

HostBridge and Novell eXtend™: Integrating CICS with Web Services

A HostBridge™ White Paper



Toll-free: (866) 965-2427
Email: info@hostbridge.com

Copyright Notice

Copyright © 2002 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: 6/13/2002

Trademarks

HostBridge is a trademark of HostBridge Technology.
eXtend is a trademark of Novell, Inc.

Table of Contents

Overview: Novell eXtend™, CICS and HostBridge	1
Types of CICS Applications	2
“Visual” vs. “Non-Visual” Transactions.....	2
Understanding CICS “Visual” Transactions.....	2
What is HostBridge?	3
Integration Architecture	3
Sample Application	4
Defining the eXtend Web Service	4
Communicating with HostBridge.....	5
Web Service Inputs and Outputs.....	5
HostBridge Requests and Responses	6
Request 1: Building and Sending a URL to HostBridge.....	6
Request 2: Managing State and Submitting Data.....	8
Request 3: Building and returning an XML document to eXtend	9
Summary	10
Appendix A: CICS Trader Application	11

List of Figures and Screen Shots

Figure 1. CICS Application Access Taxonomy Error! Bookmark not defined.	
Figure 2. HostBridge/eXtend Solution Architecture.....	3
Figure 3. Web Service Definition for Sample Application	4
Figure 4. XML Interchange to Communicate with HostBridge	5
Figure 5. XML Document Sent from Remote Application to eXtend Application	6
Figure 6. eXtend Definition for Input XML Document.....	6
Figure 7. XML Document Sent from eXtend Application to Remote Application	6
Figure 8. eXtend Definition for Output XML Document.....	6
Figure 9. XML Document Returned by HostBridge to Request 1 (abbrev.) ...	7
Figure 10. StockLookup Function	7
Figure 11. XML Interchange URL Expression.....	8
Figure 12. XML Document Returned by HostBridge to Request 2 (abbrev.).	8
Figure 13. XML Document Returned by HostBridge to Request 3 (abbrev.)	9
Figure 14. Sample eXtend Map Action	10
Figure 15. Output XML Document Returned by the Sample eXtend Application	10
Screen 1. Company selection screen	11
Screen 2. Trading options screen	11
Screen 3. Real-time Quote screen.....	12

HostBridge and Novell eXtend™: Integrating CICS with Web Services

Using HostBridge, Novell eXtend™ can easily create a web service that includes data from CICS applications. A web service in eXtend is an object that takes an input XML document, performs a series of actions and then returns an output XML document. Together, HostBridge and eXtend allow organizations to transform their existing CICS applications into web services that can be used by other eBusiness applications.

Overview: Novell eXtend™, CICS and HostBridge

Novell eXtend™ is a comprehensive, integrated services environment that simplifies and accelerates the creation and delivery of services-oriented business applications. Novell eXtend is made up of four integrated components: Director, Composer, Application Server and Workbench (in this document we will focus on Composer).

eXtend represents an emerging category of eBusiness integration software based upon the use of XML. However, the existing category of mainframe CICS applications has certainly not gone away. In fact, according to statistics from IBM and others, CICS has never been more successful:

- 30 years and \$1 trillion (per IDC) invested in CICS applications
- 14,000+ CICS customers worldwide
- 20,000+ CICS/390 licenses worldwide
- CICS is used by 490+ of IBM's top 500 customers
- 30 million end users of CICS applications
- 150,000+ concurrent users/system
- 5,000 CICS software packages from 2,000 ISVs
- 950,000 programmers earn their living from CICS
- CICS handles >30 billion transactions/day valued at >\$1 trillion/week

For companies with large investments in mainframe CICS applications, the ability to integrate these applications with products like Novell eXtend is imperative.

HostBridge is a patent pending software product that XML-enables a broad class of existing CICS applications. HostBridge does this without requiring modification to the existing applications, and without screen-scraping. As a result, HostBridge is an ideal tool for integrating CICS applications with eXtend.

This White Paper presents a case study on how you can use HostBridge to integrate existing CICS applications with eXtend.

