IBM

March 2004

Terminal Termination Technologies

Timothy F. Sipples
Senior Software IT Architect – Enterprise Modernization
timothy.sipples@us.ibm.com

Introduction

The mainframe celebrates its 40th birthday on April 7, 2004. No other computing platform approaches the IBM mainframe's continuous support for 40 years of business applications. Businesses and governments around the world have billions of dollars invested in mainframe applications – applications which are vital to their existence.

Decades-old applications continue to run without alteration on state-of-the-art zSeries mainframes alongside just-written Linux and Java applications. That unique capability saves businesses and governments countless dollars, euros, and yen. Writing a new business application, and putting it into production, is one of the most costly technology endeavors. The **durability** of IBM's solutions helps businesses and governments avoid these unnecessary costs, making IBM the industry's most valuable solution partner.

Yet while there's little reason to rewrite business rules that haven't changed in 30 years – debits must still equal credits – there's ample reason to apply these business applications to new areas in new ways. Examples include **Web self-service** and "just in time" inventory management. IBM offers tools that **extend and leverage** existing applications, to respond to new business challenges quickly and cost-effectively. This paper examines **host integration** for enterprise modernization.

The "Terminal": A Brief History

Most mainframe and iSeries (AS/400) applications work with terminals such as the classic model from the mid-1970s pictured at right with its green phosphorescent display. That's why such applications are often called "green screen" applications (or "legacy" applications). The business value of these still-running applications is enormous.

By the mid-1980s the popularity of dedicated terminals waned. Soon after the IBM Personal Computer debuted, IBM and other companies produced **terminal emulation** software, to provide the same experience as a dedicated 3270 or 5250 terminal. Healthy demand for such software continues today, with IBM's Personal Communications the most popular emulation software.

In 1984 IBM introduced HLLAPI, the High Level Language Application Programming Interface. HLLAPI and its successor, **EHLLAPI**, provide PC-based applications the ability to interact with terminal-based applications. Programmers use EHLLAPI functions such as "Send Key" and "Copy Field to String" in order to "drive" a host application, usually without displaying a terminal screen. This technique is often called **screen scraping**. Many IBM customers built client/server applications using EHLLAPI, and many remain in production. IBM still supports EHLLAPI in many of its host access software products, including Personal Communications and Host On-Demand.

The 1990s saw the emergence of the Web and a trend toward "thin" clients running browsers. That trend continues due to the high cost of maintaining and supporting "fat" clients with lots of software installed at the desktop. IBM introduced **Host On-Demand** in the mid-1990s to help businesses achieve these cost savings. Host On-Demand provides terminal emulation (and file transfer) in a browser using a Java applet. Users simply click on a Web link within their browser to get capabilities similar to Personal Communications but without installing software. HOD works with many different Web browsers, including browsers running on Linux and the Macintosh. Now in its eighth major release, WebSphere Host On-Demand is a popular initial step toward e-business – a step all businesses with "fat" terminal emulators should consider.

Host On-Demand introduced a new programming interface called **HACL** (Host Access Class Libraries). HACL better addresses the needs of Java programmers, with an object-oriented design (Java beans) and improved access to non-visual aspects of terminal applications such as color attributes. These innovations help programmers work more reliably with host screens using modern development tools and practices, including WebSphere Studio and Rational tools.

In the late 1990s, IBM added stronger security features to its terminal-oriented software products. Many organizations face regulatory and audit requirements for security. IBM zSeries and iSeries servers include built-in certificate- and key-based **encryption** and **authentication** features, such as SSL, that work together with IBM host access software to keep critical applications secure.

Terminal-Level Programming

Screen scraping acquired a bad reputation because of this sequence of events:

1. Programmers embedded screen-driving code in major parts of their applications, intertwined with new business logic.

2. Support staff then installed these "finished" client/server applications on every PC.

3. Mainframe programmers changed a screen, perhaps by adding an input field to an order entry screen, to satisfy business needs.

