WHITE PAPER

Attachmate

Accelerating Integration with Verastream Host Integrator

High-level abstraction is the key

CONTENTS

Accelerating Integration with Verastream Host Integrator

High-level abstraction is the key

Time is money, and today's IT organizations have less of both. Even so, the explosion of customer-facing applications and the high expectations of technically savvy customers leave you no choice. You must give partners and customers instant access to high-demand information and services. And you must somehow do so with the existing skills of your dwindling staffs.

There's also a significant technical challenge: The majority of business logic and data to support new customer-facing applications still resides in host applications. And because most host applications do not provide a clear separation of logic and data, accessing either for integration purposes has traditionally been a slow, difficult job.

Today, Attachmate® Verastream® Host Integrator offers a fundamentally new way to integrate mainframe applications. In short, Verastream Host Integrator transforms unstructured business logic and data into services that can be accessed by application developers in a structured way. This paper provides a technical overview of Verastream Host Integrator's high-level abstraction approach, including how it uses tables and procedures to accelerate the integration process.

Host Applications: The Nature of the Beast

Programmatic access of host logic and data through the terminal interface typically begins with a modeling process. During this process, individuals familiar with the host application select screens, define data inputs and outputs, and record user interactions. The result of the process is a service that encapsulates host functionality. The service is deployed to a middle-tier server and accessed through an application programming interface (API).

Manipulating host applications through an API can be a complex undertaking. As Figure 1 shows, a typical host application screen may contain several input fields, data record sets that scroll across multiple screens, and a range of navigation options. What's more, completing a host transaction usually involves interacting with a series of application screens.

Most host integration products provide APIs for addressing the application at a relatively low level of abstraction. As a result, the application developer needs to have a detailed understanding of the host application in order to navigate application screens, enter data, and extract data.

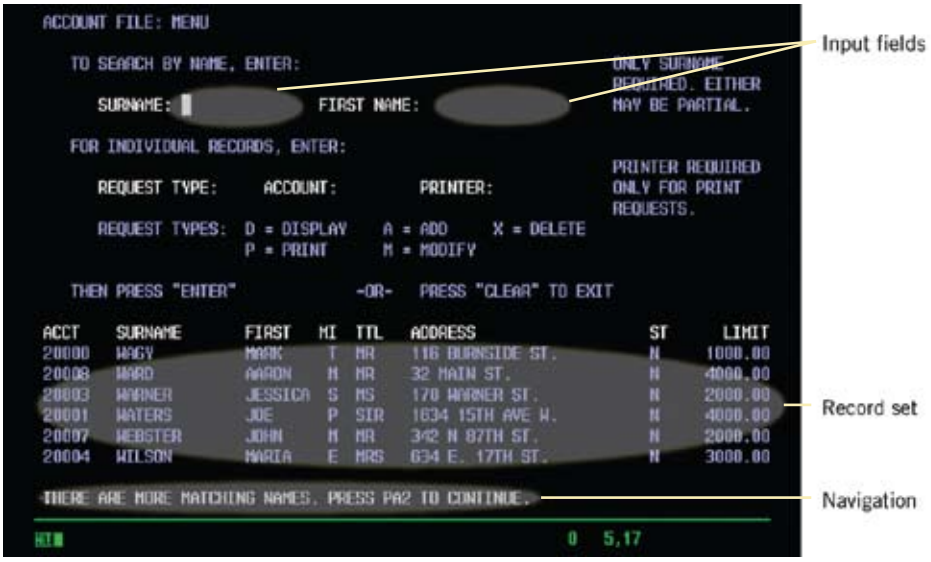

Figure 1: Host Application Screen

Verastream Host Integrator, on the other hand, combines a rich set of Java and COM APIs with added levels of abstraction that completely insulate application developers from the complexity of the underlying host applications. The hierarchy of abstraction in a Verastream Host Integrator service is shown in Figure 2. Verastream Host Integrator APIs provide access at each abstraction level.

