

Opening Your Doors to the Internet

Customer Accessibility for E-Business Applications

Contents

- Introduction 3**
 - The Progress Universal Application Architecture Defines E-Business Accessibility 4
- Accessing Applications During One E-Business Day 5**
 - Internal and External Accessibility 5
- Accessibility and Integration 10**
 - Applications Accessing Other Applications within an Enterprise 11
 - Applications Accessing Other Enterprises 11
- Appendix A: Progress Prescriptions..... 12**

Introduction

Historically business applications were designed and developed to solve very specific business issues with little regard to connectivity to other applications. Applications were built or purchased by the company and were deployed within the four walls of the enterprise—their users were the employees of the firms. Little attention was paid to allowing other users access to these solutions.

Thanks to diminishing communications costs, companies can take advantage of the Internet to position themselves for extended growth. By integrating applications within the enterprise and opening their previously isolated internal business processes to customers and suppliers, they can exchange information with other businesses' processes (B2B) to execute transactions much more effectively. Additionally, individuals are no longer cut off from information and tools that would be useful to them in the field or that require them to have a dedicated intermediary supporting them back at the office. Application accessibility becomes the cornerstone of the new e-business paradigm.

The key to successful e-business applications is to provide users with access to whatever they need to get their jobs done efficiently. The challenge to be met by today's e-business applications is to provide all its possible users with access to business functionality to complete complex business transactions.

Internet e-business applications have to be accessible to a wide variety of users and applications—anywhere, anytime. They must present the appropriate interfaces to meet the needs of the 'clients' that wish to connect to the applications. There are various types of user scenarios that your e-business solution must consider:

- **Casual users** require loose connectivity and occasional access.
- **Intensive users** require tight connectivity and frequent access.
- **Internal users** are generally employees accessing applications from within the firewall.
- **External users** are typically customers and partners who gain access from outside the firewall.
- **User-less** scenarios also exist where applications access other applications, driven by an application's workflow or business logic.

Given the variety of user scenarios and the heterogeneous nature of most businesses today, e-business applications should be constructed in a way that allows for a range of options to coexist. These may include cutting-edge, as well as established, more traditional, technologies. For example:

- Rich graphical user interfaces are the best choice to meet the requirements of intensive users that have to perform sophisticated business transactions.
- HTML Web browser interfaces are the best choice for casual users interfacing with a PC.
- WML interfaces are for casual, mobile users accessing data with a personal data assistant (PDA) or mobile phone.
- Character interfaces are often the best solution for intensive data entry application tasks performed by internal users.
- Application-to-application integration scenarios provide real-time intra-enterprise access based on dedicated APIs exchanging information using XML or other technologies.

As a result, applications must provide the architecture, functionality, and interfaces to enable all types of access required by an e-business model.

The Progress Universal Application Architecture Defines E-Business Accessibility

A fundamental goal of the Progress product strategy has been to deliver on its vision of the Universal Application Architecture (UAA) to help customers build Future Proof™ business applications. Today, Progress Version 9 provides an e-business platform that meets all accessibility requirements. The UAA architecture enables business applications to present ANY type of client, exchange data with ANY type of data source, and interact with ANY other application or business.

UAA applications are network- and user-interface-independent, allowing developers to build upon the n-tier model using 4GL Character, GUI (graphical user interface), HTML, Java, or ActiveX client interfaces dependent on customer needs and transparently and seamlessly deployed to the Internet, intranet, extranet, LAN, or host-based architectures and in any model, including the ASP (Application Service Providers) model.

In Figure 1 below, you can see how distributed Progress applications provide accessibility to an application's business logic and data through a wide variety of technologies. The following scenarios will illustrate how the UAA helps to meet the accessibility challenges of the e-enabled enterprise. For more detail on specific products, turn to "Appendix A: Progress Prescriptions."

