



# PowerTerm<sup>®</sup> Host Publisher

## Overview

Version 5.2

**Ericom North America**  
Ericom Software Inc.  
231 Herbert Ave., Bldg. #4  
Closter, NJ 07624 USA  
Tel: +1 201 767 2210  
Fax: +1 201 767 2205  
Toll Free: 1 888 769 7876  
Email: [info@ericom.com](mailto:info@ericom.com)

**Ericom Europe**  
Ericom Software (UK) Ltd.  
11a Victoria Square  
Droitwich, Worcestershire  
WR9 8DE United Kingdom  
Tel: +44 (0)1905 777 970  
Fax: + 44 (0)1905 777 972  
Email: [info@ericom.co.uk](mailto:info@ericom.co.uk)

**Ericom International**  
Ericom Software Ltd.  
8 Hamarpeh Street  
Har Hotzvim  
Jerusalem 91450 Israel  
Tel: +972 (0)2 571 4774  
Fax: +972 (0)2 571 4737  
Email: [info@ericom.com](mailto:info@ericom.com)

# Important Notice

This document is subject to the following conditions and restrictions:

- The proprietary information belonging to Ericom<sup>®</sup> Software is supplied solely for the purpose of assisting explicitly and properly authorized users of PowerTerm<sup>®</sup> Host Publisher.
- No part of its contents may be used for any other purpose, disclosed to any person or firm, or reproduced by any means, electronic and mechanical, without the express prior written permission of Ericom<sup>®</sup> Software.
- The text and graphics are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of that agreement.
- Information in this document is subject to change without notice.
- Corporate and individual names and data used in examples herein are fictitious unless otherwise noted.

Copyright © 2000-2004 Ericom<sup>®</sup> Software

Ericom<sup>®</sup> and PowerTerm<sup>®</sup> are registered trademarks of Ericom<sup>®</sup> Software, which may be registered in certain jurisdictions. Other company and brand, product and service names are trademarks or registered trademarks of their respective holders.

# Table of Contents

<b>IMPORTANT NOTICE</b>	<b>2</b>
<b>TABLE OF CONTENTS</b>	<b>3</b>
<b>INTRODUCTION</b>	<b>5</b>
<b>POWERTERM HOST PUBLISHER OVERVIEW</b>	<b>7</b>
<b>IMPLEMENTING POWERTERM HOST PUBLISHER SOLUTION</b>	<b>9</b>
Define The User Experience	10
Define Data and Business Logic Requirements	10
Create Functions using ApplicationWizard	10
<b>POWERTERM HOST PUBLISHER MODULES</b>	<b>11</b>
<b>IntegrationHub</b>	<b>14</b>
<b>IntegrationServer</b>	<b>15</b>
Integration	15
Abstraction	16
Isolation	16
<b>ApplicationWizard</b>	<b>17</b>
<b>IntegrationServer Administrator</b>	<b>18</b>
<b>PresentationWizard</b>	<b>19</b>
<b>Web Services</b>	<b>20</b>
<b>Session Pooling</b>	<b>21</b>

<b>Instant Deployment</b>	<b>22</b>
<b>FlowRecorder</b>	<b>23</b>
<b>HostSimulator</b>	<b>24</b>
<b>Wrappers</b>	<b>25</b>
<b>WINDOWS VS. NON-WINDOWS EDITIONS OF POWERTERM HOST PUBLISHER</b>	<b>26</b>
PowerTerm Host Publisher Workflow	26
<b>Comparison</b>	<b>28</b>
<b>SYSTEM CONSIDERATIONS</b>	<b>30</b>
Rollback	30
Unexpected Errors or States	30
Security	30

# Introduction

Ericom provides three different middleware tool sets that allow you to access legacy system information, including business logic, with no modifications to the legacy system. The connection to the legacy system is accomplished by means of terminal I/O. The three products are:

- **WebConnect** along with thin (Java) or fat (HostView) downloadable clients. A host access server that enables immediate Web access to legacy systems.
- **WebConnect WebView** An additional downloadable client of the WebConnect Server with the added capability of beautification plus the ability to add some additional functionality.
- **Host Publisher** You can create any user experience you wish while reusing information and business logic from one or more legacy systems.