4. The just-finished client/server application failed, triggering a costly and disruptive return to step one.

IT managers reacted to this architectural problem in a variety of ways. Some embarked on costly rewrites of their host-based applications, bringing them to new platforms. (Many failed.) Others ended their client/server application projects and returned to green screen access using terminal emulation software. Some found alternative ways to leverage and extend their host-based application investments, as discussed later in this document. The remainder put up with the shortcomings and kept their client/server applications.

A clear and demonstrable Web Services strategy is a crucial factor for vendor viability.

Today, through a combination of technical progress and IBM's focused efforts to advance beyond screen scraping, terminal-level access to host applications is now a viable and appropriate methodology. In November, 2003, Giga Group recognized this progress in its *IT Trends 2004* report on Legacy Modernization:

> ### Web-to-host continues to gain ground within IT as a viable, longer-term technology option
> *Business pressures to Web-enable applications force skeptics to accept Web-to-host as a viable and financially compelling (if tactical) business solution. Giga reiterates its position that the Web-to-host market is currently less than 15 percent to 20 percent of its eventual size.*
>
> ### Web Service-to-host evolves from Web-to-host and accelerates vendor consolidation
> *Long the object of scorn by its EAI/MOM/other integration siblings, Web Service-to-host finally earns some respect because of a compelling three-point capability - the ability to: (1) leverage existing applications via the presentation layer with standards-based (SOAP/WSDL) access, (2) deliver business value first (wrapped service to front-end application) and (3) insert a standards-based layer of abstraction that allows back-end application changes without upsetting the front end (wrapped service). The layer of abstraction helps organizations "migrate while they operate," replacing wrapped services with new technology gradually, at a manageable pace in lieu of big bang migrations. The small subset of Web-to-host vendors that also offer Web Service-to-host tools become the preferred choice of customers looking for integration tooling that solves a wide variety of integration problems. This buying preference adds to the pressures already on the 40 or more Web-to-host vendors - the current economic conditions, overlapping/converging capabilities, consolidation not only within Web-to-host but also across integration, EAI and Web Services vendors. The result will be accelerated consolidation among vendors. A clear and demonstrable Web Services strategy is a crucial factor for vendor viability.*

Beyond Screen Scraping

As Giga points out, there are several factors at work conspiring to promote "presentation layer" integration of host applications:

1. **Ever-increasing cost pressures**. Businesses are under unprecedented pressure to reduce costs, and only changing and improving core business processes will result in substantial savings. (Few businesses have "easy" savings left to make.) That often means Web self-service projects get high priority, for example.

2. **Maturity of Web architectures**. Web application design finally incorporates essential features familiar to mainframe application architects, such as server-based business logic (avoiding costly and risky software distribution to thousands of desktops), a "Model-View-Controller" paradigm (separating presentation logic from business logic), scalability, and strong security. Now businesses have the mature tools and technologies to extend vital line-of-business applications to the Web.

3. **Deceleration of host screen changes**. Terminal-based applications are typically more stable, while application designers deploy more new business logic to platforms such as J2EE (Java 2 Enterprise Edition).

4. **Better host access software**. IBM's WebSphere Host Integration software, for example, allows developers to package specific green screen application interactions as reusable Web Services. Mainframe and iSeries developers with no previous Java experience can use these tools thanks to built-in code generators and wizards.

5. **Other techniques tried and failed**. Terminal-level access is universal: it works with any host application with screens, even "ugly" screens. If the function or feature exists in the terminal presentation, WebSphere Host Integration can leverage it. Some mainframe and iSeries applications are simply unapproachable (from an application integration viewpoint) any other way. Sometimes a green screen application's complex business logic, formed over years of experience, has no other access point.

6. **Time, budget, and skills**. WebSphere Host Integration is among the most quickly implemented and lowest cost application integration solutions, easing the J2EE learning curve. Yet if wisely deployed with an eye toward a Services-Oriented Architecture (SOA), WebSphere Host Integration helps "get the plumbing right" so businesses "migrate while they operate." Too many superb application architectures never become reality because they lack transition strategies or take too long to deploy.