Universal Application Architecture

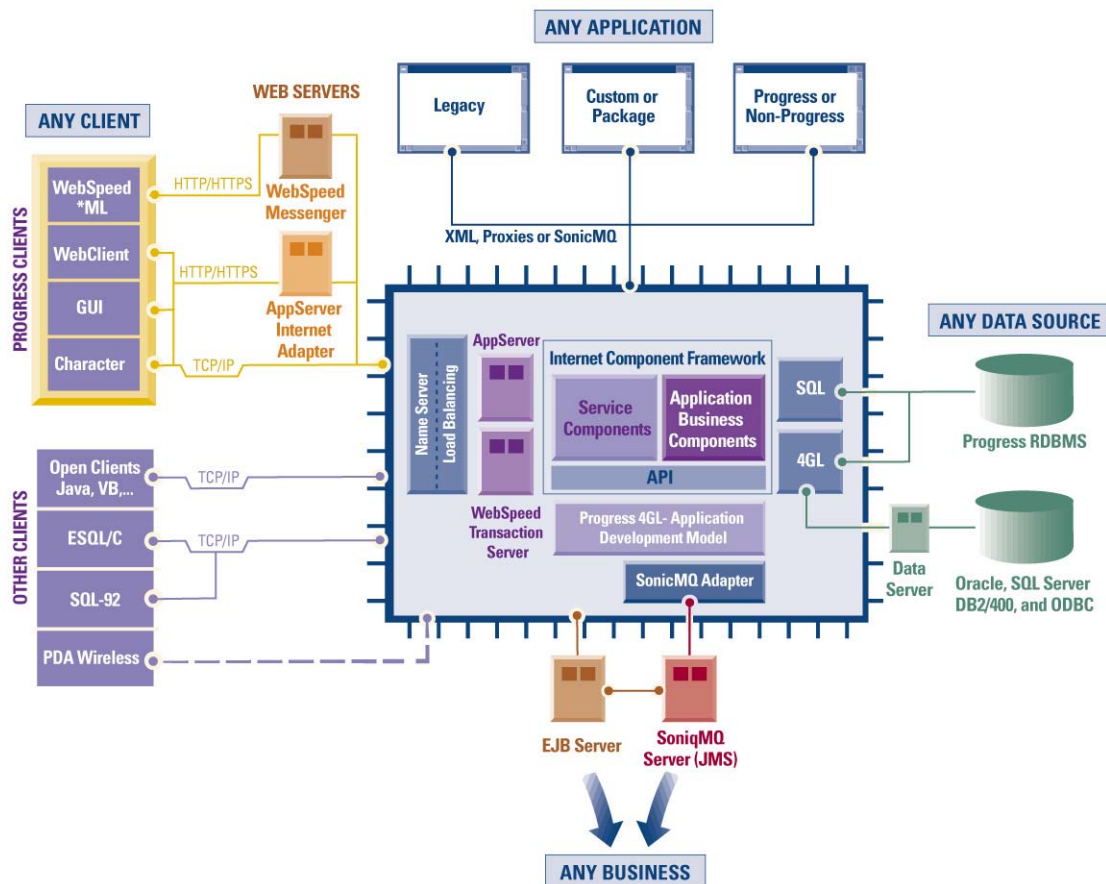


Figure-1: Progress Universal Application Architecture

This flexibility coupled with the integrated, highly productive Progress 4GL Application Development Environment shields solution providers from steep learning curves and excessive deployment costs. The resulting low cost of ownership is a critical advantage unparalleled in the industry.

Accessing Applications During One E-Business Day

To provide concrete examples of the different types of accessibility that e-business demands, this paper presents scenarios that show how enterprises implement universal accessibility. Although the companies described are fictitious, their e-business applications are based on real Progress applications that actual customers have built in the UAA model. The scenarios illustrate a common theme: *Accessing business applications during one e-business day*.

The “Internal and External Accessibility” section discusses the types of access that the applications have to provide to employees (internal) and to customers and suppliers (external). The “Accessibility and Integration” section focuses on the types of access that meet Business to Business (B2B) requirements for enterprise application integration (EAI).

