

SOA and Web Services for Advantage™ CA-Ideal® for CA-Datcom®

A HostBridge® White Paper



Copyright Notice

Copyright © 2005 by HostBridge Technology. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission. You have limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry this copyright notice. No other rights under copyright are granted without prior written permission. The document is not intended for production and is furnished "as is" without warranty of any kind. All warranties on this document are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Revision date: 11/8/2005

Trademarks

HostBridge is a registered trademark of HostBridge Technology. Advantage is trademarked by Computer Associates. CA-Ideal, CA-Datacom, and ca smart are registered trademarks of Computer Associates. CICS is a registered trademark of IBM.

Table of Contents

Accessing Terminal-oriented Applications	1
Screen Scraping	1
HostBridge for Advantage CA-Ideal	2
Architecture.....	3
Using HostBridge with Advantage CA-Ideal	3
Creating the Panel Data File.....	3
Invoking HostBridge.....	4
CICS-Based Scripting and Process Automation.....	7
Conclusion.....	7

List of Figures

Figure 1. Typical middle-tier screen scraper data path	2
Figure 2. Typical HostBridge data path	2
Figure 3. HostBridge and Advantage CA-Ideal architecture	3
Figure 4. Sample Advantage CA-Ideal application screen	4
Figure 5. Sample XML output (without field rename).....	5
Figure 6. Sample XML output (with field rename).....	5
Figure 7. Sample XML input document	6
Figure 8. Sample SOAP request	6
Figure 9. CICS-based process automation vs. middle-tier scripting	7

Integration Using SOA and Web Services

Service-Oriented Architectures (SOAs) and Web services combine to provide an opportunity for organizations to reduce the costs and complexities of integrating applications both inside and outside the firewall. This white paper explains how you can use SOA and web services to integrate Advantage™ CA-Ideal® for CA-Datcom® (Advantage CA-Ideal) with other enterprise applications.

HostBridge® is software that runs under CICS® on the mainframe and allows web server applications to invoke Advantage CA-Ideal transactions using standard interfaces and returns data as XML documents or web services in a service-oriented architecture (SOA). HostBridge has the unique ability to allow developers to integrate legacy Advantage CA-Ideal applications using field names from panel definitions rather than row/column coordinates from screen scraping and without changes to the legacy applications.

Accessing Terminal-oriented Applications

Advantage CA-Ideal allows you to develop terminal-oriented applications for CA-Datcom quickly and easily. Each screen of an application derives from panel definitions that act as templates for screen layout. These panel definitions contain names for every input field, but these names are lost when Advantage CA-Ideal generates terminal output.

The loss of field names means that when integration software receives output from Advantage CA-Ideal, there is no metadata with which to construct a meaningful HTML, XML, or SOAP document. For this reason, organizations have traditionally relied upon screen scraping gateway technologies to generate web interfaces or to integrate their legacy applications other enterprise applications.

Screen Scraping

Screen scrapers typically run on a physical or logical middle tier. The solution is tightly coupled in that the integration is between the scraper and a specific application. Any changes to the application will break the integration.

When screen scrapers communicate with terminal-oriented legacy applications they open a terminal session with the legacy application, send a request to the application, receive the terminal datastream, use HLLAPI to capture the screen data, process the screen data, convert the contents to XML, and ship the XML document to the requester. (A variation of the model is to use FEPI on the mainframe instead of a middle-tier terminal emulation client. This simply moves the screen scraping onto the mainframe.) The most common components of the screen scraper model appear in the figure below.

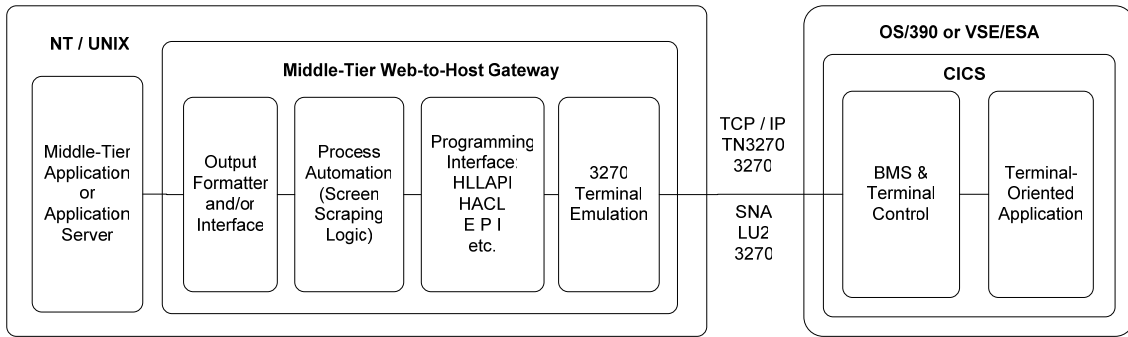


Figure 1. Typical middle-tier screen scraper data path

Screen scrapers allow you to get to your legacy applications, but they create performance bottlenecks and multiple points of failure between the legacy application and the web service. For this reason, screen scrapers are best for short-term projects, either as a transition to using more flexible and reliable integration solutions or as a stopgap measure during application reengineering or platform migration.