The capabilities of the respective products are summarized in the following table:

	<b>WebConnect</b>	<b>WebConnect WebView</b>	<b>Host Publisher</b>
<i>Changes required to legacy system</i>	None	None	None
<i>Time to deploy</i>	Immediate	Immediate but can also enhance appearance	Development is situation dependent
<i>End user look and feel</i>	Exactly the same look and screen appearance but through a browser	Beautification and customization is possible.	You can create any user experience you wish

<i>Can business logic and information from multiple applications or systems be integrated into a single user experience</i>	No	No	Yes
<i>Look and feel</i>	Emulation	Between emulation and Web	Pure Web

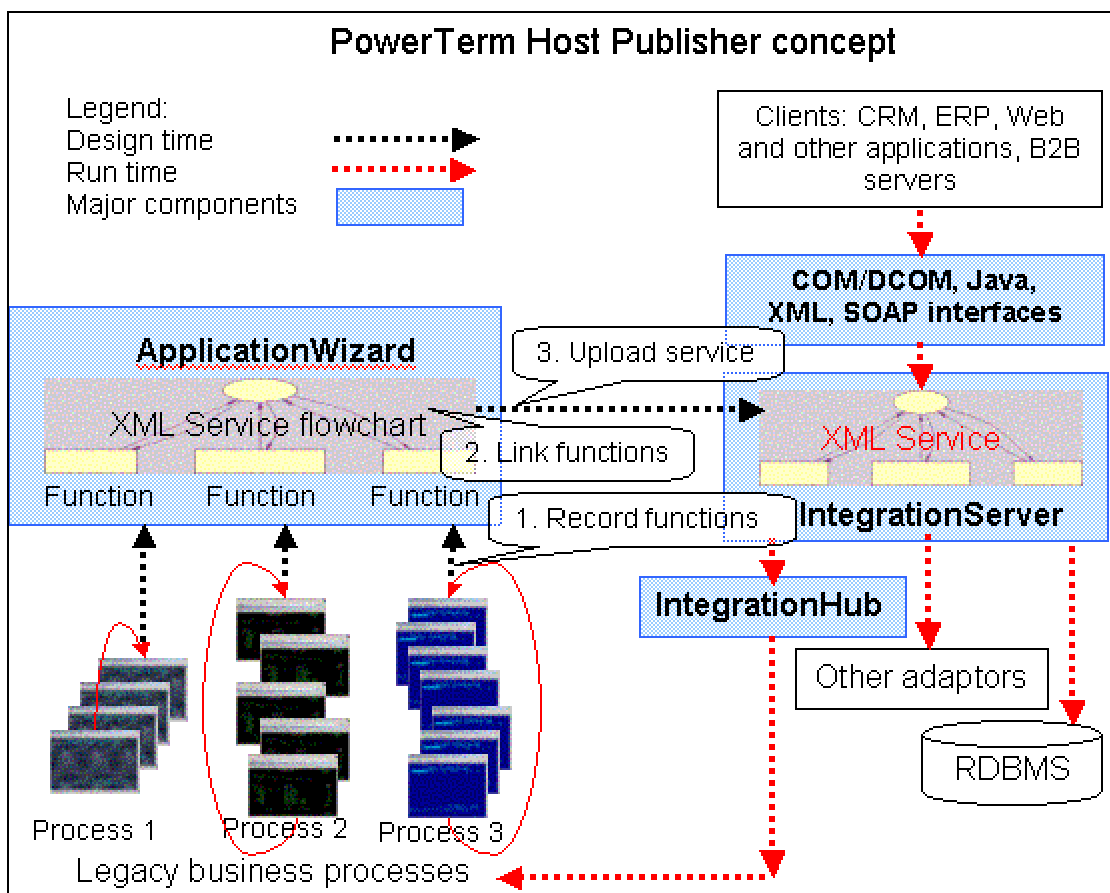
The legacy systems supported at this time by all three products include:

- VAX/Alpha OpenVMS
- IBM Mainframe
- IBM AS/400
- UNIX
- HP 3000
- Data General
- Tandem

This document provides an overview of the PowerTerm Host Publisher. For information on the PowerTerm WebConnect and the PowerTerm WebConnect WebView, please see respective manuals.

# PowerTerm Host Publisher Overview

The following illustrates the process of implementing an application:



These modules can be logically divided into four functional groups.

**N**OTE *The IntegrationHub is common to all.*

**Online Application Development** (online access to a host is available):

- IntegrationHub
- ApplicationWizard
- PresentationWizard

**Offline Application Development** (online access to a host is available for only a short period of time):

- FlowRecorder
- HostSimulator
- IntegrationHub
- ApplicationWizard
- PresentationWizard