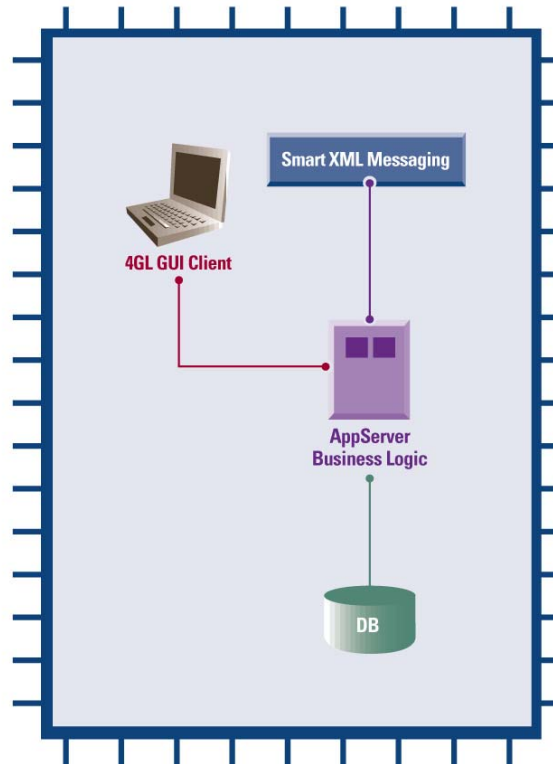
Internal and External Accessibility

Mary Clark works in purchasing at Groundskeeper, Inc., a landscaping company. She needs access to a variety of applications including the company’s inventory control, order entry, and Customer Relationship Management (CRM) systems. The intensity of access to these applications can be casual or intensive, depending on Mary’s requirements for each.

As a purchasing agent, Mary makes sure that the company’s stock levels stay at or above their minimum. This morning, Mary received a message from her inventory control system notifying her that the company’s supply of black safety glasses has dropped below a total quantity of 10. The inventory control system has a tightly-coupled, feature-rich client user interface so that Mary can pose complex queries to track and report on the inventory in real-time. Mary checks her stock levels on all glasses. She runs a report to determine how many the company has gone through since her last purchase and to forecast Groundskeeper’s needs for the next six months. The report illustrates that they have gone through a total of 62 pairs of glasses: 5 red, 14 blue, and 43 black.

Figure 2 shows the components that make up Groundskeeper’s Inc.’s application. The application is characterized by being internally accessible to intensive users. It features a 4GL GUI client connected to an AppServer running in a distributed architecture.