Types of CICS Applications

Not all CICS transactions operate the same way. As a result, the integration approach will depend on how the CICS transaction operates. The following diagram shows a high-level taxonomy of CICS transactions.

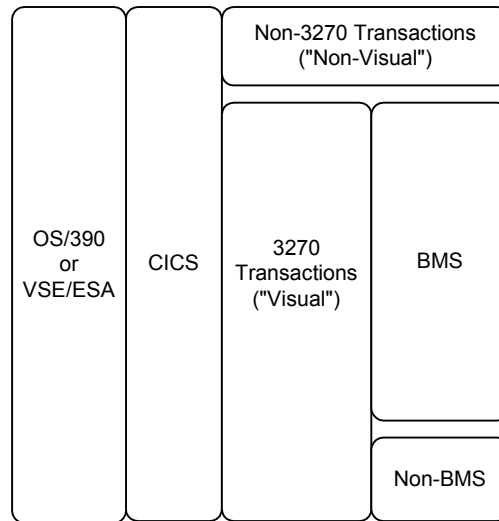


Figure 1. CICS Application Access Taxonomy

“Visual” vs. “Non-Visual” Transactions

CICS transactions fall into two broad categories: “visual” and “non-visual.” A “visual” transaction is one that expresses a presentation interface to an end-user at a terminal. You could also refer to a “visual” transaction as a “terminal-oriented” transaction. In contrast, “non-visual” transactions do **not** interact with an end-user. Instead, another program invokes these transactions. (This type of transaction is also referred to as “COMMAREA transaction” because the input/output parameters are passed to/from the transaction using an area of storage referred to as the “communication area,” or COMMAREA.)

The distinction between non-visual and visual transactions is important because integration possibilities exist for non-visual transactions that do not exist for visual transactions. As you might imagine, non-visual transactions are far easier to integrate with other programs than are visual transactions. However, visual transactions are far more common than non-visual transactions.

Understanding CICS “Visual” Transactions

In order to understand the integration possibilities for visual transactions, we need to further define this category.

CICS application developers have always had a number of choices in how to design their transaction to interact with an end-user at a terminal. The majority of applications use a component of CICS called Basic Mapping Support (BMS). BMS essentially handles the presentation logic of the

transaction and relieves the application developer from having to encode and decode 3270 terminal data streams. The minority of applications that do not use BMS either include code to process 3270 data streams, or rely upon a non-IBM solution to handle presentation logic.

The distinction between BMS and non-BMS applications is important because integration possibilities exist for BMS applications that do not exist for non-BMS applications. For non-BMS applications, integration is based upon the terminal-oriented data stream generated by the application. As described elsewhere, this approach can have serious limitations.

What is HostBridge?

HostBridge is a patent pending software product that XML-enables existing CICS transactions. HostBridge runs under CICS on the mainframe. With HostBridge, any CICS transaction can be XML-enabled automatically. HostBridge does this without requiring modification to your existing applications, and without screen-scraping. As a result, HostBridge is the perfect tool for integrating CICS applications with eXtend.

Integration Architecture

Novell eXtend™ is an integrated services environment that simplifies and accelerates the creation and delivery of web services. A web service in eXtend is an application that takes an input XML document, performs a series of actions and then returns an output XML document. HostBridge allows an eXtend application to easily integrate with existing data and business logic accessible through CICS BMS transactions.

The basic architecture of this approach looks like this:

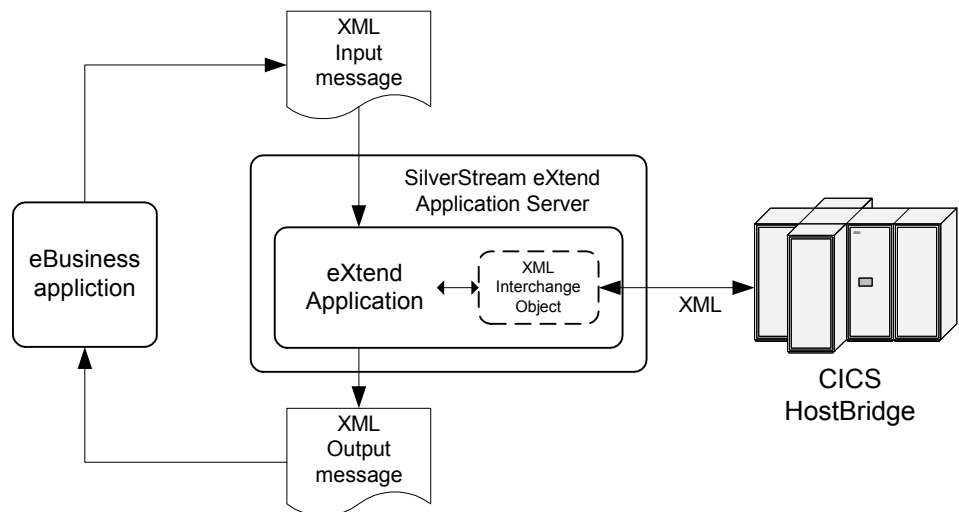


Figure 2. HostBridge/eXtend Solution Architecture

The eBusiness application initiates the process by sending a request, along with an XML input message, to the eXtend application server. This request invokes an eXtend application (web service). When the eXtend application requires data from CICS, it sends a request to HostBridge using the eXtend

XML Interchange object. Communication between the eXtend application and HostBridge continues until the required CICS data elements are obtained. The eXtend application then sends the XML output message back to the originating application.

Sample Application

Using eXtend and HostBridge, the sample application described in this section illustrates how to create a web service that integrates with a CICS BMS application. See Appendix A for a description of the CICS transaction used in this sample application.

Defining the eXtend Web Service

For this sample application we will create an eXtend web service that receives a request for a stock quote, executes multiple CICS transactions to satisfy the request (using HostBridge), and returns the results to the requestor. Figure 3, below, illustrates the definition of an eXtend web service consisting of three XML Interchanges and three Map actions.

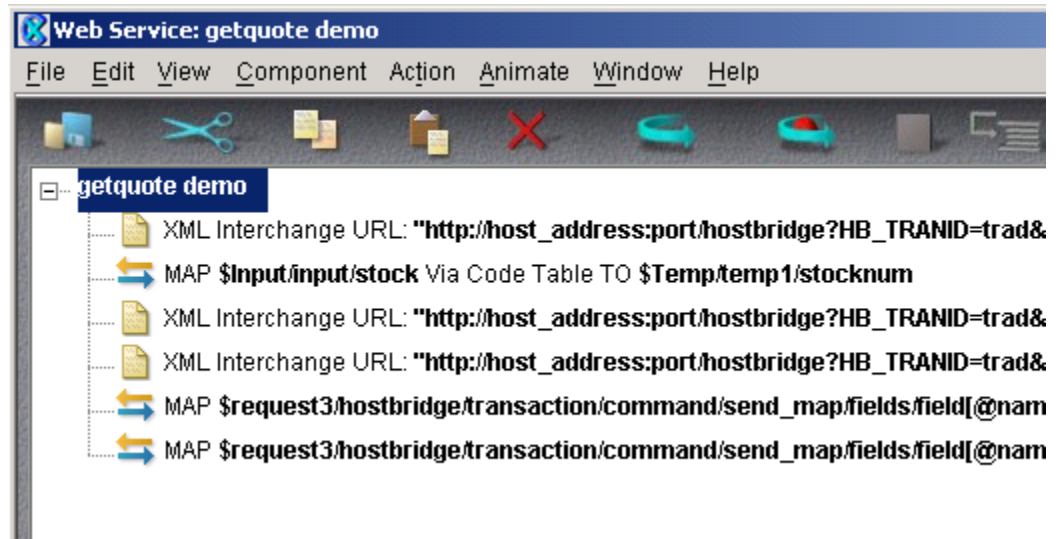


Figure 3. Web Service Definition for Sample Application

Communicating with HostBridge

In eXtend, an XML Interchange action/object defines how eXtend is to communicate with an external XML-based resource like HostBridge. You must define one XML Interchange for each unique interaction with HostBridge. An XML Interchange consists of:

- an HTTP request method (GET|POST|HEAD),
- the URL you will use to communicate with HostBridge,
- a name for the connection/transaction, and
- name for the XML document returned by HostBridge

The XML Interchange is responsible for building and sending the HTTP request to HostBridge, and receiving the XML output from HostBridge.

