Appeon in a .NET World

An Appeon Whitepaper

Appeon® for PowerBuilder®

September 2004

LAST REVISED: September 16, 2004

The information contained in this document represents the current view of Appeon Corporation on the issues discussed as of the date of publication.

This whitepaper is for informational purposes only. APPEON MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

Appeon may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Appeon, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Table of Contents

1 Introduction

1.1 Overview

This whitepaper provides a high-level overview of the .NET integration capabilities that Appeon Server's flexible and open middle-tier provides.

Appeon's middle-tier is based on the J2EE standard, and supports other industry favored open standards, such as Web Services/SOAP, XML, CORBA/IIOP, and JDBC. In addition, Appeon Server provides extensive support for server object technologies, including C/C++ COM/DCOM, C/C++ DLLs, Java EJBs and Servlets, and PowerBuilder NVOs.

Appeon's flexible and open middle-tier enables you to leverage your existing investments in PowerBuilder and standardize your enterprise on the .NET platform at the same time. You can freely mix PowerBuilder and Appeon with .NET technologies to build your application's presentation and business logic, choosing the right language and tool for the job.

1.2 .NET and integration

Microsoft debuted its .NET technology in June 2000 with the message "The Internet for Everything." .NET encompasses a wide range of individual technologies, including an application framework, the Visual Studio .NET development environment, the .NET Application Server, and a suite of Web Services such as .NET Passport.

A general definition of .NET is a program development and execution platform. The programs run wherever .NET resides. Microsoft .NET is targeted at addressing a broad class of applications including standalone, client/server, mobile, smart client, and Web-deployed applications.

A central theme of .NET is the extensive use of Web Services for software integration. .NET focuses on the use of Web Services for nearly all inter-application communication, including client-to-client, client-to-server, and server-to-server.

1.3 Appeon compatibility with .NET

Appeon Server and PowerBuilder provide extensive support for Web Services technology, including a Web Services Toolkit that allows developers to consume and expose services quickly and easily without extensive knowledge of SOAP or WSDL. With these powerful capabilities an Appeon-deployed Web application can integrate seamlessly into the Web Services-focused .NET environment.

Of course, there are tasks for which Web Services are not the most appropriate integration technology, which is why Appeon also provide support for a range of other integration options. All editions of Appeon Server supports all of the technologies commonly used to integrate .NET with J2EE, and even provides native Microsoft component support that no other leading J2EE application server provides.

	Developer Edition	Advanced Edition	Enterprise Edition
.NET Integration Approach			
Web Services/SOAP	✓	✓	✓
COM+	✓	✓	✓
CORBA/IIOP	✓	✓	✓
URL Parameters	✓	✓	✓

2 Unlimited .NET Integration Possibilities

Appeon allows you to pick the most appropriate integration technology for your needs. This following section discusses in more detail how to integrate your Appeon Web application in a .NET environment using the various integration approaches that Appeon Server supports:

- Web Services/SOAP
- COM+
- CORBA/IIOP
- URL Parameters

2.1 Web Services integration

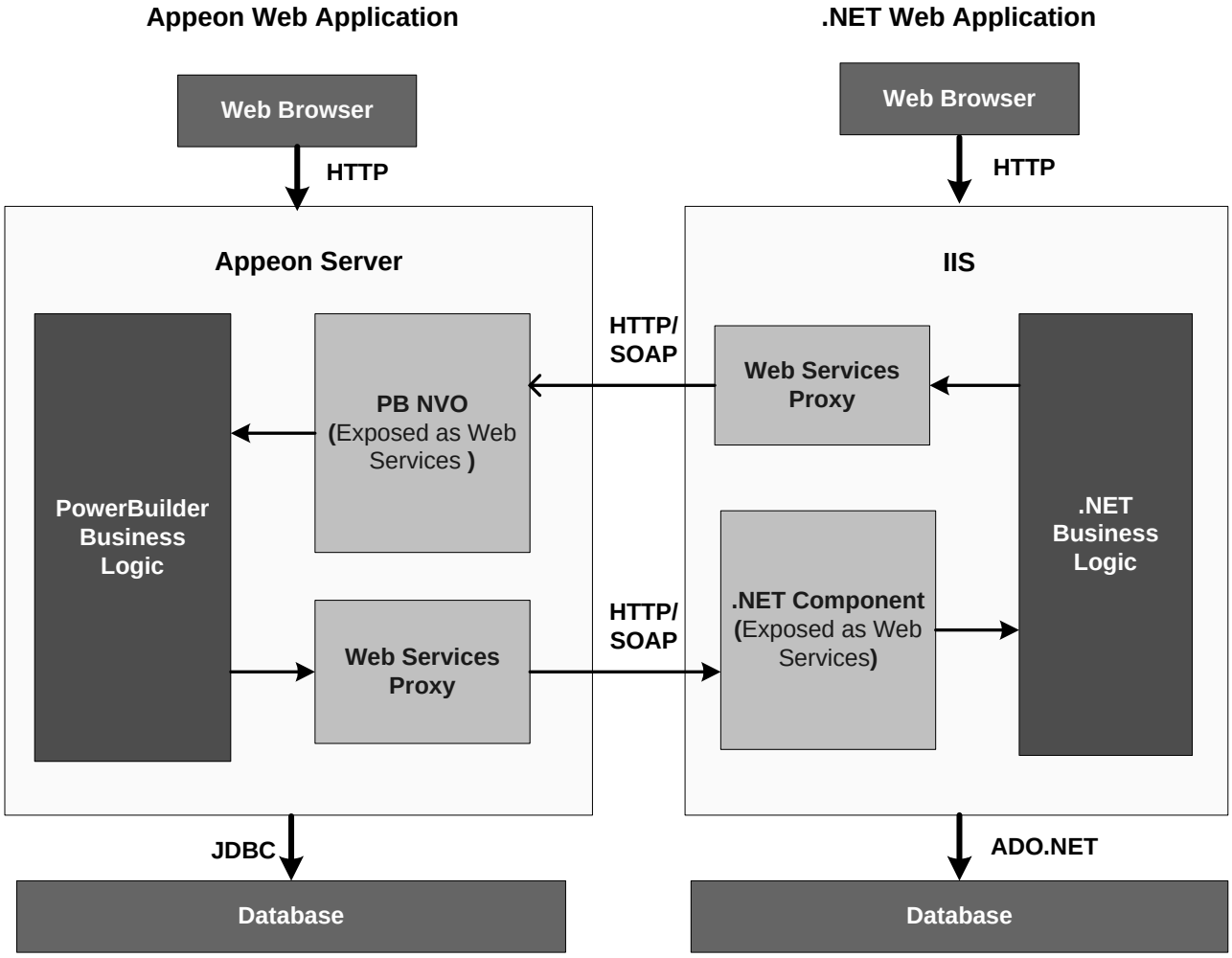
Web services are a standardized way of integrating applications using the XML, SOAP, WSDL and UDDI open standards over the HTTP protocol. Web services are part of the foundation of Microsoft's .NET strategy.

With Web services, different applications from different sources can communicate with each other without time-consuming custom coding, and are not tied to any one operating system or programming language. For this reason, they are ideal for allowing organizations to communicate data without intimate knowledge of each other's IT systems.

With Appeon you can easily consume or expose Web Services:

- PowerBuilder enables developers to easily build applications that consume Web services with unbeatable PowerBuilder 4GL productivity.
- Appeon Server offers a Web services toolkit that makes it easy to take an existing piece of business logic and expose it to the world as a Web service.

With .NET's ubiquitous support for Web services, you can integrate the Appeon Server middle tier with nearly any .NET program, whether a browser-based Web application, smart client, or standalone program. The following diagram shows an example of how Web Services can be used to integrate the business logic of an Appeon Web application and a .NET Web application:



The components used for integration are explained below:

Calling Appeon Server PowerBuilder components from .NET:

1. A PowerBuilder NVO deployed to Appeon Server exposes a Web Service, using the Appeon Server Web Services Toolkit. This NVO may itself contain business logic and/or can interact with other server-side or client-side objects.
2. .NET components access the Web Services exposed by Appeon Server through a Web Services proxy object generated using the standard .NET tools.