Groundskeeper Inc.'s Application



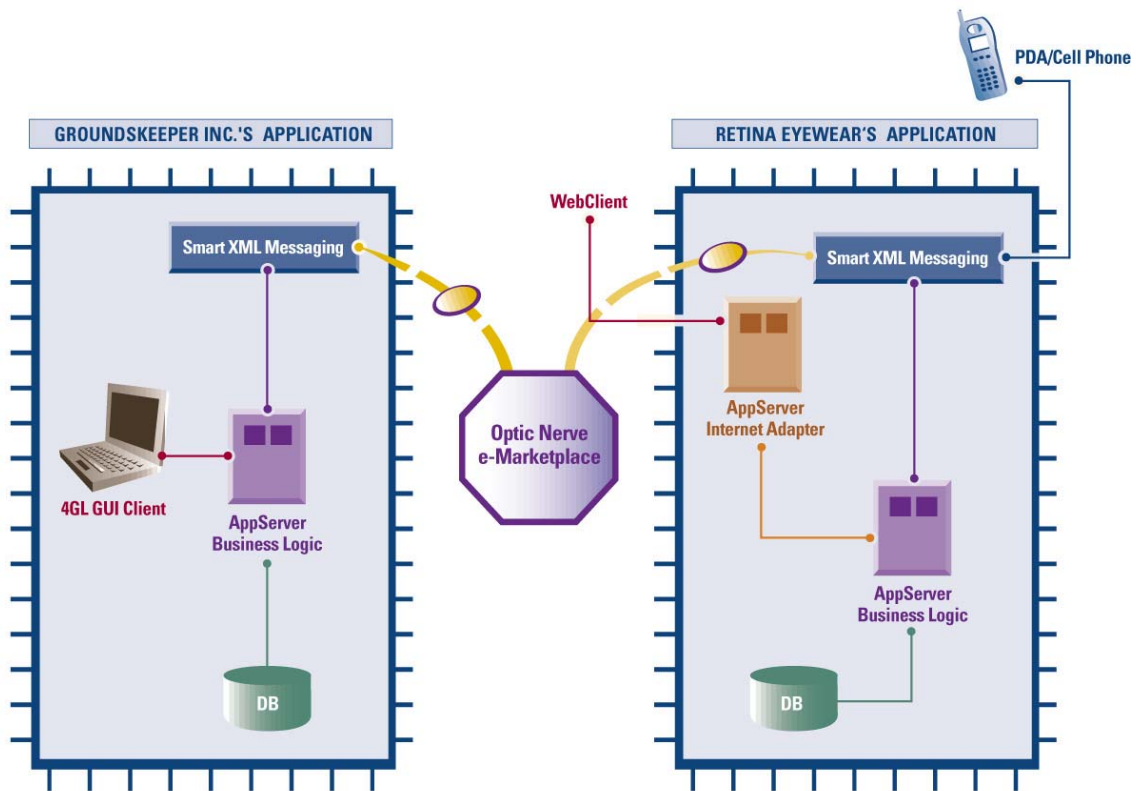
© 2001, Progress Software Corporation

Figure-2 Groundskeeper Inc.'s Application

Mary switches to her order entry screen, which is linked to many of the suppliers that Groundskeeper deals with on a regular basis. She logs in to OpticNerve, an e-marketplace for eyewear distributors and manufacturers, and submits a bid request to locate the vendor with the lowest pricing at the quantity she needs.

OpticNerve is a Progress WebSpeed application. The HTML interface makes the application both accessible and suitable for use by a wide variety of users, at different locations. All that is required is a Web browser, which is available on a variety of platforms, and support for many user interface devices (like PCs, PDAs, and cell phones). Although some users of the e-marketplace will be casual like Mary, other users are intensive intermediaries. The e-marketplace is accessible around the world via the Internet. By using the Unicode character standard internally, the application easily supports access by users who speak different languages.

Figure 3 illustrates the OpticNerve e-Marketplace and how other applications integrate into it. The OpticNerve application is characterized by its accessibility by external users. These users access the application from different clients, depending on their own work requirements. The intensive user accesses it using the Progress WebClient running on their PC or laptop. The casual user accesses it using an HTML interface presented on a PDA. The e-Marketplace features Smart XML Messaging and the SonicMQ E-Business Messaging Server.



© 2001, Progress Software Corporation

Figure-3 OpticNerve's e-Marketplace

Retina Eyewear is one of the participating vendors in the OpticNerve e-marketplace. Retina, a facility specializing in the injection molding, metalizing, and coating of safety glasses, has a legacy application that controls the manufacturing process without human intervention.

Carmen Montoya, Account Executive at Retina, is based in a branch office, but she spends more than 40% of her time on the road between customer visits and trade shows. People at branch offices, telecommuters, and people who are traveling still need access to the corporation's network to get their job done.

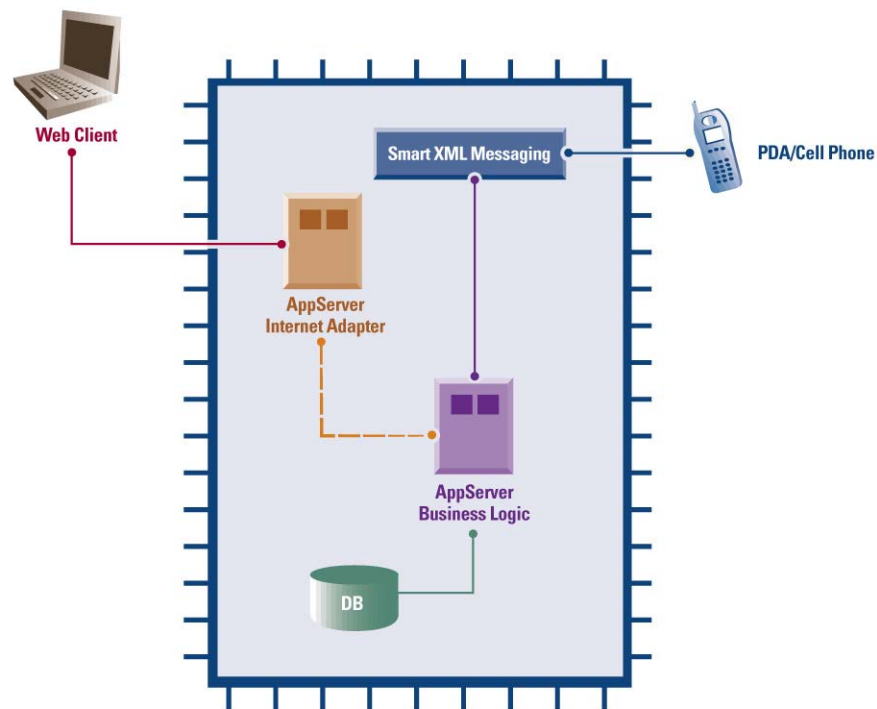
Carmen has wireless access (PDA, Cellular) to certain applications to help her in her day-to-day activities. Carmen receives a message on her Palm™ VII PDA shortly after Mary places her request. She uses her Palm VII to check her supply levels and flags the bid request for follow up once she returns to the office.