7. **Continuing need for "dual access."** Businesses and governments discovered that many users are more productive with green screen interfaces for select tasks.

Instant e-business

WebSphere Host Integration, specifically WebSphere **Host Access Transformation Services (HATS),** has a rare (if not unique) capability to automatically, on-the-fly, convert 3270 and 5250 terminal data into Web pages (HTML). HATS has a rules-based engine that performs this conversion without any preparation of the host applications. Web designers specify how host fields should look, whether function key menus should appear as clickable web links, what company logo should appear on every page, etc. When used for conversion, HATS tolerates screen changes without business interruption.

HATS means that any 3270 or 5250 green screen application can become a browser-based application instantly.[1] Few developers will stop there. HATS can also package any green screen application interaction, e.g. entering a shipment number to retrieve an order status, as reusable Web Services.

Non-IBM Terminal Protocols

IBM's competitors designed host computers and terminals that utilize different protocols. Green screen applications written for UNIX, Linux, and VMS operating systems use the DEC VT standard. Unisys and Tandem (now HP NonStop) use other terminal protocols. IBM offers software to reach these terminal-based systems, to leverage and extend those applications as well.

There's renewed demand for VT terminal emulation software among Linux system administrators given the growth of that operating system. WebSphere Host On-Demand fits their needs well.

Sample Architecture

Since host integration software strategies inherently depend on host-based applications, there is a convincing argument that the host integration software itself should run on the mainframe or iSeries servers. Co-location of integration software and green screen applications greatly simplifies the reliability engineering and availability, with no mid-tier servers and network connections to worry about. ("If the mainframe is running, everybody is happy. If the mainframe is down – is it ever down? – everybody is screaming.") Co-location also encourages mainframe and iSeries programming and support staff to take ownership of the host integration services – still not a bad idea when they're the staff changing screens.
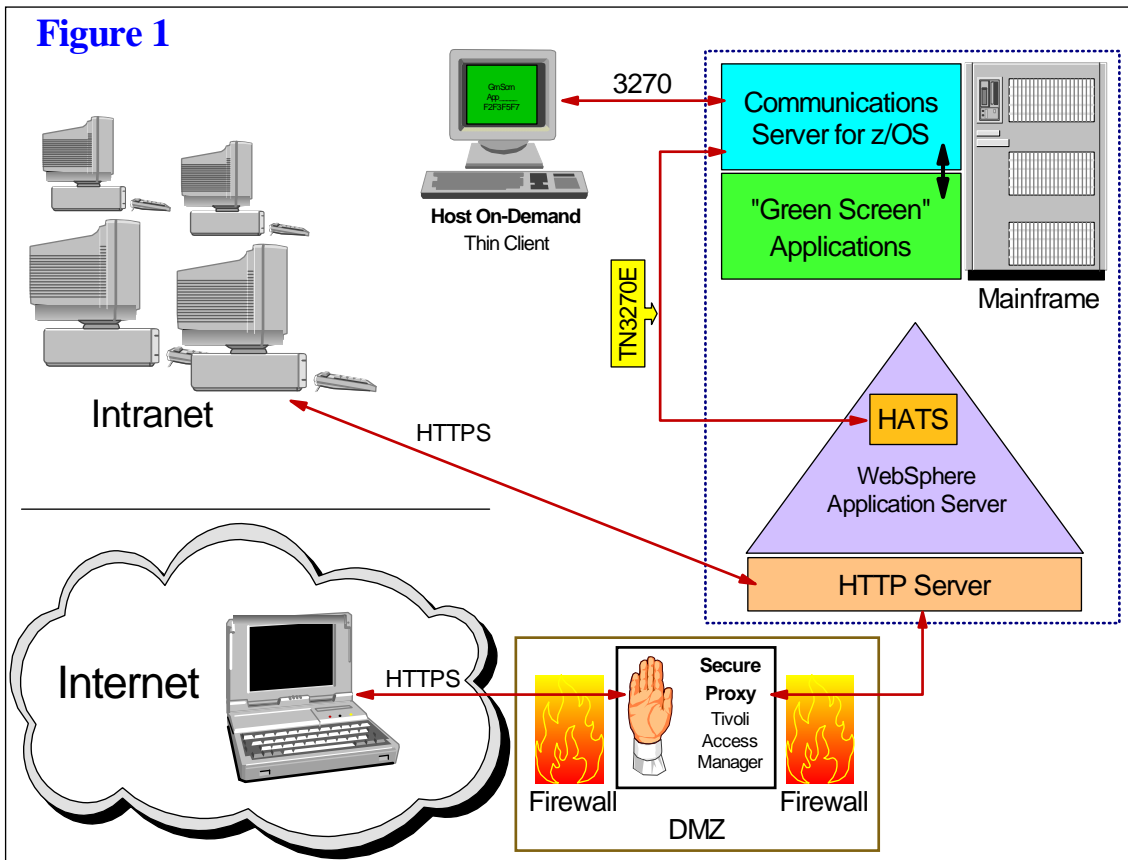
This simplified architecture diagram (Figure 1) depicts three classes of users: Internet-connected Web browser users, internal network (intranet) Web browser users, and intranet Host On-Demand (Java-enabled Web browser) users. Other users can participate in this same infrastructure. For example, WebSphere Voice Server and telephone switches provide access, via HATS, to select green screen application functions through wireless and wireline telephones. Other applications, such as business-to-business (B2B) applications,

