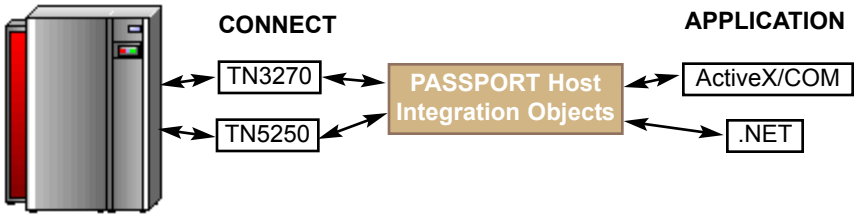
PASSPORT Host Integration Objects

PASSPORT
Host Integration Objects

Client, Server and Developer Editions Offer ActiveX and .NET 3270 and 5250 Host Integration



PASSPORT Host Integration Objects offers a programmatic method of integrating 3270 and 5250 host applications with other applications using either an ActiveX or .NET development platform.

There are three editions of the Zephyr product; a Client Edition, a Server Edition and a Developer Edition. The Server Edition is designed for a Microsoft Windows 2003 Server or Windows 2000 platform, while the Developer Edition can be used on a Windows XP or Windows 2000 desktop, and the Client Edition can be deployed to Windows XP, Windows 2000, Windows NT, or Windows 98.

PASSPORT Host Integration Objects offers a path to host information at the screen buffer level, allowing you to read and write data to the host presentation space and input fields, open and close sessions, get specific text strings from the screen, create string values, send function keys to the host, switch to other tasks and much more. In doing so, customers can reuse critical legacy applications and proven business logic without making any changes to host code.

Using Microsoft's Visual Studio Version 6 (VB, VC++, etc.) or Visual Studio .NET (VB, VC++, C# .NET, ASP .NET, etc.), composite applications can be created that require fast, reliable access to existing IBM mainframe 3270 and 5250 applications.

Available via an annual subscription-based license that includes the cost of the software, access to technical support and all product upgrades, PASSPORT Host Integration Objects offers one of the most inexpensive methods to continue the use of valued legacy applications.

PRIMARY FEATURES

- Non-intrusive integration with existing 3270 and 5250 applications
- Direct access to host screen buffers
- Object-oriented API based on IETF OHIO draft
- Does not require terminal emulation software
- Proven TN3270 and TN5250 communications module
- Designed for a Microsoft Windows environment
- Offset addressing for host screen presentation space
- Simplified model for presentation space
- Automatic generation of presentation space field list and attributes
- Keyword-based function text strings
- Event notification for host communication status
- Diagnostic tracing capabilities

INCLUDED IN THE SOLUTION

- Programming Documentation
- Test Programs in Visual Basic and VBScript that demonstrate how to use each object method, property and event
- Sample Program in Visual C++
- Sample Program in C# .NET

SYSTEM REQUIREMENTS

- Windows XP, 2000 (for developer edition)
- Windows 2003, 2000 Server (for server based applications)
- Windows XP/2000/NT/98, (for client based applications)
- Microsoft Visual Studio Version 6 or .NET
- Any Microsoft programming language that supports the ActiveX COM or .NET interface

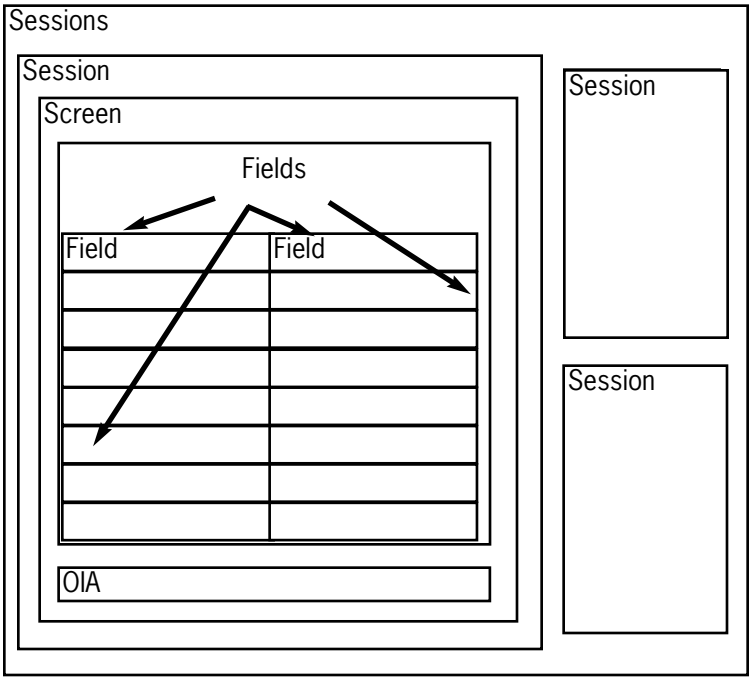
PASSPORT Host Integration Objects, based on the IETF OHIO draft, includes direct TN3270 and TN5250 connectivity to critical legacy applications using IP. Separate terminal emulation is not required for host access. Sample programs, documentation and test programs that demonstrate the use of each object method, property and event are included in the Zephyr development solution.