When she returns to the office, Carmen accesses Retina's CRM application using the Progress WebClient. Retina chose to use the WebClient at this and other remote offices for its ease of deployment and maintenance over the Internet. The Progress WebClient enables application developers to deploy a feature-rich client UI in the Progress 4GL and leverage the Internet for server functionality. The WebClient contains Progress 4GL, GUI code that is common to all 4GL interfaces. Once installed on the client PC, Carmen can quickly run a Web- or ASP-enabled application that requires less bandwidth and a much smaller footprint than the traditional GUI client, yet it retains all of the functionality.

When Carmen logs into the CRM system, she is prompted to update her client to take advantage of new functionality that has been added since she last accessed the application. She accepts the update and very quickly views a listing of the recent bid requests that have been created in her region, including Mary's. The CRM application is integrated with the e-marketplace so that Carmen can search and bid on new sales opportunities more efficiently. Using the pricing module, she prepares a quote, then selects a menu item to generate a bid to Mary at GroundsKeeper. By incorporating standards—such as an XML template based on Carmen's industry as well as Java Message Service (JMS) through the SonicMQ message server—Carmen's bid is seamlessly presented to Mary through the e-marketplace. The CRM application is 4GL-based and uses the Progress SonicMQ Adapter to communicate with the e-marketplace application.

Figure 4 illustrates Retina Eyewear's application which is designed for internal access by employees. If the employee is an intensive user, she can access it through the WebClient, while employees who need only casual use of the application can access it from their PDAs and leverage the built-in messaging services.

Retina Eyewear's Application



© 2001, Progress Software Corporation

Figure-4 Retina Eyewear's Application

Of all the proposals Mary received, Carmen's was the lowest and most attractive. Groundskeeper does not currently have a relationship with Retina Eyewear; however—since both companies are affiliated with OpticNerve—Mary is able to place the order.

Meanwhile, Carmen receives a notification via her mobile phone to indicate that she has won the bid. The message is formatted as Wireless Markup Language (WML) so that the phone's browser can easily display it.

When Carmen logs into her CRM application to process the order, she is flagged with the option to offer free ultraviolet protection to Groundskeeper as a first-time purchaser. Carmen calls Mary to offer the UV upgrade personally. During their conversation, Mary accepts the offer, doubles the amount of her purchase, and attaches a new purchase order to cover the additional charges.

Table 2 summarizes the points of internal and external access demonstrated by this scenario.

Table 2: Summary of internal and external accessibility

| Activity | User | Client | Progress Products | Application | Security | Standards |
|--|-------------------------------|-------------------------|--|--|--|---------------------------|
| Mary receives a message that the inventory level of safety glasses has fallen below the minimum. | Casual Internal | HTML on PDA | Open Client, AppServer, SonicMQ | Internal inventory control system | N/A | JMS |
| Mary accesses the inventory control system to determine stock levels. | Casual Internal | Progress 4GL GUI | Progress Client Networking, AppServer, | Internal inventory control system | User Authentication | N/A |
| Mary pulls the data to a local database where she runs reports on the data. | Intensive Internal | Progress 4GL GUI | Progress Client Networking, AppServer, CorVu CorBusiness | Internal inventory control system, with Business Intelligence (AD HOC reporting functionality) | User Authentication | JDBC, ODBC |
| Mary switches to her order entry screen and places an order. | Intensive Internal | Progress 4GL GUI Client | Progress Client Networking, AppServer | Internal inventory control system | User Authentication | N/A |
| The Groundskeeper order entry system submits a bid request to the e-marketplace. | A2A | N/A | AppServer, SonicMQ | E-marketplace hosted by a third party | User Authentication, HTTPS tunneling, SSL encryption | JMS, XML |
| Carmen receives a message notifying her of a new bid request. | Casual External and A2A | HTML on PDA | Open Client, AppServer, SonicMQ | E-marketplace hosted by a third party | N/A | JMS, WML |
| Carmen accesses the e-marketplace to look at the bid request. | Casual External | HTML on PDA | WebSpeed | E-commerce portal hosted by a third party | User Authentication | Unicode, HTML, XML, DHTML |
| Carmen checks inventory status with her PDA. | Casual Internal | HTML on PDA | WebSpeed | Inventory tracking application | User Authentication | Unicode, HTML, XML, DHTML |
| Carmen prepares and submits a quote. | Intensive External and A2A | WebClient | Progress WebClient, AppServer | Order tracking application integrated with e-marketplace hosted by a third party | User Authentication, HTTPS tunneling, SSL encryption | XML, HTTP, SSL |