**Run time:**

- IntegrationHub
- IntegrationServer
- Instant Deployment

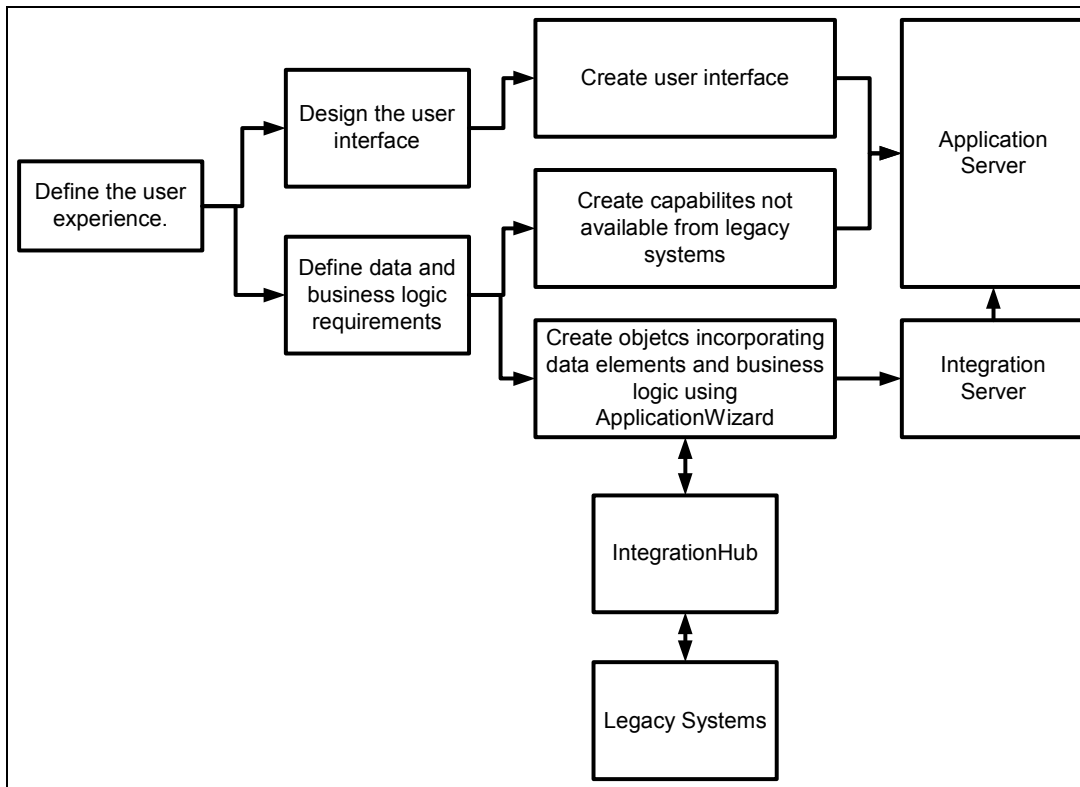
**Administration:**

- IntegrationServer Administrator



# Implementing PowerTerm Host Publisher solution

The basic steps for implementing a Host Publisher based system are illustrated below:



Due to the capabilities of the PowerTerm Host Publisher, you have the ability to design the user experience with little concern for the current legacy screens or legacy application flow. Also, even if the data and business logic is spread across multiple systems, you can use PowerTerm Host Publisher to integrate these resources into a unified user experience. While there is much more involved in fielding a system, the three basic steps for implementing a Host Publisher solution are as follows:

- 1** Define the user experience.
- 2** Define Data and Business Logic Requirements.
- 3** Record functions using the ApplicationWizard.

## **Define The User Experience**

The first step is to define the user experience. Various reports estimate that almost 70% of shopping baskets are abandoned. In many cases the carts are abandoned because the user gets frustrated and clicks to your competitor. In Intranet scenarios, a poorly designed user interface results in longer and more expensive training than necessary, excessive use of a search engine looking for missing information and there are many cases of users refusal to use the system. Getting the User experience right is essential.

## **Define Data and Business Logic Requirements**

The outputs from the user experience design include the information necessary to design the user interface as well as define the data and business processes that will be required. In most cases, all the data and business rules will already exist in the legacy systems. If not, then the additional capabilities can be added via new systems on Unix or NT or whatever system makes sense for the new processes. All of these systems can be integrated using the PowerTerm Host Publisher.

## **Create Functions using ApplicationWizard**

Once the data and business logic are defined, use the ApplicationWizard's point and click interface to capture (record) the business processes from the legacy system(s). As you capture the data and procedures, the ApplicationWizard produces either VB script or Java functions as desired. These functions are uploaded to the IntegrationServer, which exposes them to client applications (Web, XML, WAP, SOAP etc.) at run-time as appropriate. As mentioned above applications can access the IntegrationServer services via COM/DCOM, Java, TCP/IP/XML or HTTP/SOAP.