HostBridge for Advantage CA-Ideal

The diagram below shows the relatively simple data path required for conducting transactions using HostBridge. Fewer moving parts reduce latency and increases reliability. This configuration provides sub-second response times and as many concurrent transactions as the host system allows.

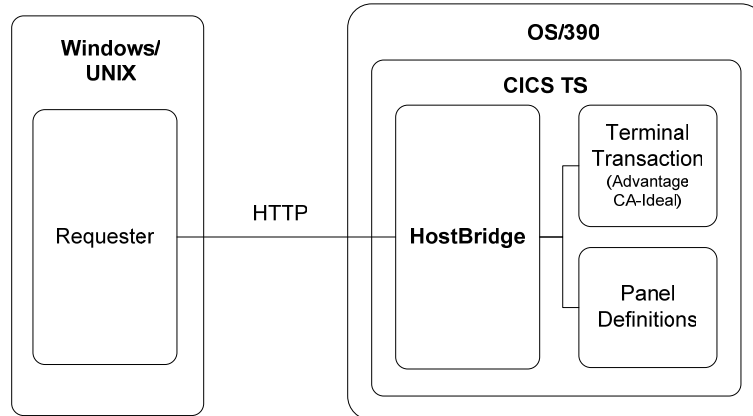


Figure 2. Typical HostBridge data path

Architecture

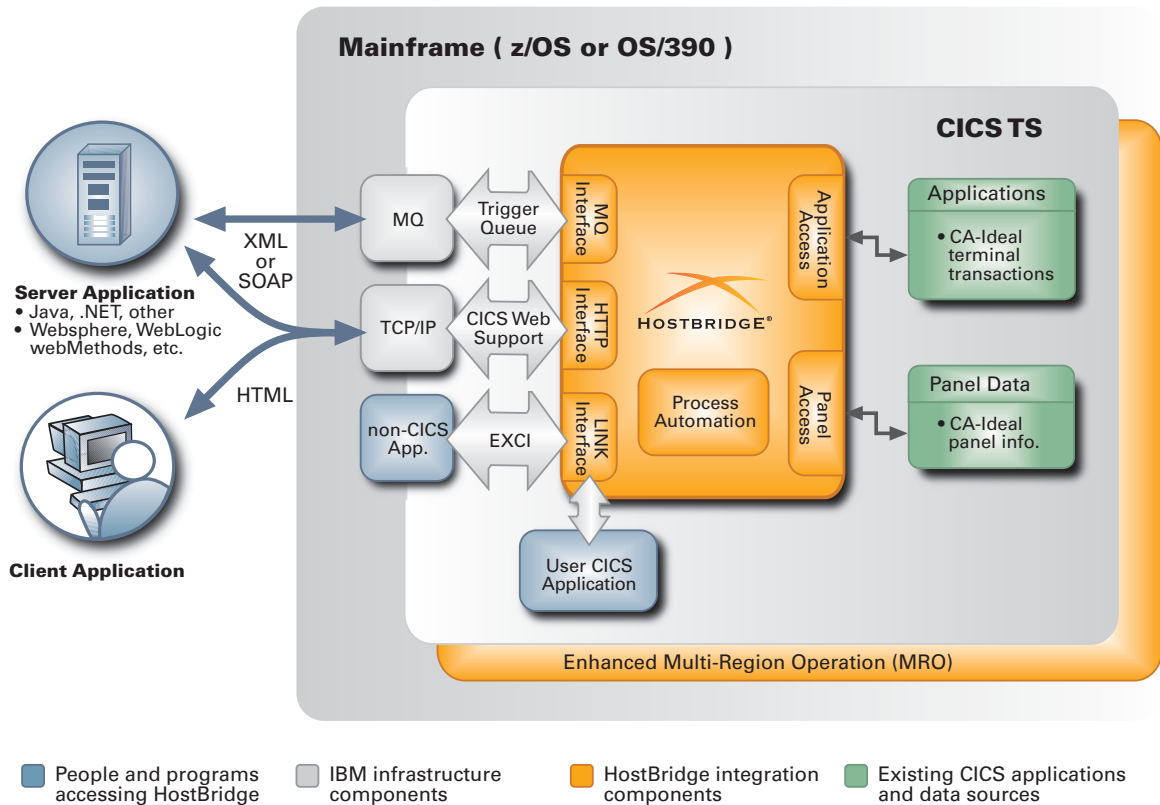


Figure 3. HostBridge and Advantage CA-Ideal architecture

HostBridge works with standard CICS components to allow other applications to access existing Advantage CA-Ideal applications. HostBridge stores panel definition information in a VSAM file, which it later uses to identify panels and field names for each transaction before generating XML output.