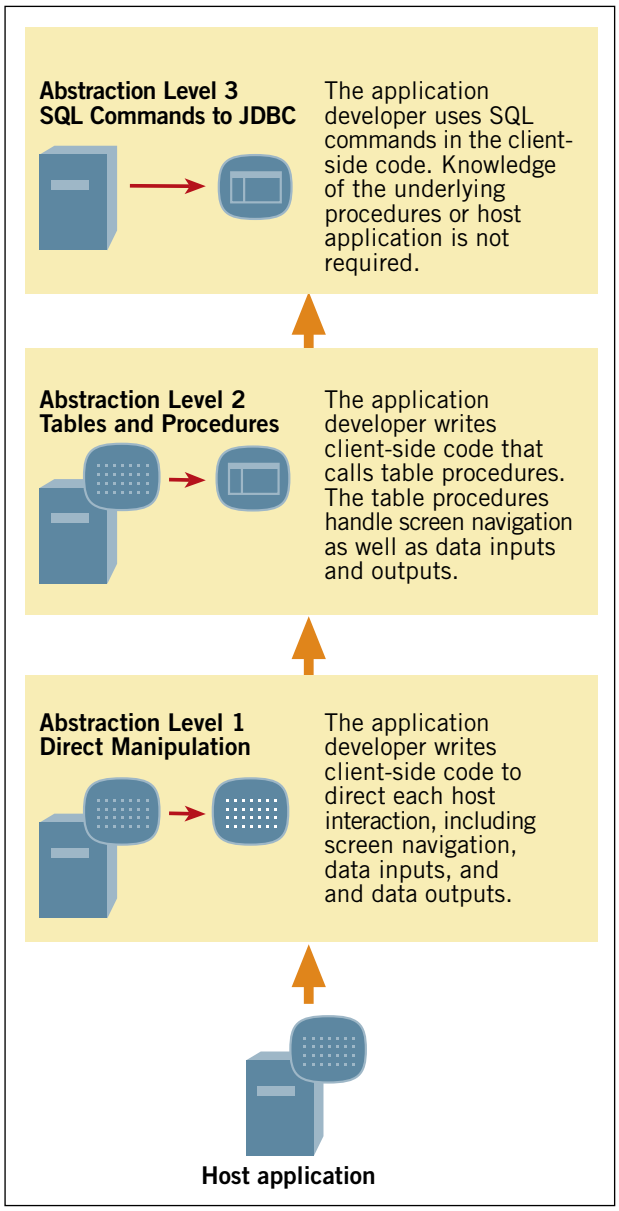

Figure 2: Verastream Host Integrator Abstraction Layers

During the Verastream modeling process, host application experts define tables by creating a list of database columns that name the data elements. Within the tables, procedures map data inputs and outputs from any host application screen and automatically manage all navigation and transaction details. Procedures—including *select*, *update*, *insert*, and *delete*—can be combined to perform a complex series of transactions.

Using the Verastream Host Integrator tables and procedures, application developers can then work with unstructured host applications just as they would with any standard database. The Verastream Host Integrator tables and procedures dialog is displayed in Figure 3.

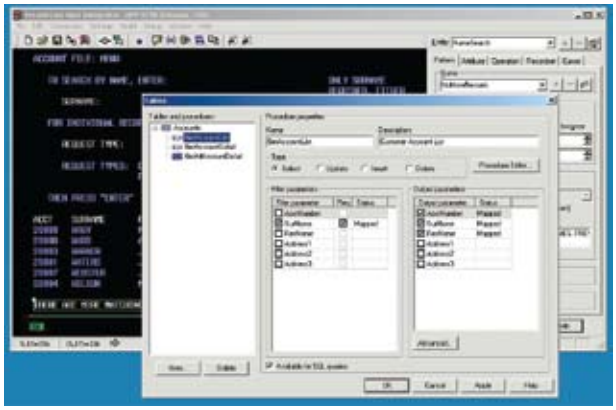

Figure 3: Verastream Host Integrator Tables and Procedures

An Application Programming Example

The value of using high-level abstraction to streamline integration can be illustrated with a simple programming example. In our example, a host application user wants to retrieve information about all customers whose last names begin with “w.”

When a non-integrated host application is involved, this transaction begins with the user entering the letter “w.” The application then returns a list of account numbers for each customer whose last name begins with “w.” At that point, the user has to manually enter each account number in order to retrieve the desired information for each one. This process requires a number of repetitive operations through a series of application screens. There’s no way to collect the information in a single display or sort results by account number.

When a newly integrated business application is involved, the user can get significantly better results just by entering the letter “w.” That one operation yields a complete and sorted list that probably resembles the table in Figure 4.

Account Number	Last Name	First Name	Address 1	Address 2
20000	Wagy	Mark	116 Burnside St	Portland, OR 97219
20001	Waters	Joe	1634 15th Ave. W	San Diego, CA 92101
20003	Warner	Jessica	170 Warner St	Warner, MN 26582
20004	Wilson	Maria	634 E 17th St	Taos, NM 98765
20006	Winters	Shelly	1400 Snowy Lane	Portland, OR 38694
20007	Webster	John	342 N 87th St	Seattle, WA 98456
20008	Ward	Aaron	32 Main St	Plymouth, IN 74856

Figure 4: Application Transaction Output

There are two ways to generate the table in Figure 4. You can use either low-level APIs or high-level abstraction. Figure 5 compares the work that goes on behind the scenes for each method. The middle column shows the considerable amount of client-side code that is required when you use low-level APIs to directly manipulate the host application. In contrast, the right column shows how you can accomplish the same result by issuing a single SQL command—a shortcut made possible by Verastream Host Integrator tables and procedures.