What the application server sees is a high level API, not the legacy system. The high-level API represents corporate business tasks, which can be executed on the legacy machine. This level of abstraction allows the Web developers to do what they know how to do. Also, the programmer that captures the various data and business logic from the legacy application does not have to know anything about the user interface. This isolation of tasks greatly simplifies the creation of new systems and reduces the skill set required by all. In many cases, a junior programmer who has practiced with the ApplicationWizard, working with an experienced user (not necessarily a programmer!) of the legacy system, can create the needed objects and functions with zero programming and zero risk for the original legacy application(s).

# PowerTerm Host Publisher Modules

As a reference, the following table contains a brief description of each module. For more information, see the module descriptions further down. For detailed information, see the different help manuals and overview (in pdf and html format) in addition to the Online helps for each module. A **Hands-On Training Center**, with samples and Labs in html, is also available as well as **Readme** and **Release Notes** files in txt format for each edition of the PowerTerm Host Publisher.

## **IntegrationHub**

The core component of both editions of the PowerTerm Host Publisher (SDK and Enterprise Server).

- Connects to the legacy system using terminal I/O.
- Communicates with the legacy system by simulating keystrokes and reading the system's responses.
- Exposes low level methods and properties via COM/DCOM, Java, or TCP/IP Sockets.
- Supports Win32 OS, Linux, Sun Solaris, and OpenVMS.

## **IntegrationServer**

- Exposes business processes of legacy application(s) as a service via HTTP/SOAP, TCPIP/XML, COM/DCOM and Java.
- Executes business processes represented by transactions created using the ApplicationWizard.
- Hides details of legacy system access from the end-user application developer.
- Supports Win32 and Java.

**ApplicationWizard**

- Creates functions and transactions (representing single business processes of an original application on the legacy machine) by either recording key presses and screen sequences via point and click, or by creating Visual Transactions. Functions and transactions are implemented in either VB Script or Java.
- Creates a service based on the library of business procedures and represents it as a flowchart in the design-time environment. The Service Flowchart illustrates the relationship between all of the legacy business processes (ApplicationWizard functions and transactions) exposed by the service.

**IntegrationServer Administrator**

- Manages the IntegrationServer and the IntegrationHub, the services on the IntegrationServer, the TCP/IP (sockets) Interface, Web Service optimization, SOAP Interface settings, etc.
- Connects to IntegrationServer running on local or remote computer.

**PresentationWizard**

- Automatically generates a fully functional Web application for services previously created with the ApplicationWizard and deployed on the IntegrationServer.

**Web Service**

- Platform independent Web Services can be created from the IntegrationServer services.

**Session Pooling**

- Provides session sharing so multiple Web users can share a single session resource.
- Handles initialization of sessions in advance of actual need based on anticipated demand thus eliminating the initialization overhead at real time and cutting the initial request responsiveness to only a few seconds (a must for most of the Internet applications).

**Instant Deployment**

- Enables the developer to deploy certain parts of the legacy application on the Web browser through the WebConnect WebView component, thus not requiring conversion of all of the legacy application into services.

**FlowRecorder**

- Records business processes within the original (legacy) application for offline development using the ApplicationWizard.

**HostSimulator**

- Plays back legacy business processes previously recorded by the FlowRecorder. Recorded processes are used for offline development of the required e-Business solution (either the new service or a Web application).

**Wrappers**

- The different wrappers supplied enable communication with the IntegrationServer from different programming technologies.

# IntegrationHub

The IntegrationHub is the core component of the PowerTerm Host Publisher and can run as an NT Service (Windows edition) as well as an “in-process” component, “IntegrationHub Native implementation” (Java edition). In the current version for Windows it is packaged as an OutProc COM (Microsoft Component Object Model) component. It exposes the low-level methods and properties of the PowerTerm Host Publisher.

The 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. However writing applications with only PowerTerm Host Publisher IntegrationHub without using other PowerTerm Host Publisher sophisticated productivity tools might be somewhat labor-intensive.

Productivity tools, such as the ApplicationWizard, are included with the SDK edition of the PowerTerm Host Publisher. Applications can call the IntegrationHub methods either using a standard COM/DCOM (Distributed COM) or Java. Activating a socket server built into the IntegrationHub enables both Java and TCP/IP interfaces.