Using HostBridge with Advantage CA-Ideal

This section describes how you can use HostBridge to access your existing Advantage CA-Ideal applications.

Creating the Panel Data File

As mentioned above, HostBridge uses a VSAM file as a repository to store and retrieve panel data. To create the repository, use the Advantage CA-Ideal export utility to export the Advantage CA-Ideal panel definition dataset. HostBridge includes a panel processor that analyzes the exported panel and stores the panel information in a VSAM file.

HostBridge reads this VSAM file with each transaction to identify the current panel and extract the field names from the dataset. HostBridge then maps the field data on the screen to the field names and returns an XML document to the requesting application. The XML tags identify the field names and contain the corresponding field data.

Invoking HostBridge

Assume you need to access information contained on an Advantage CA-Ideal screen such as the one shown below.

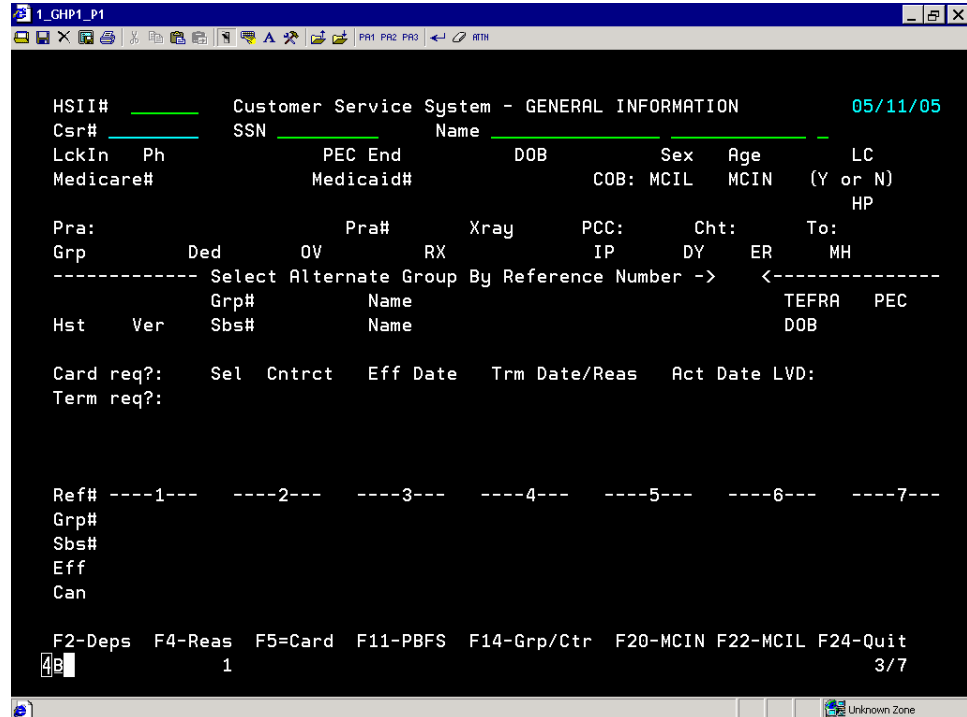


Figure 4. Sample Advantage CA-Ideal application screen

In the screen, notice the “HSII#” field with the value “_____.” Figure 5 shows how HostBridge displays the CICS data in XML after an external application sends an HTTP request with the following format:

```
http://company.com:[port]/hostbridge?HB_ENTRY=[transaction]
```