WHAT IS OHIO?

The Open Host Interface Objects (OHIO) address the need for a standardized advanced programming interface to the host data. OHIO does not modify the TN3270/TN5250 protocol or data stream but instead provides a common access method to that data once it arrives at the client. OHIO uses an object oriented approach to divide the data into logical objects, and provides methods on those objects to allow standard access to the data. You do not have to be concerned with details of structure packing and parameter command codes, but can focus on the application functions.

The OHIO API was designed as a replacement for the older IBM HLLAPI (High Level Language Application Programming Interface). The reasons include:

- OHIO was designed as an object oriented API, and thus has all the well known benefits of the object oriented programming paradigm.
- Interacting with the host generally requires less code using the OHIO API than doing the same function through the HLLAPI interface.
- OHIO objects concentrate related functions into specific classes, an easier to understand model than the one HLLAPI entry point.
- No need to have a emulation session running in order to run an OHIO application.



PERFORM THE FOLLOWING HOST FUNCTIONS

- Open a host session
- Hide host screens from the user
- Read and write information to the host presentation space
- Retrieve text strings from host input fields
- Retrieve text strings from screen
- Determine new string values
- Get associated attributes of text strings and fields
- Copy information into host input fields
- Simulate use of function keys with host application
- Read the operator information area (OIA) for status updates
- Receive notification of events (receipt of new screen, etc.)
- Close the host session

OHIO FEATURES INCLUDE

- No architectural limit to the number of sessions
- Event notification for host communications status and screen updates
- Comprehensive error trapping
- Generation of error message text
- Offset addressing for host presentation space
- Simplified model for presentation space
- Automatic generation of presentation space field list and attributes
- Keyword-based function key strings

Sample Server-Based Solutions That Can Be Created with PASSPORT Host Integration Objects

ELECTRONIC PAYMENT SYSTEM FOR CREDIT CARD PROVIDER

The PASSPORT solution can be used to create a Microsoft server-based COM module that streamlines the process of securing an approval or decline for payment transactions involving credit cards. In doing so, access to a commonly used host application can be made available by phone or web server, and the amount of time spent processing the financial transactions can be greatly reduced.

The mainframe application manages the availability of funds that are present to the account holder. New business logic can be created externally from the host application that allows a web site, telephone or hand held device to access the financial application, input the credit card type and credit card number, and begin the request for electronic payment.

The PASSPORT COM object takes this information, automatically logs onto the host system, navigates through the appropriate 3270 screens, copies the information into specific host fields and retrieves the available balance, which is presented back to the server application.

The application then compares the requested payment amount against the current balance and either approves or declines the credit card payment. If the request is declined, appropriate explanations are automatically provided.

The entire transaction runs in the background and involves virtually no intervention by the end user.

POINT OF SALE TERMINAL INTEGRATION FOR RETAILER

PASSPORT Host Integration Objects project can be used to replace outdated SNA terminal controllers with a new IP-based network that employs the use of state-of-the-art IBM 4690 POS equipment and PASSPORT HIO for 3270 application integration.

A terminal controller is used to connect multiple point-of-sale terminals to a host system, and you can find them in retail, convenience, grocery, food service and hospitality stores around the world.

Windows 2000 server can be used as an IP-based replacement for aging SNA terminal controllers. The solution is to install PASSPORT Host Integration Objects on a Microsoft Windows 2000 server, create a middleware application that manages communication with a 3270 inventory application, and ultimately display the appropriate information on the IBM 4690 point of sale terminals.

This middleware solution uses PASSPORT to launch the host session, navigate through the inventory application, parse the various 3270 screen buffers and pass the appropriate data to and from the POS terminal and the mainframe using the PASSPORT Host Integration Objects server.

This integration solution offers a very inexpensive approach, and is based on standard technology that can be easily managed throughout several hundred stores nationwide.

CUSTOMER SELF-SERVICE WEB APPLICATION

Self-service web applications can eliminate or reduce the need for customer service personnel, and in many cases, are a preferred service method for customers, especially if given 24x7 availability.

PASSPORT Host Integration Objects technology can be used by various industries and organizations – insurance companies, local governments, car rental agencies, mortgage brokers and others - to create self-service applications that are accessible via web, kiosk and wireless communication.