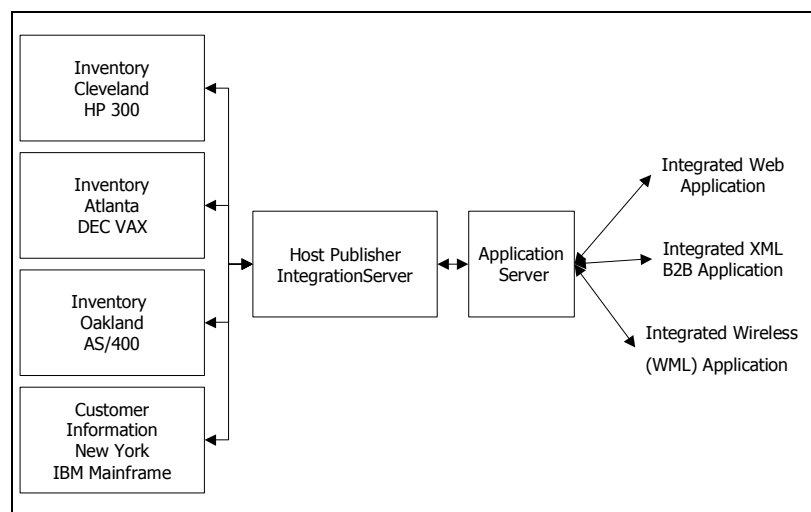
# IntegrationServer

The IntegrationServer exposes to its clients (via COM/DCOM, Java, HTTP/SOAP, TCPIP/XML) the high level API of legacy business procedures recorded with the ApplicationWizard. It also provides Integration, Abstraction, and Isolation.

- **Integration** Integrate disparate legacy applications across multiple platforms into a single cohesive enterprise-class application or service.
- **Abstraction** Design new applications with the high-level of abstraction in a process-oriented manner rather than converting screens to the Web.
- **Isolation** The IntegrationServer exposes the high-level API of the business procedures created in the ApplicationWizard. As the Web developers only see these high level APIs, they can concentrate on the user interface. On the other hand, the programmer, working with an experienced legacy system user, creates the APIs in the ApplicationWizard. Thus, not requiring specific knowledge of the user interface other than the data and business logic requirements.

## Integration

The IntegrationServer exposes the APIs of the business processes created in the ApplicationWizard. The server does not have any knowledge if the APIs access one or many legacy applications or systems. The following illustrates a run-time architecture of a system that integrates disparate systems at geographically separate locations with different applications and different functions into a single API.



## **Abstraction**

With the PowerTerm Host Publisher IntegrationServer, you can design new applications with the high-level of abstraction in a process-oriented manner rather than converting screens to the Web. This allows you to re-purpose or re-deploy the original business application(s) for use with business partners, sub-contractors, Internet consumers, etc. exposing to them desired business processes.

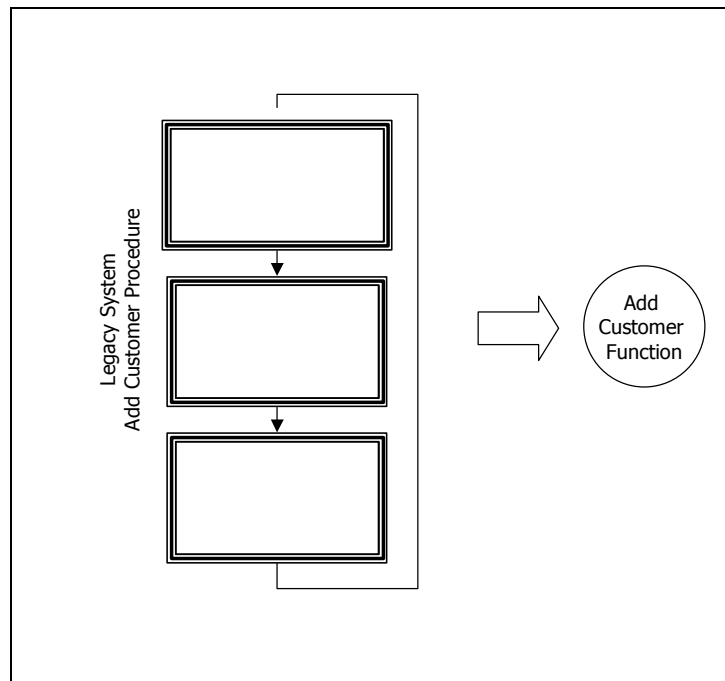
## **Isolation**

The IntegrationServer isolates the functions of the programmer from the functions of the end-user application developer. While the programmer focuses on extracting business processes from the legacy applications, the application developer can focus on the user experience and other aspects. These aspects usually are the end-user application logic and interface, or the application rules in case the target is a service (machine-to-machine scenario). Both players do what they know to do best, while the rest of application layers are transparent to them.



# ApplicationWizard

The ApplicationWizard is a design-time tool supplied with the Host Publisher SDK package, which enables you to record functions in VB Script or Java that encapsulate legacy system business procedures. For example, in the illustration below, three screens are required on the legacy system to add a new customer. Using the ApplicationWizard point and click interface, you capture the commands, input fields, and return fields necessary to complete adding a new customer. This technique also captures the business logic, error messages, and the various system responses.