Calling .NET components from Appeon Server PowerBuilder components:

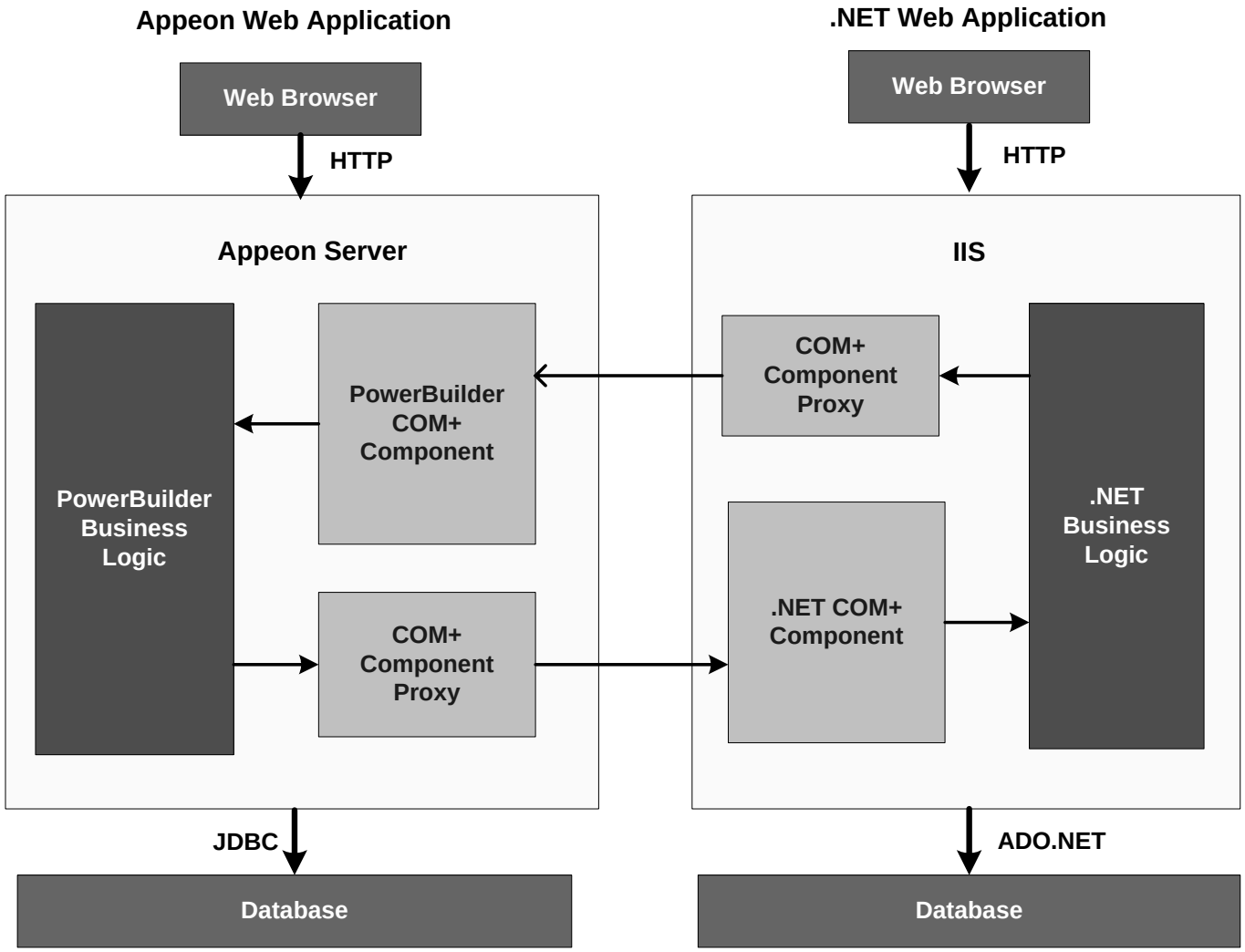
3. Server-side PowerBuilder components access the Web Service exposed by .NET through a Web Services proxy, which can be a COM+ component generated by the .NET tools or a PowerBuilder component generated by the Appeon Server Web Services Toolkit.
4. A .NET component exposes Web Services, using the standard .NET Web Services framework. This component may itself contain business logic and/or can interact with other .NET components

