Ericom Software

White Paper

PowerTerm Host Publisher

A host system web-enabling solution

Ericom North America
Ericom Software Inc.
231 Herbert Avenue, Bldg. #4
Closter, New Jersey, 07624
Tel: +201 767 2210
Toll Free: 1 888 769 7876
Fax: +201 767 2205
E-mail: info@ericom.com

Ericom Europe
Ericom Software (UK) Ltd.
11a Victoria Square
Droitwich, Worcestershire
United Kingdom, WR9 8DE
Tel: +44 (0) 1905 777 970
Fax: +44 (0) 1905 777 972
E-mail: ukinfo@ericom.com

Ericom International
Ericom Software Ltd.
8 Hamarpeh Street
Har Hotzvim Technology Park
Jerusalem 91450 Israel
Tel: +972 2 571 4774
Fax: +972 2 571 4737
E-mail: info@ericom.com

PowerTerm Host Publisher

Introduction

Recent economic and technological advances have made it essential for corporations to integrate their stable, scalable back-end host systems with their new IT infrastructure. Having withstood the test of time, these back-end systems are often at the core of the enterprise's business process. As such, developing an effective e-Business strategy without integrating them with the newer systems on the corporate network may be impossible. Even corporations that choose an evolutionary rather than revolutionary development strategy may find that it is unacceptable to leave these systems isolated from their web infrastructure. Yet despite this need, corporations often find that they cannot make the necessary adaptations to the legacy back-end systems due to the prohibitive costs involved.

Ericom Software PowerTerm Host Publisher provides an innovative solution to this dilemma. It allows external applications, such as web and application servers, to directly access both data and business processes on the host systems, and it does this without requiring the host systems to be modified in any way. As a result, the host services become instantly available to any system inside and, if required, outside the organization, allowing the organization to quickly move to the next level of e-Business and information services.

Overview

To date, legacy host systems, both mainframe and midrange, still store approximately 70% of all corporate information worldwide. In addition, sales of new host systems remains strong, due to the fact that such systems continue to offer high ROI in terms of scalability, performance and reliability. Unfortunately, these same systems are also characterized by a closed architecture, requiring both proprietary hardware and software to access their services.

The advent of new standard-based data access protocols and technologies, such as HTTP/HTML, WAP and XML, makes it highly desirable to enable their use for interacting with legacy systems. Bridging the gap between the proprietary legacy protocols and the open new ones makes the huge data repository stored on the host systems much more accessible, while retaining all the benefits that such systems offer. For example, a legacy application that controls an automated warehouse and inventory levels can be connected to an online shopping Web site, providing the foundation for a company's e-Business strategy.

When designing and implementing a connectivity solution, it is important to understand that legacy systems provide more services than simple data storage. Usually these systems implement business-logic units that comprise the business processes of the organization. These business processes are often complex and difficult to analyze or reproduce, which means that they cannot be recreated on newer systems. A host system access solution must, therefore, provide facilities for activation of the business-logic units, transporting the input parameters and reporting the results.

Ericom Software provides a complete host system web enabling solution that facilitates secure access to the corporate data and the business processes encapsulated in a host system. Ericom Software's PowerTerm Host Publisher wraps the business-logic units, utilized in an organization's workflow, and exposes them as procedures that can be activated through a high-level Application Programming Interface (API). This API is built using industry standard technologies including Java, XML, Microsoft COM, SOAP and TCP/IP sockets, thus guaranteeing simple usability and maximum availability. This makes web-enabling legacy systems both straightforward and inexpensive.

An advantage of PowerTerm Host Publisher is that it utilizes a "black box" approach, whereby the underlying legacy application does not need to be modified to support the new connectivity features. While the client applications and user interfaces connecting through Host Publisher operate at the business-logic level, the legacy system itself continues to use its standard I/O facilities through Ericom Software's mature connectivity infrastructure. The combination of high-level services accessible via a simple and standard API that translate automatically to the low-level operations, makes the processes of opening the host system to new clients simple, safe and relatively inexpensive.