A typical URL sent to HostBridge always has certain common elements, such as:

http://hostaddress:port/ hostbridge? HB_TRANID=trad& HB_AID=enter	IP address and port assigned to HB indicates this is a hostbridge request the CICS transaction name to be executed AID key used with a transaction
--	---

The above URL will be the first request that the eXtend application will send to HostBridge. Figure 4, below, illustrates the definition of this request:

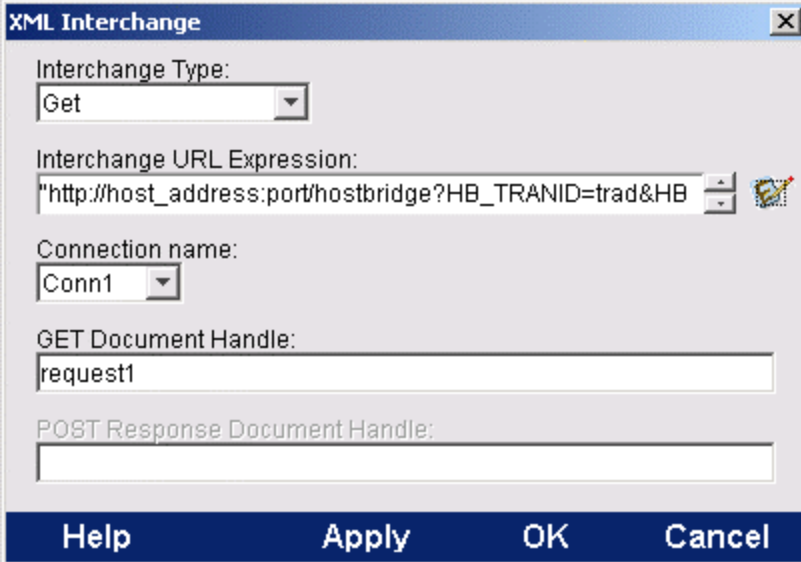


Figure 4. XML Interchange to Communicate with HostBridge

Web Service Inputs and Outputs

When the remote application invokes the eXtend web service, it must provide an input XML document. Likewise, when the eXtend application is finished it will return an XML document as output.

For our sample application, the XML input document has a single data element, <stock>, which contains the name of the stock for which a quote is being requested quote (see Figure 5, below).

```
<input>
  <stock>ibm</stock>
</input>
```

Figure 5. XML Document Sent from Remote Application to eXtend Application

The input XML document is defined to eXtend, as follows.

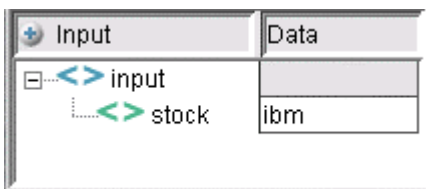


Figure 6. eXtend Definition for Input XML Document

The XML output document is shown in Figure 7, below. It contains two data values: <price> contains the current price of the requested stock, and <shares> contains the number of shares currently held:

```
<output>
  <price>163.00</price>
  <shares>1000</shares>
</output>
```

Figure 7. XML Document Sent from eXtend Application to Remote Application

The output XML document is defined to eXtend, as follows.

