



Legacy Application Modeling With Attachmate Verastream Host Integrator

Programmatic Integration vs. Traditional Screen Scraping

CONTENTS

A New Approach to Mainframe Integration	1
The Evolution of Screen Scraping	1
Not an Elegant Solution	2
Programmatic Integration with Verastream: Next-Generation Screen Scraping	2
The Verastream Advantage	3
About Attachmate	3

Legacy Application Modeling With Attachmate Verastream Host Integrator

Programmatic Integration vs. Traditional Screen Scraping

As IT professionals know, accessing mainframe applications for integration with other systems can be a slow, difficult job. That's because mainframe applications rarely provide a clear separation between logic and data. In fact, business logic, presentation logic, and a variety of data are often tightly entwined.

Furthermore, while most modern packaged applications offer well-defined interfaces that expose their data as COM, .NET, or web services, few mainframe applications do. For applications without a well-defined interface, access can be limited to these methods:

- Direct access to IMS transactions.
- Direct access to DPL-compliant CICS applications through COMMAREA.
- Direct data access.

When an application is not using an IMS or DPL-compliant CICS transaction platform, or when direct-data access is not feasible, only one point of entry remains: presentation-level integration using the terminal screen.

The traditional process for extracting data from screen-based mainframe applications is commonly known as screen scraping. Over the years, screen scraping has proved useful for simple task automation and limited application rejuvenation. However, early methods were quickly pushed beyond their limits, earning screen scraping a reputation for being slow, unreliable, and difficult to maintain.

A New Approach to Mainframe Integration

This paper examines the history and limits of traditional screen scraping and introduces programmatic integration modeling with Attachmate® Verastream® Host Integrator. Composed of server software and development tools, Verastream Host Integrator takes a fundamentally new approach to integrating terminal-bound mainframe applications.

For organizations planning to reuse legacy mainframe logic and data in web, portal, CRM, mobile, call-center, web self-service, or other new business initiatives, this paper is especially relevant. Verastream Host Integrator makes it possible to develop new, fully integrated applications – very quickly – while allowing

the mainframe interfaces to be changed at any time without affecting the new application.

The Evolution of Screen Scraping

Screen scraping emerged when terminal emulation software introduced application programming interfaces that allowed programmatic access to data and functions presented on mainframe terminal screens.

The most widely used API was defined by IBM for OS/390 system applications accessed via the 3270 data stream. It was called High Level Language Application Programming Interface, or HLLAPI. This was later extended to include IBM OS/400 systems accessed via the 5250 data stream and renamed Extended High Level Language Application Programming Interface, or EHLLAPI. (Other APIs provide access to applications running on UNIX, HP e3000, and HP OpenVMS systems.)

These APIs enabled a software program on a PC to mimic the steps taken by a human operator interacting with mainframe application screens; the program could send keystrokes and read the data stream to navigate from screen to screen, entering and extracting screen information.

Initially, screen-scraping APIs were used for simple task automation. They could allow repetitive actions, such as system logons or application traversal, to be reduced to a single keystroke or command line entry.

As GUI operating environments such as Microsoft® Windows® proliferated, users became dissatisfied with the arcane look and feel of “green screen” mainframe applications. That dissatisfaction was addressed by rejuvenation technologies, which used screen scraping to manipulate the legacy mainframe application in the background while presenting a more familiar GUI at the desktop. Rejuvenation took two forms:

- **Auto rejuvenation** - direct screen-by-screen translation of the mainframe application without changing the mainframe-application workflow at the desktop.
- **Custom rejuvenation** - minor changes made to the workflow at the desktop (such as combining information from multiple mainframe screens), without changing the original mainframe application.

Not an Elegant Solution

Screen scraping was an enticing prospect, for many valid reasons. IT professionals knew the difficulty of replicating mainframe-system reliability and scalability. They also knew the high cost of rewriting legacy mainframe applications. (In fact, it was often impossible to do this because the knowledge and expertise associated with the original application were no longer available.) And since many legacy mainframe applications didn't separate presentation from business logic, there weren't a lot of viable choices for extending and integrating these applications.

So the promise of a technology that could unlock data in legacy mainframes, without the need to replace stable systems or rewrite mainframe applications, struck a chord with IT organizations. But screen scraping raised hopes only to leave disappointment in its wake. A significant drawback of traditional screen scraping is its reliance on a direct, interactive, and synchronous connection between the visible GUI client application at the desktop and the hidden mainframe application. This relationship sets several key problems into play:

- **Performance.** The complete data stream representation of each mainframe terminal screen must flow to the client before it can be recognized and processed. This involves a round trip from the client to the mainframe and back again, for each screen. Consider, for example, navigating five or six mainframe screens in the background, at the rate of two or three seconds per screen; the user is left to wait and wonder when the response will be presented. In the meantime, if a lot of users are doing screen scraping, imagine the potential for dragging the entire network down. As these performance issues became more and more apparent, screen scraping developed a reputation for sluggishness.
- **Synchronization.** Even more problematic is maintaining the synchronization between the GUI client and the mainframe application. When the client application sends keystrokes or data to the mainframe, the mainframe application must be at the screen that knows how to respond correctly. But the mainframe application behaves independently and is unaware of the client application. It may get disconnected, receive an unexpected error message, or follow a path that the client application didn't expect.