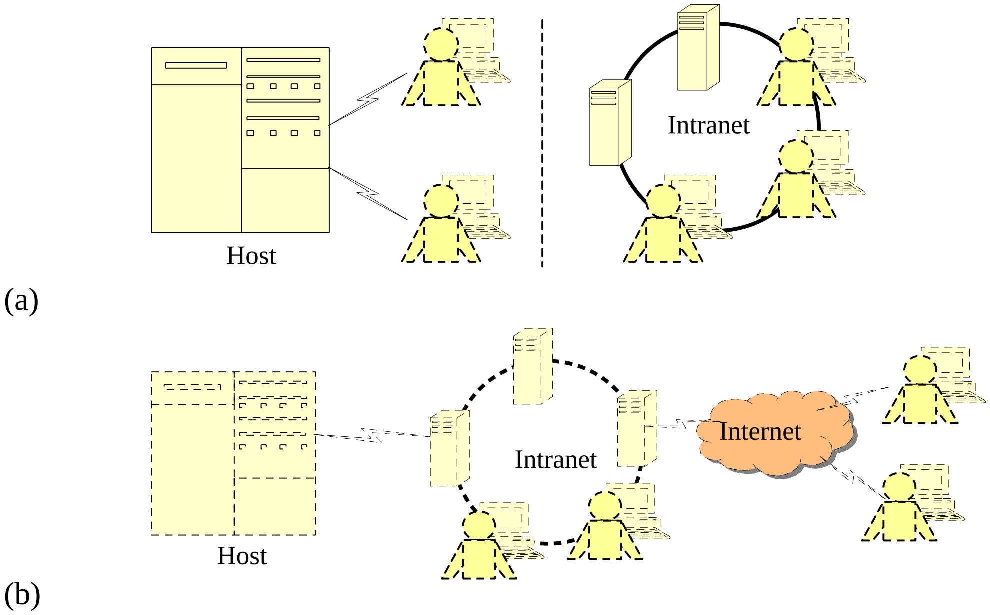


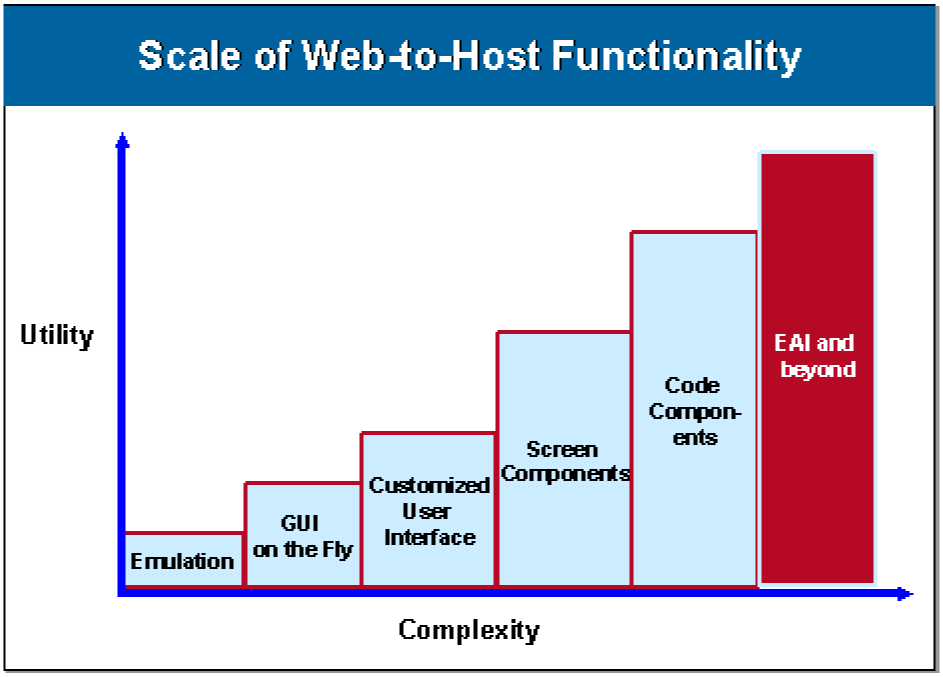
Figure 1: In the original configuration (a), the host system is separate from the rest of the Intranet, limiting exchange of information. Installing PowerTerm Host Publisher integrates the host system into the network and through it to a web server.