Rather than call or interact with customer service personnel to request information about their account, secure services, etc., customers can help themselves electronically. Instead of a service representative requesting the client's name, account number and other required information, the customer can directly access a CICS or IMS application using PASSPORT Host Integration Objects, and any UI or middleware software that may or may not be a component of the customer experience.

This inexpensive solution is desirable for banks, state and local governments, airlines, car rental agencies and others that can place self service equipment in secure locations under their control, or for organizations that offer secure wireless connectivity for employees and agents. With standard IP security, it can also be an inexpensive way to create self service web applications.

Sample Client-Based Solutions That Can Be Created with PASSPORT Host Integration Objects

APPLICATION INTEGRATION FOR INSURANCE AGENTS

Using our legacy integration solution, Zephyr can help insurance companies simplify access to data by sales personnel and insurance agents by integrating desktop and host applications.

When underwriting different types of insurance policies, most companies rely on multiple host applications, not to mention other client and server applications, to complete transactions.

In order to reduce the amount of training required for new agents and to increase the productivity of existing agents, a client application can be written, using PASSPORT Host Integration Objects, to integrate and consolidate various component applications.

By automatically establishing sessions, traversing host screens, automatically switching between host applications and simplifying the overall processes required with multiple document interfaces, a complex business transaction can be significantly streamlined.

This type of solution results in increased productibity, lower costs and satisfied customers.

SINGLE SIGN ON AND ELIMINATION OF DUPLICATE DATA ENTRY

It is not uncommon for call center personnel at national banks, utility companies, phone companies and government agencies to access multiple host databases and host applications in order to service clients, especially of the call center personnel supports multiple states and/or the customer base has been acquired through company mergers and acquisitions.

With PASSPORT Host Integration Objects, a single sign on solution can be created to launch multiple host applications (to multiple host systems), along with multiple client and server applications, based on the user ID and password.

This time saving attribute can be complimented with efforts to eliminate duplicate data entry. With PASSPORT Host Integration Objects, information can also be copied and automatically pasted into specific host input fields, even copied into input fields in multiple host applications. Navigation from one host screen to another can be controlled and modified by the client application.

Collectively, the end user is spared duplicate data entry – and potential data entry errors, but more importantly saves valuable time using the Zephyr host integration resources.

MIGRATION PATH FOR EXISTING HLLAPI APPLICATIONS

There are literally thousands of existing client and server applications in use today that connect to legacy applications by way of terminal emulation and an API known as HLLAPI (High Level Language Application Programming Interface). Most HLLAPI applications have been used for a decade or more. Their proven contribution makes them hard to displace, especially if they provide important business processes. Yet almost all can be migrated for use with PASSPORT Host Integration Objects, replacing the use of the traditional terminal emulator.

HLLAPI applications require a terminal emulation that's connected to a host session. The actual HLLAPI application runs as a separate process and communicates with the terminal emulation program, that in turn communicates with the host system. This approach to legacy access requires a large amount of PC memory and resources per host session, can be slow and has inherent scalability issues. HLLAPI applications can also be very expensive to maintain.

Migrating a HLLAPI application to PASSPORT Host Integration Objects removes all of these issues or significantly improves upon them. To ease the transition, Zephyr has sample source code, documentation and tools that can make the effort as painless as possible.

Containment Hierarchy

Sessions Contains a collection of Session objects.

Session A host session containing one Screen.

Screen Encapsulates the host presentation space containing both OIA and Fields.

OIA The operator information area of a host session.

Fields Contains a collection of Field objects.

Field A field in the presentation space. A field is the fundamental element of a virtual screen.

Note: "1" based counting is used for indexing (the first item in a collection is item 1, not item 0) and "0" based counting is used for positioning (the first position on the screen is position 0, not position 1).

SESSIONS OBJECT

A collection object consisting of sessions.

PROPERTIES

Count Returns the number of items in the collection of objects. Read-only.

SessionLimit Returns an integer representing the number of licenses. Read only.

Version Returns a string representing the current version. Read only.

ExpireDate Returns a string representing the date when the software will expire. Read only.

METHODS

AddSession Adds a session to the Sessions collection.

OpenSession Returns an existing session and adds it to the Sessions collection.

CloseSession Closes a session in the collection.

Item Returns an element in the collection.

UpdateLicense Updates the PASSPORT Host Integration license code.

SESSION OBJECT

Provides access to host data and PASSPORT HIO functionality. The Session object contains sub-objects, such as the Screen object and the Fields object.

PROPERTIES

ConfigurationResource The configuration resource for the Session object.

IsConnected Indicates whether the Session object is connected to a host. True means connected, false means not connected.

SessionName The session name for the Session object. The session name is unique among all instances of the Session object.

SessionType The session type for the Session object, 3270 or 5250.

