing
- ◆ Case management

The API also enables the creation of “Plug-Ins” to **The Raiser’s Edge**. A Plug-In is a small software program that adds features to a larger piece of software. In **The Raiser’s Edge**, Plug-Ins are accessed through the Plug-In icon and are displayed directly in the Shell. This flexible technology facilitates adding custom applications and extensions directly within **The Raiser’s Edge**. The interface of a Plug-In can be implemented in .NET, pure HTML or as an ActiveX document. Plug-Ins share the database connection and runtime code resources with **The Raiser’s Edge**, making them an excellent alternative to a standalone API application. For example, a systems staff member can create a Plug-In that instantly takes all gifts made in the past month and displays them by fund in a colorful Microsoft Excel pie chart that appears within **The Raiser’s Edge** shell.

Business Intelligence and OLAP Integration

The Raiser’s Edge features tight integration to Blackbaud’s Business Intelligence solution **The Information Edge**. **The Information Edge** Extract, Transform and Load (ETL) engine can be

customized by developers via an XML definition file, .NET code, and direct database and Online Analytical Processing (OLAP)-level manipulation. **The Raiser's Edge** is designed to aggregate data from any solution, so this platform can be ideal for organizations looking to integrate and report on data from multiple systems.

RODBA

RODBA enables external applications to have read-only access to **The Raiser's Edge** data. Because **RODBA** provides access directly to the data tables and does not use an object model like **RE:VBA** or **RE:API**, it is best suited for reporting applications, such as Crystal Reports, where timeliness of information is crucial.

RE:VBA vs. RE:API vs. RODBA

Although all these options are similar in that they are used to customize and extend **The Raiser's Edge**, each is best suited for meeting specific needs. The following lists highlight the major differences among these three options:

RE:VBA

- ♦ Ideal for creating custom screens, business rules and routines within **The Raiser's Edge** interface
- ♦ Provides a built-in programming environment and tools
- ♦ Based on the ubiquitous Visual Basic programming language
- ♦ Must be "hosted" by a running version of **The Raiser's Edge** on the client machine to execute
- ♦ Cannot be used with **RE:Anywhere for Remote Access**

RE:API

- ♦ Ideal for linking **The Raiser's Edge** to external applications
- ♦ Developers can use their object-oriented programming language of choice, such as C#, VB.NET, Visual Basic, VBScript, Java, JavaScript or C++
- ♦ Does not require a running version of **The Raiser's Edge** system to execute, making API a better alternative than VBA when data is to be shared during non-operational hours

RODBA

- ♦ Ideal for creating ad hoc reports with Crystal Reports or other report-writing tools
- ♦ Allows direct read-only, SQL-based access to **The Raiser's Edge** data tables
- ♦ A fully documented schema is provided. In addition, RODBA can be used to integrate custom reports and interfaces directly into **The Raiser's Edge**

Multi-tier Architecture

The Raiser's Edge design is based on a multi-tier architecture. These tiers include:

Presentation Logic

The presentation logic tier, also known as a user interface, is what **Raiser's Edge** users see on their screens while they are working in the program. It is the top tier in the architecture and is where users can set unique preferences as to how certain aspects of **The Raiser's Edge** appear and behave. For example, User One might prefer cash gifts to appear in the color green on the

constituent Gift tab, while User Two might prefer cash gifts to appear in blue. With this type of processing performed at the presentation logic tier, User One and User Two can each see cash gifts in the color they choose.

Business Logic

The business logic tier is where most of the processing takes place in **The Raiser's Edge**. Business logic drives functions and procedures, such as directing data to the correct place in the database and enforcing security restrictions.

Database

The database is where constituent, gift, event, action and other types of **Raiser's Edge** data are stored.

Creating distinct tiers with independent functions simplifies product development and makes the program's code easier to maintain, revise and expand. The multi-tier architecture is similar to a stereo system — you can hook up a CD player, a receiver and speakers. Each component doesn't know how the others work but as long as they are hooked up correctly, the CD player will send a signal to the receiver and the receiver will amplify and send that signal to the speakers. If one of the components falters, it is easy to determine where the problem is, and fix only the component that is broken. You could also hook up a DVD player to your receiver as well at a later date. This would not cause any problems with your system that is already in place.

Database Options for The Raiser's Edge

The Raiser's Edge supports the Microsoft SQL Server 2000 and Oracle 9i database platforms. Performance, scalability and product features are virtually identical across the two databases. The decision on which database to deploy should be based solely on an organization's level of expertise with supporting a given platform.

Crystal Reports

Crystal Reports, developed by Crystal Decisions (formerly Seagate Software), is the most widely used report writer in the world. Although more than 100 customizable reports are available in **The Raiser's Edge**, Blackbaud provides Crystal Reports 8.5 with each copy of **The Raiser's Edge** to handle unique reporting needs.

Crystal Reports offers powerful analytical features within an easy-to-use interface. By using Crystal Reports, development professionals can create presentation-ready, custom reports and report templates using any information in **The Raiser's Edge**.

For creating new reports, wizards provide step-by-step assistance in connecting to data, selecting fields and records, sorting and formatting. Crystal Reports' flexibility means that virtually any report is possible, including sub-reports, conditional reports, summary reports, cross-tabs, form reports, multiple details and mailing labels. Manipulate data within Crystal Report by using more than 160 built-in functions and operators, such as mean, median, mode, range and standard deviation.

The report-design screen offers a “drag-and-drop” solution for positioning data elements anywhere on the page. Graphics tools are available for adding color, pictures, lines, borders, highlights and text to reports.

Blackbaud offers training options, customer support and report-design services for our clients that use Crystal Reports with **The Raiser’s Edge**.

Conclusion

The technologies discussed in this document, along with many others not included here, come together to make **The Raiser’s Edge** the ultimate fundraising software solution — powerful standard features, with customization options to meet every need.

If you have specific questions about the topics covered in this white paper or if you would like general information about **The Raiser’s Edge**, please contact Blackbaud at 1-800-443-9441 or solutions@blackbaud.com.

About Blackbaud

Blackbaud is the leading global provider of software and related services designed specifically for nonprofit organizations. More than 12,500 organizations use Blackbaud products and consulting services for fundraising, financial management, business intelligence and school administration. Blackbaud’s solutions include The Raiser’s Edge®, The Financial Edge™, The Education Edge™, The Information Edge™, WealthPoint™ and ProspectPoint™, as well as a wide range of consulting and educational services. Founded in 1981, Blackbaud is headquartered in Charleston, South Carolina, and has operations in Toronto, Ontario, Glasgow, Scotland, and Sydney, Australia.

For more information, visit www.blackbaud.com

2000 Daniel Island Drive
Charleston, SC 29492
800.443.9441
www.blackbaud.com
solutions@blackbaud.com



© April 2005, Blackbaud Inc

This publication, or any part thereof, may not be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, storage in an information retrieval system, or otherwise, without the prior written permission of Blackbaud, Inc.

Blackbaud, Inc., reserves the right to make improvements in this document and the products it describes at any time, without notice or obligation.

- ♦ Microsoft is a registered trademark, and Windows, Visual Studio, Visual Basic, Visual Basic for Applications, SQL Server, C++, ActiveX, Microsoft Office, Microsoft Word and Microsoft Excel are trademarks of Microsoft Corporation.
- ♦ Java is a trademark of Sun Microsystems, Inc.
- ♦ Crystal Reports is a registered trademark of Seagate Software, Inc.
- ♦ All other products and company names mentioned herein are trademarks of their respective holders.