An additional factor to be taken into account when evaluating a legacy access solution is that of scalability. Such systems usually support a fixed number of simultaneous user sessions. Indeed, direct access to the host system's services was often limited to specially trained personnel. When a host system is opened up for general access via the Intranet, Extranet or Internet, the number users can grow dramatically, consequently overloading the system's resources. Host Publisher offers a solution in the form of the optional SessionManager component. By enabling simultaneous session reuse, Host Publisher SessionManager reduces system load while improving response times and overall performance.

An integration solution, built using PowerTerm Host Publisher, allows organizations to leverage their existing hardware and software infrastructure to meet the growing IT demands. The PowerTerm Host Publisher solution lets an ever-increasing number of people, both inside and outside the organization, access the information they need, when they need it, wherever they may be.

Host Connectivity Solutions

When considering any form of a legacy connectivity solution (i.e., web-to-host integration), it is important to understand the various technological alternatives. Each alternative offers distinct tradeoffs between ease of implementation and the functionality it provides:



Source: Giga Information Group

Figure 1

At the low end of the scale, terminal emulation software offers the simplest level of connectivity solutions. While this solution reduces the organization's TCO by relieving it of the need to service and support dedicated terminal stations, it does little to improve system accessibility or usability.

A "GUI on the Fly" type solution improves system usability by replacing the dated legacy display services with an attractive graphical user interface (GUI). If this user interface is also made customizable, productivity can be improved even more because the display can be updated to support new usage methodologies that were not available when the original system was designed. In addition, these solutions also improve accessibility by leveraging modern communication protocols such as HTTP. However, a "GUI on the Fly" solution preserves the original legacy user logic, which more often then not, is not an acceptable solution for general web applications.

At the high end of the scale, code components and Enterprise Application Integration (EAI) solutions offer the greatest amount of flexibility by providing direct programmatic access to the host system's services, both for data access and activation of business-logic units. This allows a user interface design that is not constrained by the original application sequence of "green screens". In addition, the host system can be utilized by other applications. As a result, a specific procedure within the host can participate in a business transaction that spans several systems, even across several organizations. For example, a new corporate CRM package can be integrated with a billing application running on a legacy computer.

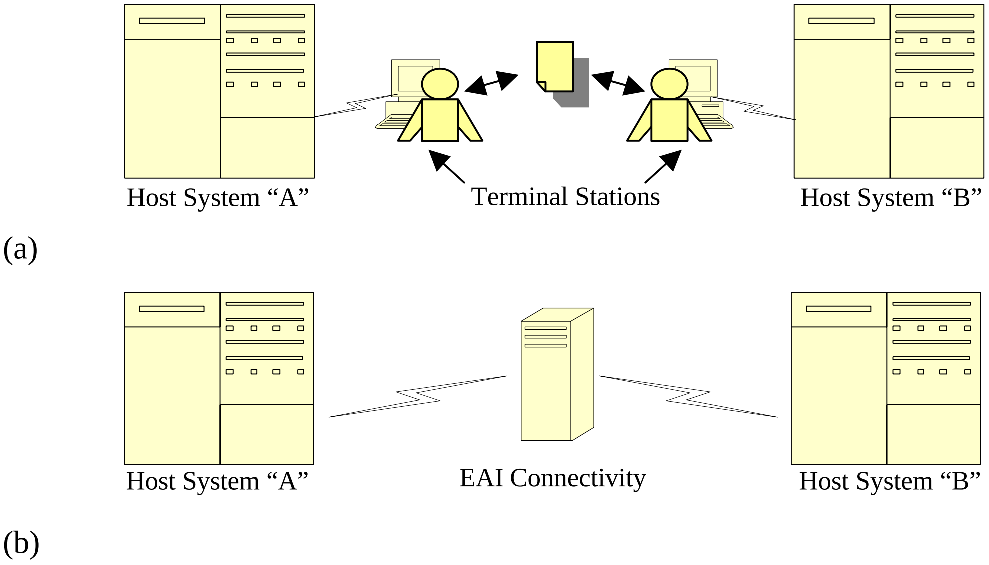


Figure 3: EAI implementation replaces a manual application interface (a) with an automated one (b). This change reduces costs, improves response time and decreases likelihood of error.

Given the above-mentioned benefits, an EAI solution is ideal from a system integration point of view. The question then becomes: is it possible to implement such a solution without incurring the high development costs that are usually associated with such an undertaking? Also, can this solution be implemented in such a way that the various risks of major IT projects, such as data corruption and system downtime, are negated?