Screen The Screen object for this session.

METHODS

Connect Starts the communications link to the host.

Disconnect Stops the communications link to the host.

EnableTrace Enables or disables data stream and/or low level trace for the session.

EVENTS

OnSessionChanged This event will be generated whenever the session state changes.

OnScreenChanged This event is generated whenever the virtual screen is modified.

OnOIAChanged This event will be generated when anything on the Operator Information Area (OIA) changes.

SCREEN OBJECT

Provides access to the contents of the host screen's presentation space.

PROPERTIES

Cursor The location of the cursor in the presentation space.

OIA The OIA object associated with this presentation space. This object can be used to query the status of the operator information area.

Fields The Fields object associated with this presentation space. This provides another way to access the data in the virtual screen.

Rows Returns the number of rows in the presentation space. Read-only.

Columns Returns number of columns in the presentation space. Read-only.

String The entire text plane of the virtual screen as a string. All null characters and Field Attribute characters are returned as blank space characters.

METHODS

FindString Searches the text plane for the target string.

GetData Returns a character array containing the data from the Text, Color, Field or Extended plane of the virtual screen.

PutString Sends a string to the virtual screen at the specified location.

SendKeys Sends a string of keys to the virtual screen.

SendAid Sends an "aid" keystroke to the virtual screen.

FIELDS OBJECT

Contains a collection of the fields in the virtual screen.

PROPERTIES

Count Returns the number of Field objects contained in the collection.

METHODS

Item	Returns Field object at the given index.
Refresh	Updates the collection of Field objects.
FindByString	Searches the collection for the target string and returns the Field object containing that string.
FindByPosition	Searches the collection for the target position and returns the OhioField object containing that position.

FIELD OBJECT

A Field is the fundamental element of a virtual screen. A field includes both data and attributes describing the field. The OhioField class encapsulates a virtual screen field and provides methods for accessing and manipulating field attributes and data. Field objects can be accessed only through the Fields object.

PROPERTIES

Start	The starting position of the field.
End	The ending position of the field.
Length	The length of the field.
Attribute	The attribute byte for the field.
Modified	Indicates whether or not the field has been modified (TRUE or FALSE).
Protected	Indicates whether or not the field is protected (TRUE or FALSE).
Numeric	Indicates whether or not the field is numeric (TRUE or FALSE).
HighIntensity	Indicates whether or not the field is high-intensity (TRUE or FALSE).
PenSelectable	Indicates whether or not the field is pen-selectable (TRUE or FALSE).
Hidden	Indicates whether or not the field is hidden (TRUE or FALSE)..
String	The text plane data for the field.

METHODS

GetData	Returns data from the different planes (text, color, extended) associated with the field. The data is returned as a character array

OIA OBJECT

The operator information area of a host session. This area is used to provide status information regarding the state of the host session and location of the cursor. An OIA object can be obtained using the OIA property on an instance of the Screen object.

PROPERTIES

Alphanumeric	Indicates whether the field which contains the cursor is an alphanumeric field (TRUE or FALSE).
InputInhibited	Indicates whether or not input is inhibited. If input is inhibited, SendKeys or SendAID calls to the Screen object are not allowed.
Numeric	Indicates whether the field which contains the cursor is a numeric-only field. (TRUE or FALSE).
Owner	Indicates the owner of the host connection.
String	The entire text plane of the selected object as a string.

Zephyr specializes in TN3270, TN5250 and VT100/VT220 host access solutions for Microsoft Windows. Using our PASSPORT ActiveX or.NET legacy integration, PC or browser-based terminal emulation, client or server legacy extension, Zephyr lets you connect, integrate and web enable valued legacy applications.

Our impressive client list includes many notable U.S. and international firms and organizations, including Canon, Cessna, Huntington Banks, Jefferson Pilot Financial, Liberty Mutual, Merrill Lynch, Nationwide Insurance, Otis Elevators, Rite Aid, Saks, State of California, United Space Alliance (NASA) and many others.

DOWNLOAD A FREE TRIAL COPY

The opinion we value the most is yours. Download a free, fully-functional 30-day trial copy of the software at:

http://www.zephyrcorp.com/downloads/

ZEPHYR®

8 E GREENWAY PLAZA
SUITE 1414
HOUSTON, TX 77046-0801

800-966-3270
PHONE: 713-623-0089
FAX: 713-623-0091

info@zephyrcorp.com

IN THE UK/EMEAI, CONTACT
ZEPHYR EMEAI

6 ROSE COURT
OLNEY
BUCKINGHAMSHIRE
MK46 4BY
UNITED KINGDOM

PHONE: 44 (0) 1234-710040
FAX: 44 (0) 1234-711547

zephyr@integranet.co.uk