The output from the ApplicationWizard is a Service flowchart stored as an XML file and represented in the design-time environment. The Service flowchart is then uploaded (deployed) to the IntegrationServer.

With the ApplicationWizard, you can record functions that wrap legacy system procedures without the need to write a single line of code.

The ApplicationWizard also supports the creation of “codeless” Visual Transactions for IBM block-mode emulations. Such transactions are developed using point-and-click on fields in an emulation view.

# IntegrationServer Administrator

The IntegrationServer Administrator is the main administration tool of the PowerTerm Host Publisher, which allows managing the Services on either a local or a remote IntegrationServer.

The IntegrationServer Administrator performs such essential administration activities as setting of a default Service, adding (loading) of a new Service, monitoring of statistics, etc. All global settings are also accomplished by it, as well as the administration of the Session Pooling feature and creating Web Services.

The IntegrationServer Testing Utility can also be launched from within the IntegrationServer Administrator.

# PresentationWizard

The PresentationWizard automatically generates a fully functional Web presentation layer for a Service previously designed with the ApplicationWizard and deployed on the IntegrationServer.

The Web presentation layer (GUI) is generated according to the selected server-side scripting platform and a form-factor of the target client device/browser combination. Once generated in a very attractive default layout, if needed, the resulting code can be customized to meet specific end user requirements.

The PresentationWizard can generate both ASP and JSP.

# Web Services

PowerTerm Host Publisher enables developers to easily deploy any IntegrationServer service as a Web Service. The IntegrationServer Administrator creates all the files needed in order to run the IntegrationServer services as Web Services. Every transaction existing in the IntegrationServer service is implemented as a specific method in the created Web Service. This means that for every IntegrationServer service a distinct WSDL file is generated, which uniquely describes that service.

IntegrationServer services often need to maintain state and as result require the concept of a session. SOAP, in particular when using HTTP as the transport protocol, is stateless – each Web Service method can be invoked independently of any other. PowerTerm Host Publisher provides two facilities to bridge this gap:

- HTTP cookies
- Session id within the SOAP header

In the **Windows edition**, four files are created for each IntegrationServer service:

- **WSDL file** contains the description (object and methods) of a Web Service for a certain IntegrationServer service. The file is a reference for clients accessing the Web Services on the Web Server.
- **Global.asa** keeps the SoapDispatcher through a session scope.
- **HostPublisherWSDLServer.asp** runs the SoapDispatcher.
- **XML file** helps the SoapDispatcher to transform the different XML messages between the Web Service and the IntegrationServer.

The files are created in a new **Output** directory. A new Virtual Directory on the IIS, which points to the Output directory, is also automatically created.

In the **Java edition** a WAR (Web Application Archive) file is created. This file contains all that is required to run transactions on the IntegrationServer using Web Services. It also includes the WSDL file with the description of the IntegrationServer service. The WAR file should be copied to the WebApps library on the Tomcat server.

# Session Pooling

The Session Pooling feature of PowerTerm Host Publisher shortens the waiting period of the initial host connection. The initialization stages of connecting to a legacy system can take up a lot of time while the end-user is waiting and impose a burden on the Web Server and on the host to which the end-user is connecting.

The PowerTerm Host Publisher IntegrationServer maintains a pool of sessions prepared in advance that have already completed the login transaction process. When a service requires to run the login transaction, the IntegrationServer provides a connection from the Session Pool thus enabling instantaneous host connection.

The Session Pool contains sessions created from transactions that originate in the Start state.

# Instant Deployment

The Instant Deployment architecture enables the developer to deploy certain parts of the legacy application on the Web browser through the PowerTerm WebConnect OCX component (Ericom Software's browser-based Web-to-host connectivity solution), thus not requiring conversion of all of the legacy application into IntegrationServer services.

The Instant Deployment utilizes the OCX Control, which can be deployed immediately as well as be customized at a later stage. It provides a web look and feel for legacy applications, while no programming is required.

There is an ongoing communication between the OCX Control and the legacy system through the PowerTerm Host Publisher, thus determining what is being displayed in the Web browser.

# FlowRecorder

The FlowRecorder records screen flows from the original applications as qualified internal personnel execute them. Ideally the legacy computers will be accessible during design and development. However, in real life sometimes it is not possible. Once the desired screen flows are recorded, they can be played back through the HostSimulator.

The combination of the FlowRecorder and the HostSimulator enables offline development with the PowerTerm Host Publisher SDK.

# HostSimulator

The HostSimulator enables offline development of the application.

The HostSimulator plays back screen sequences previously recorded with the FlowRecorder. Although some live-connection adjustments are still necessary at the later stages of the development process, about 70 – 80% of the development activities can be performed offline.