Application Program Interface for EAI Solutions

Enterprise-level integration relies on the ability of separate applications, usually running on different computers, to exchange data and services. The basis for this interoperability is the Application Program Interface (API), a set of routines and protocols that enables applications to invoke operations inside each other. An API allows one application, referred to as the client, to request and receive services from another application, referred to as the server. This is a many-to-many relationship because a specific client can receive services from several servers, and a single server can grant services to multiple clients. EAI transforms a closed legacy system into an API server, thus allowing other applications on the network to access its functionality as clients.

For an API to be effective it must utilize a communication protocol that is usable from a variety of hardware platforms and software environments. Because of this, common frameworks for implementing APIs have evolved. These frameworks include RPC, COM, CORBA, SOAP and others. Such frameworks define a common type system and also provide an infrastructure for Remote Procedure Calls (RPC) or other forms of Inter Process Communication (IPC). Above these frameworks, common usage protocols have evolved. Such protocols define standard sets of functions, task specific data-types and, in the case of object oriented frameworks, standard objects and interfaces.

It is important to understand that, by default, distinct frameworks cannot interoperate with each other. For example, a COM client cannot directly receive services from a CORBA server. For this reason, PowerTerm Host Publisher supports several industry standard API frameworks. Its services can be invoked via Microsoft COM (DCOM), using XML over TCP/IP, through Java classes and using SOAP. Moreover, PowerTerm Host Publisher IntegrationServer describes the services provided by the host, using XML metadata. This allows a wide variety of applications to easily connect to a host system through Host Publisher. For example, a web site developed using ASP over Microsoft IIS would connect via COM; a site developed using JSP over IBM WebSphere would connect via the Java classes, and a site build over Apache using PHP would connect via TCP/IP sockets.

A feature of modern API frameworks is that they abstract the relationship between the client and the server. This means that they place the emphasis on the type of services they provide rather than on a specific implementation. Thus modern APIs allow a client to switch servers without requiring code modifications, provided that they belong to the same category. For example, a web browser uses HTTP to communicate with any type of web server. In the context of EAI, this means that the API should obscure the type of the host system. PowerTerm Host Publisher fulfills this premise by presenting the same interface whether the host is an OpenVMS, AS/400 or any other platform.

Implementing EAI Over Host Systems and Legacy Applications

There are two main approaches to providing an API for applications to access the business-logic units encapsulated in a host system. The “white box” solution is to restructure the host so that these services become accessible via modern communication protocols. While there are tools to ease this process, such as code libraries that provide data access services and communication gateways that translate packets and data types, this process is still complex and far from being risk-free. Consider the cost incurred by the various Y2K solutions; yet these only required changes to those specific portions of the legacy code that dealt with date storage and calculation. Providing programmatic access to any service within the host system requires much more significant code modifications.

In many cases a “white box” approach to legacy renovation is simply not practical because:

1. Source code for host applications is often in a dated programming language such as COBOL or RPG. It is difficult to find programmers that have the necessary development skills in these languages, and those programmers that are available command a premium price.
2. In many cases the source code itself has been lost. When it is available it is not documented, and is implemented in an interdependent, monolithic, non-modular design.
3. Even the business processes that the code implements are not documented and are wholly familiar only to a few selected individuals within the organization.
4. Changes to one portion of the legacy code may have adverse effects on other, seemingly unrelated, portions. Some of these effects may not be immediately noticeable, resulting in cumulative long-term damage.

Given these factors it is obvious that a “black box” approach, where features are implemented externally, without modifying the host application, is generally preferable.

A zero-risk “black box” solution utilizes the communication facilities provided intrinsically by the host system. These communication facilities involve the I/O devices supported by the host system, such as terminal interfaces and printers. In a technique commonly referred to as screen-based data capture, input is fed into the host by simulating keystrokes and output is read from the data stream originally intended for display. Using this technique, PowerTerm Host Publisher offers a non-invasive system renovation approach.

PowerTerm Host Publisher ApplicationWizard makes it easy to create high-level wrappers for business processes over the low-level I/O facilities. It does this by providing a complete development environment where host sessions are recorded and automatically converted into script functions that encapsulate complete business-logic units. These script functions can then be activated by name in order to automate the host.