| Activity | User | Client | Progress Products | Application | Security | Standards |
|--|--|----------------------------|--|---|--|------------------------|
| Mary accepts the bid and submits an order to Retina Eyewear for 100 pairs of safety glasses. | Casual External, Internal and A2A | Progress 4GL GUI Client | Progress Client Networking, AppServer, SonicMQ | CRM application coupled with e- marketplace hosted by a third party | User Authentication, HTTPS tunneling, SSL encryption | XML, JMS, HTTP, SSL |
| Carmen receives a message stating the bid was accepted. | Casual External | HTML on PDA | SonicMQ | E-commerce portal hosted by a third party | N/A | WML, JMS, XML |
| Carmen accesses the CRM system to process the order and is notified about new promotion. | Casual Internal | WebClient | Progress WebClient, AppServer | CRM application | User Authentication, HTTPS tunneling, SSL encryption | HTTP, SSL |
| Mary modifies the order and attaches a new purchase order number. | Casual External, Internal and A2A | Progress 4GL GUI Client | Progress Client Networking, AppServer, 4GL Business logic | CRM application coupled with e- marketplace hosted by a third party | User Authentication, HTTPS tunneling, SSL encryption | XML, HTTP, SSL |
| Carmen releases the order to production where it is assembled and readied for shipment. | Intensive Internal and A2A | WebClient | Progress WebClient, AppServer | CRM application, legacy manufacturing application | User Authentication | N/A |

Accessibility and Integration

When one talks about an application accessing another application or one application acting as a client to another (A2A), we are actually talking about application integration. In e-business scenarios, the distinction between accessibility and integration can be an academic one. If your goal is to open your back-office applications or data resources to customers or partners so that they can complete complex business transactions, whether you achieve this by allowing external clients to access your applications or external applications to access your back office is a question of design.

There are two basic integration schemes, one where the applications are all within an enterprise and the other where an application is accessing an application owned by another enterprise. In the first case, applications can be tightly integrated since they operate as part of one business environment. In the second case, applications are more loosely integrated and, in some cases, have to interoperate with a wide range of applications. Standards-based integration technologies, such as XML and messaging make this loose integration feasible.

Applications Accessing Other Applications within an Enterprise

A typical reason for having to integrate applications within an enterprise is the recent purchase or acquisition of software that might not be directly compatible with other enterprise software. Take as an example a company whose order entry system is written in Progress but has recently acquired warehouse software written in another programming language and using an Oracle database. These applications can be linked such that an order placed by someone in the order entry system will trigger a search for and relocation of the products in the warehouse required to fill the order. The company's IT group can share the design of the applications and the structure of the databases so they can tightly couple their applications through the Progress 4GL and Host Language Calls (HLC). This will provide the least latency and the highest performance for their customers.

Some enterprises are discovering the advantages of loosely coupling the integration of applications within their enterprise. The loose coupling makes it possible to modify or upgrade an isolated component of the integrated solution without also changing the integration interfaces. An organization has to weigh which solution works best for them in terms of the high performance of a tightly-coupled integration versus the flexibility provided by a loosely-coupled integration.

Applications Accessing Other Enterprises

Participating in today's e-marketplaces is often the motivation for making an enterprise's applications accessible by another enterprise. A distributor's success relies on efficient responses to requests for quotes and inventory status. One way to achieve this is to integrate the distributor's application with their supplier's order entry and warehouse systems. Even in cases where a tight integration is possible (because of compatible software), many enterprises opt for a loose coupling due to the uncertainty of the reliability of the Internet provider and the added complexity of direct coupling. Loose coupling has the added benefit that applications can be modified without requiring constant synchronization. Using the Java-based E-Business Messaging Server provided by SonicMQ, the Progress SonicMQ Adapter, and the Progress SmartBusinessObjects makes it easy to send XML messages to and from the applications.