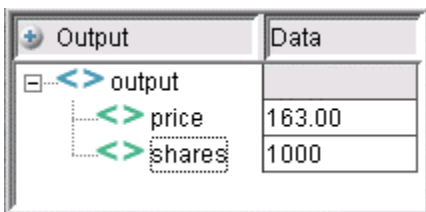


Figure 8. eXtend Definition for Output XML Document

HostBridge Requests and Responses

Request 1: Building and Sending a URL to HostBridge

The first request sent to HostBridge by the eXtend application causes HostBridge to start a CICS transaction named “trad.” In response, HostBridge returns an XML document that represents the first screen of the CICS application (Screen 1 in Appendix A). Figure 9 shows an abbreviated version of the XML document.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--HostBridge Copyright 2000, 2001 HostBridge Technology, U.S. Patent Pending-->
<hostbridge>
  <token>86d946b5</token>
  <fields>
    <field name="COMP1">
      <value>Casey_Import_Export</value>
    </field>
    <field name="COMP2">
      <value>Glass_and_Luget_Plc</value>
    </field>
    <field name="COMP3">
      <value>Headworth_Electrical</value>
    </field>
    <field name="COMP4">
      <value>IBM</value>
    </field>
    <field name="OPTION">
      <value></value>
    </field>
    <field name="MESS2">
      <value></value>
    </field>
  </fields>
</hostbridge>

```

Figure 9. XML Document Returned by HostBridge to Request 1 (abbrev.)

While the input XML document requests a specific stock by name, the CICS application expects the end-user to use a numeric value to select the company from a list of company names. Thus, in order for the eXtend application to select the company it must determine the corresponding item number from the list. An eXtend Code Table Map action is used to determine the item number of the stock from the input XML document (Figure 10, below). This value is then mapped into the temporary variable \$Temp/temp1/stocknum.

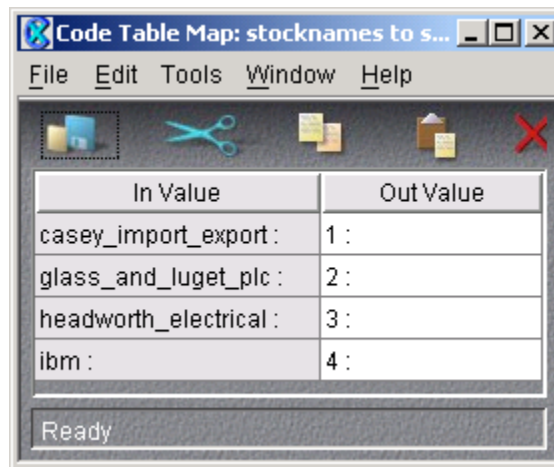


Figure 10. StockLookup Function

Once the code table correlates the company name to an item number in the list, the eXtend application sends the next request to HostBridge.

Request 2: Managing State and Submitting Data

The CICS transaction used in this example is a “pseudo-conversational” transaction. Without going into a detailed explanation of what this means, whenever HostBridge is used to invoke a pseudo-conversational transaction it returns a “token.” HostBridge must receive this token on the next request.

To prepare to for the next XML Interchange with HostBridge, the eXtend application uses the XPath expression `request1.XPath("hostbridge/token")` to retrieve the `<token>` value from the first XML document returned by HostBridge. In our example, the token value is `86d946b5`. Next, the eXtend application appends the string “HB_TOKEN=” and the `<token>` value to the URL. This will tell HostBridge to associate this next transaction with the previous transaction.

Next, the eXtend application appends the string “&option=” and the value of `$Temp/temp1/stocknum` to the URL. This specifies the stock number to the CICS transaction. Note that the input field `option` was included in the first XML document returned by HostBridge. Figure 11 shows the XML Interchange action for this request.

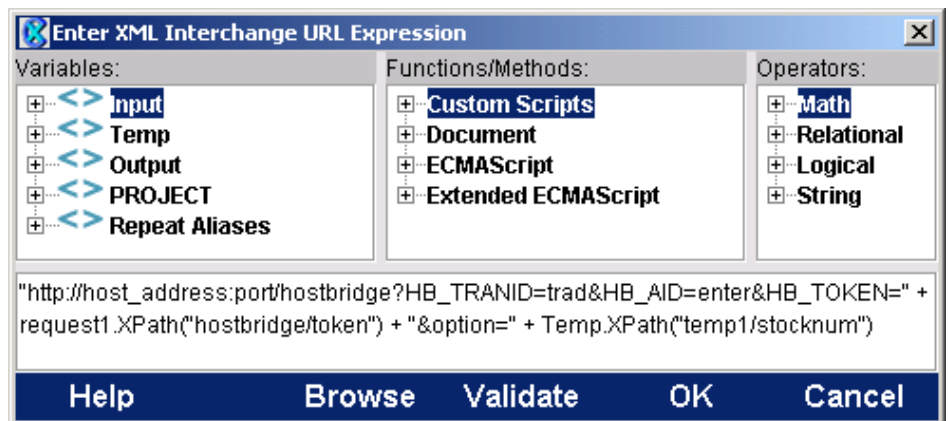


Figure 11. XML Interchange URL Expression

The eXtend application sends the resulting URL to HostBridge as follows:

```
http://hostaddress:port/hostbridge?HB_TRANID=trad&
HB_AID=enter&HB_TOKEN=86d946b5&option=4
```

This request tells HostBridge to execute the next leg of the `trad` transaction, selecting company number 4. In response, HostBridge returns an XML document that represents the second screen of the CICS application (Screen 2 in Appendix A). Figure 12 is an abbreviated version of the XML document.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--HostBridge Copyright 2000, 2001 HostBridge Technology, U.S. Patent Pending-->
<hostbridge>
  <token>86d946b5</token>
  <fields>
    <field name="OPT2">
      <value></value>
    </field>
    <field name="MESS3">
      <value></value>
    </field>
  </fields>
</hostbridge>
```

Figure 12. XML Document Returned by HostBridge to Request 2 (abbrev.)

Request 3: Building and returning an XML document to eXtend

After selecting a company, the eXtend application needs to tell the CICS application what to do: get a quote, buy shares, or sell shares. The eXtend application must select from a list of three items where the order of these items in the list is fixed. Selecting item 1 from the list indicates that we want a quote. Thus, the eXtend application appends the string "&opt2=1" to the base URL (along with the token). Note that the input field opt1 was included in the second XML document returned by HostBridge.

The eXtend application uses XML Interchange to send the resulting URL to HostBridge as follows:

```
http://hostaddress:port/hostbridge?HB_TRANID=trad&
HB_AID=enter&HB_TOKEN=86d946b5&opt2=1
```

This request tells HostBridge to execute the next leg of the trad transaction, selecting action item 1 (get quote). In response, HostBridge returns an XML document that represents the third screen of the application (Screen 3 in Appendix A). Figure 13, below, is an abbreviated version of the XML document.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--HostBridge Copyright 2000, 2001 HostBridge Technology, U.S. Patent Pending-->
<hostbridge>
  <token>86d946b5</token>
  <fields>
    <field name="SHRNOW">
      <value>00163.00</value>
    </field>
    <field name="HELD">
      <value>1000</value>
    </field>
  </fields>
</hostbridge>
```

Figure 13. XML Document Returned by HostBridge to Request 3 (abbrev.)

The XML document returned by HostBridge contains the share price and number of shares held. In the BMS map used by the TRAD transaction, the fields containing that information are called SHRNOW and HELD. HostBridge uses these same names in its XML document. The eXtend application now uses the Map actions to map these values into the <price> and <shares> elements of the output XML document. Figure 14, below, illustrates the eXtend Map action for SHRNOW.

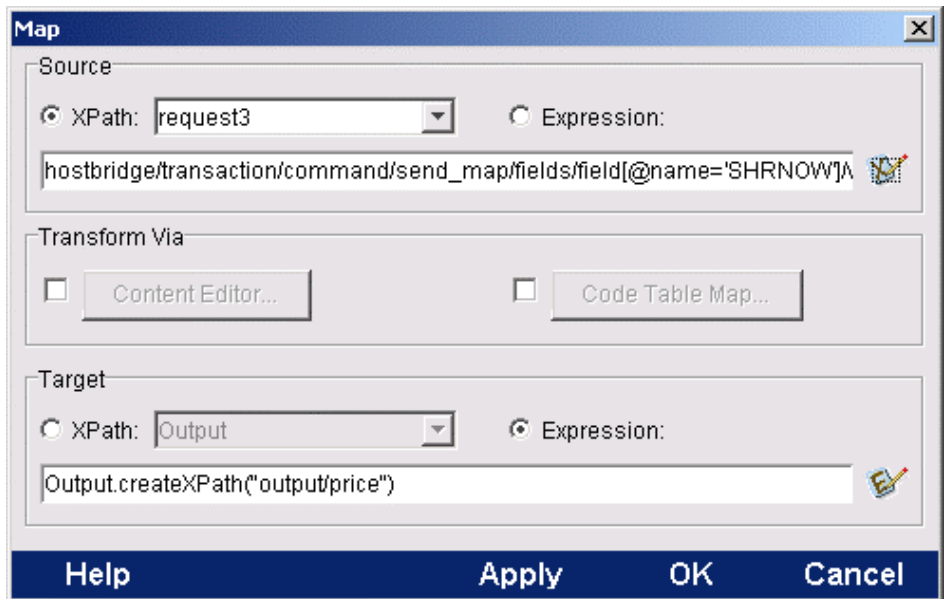


Figure 14. Sample eXtend Map Action

After extracting the values of `SHRNOW` and `HELD` out of the XML document returned by HostBridge, the eXtend application returns the output XML document to the remote application.

In our example, the XML output would be:

```
<output>
  <price>163.00</price>
  <shares>1000</shares>
</output>
```

Figure 15. Output XML Document Returned by the Sample eXtend Application

Summary

Novell eXtend™ is a comprehensive, integrated services environment that simplifies and accelerates the creation and delivery of services-oriented business applications. While eXtend represents an emerging category of eBusiness integration software, the existing category of mainframe CICS applications is alive and well within large organizations.

HostBridge is a patent pending software product that XML-enables CICS BMS applications. HostBridge does this without requiring modification to the existing applications, and without screen-scraping. As a result, HostBridge is an ideal tool for integrating CICS applications with eXtend.

Working together, HostBridge and eXtend provide a simple, scalable, and secure solution for integrating existing CICS BMS applications with eBusiness processes.

Appendix A: CICS Trader Application

IBM provides a sample CICS BMS application that simulates a stock trading application. The Share Trading Demonstration, or TRADER, consists of only a few screens that allow you to choose a company, get a stock quote, or buy/sell shares. The case study presented in this White Paper uses this application. The application transactions are pseudo-conversational and use the BMS commands SEND MAP and RECEIVE MAP to communicate with the end-user. The application is simple to use. The example below shows how to get a stock quote for IBM using the demonstration application.

1. Logon to CICS.
2. Enter TRAD to start the application. (The following screen should appear.) This screen allows you to select the company whose stock you want to act upon.

```
Share Trading Demonstration                                TRADER.T002
Share Trading Manager: Company Selection

1. Casey_Import_Export
2. Glass_and_Luget_Plc
3. Headworth_Electrical
4. IBM

Please select a company (1,2,3 or 4) :

-----
PF3=Exit                                                    PF12=Exit
```

Screen 1. Company selection screen

3. Enter 4 to select IBM. (The following screen should appear.) This screen allows you to get a stock quote, buy shares, or sell shares.

```
Share Trading Demonstration                                TRADER.T003
Share Trading Manager: Options

1. New Real-Time Quote
2. Buy Shares
3. Sell Shares

Please select an option (1,2 or 3): _

-----
PF3=Return                                                    PF12=Exit
```

Screen 2. Trading options screen

4. Enter 1 to retrieve the stock quote. (The following screen should appear.) This screen displays the stock quote for the selected company.

```
Share Trading Demonstration                                TRADER.T004
Share Trading Manager: Real-Time Quote

User Name:      RUSS
Company Name:   IBM

Share Values:
NOW:            00163.00
1 week ago:    00157.00
6 days ago:    00156.00
5 days ago:    00159.00
4 days ago:    00161.00
3 days ago:    00160.00
2 days ago:    00162.00
1 day ago:     00163.00

Commission Cost:
for Selling:    015
for Buying:     010

Number of Shares Held: 5033
Value of Shares Held: 000820379.00

Request Completed OK
-----
PF3=Return                                           PF12=Exit
```

Screen 3. Real-time Quote screen

5. Press PF12 to exit the application.