In addition, the optional PowerTerm Host Publisher IntegrationServer organizes these functions into an application flowchart. This flowchart defines a virtual state-machine that describes the business processes implemented in the host application as well as the relations between them. A high-level API enables stateful navigation of the application flowchart. In this manner, the organization’s business processes can be completely automated.

The PowerTerm Host Publisher Solution

Providing an integrated solution to host connectivity scenarios, the PowerTerm Host Publisher suite has several distinguishing features. It boasts an extremely wide coverage of mainframe, midrange and other enterprise systems, including: IBM OS/390, AS/400, Hewlett Packard, Compaq OpenVMS and VAX/VMS, any type of UNIX, Data General and Tandem. Moreover, PowerTerm Host Publisher development tools work identically over all these systems. Even more importantly, the API for manipulating data and business-logic components in the host works the same for all these systems. This means that an application created to access one host system can be used to access another with only minor modifications. For example, a web server, using ASP to generate dynamic web pages off of content coming from an IBM OS/390, can be retargeted to retrieve content from an OpenVMS host with little or no changes. This feature is of particular importance to system integrators and can result in significant cost reductions for any organization that has more than one host type.

A major design feature of PowerTerm Host Publisher is that it allows its clients to work at the business logic level, rather than at the much lower level of keystrokes and screen text. Complete sessions can be easily recorded using the PowerTerm Host Publisher ApplicationWizard, which will generate wrapper functions for the business processes. These functions can then be activated as procedures through the PowerTerm Host Publisher API using high-level data types such as string and arrays. By being process-oriented rather than screen-oriented, PowerTerm Host Publisher allows development to occur at a much higher level of abstraction.

In addition, the optional PowerTerm Host Publisher IntegrationServer component models the relationship between the various business-units that comprise the organization’s entire business process. The model is presented in the form of a flowchart that represents a state-machine. A high-level API can be used to activate the procedures that form the links in the flowchart, moving between the states that are its nodes. Additional API functions provide metadata describing the current state, the procedures that can be activated in that state, and their input and output parameters.

Because all the services provided by PowerTerm Host Publisher are accessible through an API, they can be used to develop any type of application over the host system. “On-the-fly GUI” and screen components, on the other hand, are limited to the generation of Business-to-Human type applications, such as web pages that reflect host data. The PowerTerm Host Publisher API also enables the generation of Application-to-Application type solutions. In the latter scenario, two or more systems can participate in a single transaction, automating a complete business task. In this way, PowerTerm Host Publisher provides the basis for transactional EAI.

The PowerTerm Host Publisher API is built using industry standard protocols and technologies, including XML, TCP/IP sockets, Java, Microsoft COM and SOAP. This means that its services can be accessed simply and directly from almost any development environment. In addition, these services can be activated across the network and even across the Internet. PowerTerm Host Publisher employs a modular framework specifically designed to speed up the development process and simplify ongoing management, ensuring high performance and rapid deployment of the final solution. The modular design allows projects to be completed faster because a project can be broken down into task related components, and development can then occur in parallel. In addition, this approach permits parts of the project to be outsourced.