Steps Tables and Procedures	Application Code Using a Low-Level API	Application Code Using Verastream
	HashMap attributesMap = new HashMap(); Vector fieldNames = new Vector(); Vector detailRecords = new Vector(); attributesMap.put("LastName", "w");	

Enter "w" in host application	// write w to the LastName attribute
	mySession.setAttributes(attributesMap);	

Navigate to the search results screen	// navigate to entity NameSearchResults
	mySession.setCurrentEntity("NameSearch Results");
	fieldNames.add("AcctNum");
	// fetch records with the field name AcctNum
	records = mySession.fetchRecords(0, fieldNames, null);
	// create an array of acct numbers	

Retrieve the returned account numbers	String[] acctArray = new String[records.get Count()];
	for (int i = 0; i < records.getCount(); i++)
	{	

Sort the list of account numbers	acctArray[i] = records.get(i).get("AcctNum");
	}
	// sort the array of acct numbers
	Arrays.sort(acctArray);
	// addtitional attributes to get for each record in the recordset
	fieldNames.add("LastName");
	fieldNames.add("FirstName");
	fieldNames.add("Address1");
	fieldNames.add("Address2");
	// iterate thru the sorted acct number array from name search to get acct details
	for (int i = 0; i < acctArray.length; i++)
	{
	// navigate to the entity Main
	mySession.setCurrentEntity("Main");	

Navigate back to the main menu screen	// get the acct number from the current record to write to the AcctNum attribute	AppConnRecordSet records =
Enter an account number from the list	attributesMap.put("AcctNum", acctArray[i]); // write the AcctNum attribute	mySession.executeSQLStatement("Select AcctNum, LastName, FirstName, Address1,
	mySession.setAttributes(attributesMap);	Address2 from Accounts where LastName like

Write account number to host application	// navigate to the entity RecordDisplay
	mySession.setCurrentEntity("RecordDisplay");
		'w' order by AcctNum");

Navigate to the record display screen	// read the attributes for the acct number
	record = mySession.getAttributes(fieldNames);	

Retrieve corresponding account details	// store account details in an array of records
	detailRecords.add(record);	

Store account details for future use Navigate back to the main menu screen Repeat for all of the account details	}	

Figure 5: Coding Comparison

The transaction example presented here is fairly straightforward. As your transactions get more complex, the benefits of accelerated development afforded by Verastream tables and procedures will increase dramatically.

Reduced Network Traffic: An Added Bonus

Rapid application development is only part of the acceleration that occurs with Verastream tables and procedures.

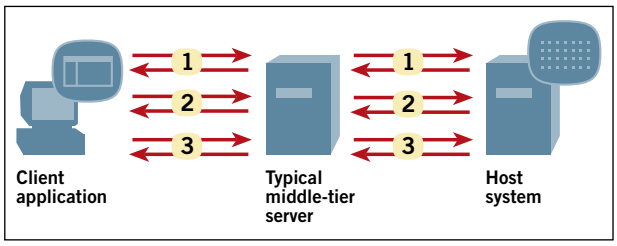

Figure 6: Network Traffic Generated by Low-Level DPIs

When client applications are written to interact directly with host applications, a full network round trip is required for each interaction (see Figure 6). For example, the low-level API code in Figure 5 creates more than 20 network round trips.

Conversely, when client applications are written to take advantage of Verastream tables and procedures, network traffic is reduced and transaction processing is shifted from the client application to the Verastream middle-tier server.

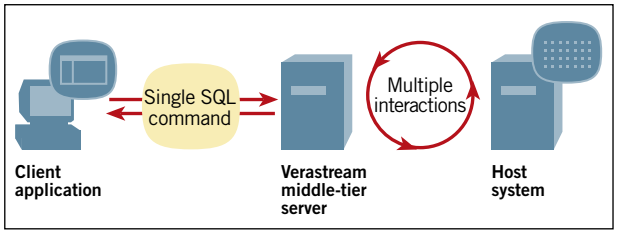

Figure 7: Network Traffic Generated by a Single SQL Command

As Figure 7 illustrates, the result of one SQL command from the tables and procedures application code is a single network round trip between the client application and the Verastream Host Integrator server. The balance of the transaction processing is carried on only between the server and the host application, significantly reducing overall network traffic and improving performance.

Rapid Host Integration: Don't Settle for Less

The speed of business innovation is accelerating at the same time that IT organizations are being pressed to do more with less. Verastream Host Integrator's unique database representation of unstructured host applications drives rapid integration by reducing development complexity and optimizing individual skill sets. After deployment, Verastream Host Integrator continues to accelerate new application processing via enhanced network performance.

A Range of Integration Solultions

Verastream Host Integrator is part of the Attachmate Verastream legacy integration suite, a complete range of mainframe, desktop, and web modernization tools. Our solutions deliver the full spectrum of basic rejuvenation to customized presentation and sophisticated high-performance integration.

Verastream-generated services can be mixed, matched, and reused selectively to extend legacy functionality to new applications or new users. No code changes to legacy applications are required.

About Attachmate

Attachmate delivers advanced software for terminal emulation, application integration, and secure communications. Our NetIQ business provides solutions for automating IT processes and managing performance, security, and compliance of distributed IT. With our technologies, more than 65,000 businesses worldwide are putting their IT assets to work in new and meaningful ways. **www.attachmate.com**.

Attachmate

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 +31 172 50 55 55
FAX +31 172 50 55 51

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**.