Any of these conditions could cause the client and mainframe applications to get out of sync and produce an error condition. Error exception handling can be addressed only on the client-application side and is usually the most labor-intensive part of

a screen-scraping development project. It should come as no surprise then, that poor handling of synchronization failures is the greatest threat to the reliability of traditional screen-scraping applications.

- **Change control.** Another shortcoming of traditional screen scraping is the way it handles application changes. Because the code controlling the interaction between the client and mainframe applications is embedded in the client application, all but the simplest changes to mainframe screens will cause the client application to fail. That means the client application has to be rewritten and redeployed. Embedding the mainframe manipulation code in the client application also requires that the client-side developer have intimate knowledge of the operational aspects of the mainframe application, which may make the client application more difficult to maintain in the future.

Programmatic Integration with Verastream: Next-Generation Screen Scraping

Verastream Host Integrator extends the value of legacy mainframe systems, while overcoming the limitations of traditional screen scraping. Verastream Host Integrator doesn't use an API like EHLLAPI to embed legacy mainframe interactions directly in a client application. Instead, it employs a graphical design environment where expert mainframe-application users create models that define how to navigate mainframe applications and access critical information. These models encapsulate mainframe interactions into reusable web services (J2EE, .NET, XML), JavaBeans, EJB, .NET components, or COM objects that are deployed to the Verastream Host Integrator server, where they can be accessed by any client application.

Verastream Host Integrator legacy application modeling and programmatic integration effectively eliminate the direct, synchronous reliance between client and legacy mainframe applications. This functionality resolves the limitations of traditional screen scraping across all the above problem areas:

- **Performance.** Verastream Host Integrator models can execute multiple operations based on a single call from a client application. Since the models reside on the Verastream Host Integrator server rather than on the client desktop, round trips between the client and mainframe applications are significantly reduced. Take the creation of a new customer record, for example: While this task might require traversing a number of mainframe application screens, the Verastream Host Integrator server handles all the background complexity; the

client application needs to make only one high-level function call rather than performing low-level mainframe screen navigation.

The Verastream Host Integrator server can manage almost any number of mainframe-session connections. It can be configured to share mainframe sessions among many users, ensuring low latency by using session pools. It can also be configured to dedicate sessions to particular users when per-user security is required. Models can be deployed across multiple Verastream Host Integrator servers for load balancing and failover protection, so thousands of concurrent users can be supported.

- **Synchronization.** The Verastream Host Integrator approach completely decouples the client and mainframe applications. Client applications can issue multiple simultaneous requests to the Verastream Host Integrator server, which accesses multiple mainframe applications; the Verastream Host Integrator server manages the mainframe communications and returns the results as soon as they are available.

With the Verastream Host Integrator transactional-component process, client applications are concerned only with specific, defined paths through the mainframe application, instead of random user selections. This means that all possible navigation paths to required mainframe screens can be accounted for, while error-exception handling and recovery can be designed directly into Verastream Host Integrator mainframe models. Client and mainframe applications can operate autonomously, with predictable and reliable results, while laborious exception-handling efforts are dramatically reduced.

- **Change control.** Verastream Host Integrator offers superior change control management along with unprecedented flexibility to support future system development. A multi-tier architecture allows client applications to be developed with complete independence from back-end legacy systems. Access to legacy data and business logic is gained through function calls or SQL queries that need no awareness of the mainframe system. This level of abstraction means that client application developers

can create new business systems without any mainframe application knowledge or expertise.

Changes to mainframe applications should require no reciprocal modification to client applications. Because Verastream Host Integrator models address discrete functions, mainframe application changes can be addressed easily, with updates to only specific model components. The models can be reused to speed the development of new client/server and web applications, with the assurance that applications deployed today will not be disrupted by future re-engineering or replacement of legacy mainframe systems.

Verastream Host Integrator supports the full range of legacy hosts, including IBM zSeries (S/390), IBM iSeries (AS/400), Unix, OpenVMS, and HP e3000 systems. Verastream Host Integrator interfaces enable client interaction with Verastream Host Integrator models. The interfaces support Java, JDBC, COM, .NET, SQL, Microsoft BizTalk Server 2004 and Microsoft BizTalk Server 2006.

The Verastream Advantage

Verastream Host Integrator is part of the Verastream integration suite, which provides a complete range of mainframe, web, and desktop integration tools that speed the reuse of legacy applications in service-oriented architectures. Verastream Host Integrator delivers rapid results because you can use existing development skills, familiar IT tools, and proven mainframe investments. With Verastream Host Integrator, you can give your users a new look and feel, without disturbing mainframe-application code or associated business processes.

About Attachmate

Attachmate helps businesses extend, manage, and secure their IT investments. We offer a broad range of solutions—from terminal emulation, legacy integration, and PC lifecycle management products to innovative systems and security management tools. With our technology, more than 40,000 businesses worldwide are putting their IT assets to work in new and meaningful ways. Learn more at www.attachmate.com.



Corporate Headquarters
1500 Dexter Avenue North
Seattle, Washington 98109
TEL 206 217 7500
800 872 2829
FAX 206 217 7515

EMEA Headquarters
The Netherlands
TEL +31 71 368 1100
FAX +31 71 368 1181

Asia Pacific Headquarters
Australia
TEL +61 3 9825 2300
FAX +61 3 9825 2399

Latin America Headquarters
Mexico
TEL +52 55 9178 4970
FAX +52 55 5540 4886

WEB attachmate.com
E-MAIL info@attachmate.com

For regional office information, visit www.attachmate.com.