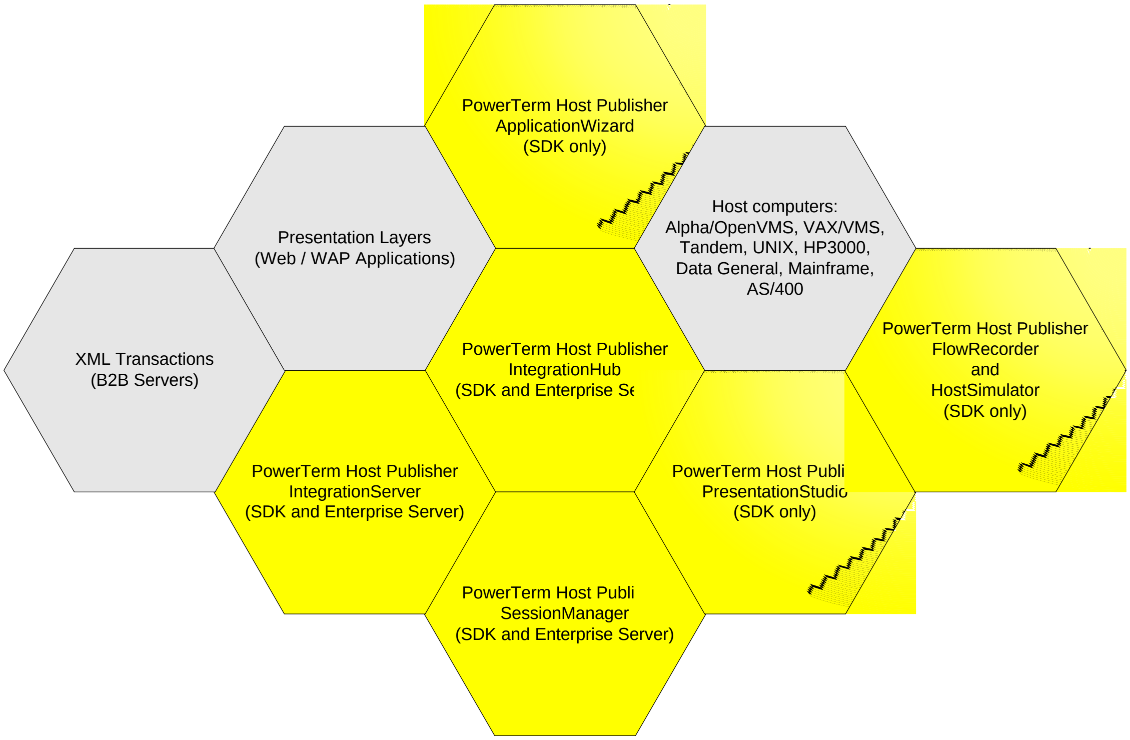


Figure 4: The components of PowerTerm Host Publisher (yellow) and the components they interact with (gray).

PowerTerm Host Publisher IntegrationHub

PowerTerm Host Publisher IntegrationHub is the core component of the Host Publisher, which usually runs as a Windows service. IntegrationHub is sometimes referred to as PowerTerm Host Publisher Server. It exposes the low-level methods and properties of the Host Publisher. Using these methods it is possible to send keystrokes to a host system and read content of a virtual screen. PowerTerm Host Publisher IntegrationHub can be used on its own to either Web-enable a legacy application or to integrate multiple legacy applications into a new corporate or business-to-business solution.

PowerTerm Host Publisher Administrator

PowerTerm Host Publisher Administrator is the main administration tool, synchronizing all global settings for the IntegrationHub. Licensing activities, Remote Interfaces for Java/Socket Server settings and control etc., are managed by the PowerTerm Host Publisher Administrator. It can connect to either a local or a remote PowerTerm Host Publisher Server.

PowerTerm Host Publisher ApplicationWizard

PowerTerm Host Publisher ApplicationWizard automatically generates the code for the Host Publisher functions by recording user sessions. As a result, these functions are snapshots of processes and procedures originally built into the legacy applications. By using PowerTerm Host Publisher ApplicationWizard, extremely complex applications can be developed with only “point-and-click” skills.

Because the functions generated by the ApplicationWizard are implemented in VBScript, they can be modified and extended through the VBA development environment (directly accessible from PowerTerm Host Publisher ApplicationWizard.)

The export feature of the ApplicationWizard creates a basic web application for the recorded business logic of the original legacy application, a process referred to as web-enabling. In this way the ApplicationWizard provides automatic web-enablement of legacy applications.

PowerTerm Host Publisher FlowRecorder

Communicating with the legacy computers is in the very nature of every application designed with the PowerTerm Host Publisher. Ideally the legacy computers need to be accessible also at design-time. However, in reality it is not always possible. PowerTerm Host Publisher FlowRecorder allows you to record business processes from the original applications while they are being executed for off-line usage.

PowerTerm Host Publisher HostSimulator

Once the desired business processes are captured, off-line development of the application is possible with PowerTerm Host Publisher HostSimulator. HostSimulator enables the off-line playback of business logic processes recorded with the PowerTerm Host Publisher FlowRecorder. Although some live-connection adjustments are still necessary during the later stages of the development process, about 70 to 80% of the development activities can be performed off-site.