To learn more specific information about the roles played by various Progress products in providing accessibility for your e-business applications, see the following section, "Appendix A: Progress Prescriptions."

Appendix A: Progress Prescriptions

This section reviews the Progress products and provides guidelines for choosing one over the other based on scenarios described in this paper.

Progress Version 9

Applications architected for e-business with UAA provide the flexibility to combine available and future application interfaces by accessing, reusing, and sharing the most valuable component of the solutions, their business logic.

In this manner, solution providers have the opportunity to take advantage of the interface that best meets application and market requirements (i.e. character UI for data-entry modules, Windows WebClient GUI and GUI client/server mode for highly-productive-OLTP modules, Web *ML UI for Internet B2C casual users, etc).

With the Internet, e-business, and distributed environments where the interfaces are very diverse and less predictable, it is necessary to have flexible development models in which the choice of the interface can be postponed. These needs are better addressed with a deeper component-oriented model, not only for user interface components but for business logic components as well. Database dependencies are removed from the interfaces, and user interfaces are created based on input/output requirements (Web, graphical, character, PDA, and wireless) no matter if they will be deployed on LAN, WAN, Internet, intranet, or using the ASP model.

Progress Version 9 provides the platform to serve as the model for user-interface-independent and distributable business object components that can be accessed from any interface and used by any client, data source, application, or business.

Progress WebClient

Progress WebClient provides the ability not only to run a GUI client over the Internet, but also to host the installation on a Web server. The application can then be remotely deployed, installed, and even updated on a client machine automatically, and transparently, when the user runs the application, thus reducing the deployment and maintenance costs. WebClient can be used also for LAN or intranet models, thereby significantly reducing the application deployment and maintenance costs while retaining the functionality of 'conventional' client/server applications required by intensive users, including local printing and integration with Microsoft Office suite and other Windows applications or services residing on the end user's PC. Progress WebClient leverages the business logic running on the Progress AppServer, which can be accessed as client/server (Client Networking), host-based (Character Client), OpenClient, WebSpeed, B2B/EAI APIs, and business messaging middleware modules.

Prescribed for all types of intensive users (Internal and External): LAN, intranet, and Internet.

Progress WebSpeed

Progress WebSpeed provides an optimized transaction-processing environment for high transaction volumes and rapid responses for HTML-based applications. Dynamic load balancing ensures high availability of transaction processing resources in a distributed, multi-tier environment. WebSpeed allows developers to leverage common business logic for Web-based applications. Using it in combination with the Progress AppServer guarantees reusability of the application logic.

Prescribed for all types of casual users (Internal and External): Internet, intranet, and extranet.

Progress Open Client

The Progress Open Client Toolkit enables developers to easily change the interface to an application to be a Java or an ActiveX application (Visual Basic or C++ for example). The Proxy Generator, a component of the Open Client Toolkit, allows a 4GL developer to identify AppServer functionality to be *exposed* to Open Clients and generate proxies for these 4GL procedures. The generated proxies enable Java, ActiveX, Visual Basic, C, and C++ interfaces to transparently access the Progress components on the Progress AppServer.

Progress AppServer

Progress AppServer is the foundation for Progress' distributed computing strategy, providing a framework for developing and deploying 4GL procedures as distributed application components. Progress AppServers provide flexible deployment options to take advantage of the infrastructure. In addition, by separating the business logic from the UI logic it is possible to connect and provide access from a variety of interfaces to the application logic. The Progress AppServer can be accessed using TCP/IP or HTTP, and for secure connections the Secure AppServer supports HTTPS.

Prescribed as the platform for distributed computing, regardless of the deployment model.

SonicMQ, Progress SonicMQ Adapter and Smart XML Messaging

SonicMQ, the leading solution for E-Business Messaging (EBM), a rapidly growing category of messaging middleware, is designed to meet the unique requirements of information integration and exchange over the Internet. Offering a highly efficient, robust, and cost effective messaging solution to address enterprise application integration and secure transport of business critical data over the Internet, it allows companies to integrate their internal applications and provide a mechanism for them to interoperate with their business partners. SonicMQ can scale up to support full-blown highly-distributed, highly-scalable Internet messaging requirements along with guaranteed message delivery.