In the URL above, the “port” is a port you assign to HostBridge under CICS Web Support (CWS) and “transaction” is the name of the transaction you want to execute. For our purposes, assume the transaction name is “MGET.” Below is the HostBridge output sent back to the requesting application after executing the transaction.

```
<?xml version="1.0" standalone="yes" ?>
<hostbridge sysid="TPT1" build_date="Mar 28 2005 10:56:06">
  <token>3664ef41</token>
  <timestamp>20050511211042</timestamp>
  <status>
    <response>0</response>
    <desc>ok</desc>
  </status>
  <parameters>
    <trandid>MGET</trandid>
    <entry>mget</entry>
    <userid>RACFUSR</userid>
  </parameters>
  <fields count="178">
    <field name="field_r1_c2">
      <name len="11">field_r1_c2</name>
      <value maxlen="79" len="0" />
      <attr byte="e8" justify="n" disp="y" prot="y" num="n"
        int="y" mdt="n" />
    </field>
    <field name="field_r2_c2">
      <name len="11">field_r2_c2</name>
```

```

        <value maxlen="6" len="5">HSII#</value>
        <attr byte="e8" justify="n" disp="y" prot="y" num="n"
            int="y" mdt="n" />
    </field>
    <field name="field_r2_c9">
        <name len="11">field_r2_c9</name>
        <value maxlen="6" len="6">_____</value>
        <attr byte="40" justify="n" disp="y" prot="n" num="n"
            int="n" mdt="n" />
    </field>
    ...
    <field name="field_r24_c2">
        <name len="12">field_r24_c2</name>
        <value maxlen="79" len="76">F2Deps F4Reas F5=Card F11PBFS
            F14Grp/Ctr F20MCIN F22MCIL F24Quit</value>
        <attr byte="f0" justify="n" disp="y" prot="y" num="y"
            int="n" mdt="n" />
    </field>
</fields>
<cursor type="pos" pos="166" row="3" col="7">166</cursor>
<copyright info="HostBridge Copyright 20002005 HostBridge
    Technology, U.S. Patent Pending">
</copyright>
</hostbridge>

```

Figure 5. Sample XML output (without field rename)

Because MGET is a non-BMS transaction, HostBridge assumes the transaction is a standard CICS 3270 transaction. Notice that in this case, HostBridge identifies the HSII# field using its row/column location (<field name="field_r2_c9">) in the 3270 datastream. To alert HostBridge that panel definitions exist for the transaction, simply add the command HB_FIELD_RENAME=1 to the URL:

[http://company.com:\[port\]/hostbridge?HB_ENTRY=MGET&HB_FIELD_RENAME=1](http://company.com:[port]/hostbridge?HB_ENTRY=MGET&HB_FIELD_RENAME=1)

Now, the XML output includes the field names instead of the row/column values as shown in the figure below.

```

<?xml version="1.0" standalone="yes" ?>
<hostbridge sysid="TPT1" build_date="Mar 28 2005 10:56:06">
  <token>3664ef41</token>
  <timestamp>20050511211707</timestamp>
  <status>
    <response>0</response>
    <desc>ok</desc>
  </status>
  <parameters>
    <trandid>MGET</trandid>
    <entry>mget</entry>
    <userid>RACFUSR</userid>
  </parameters>
  <fields count="118">
    <field name="P1HSII#" index="0">
      <name len="8">P1HSII#</name>
      <value maxlen="6" len="6">_____</value>
      <attr byte="40" justify="n" disp="y" prot="n" num="n"
          int="n" mdt="n" />
    </field>
    ...
    <field name="P1MESSAGE" index="0">
      <name len="10">P1MESSAGE</name>
      <value maxlen="79" len="62">Consumer number 0000MGET
          invalid, please enter a valid number.</value>
      <attr byte="f8" justify="n" disp="y" prot="y" num="y"
          int="y" mdt="n" />
    </field>
  </fields>
  <cursor type="pos" pos="166" row="3" col="7">166</cursor>
  <copyright info="HostBridge Copyright 20002005 HostBridge
      Technology, U.S. Patent Pending" />
</hostbridge>

```

Figure 6. Sample XML output (with field rename)

Now, that same HSII# field appears as `<field name="P1HSII#" index="0">` which is the name assigned to the field in the Advantage CA-Ideal panel definition. If you invoke the initial transaction using `HB_FIELD_RENAME`, each subsequent transaction during a session will honor the command.

To continue interacting with the application, let us look at a URL that you could send to the application to change the contents of the P1HSII field from “____” to “012345.”

```
http://company.com:4041/hostbridge?HB_TOKEN=3664ef41&P1HSII=012345
```

This part of the URL...	Does this...
<code>http://company.com:4041/</code>	Identifies the address and port where HostBridge runs. The HTTP listener passes URLs on this port to the HostBridge analyzer.
<code>hostbridge?</code>	Identifies this as a HostBridge session. Everything that follows the “?” in the URL is the command string that controls HostBridge behavior.
<code>HB_TOKEN=3664ef41</code>	After an initial connection to the host application, HostBridge returns a session ID called a state token. Seen in the XML above as: <code><token>3664ef41</token></code> Subsequent transactions with the host must include this token in the URL so CICS will recognize the transaction as part of an existing session.
<code>P1HSII=012345</code>	Sends the field name and the data value we want to enter in the transaction. HostBridge interprets any name/value pair sent along the URL as input to a field unless the name starts with “HB_” (e.g., <code>HB_TOKEN</code>).