PowerTerm Host Publisher SessionManager

SessionManager is an add-on to the PowerTerm Host Publisher IntegrationHub. Although any Host Publisher application can be deployed without SessionManager, this may result in not taking full advantage of your run-time licenses, not achieving the best possible response times and performance, and causing extra overhead to the host system. SessionManager helps eliminate these problems by bridging the gap between the stateful nature of legacy connections on the backend systems (i.e,. VT, 3270, 5250 sessions), with the stateless nature of client requests on the front-end (such as HTTP).

In essence SessionManager enables the sharing of each legacy application session between multiple requests. This is accomplished by breaking down monolithic legacy applications into atomic procedures at design time. These procedures are also referred to as business tasks. The SessionManager administrator defines a session pool for each business task. At run time a client request is served by a session retrieved from the appropriate session pool, based on which task is being executed. Upon completion the session is returned to the pool and is ready to serve the next client request. With this time-sharing mechanism, a single host session can service multiple client requests. Also, because these sessions can be initialized in advance additional host overhead is eliminated and response time is improved.

PowerTerm Host Publisher IntegrationServer

PowerTerm Host Publisher IntegrationServer, formally known as the TransactionServer, is an add-on to Host Publisher running as a service on top of the IntegrationHub. IntegrationServer allows business tasks within the original host application to be re-packaged into new applications in a process-oriented manner rather than simply converting screens to the web. This allows organizations to re-purpose the original application(s) for use by suppliers, business partners, sub-contractors, and customers over the Internet or Extranet. Host Publisher IntegrationHub can also be used to integrate internal, disparate legacy applications as new needs emerge.

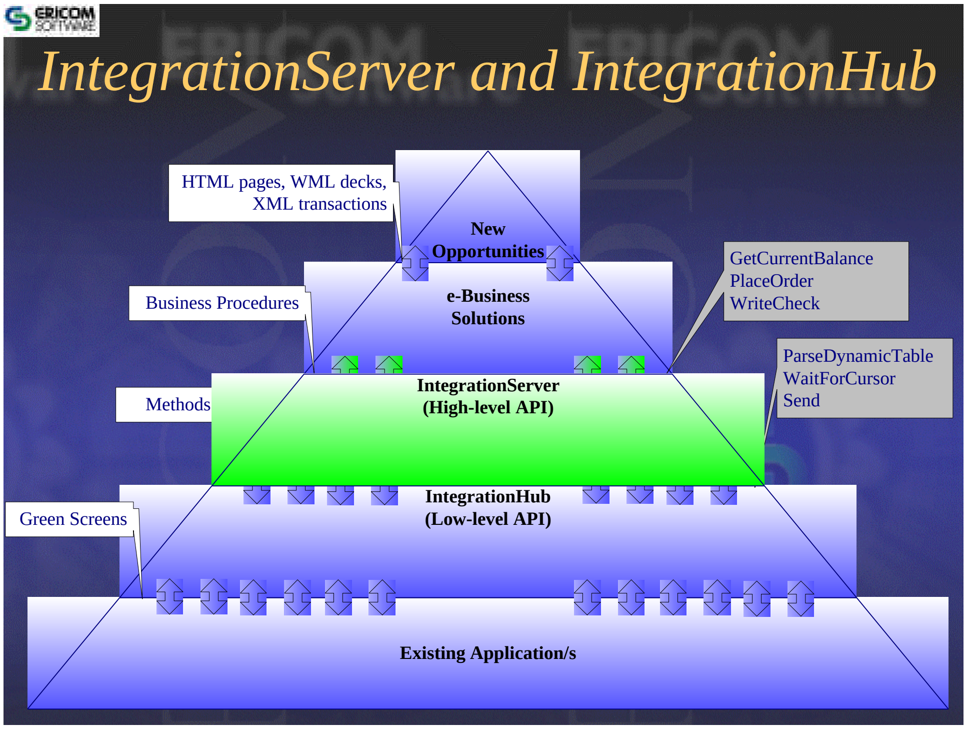


Figure 5: Built as a layer above PowerTerm Host Publisher IntegrationHub, IntegrationServer provides high-level access to host services through an API that works at the business process level.

The key feature of PowerTerm Host Publisher IntegrationServer is that it allows the functions generated by the ApplicationWizard to be combined into an application flowchart. This flowchart (figure 6) has two types of nodes: states (represented by ovals) and operations (represented by rectangles). A state is a holding position for the legacy application where it waits for input. In terms of the original legacy application a state represents a screen from which a user can select which operation to perform, for example a menu. For each state, there exists one or more operations that can be initiated from it.