Progress SonicMQ Adapter and Smart XML Messaging add secure A2A and B2B communication and facilitate the accessibility and integration of 4GL applications to and from other applications as well as e-marketplaces.

Smart XML Messaging allows seamless integration of the application business logic and SonicMQ Adapter. Its 4GL-to-JMS API and XML-to-4GL mapping shield the developer from the underlying technologies (JMS and XML) allowing them to focus on their core competency, that is, handling and managing the business processes associated with the messages and their contents expressed in XML format:

SmartB2BObject — This object transforms XML business messages to and from business logic and provides an open XML API to the enterprise application. The SmartB2BObject reads data from or updates data in the other objects dedicated to handling business logic (e.g., the SmartBusinessObject or SmartDataObject). The SmartB2BObject provides a mechanism by which disparate applications can exchange XML data, in spite of the wide variety of business message semantics.

SmartSender and SmartReceiver Objects — These objects enable the sending and receiving of messages used in SmartB2B Objects. The SmartSender packages content and calls the SmartProducer to send the message, and likewise, the SmartReceiver accepts inbound messages from the SmartConsumer object before passing them to a SmartBusinessObject, SmartDataObject, or SmartB2BObject for storage or further handling.

SmartProducer and SmartConsumer Objects — To be used in conjunction with the SmartB2B Objects (or SmartSender and Smart Receiver mentioned above), the SmartProducer and SmartConsumer objects handle the message transfer to and from the SonicMQ message server through the Progress SonicMQ Adapter.

SonicMQ is prescribed for message-based information integration between disparate applications, XML being the prescribed data format to use for the message-documents that have to be exchanged.

Progress DataServers

When access to non-Progress data sources, including Oracle, Microsoft SQL Server, DB2/400 and any ODBC- or JDBC-compliant data source, is required, the Progress DataServers provide direct, native, and high-performance access, the DataServers simplify the application development, deployment, and integration of legacy applications. They enable the developer to read, write, update, and delete data from non-Progress databases with consistent application behavior and hide much of the complexity normally associated with the development and maintenance of foreign database definitions.

Prescribed when full, direct, and real-time data access is required from 4GL applications to non-Progress data sources.

Progress ODBC and Progress JDBC Drivers with Progress SQL Database Engine

Progress ODBC and JDBC drivers provide direct data access to the Progress database via the Progress SQL database engine from any non-Progress language such as VB, Java, C++, and tools, including reporting, business intelligence, data mining, and Microsoft Office programs like MS-Excel and MS-Word.

Prescribed when non-Progress applications and tools require full, direct, and real-time data access to the Progress database.

XML

XML (eXtensible Markup Language) is a set of rules, guidelines, or conventions for designing text formats for data in a way that produces files that are easy to generate and read by a computer. It is unambiguous and avoids common pitfalls such as lack of extensibility, lack of support for internalization/localization, and platform-dependency. XML looks like HTML in format and it is a text file, but it is not meant to be read directly by users, just by other software applications. Where appropriate, there are programs that accept XML and format the contents for display.

XML's flexibility is superior compared with ad-hoc API-based protocols to exchange data between applications that require changes in the APIs and the applications whenever an extension is required.

Corporate Headquarters

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA Tel: 781 280 4000 Fax: 781 280 4095

Europe/Middle East/Africa Headquarters

Progress Software Europe B.V. Schorpioenstraat 67 3067 GG Rotterdam, The Netherlands Tel: 31 10 286 5700 Fax: 31 10 286 5777

Latin American Headquarters

Progress Software Corporation, 2255 Glades Road, One Boca Place, Suite 300 E, Boca Raton, FL 33431 USA Tel: 561 998 2244 Fax: 561 998 1573

Asia/Pacific Headquarters

Progress Software Pty. Ltd., 1911 Malvern Road, Malvern East, 3145, Australia Tel: 61 39 885 0544 Fax: 61 39 885 9473

Progress is a registered trademark of Progress Software Corporation. All other trademarks, marked and not marked, are the property of their respective owners.

PROGRESS
SOFTWARE

www.progress.com

Specifications subject to change without notice.
© 2001 Progress Software Corporation.
All rights reserved.