2.2 Microsoft COM+ integration

COM+ is a Microsoft component technology that predates Web services. Microsoft continues to believe in this technology, and COM+ is strongly supported by Microsoft's .NET Enterprise Server.

Like Web Services, COM+ components are very versatile because they are completely language-independent and easily fit into an object-oriented program design. Their main drawback is that they are limited to the Microsoft Windows platform. COM+ is still a good choice where integration is between tightly coupled systems, or where an existing system uses COM technology and the ROI for moving to Web Services is not high.

PowerBuilder provides native support for building and calling COM+ components. The following diagram shows an example of how COM+ can be used to integrate an Appeon Web application and a .NET Web application:



The components used for integration are explained below:

Calling Appeon Server PowerBuilder components from .NET:

1. A COM+ component developed in PowerBuilder exposes an interface that can be accessed in any language supporting COM+ integration. The component can itself contain business logic and/or can interface with business logic in other client-side or server-side objects.

2. .NET components access the PowerBuilder COM+ component through a proxy generated by the Windows Component Services Management tool.

Calling .NET components from Appeon Server PowerBuilder components:

3. Server-side PowerBuilder components access the .NET COM+ component through a proxy generated by the Windows Component Services Management tool.
4. A COM+ component is created using standard .NET tools. This component may itself contain business logic and/or can interact with other .NET components

2.3 CORBA integration

CORBA is an open standard for defining components that can be integrated easily between processes on a machine or between machines over a network. It is similar to Microsoft's COM+, but is based on a vendor-independent open standard that is supported by a number of J2EE application servers, open-source frameworks, and commercial products. CORBA objects can communicate with one another regardless of what programming language they were written in or what operating system they are running on.

Appeon Server provides built-in support for this powerful standard, and no purchase of third-party software is required.

It is possible to add this powerful integration technology to your Microsoft .NET environment, in order to take advantage of this open standard that has been used heavily to in the J2EE world to integrate with non-J2EE applications. There are a number of third-party products that can add CORBA support to the .NET Enterprise Server, such as MiddTec's MiddCor product.

2.4 URL parameter integration

A common way to pass commands or short messages to Web applications is to add them as parameters in the URL used to start the Web application or bring up another Web page of the application. Since all Web applications are accessed via a URL, this is an excellent way to achieve cross-platform integration without any need to understand the internals of the application. For example, an ASP.NET Web page can pass commands to a separately developed JSP Web page, and vice versa.

This approach is very straightforward to implement, and most developers are comfortable with this technique.

Appeon fully supports this method of integration. Appeon Web applications can receive parameters in the URL that will then be processed by the PowerScript logic. They can also invoke ASP.NET Web pages with parameters using the PowerBuilder HyperlinkToURL function or the Hyperlink controls.

3 Conclusion

Enterprises can continue leveraging the substantial investment they have made in PowerBuilder and standardize on .NET. Appeon and PowerBuilder have provided powerful support for Web Services – the cornerstone of .NET – as well as a range of other integration capabilities that can fit nearly every task. With extensive support for integration technologies built into Appeon Server and PowerBuilder, it is easy to make an Appeon-deployed Web application integrate seamlessly into a .NET environment. In the future, with PowerBuilder 11.0, PowerBuilder will become a .NET language, compiling to the Microsoft Intermediate Language (MSIL). As PowerBuilder continues to evolve to support .NET, Web developers using Appeon will have an ever-increasing number of options for tightly integrating with .NET at every architectural tier.

Appeon Corporation
1/F, Shell Industrial Building
12 Lee Chung Street
Chai Wan District, Hong Kong
www.appeon.net