# Wrappers

The Wrappers supplied with the PowerTerm Host Publisher allows easy communication with the IntegrationServer in different environments. The following wrappers are supplied:

<b>IntegrationServer WinWrapper</b>	Enables work with Visual Basic, VBScript, and other COM clients, for example Microsoft Office applications.
<b>IntegrationServer JWrapper</b>	A Java package that works with both Windows and non-Windows versions in any Java environment, for example Java applications, Servlets, and JSP.
<b>IntegrationServer DBWrapper</b>	Enables work with the IntegrationServer as if it is a relational Database. Allows invoking transactions as stored procedures and retrieving data as record-sets from any OLEDB, ADO, or ADO.NET compatible environment.
<b>IntegrationServer PHPWrapper</b>	Enables work in a PHP environment.
<b>EJB</b>	Looks at the IntegrationServer as a Session Bean.
<b>Web Services</b>	Looks at the IntegrationServer as a Web Service.

# Windows vs. non-Windows editions of PowerTerm Host Publisher

This chapter describes a typical PowerTerm Host Publisher workflow and compares the Windows and non-Windows editions of the PowerTerm Host Publisher, pointing out similarities and differences.

## PowerTerm Host Publisher Workflow

### **1 ApplicationWizard**

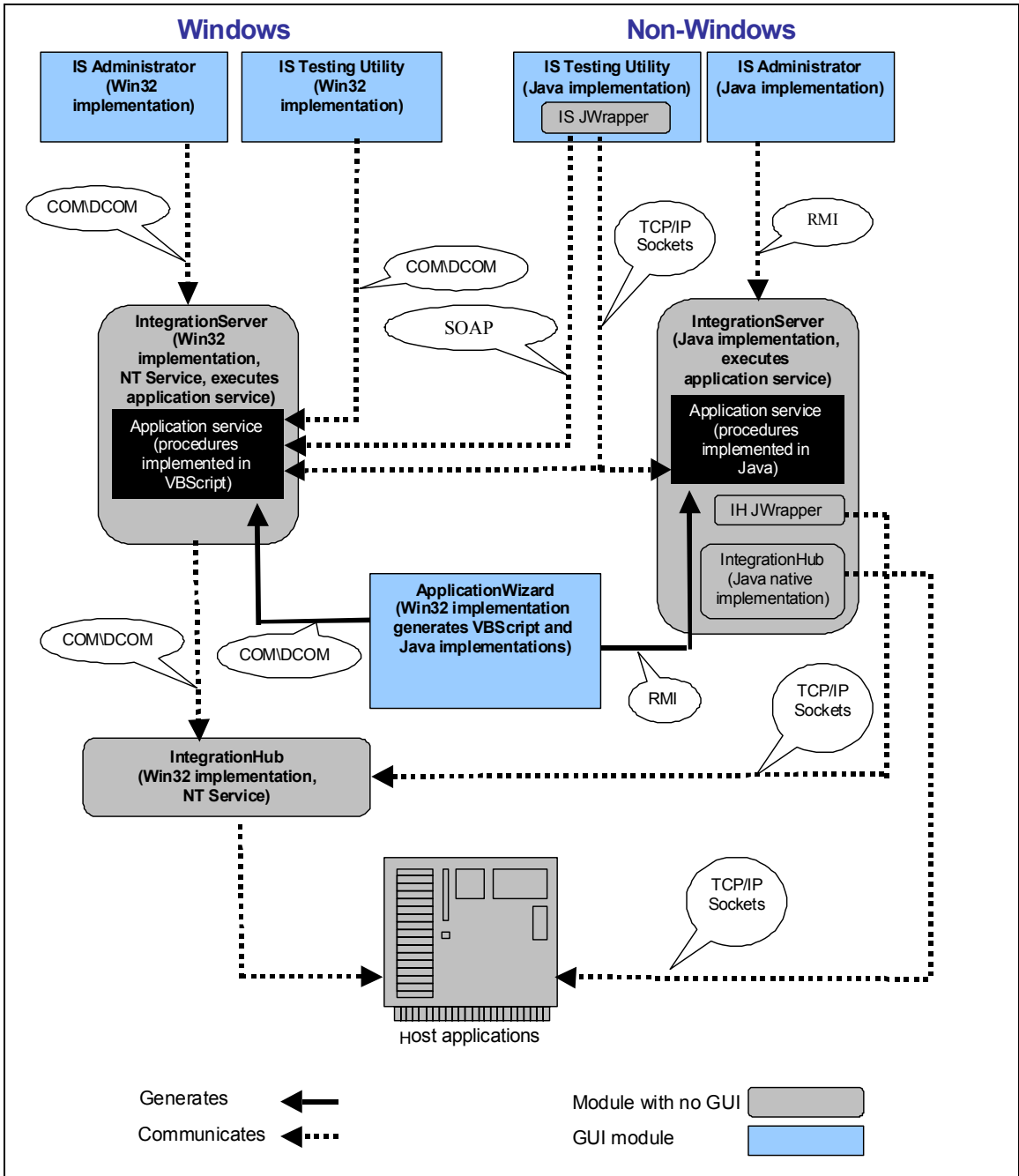
The most common place to begin is at the formation of transactions. ApplicationWizard creates transactions, which use the IntegrationHub API to control the host and perform actions.