---

[1]In addition, recent versions of OS/400 include **HATS Limited Edition** and **iSeries Access for Web**. Both products also provide instant conversion of 5250 (only) into HTML.

participate via the Web Services interfaces in WebSphere Application Server and WebSphere HATS. Still other users might access host applications via the HATS and HOD portlets for WebSphere Portal.

Note that HATS can tap into more than one mainframe, iSeries, or other host system. This capability helps businesses accelerate the process of reengineering workflow, to better reflect current and future business needs that may require multiple backend applications. HATS can also "ration" a finite number of host connections in a **connection pool**, queuing transactions through a regulated number of interactive sessions. Limited iSeries interactive session licensing might argue for using a connection pool.

Figure 1

3270

Communications Server for z/OS

"Green Screen" Applications

Mainframe

Host On-Demand
Thin Client

TN3270E

Intranet

HTTPS

HATS

WebSphere Application Server

HTTP Server

Internet

HTTPS

Secure Proxy
Tivoli Access Manager

Firewall   Firewall

DMZ

Security and Performance

HATS must respect all existing security systems, including RACF and any application-level controls. As shown in Figure 1, HATS never permits direct terminal-level access to host systems. Web-based controls and HATS application design may add extra security protections.

While HATS probably would not be the best vehicle to retrieve everything stored in a large mainframe database within five minutes, performance is better than ever thanks to some recent innovations. IBM worked through the IETF (Internet Engineering Task Force) RFC (Request for Comments) standardization process to enhance the TN3270E specification to add a feature called **contention resolution**. Then IBM implemented this revision of the

TN3270E specification in both z/OS and HATS. With contention resolution HATS no longer must wait a fixed period of time for each screen to "settle." This enhancement has a dramatic effect on throughput. At this writing, only IBM's products implement the new TN3270E standard with contention resolution.

Other Host Integration Strategies

While it is true that HATS is a "universal adapter" – any host application with green screens qualifies – there are numerous other options for host integration. Please note that most enterprises will choose more than one approach as they leverage their investments in host-based applications. Major categories include:
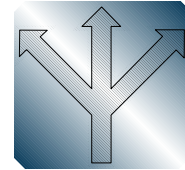
1. **Messaging** (WebSphere MQ, JMS). Many organizations wish to standardize on a common messaging infrastructure and are working to enable as many applications as possible, including their host-based applications, for WebSphere MQ and JMS (Java Message Service). An enterprise messaging infrastructure has attractive inherent capabilities, such as **assured delivery** once and only once across a range of systems and networks.