HostBridge can also receive requests in the form of XML documents that conform to a fixed schema. The following XML document submits the same transaction data as the URL above.

```
<?xml version="1.0" ?>
<hostbridge>
  <transaction>
    <parameters>
      <hb_token>3664ef41</hb_token>
    </parameters>
    <fields>
      <P1HSII>012345</P1HSII>
    </fields>
  </transaction>
</hostbridge>
```

Figure 7. Sample XML input document

For companies integrating applications as web services, HostBridge supports SOAP messages as input to invoke HostBridge. The following SOAP message submits the same transaction data as the XML and URL requests above.

```
<?xml version="1.0" ?>
<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <hostbridge>
      <hb_token>3664ef41</hb_token>
      <P1HSII>MICHAEL</P1HSII>
    </hostbridge>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 8. Sample SOAP request

CICS-Based Scripting and Process Automation

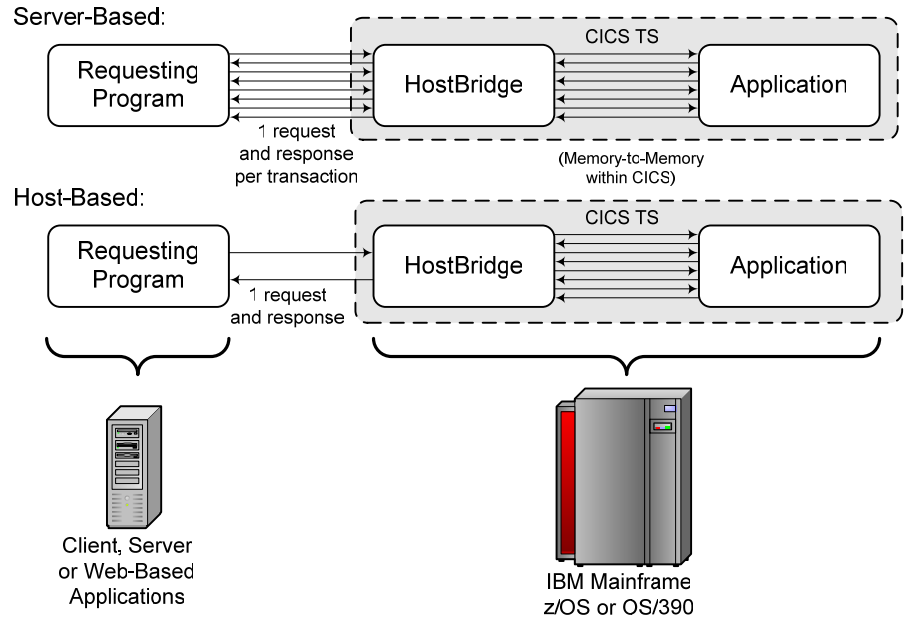


Figure 9. CICS-based process automation vs. middle-tier scripting

The diagram above illustrates the advantages of using host-based scripting to automate interactions with Advantage CA-Ideal applications. Server-based scripting requires an HTTP request to drive each transaction within a business process. Thus, complicated business processes can produce large numbers of HTTP requests and increase latency. Host-based scripting reduces latency and remove the burden from the middle-tier developer from having to handle each individual transaction within a business process.

HostBridge supports integrated process automation based on a highly efficient, industry standard ECMAScript (a.k.a., JavaScript) engine that runs under CICS on the mainframe. Web developers compose the automation scripts at their workstations, compile them, and deploy them to HostBridge.

Conclusion

Advantage CA-Ideal continues to provide rapid development of mainframe applications. HostBridge allow companies with legacy applications built using Advantage CA-Ideal to protect their investments by integrating those applications with other enterprise applications. Using industry standards such as XML, SOAP, and JavaScript, HostBridge allows web developers to use existing skills and knowledge to build the integration framework, while shielding mainframe Advantage CA-Ideal developers from integration projects because HostBridge does not require changes to the legacy applications.

HostBridge lets developers interact with Advantage CA-Ideal applications using field names, not field locations on a screen. Therefore, if mainframe developers make changes to their applications by adding or removing fields on a screen, the changes will not affect web applications. This saves time and money for development staff and reduces the chances that users will receive errors when they access your web applications.

Contact us to find out how HostBridge can help integrate your Advantage CA-Ideal applications.



100 E Seventh Ave, Stillwater, OK 74074
Ph. 405.533.2900
www.hostbridge.com