### **2 Upload**

After creating the Service flowchart, the “Upload” action adds the created Service to the IntegrationServer’s services.

### **3 Testing Utility**

You can then test your Service. The Testing Utility is a tool that uses IntegrationServer API in order to execute any service’s transactions as selected by the user. The **IntegrationServer** executes the requested transaction, using the created Service.



# Comparison

	Windows	Non-Windows
<b>IntegrationHub</b>	IntegrationHub is a Windows NT service, which communicates with its Administrator through Microsoft COM. In order to invoke IntegrationHub API methods, client applications communicate with it through either Microsoft COM or IH JWrapper (sockets). For each client, a new thread is created.	IntegrationHub is an integral part of the IntegrationServer and runs as an in-process component, IntegrationHub Native implementation. Since it is not an independent service, client applications cannot communicate with it.
<b>IntegrationServer</b>	IntegrationServer is a Windows NT service, which communicates with all the Host Publisher Windows version components through Microsoft COM technology. Therefore the Testing Utility and the Administrator are COM clients of the IntegrationServer service and the IntegrationServer is a COM client of the IntegrationHub service. The IntegrationServer executes the IntegrationHub API methods by running VBScript code contained in the hpts file.	The IntegrationServer is a Java Application, which communicates with Host Publisher components either through sockets or through RMI. It invokes the IntegrationHub API methods by running the Java class that is contained in the hpts file. That class invokes IntegrationHub API methods, which can be accomplished in two ways, by using either the IntegrationHub Native implementation or the IntegrationHub JWrapper. The later is a Java package created to allow Java applications to invoke IntegrationHub API through sockets. It can communicate with the IntegrationHub Windows version. The IntegrationServer Administrator determines which implementation to use.

<b>ApplicationWizard</b>	The code is a VB Script code saved in ASCII format.	The code is a compiled Java code and the hpts file includes the compiled class bytes.
<b>Upload</b>	ApplicationWizard communicates with the IntegrationServer through Microsoft COM technology.	The non-Windows version of PowerTerm Host Publisher cannot use COM for communication as the ApplicationWizard is a Windows application and the IntegrationServer is a Java service. Therefore a Java class, which communicates with the IntegrationServer through RMI, invokes the required actions in order to upload new services by the ApplicationWizard, when needed.
<b>Testing Utility</b>	Use of IntegrationServer API through Microsoft COM technology.	Use of the IntegrationServer JWrapper package in order to use IntegrationServer API. The IntegrationServer JWrapper can work with both Windows and non-Windows versions.
<b>IntegrationServer Administrator</b>	A Win32 GUI application, that communicates with the IntegrationHub and the IntegrationServer through COM/DCOM.	A Java GUI application, that communicates with the IntegrationServer through RMI technology.

# System Considerations

## Rollback

In many cases, provision must be made if a procedure fails. A failure might be due to a variety of reasons. In most cases, rollback is implemented by modifying the VB Script or Java code to implement the needed logic to negate a partial execution.

## Unexpected Errors or States

No matter how well you plan or anticipate system states, the unexpected will always happen eventually. In every Service flowchart generated by the ApplicationWizard, there is an “Unexpected” state. An appropriate “Escape” or “Get-In-Sync” procedure for the “Unexpected” state should be made to add appropriate logic to perform whatever actions are necessary to return the service to a known state.

## Security

In the three-tiered architecture of the PowerTerm Host Publisher, corporate hosts are never directly exposed to the Web. There are two levels of security: access security and information security. This ensures that legacy information is encrypted at all times and not exposed to potential hackers.

- **Access security** Access to the legacy system is limited to the high level APIs in the IntegrationServer. Thus, the functions that an end user can perform are limited. Further, access via the Web or other method is controlled via a wide variety of regular security measures such as Secure Server ID and 128-bit Global Server ID from VeriSign Inc etc., as you deem appropriate.
- **Information security** IntegrationServer allows operations in the 128-bit encryption secure mode (SSL 2.0, SSL 3.0, TLS 1.0, SSH 1.0, and SSH 2.0).