   Whether this approach makes sense will largely depend on the degree of effort required to enable existing host applications (by partial recoding). If that effort is too costly or difficult, HATS still offers "last mile" connections to unmodified host applications so they can participate in an enterprise messaging system.

2. **Published interfaces** (e.g. JCA adapters). Certain host-based applications and middleware, such as CICS, offer published APIs (Application Programming Interfaces) of various kinds, often accessible through WebSphere Business Integration adapters or other connectors, such as the CICS Transaction Gateway. Well-structured host applications lend themselves to integration using these interfaces.

   A common strategy is to use terminal-level integration for less well-structured host applications (and for applications with a heavy investment in presentation-oriented business logic such as input validation) while incorporating more middleware API-level integration as the host applications evolve.

3. **Data-level integration** (JDBC, data federation). If most of the value in the host-based applications rests in the stored data rather than in business logic, then data-level integration makes sense. Tools such as the DB2 Information Integrator products combine multiple relational and nonrelational databases into a single **federated** view, more easily accessible for new Web applications. These software tools embrace both IBM and non-IBM databases, including DB2, IMS, VSAM, Informix, Datacom, Adabas, Oracle, Sybase, Microsoft SQL Server, and others.

   Application architects using these mechanisms should plan carefully if host-based applications will continue to access and update the same data, anticipating all scenarios in which each application accesses the others' data.

4. **iSeries features** (WebFacing, iSeries Toolbox for Java). OS/400 offers unique and potentially attractive integration features. For example, **WebFacing** provides similar capabilities to HATS's automatic default rendering but without consuming interactive sessions on many iSeries models. WebFacing currently has some limitations, notably: (a) end users must have Microsoft Internet Explorer; (b) DDS (Data Description Specifications) source files must be available for conversion; (c) no provision for building Web Services. However, if those limitations are acceptable, WebFacing provides a low cost mechanism for extending iSeries applications to the Web. The **iSeries Toolbox for Java** offers a collection of powerful interfaces to access iSeries databases, execute RPG and other applications, and collect system information.

5. **Hybrid applications**. New mainframe and iSeries application development tools from IBM's WebSphere Studio and Rational product lines allow programmers to write code in languages they prefer (COBOL, RPG, etc.) that invokes business logic written in Java, and vice versa. This approach helps programmers maintain and add value to existing applications using Java when appropriate. For example, WebSphere Application Server for z/OS and WebSphere Studio Enterprise Developer introduced a preview technology in early 2004 that wraps COBOL programs as session EJBs (Enterprise Java Beans) for access by other EJBs.

To reiterate, the HATS "universal green adapter" often serves well combined with other integration strategies, such as WebSphere Business Integration and WebSphere MQ.

Summary

Given recent innovations, terminal-level integration of host applications is compelling. Organizations with investments in line-of-business host applications should consider these solutions, particularly in the context of a business-friendly "migrating while operating" drive toward modern Web Services and Services-Oriented Architectures (SOAs).

Resources

IBM WebSphere Host Integration   http://www.ibm.com/software/webservers/hostintegration
Host Access Transformation Services   http://www.ibm.com/software/webservers/hats
Giga Information Group (unit of Forrester Research)   http://www.gigaweb.com
COBOL as EJBs   http://www.ibm.com/software/awdtools/cobol/zos
IBM DB2 Information Integration   http://www.ibm.com/software/data/integration
IBM WebSphere Studio   http://www.ibm.com/websphere/studio
IBM Rational   http://www.ibm.com/software/rational
HATS Limited Edition   http://www-1.ibm.com/servers/eserver/iseries/access/hatsle
IBM iSeries Access for Web   http://www.ibm.com/servers/eserver/iseries/access/web
iSeries WebFacing   http://www.ibm.com/software/awdtools/wdt400/about/webfacing.html
IBM iSeries Toolbox for Java   http://www.ibm.com/servers/eserver/iseries/toolbox
IBM WebSphere Studio Asset Analyzer   http://www.ibm.com/software/awdtools/wsaa