These operations are the functions generated by the ApplicationWizard. When an operation finishes it places the application in a state depending on its result.

PowerTerm Host Publisher SDK allows the developer to visually model the application flowchart. IntegrationServer provides a high-level API that allows its clients to query their current state, determine which operations can be invoked in that state, activate a specific operation, and receive its output. XML is used as the serialization format for this information.

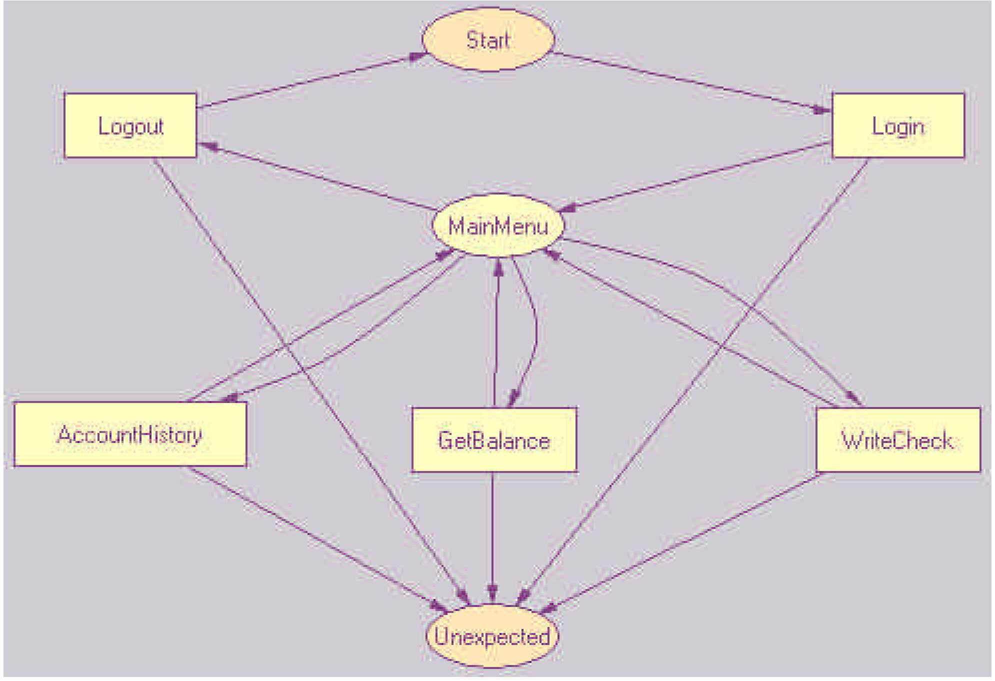


Figure 6: A small IntegrationServer application flowchart. This flowchart defines a simple checking account application.

Conclusion

Selecting the appropriate host system and legacy application integration strategy can jump-start a corporate e-Business initiative. Conversely, selecting the wrong integration strategy will not only impede such an initiative, it can actually have an adverse effect on the existing IT infrastructure. To succeed the integration solution should have provide the following features:

- Provide the foundation for complete EAI enabling programmatic access to the data stored on a host system through its business processes. Such a solution goes beyond putting an HTML face on the existing legacy screens.
- Utilize non-invasive (“black box”) techniques to get to the information and services stored in the host. This eliminates the danger of data corruption due to insufficient or incorrect data validation in new modules added to the host in a “white box” approach, and prevents system downtime.
- Support a wide variety of host systems. In addition, the same set of APIs and services should be available for all types of hosts. This minimizes the cost of

integrating all the host systems in an organization. It is also essential for system integrators because it facilitates reuse development resources.

- Provide an API that can be used by any potential client application: web and application servers, business integration servers, etc. Such an API should match client usage patterns, thus reducing the learning curve and speeding development.
- Supply a robust development environment for leveraging the API and modeling business processes on top of the host services.
- Enable optimal use of host system resources such as sessions.
- Security and scalability.

By providing a complete host system and legacy application integration solution that satisfies all these requirements, PowerTerm Host Publisher provides a solid foundation upon which any organization can build current and future e-Business projects.

About Ericom Software