Integrating CICS Using HostBridge and Microsoft® Windows® DNA

A Tier Technologies White Paper

Copyright Notice

Revision date: 7/4/2002

Trademarks

Table of Contents

Integrating CICS Using HostBridge and Microsoft Windows DNA

Overview: Microsoft DNA, CICS and HostBridge

Microsoft's Distributed Internet Architecture (DNA) enables the rapid construction and deployment of web-based business applications. It leverages an n-tier application development philosophy—with a presentation layer, business logic layer, and data layer—to provide two key benefits:

1. Developer independence, which results in increased efficiency in coding, and
2. The ability to distribute different system pieces (such as a relational database or web pages) to different physical machines, which improves scalability and performance.

While Microsoft intends its .NET platform as an eventual replacement for DNA, several reasons exist for continuing to develop applications using DNA methodologies:

- The core technologies underlying DNA—Microsoft Visual Basic 6.0 and Microsoft Active Server Pages 3.0—are simple to learn and require less computing power than comparable .NET development tools to use.
- DNA applications have been developed for almost 5 years now, making it a mature and time-tested technology with both a sizable knowledge base available for troubleshooting issues and a significant number of technology workers with the appropriate training to use the technology.
- Conversely, the .NET platform represents a new technological paradigm. Enough significant issues have been identified within the platform to necessitate the release of a service pack upgrade to the foundation software after only 4 months. Thus, .NET development appears desirable for the mid-to-long range future, when most of the flaws in the initial release will have been resolved, but leaves developers wanting in the short term.
- When a determination has been made that .NET is a stable enough platform for production-level development, DNA applications can be easily upgraded with only moderate amounts of rework.

DNA serves as an extremely flexible model with which to build and deploy web-based business applications. However, the existing category of mainframe CICS business applications is not dead. 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 license 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 values at $1 trillion/week

For companies with large investments in mainframe CICS applications, leveraging exisitng business logic and processes through a web-based interface becomes one of the most cost-efficient methods of moving into the eBusiness realm.

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 the overhead of "screen scraping". As a result, HostBridge is an ideal tool for leveraging CICS applications as part of a DNA application.

This White Paper presents (1) a case study on how you can use HostBridge to leverage existing CICS application as part of a DNA application and (2) information about how the consulting firm Tier Technologies can help you build and deploy your HostBridge-based DNA applications in record time.

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.

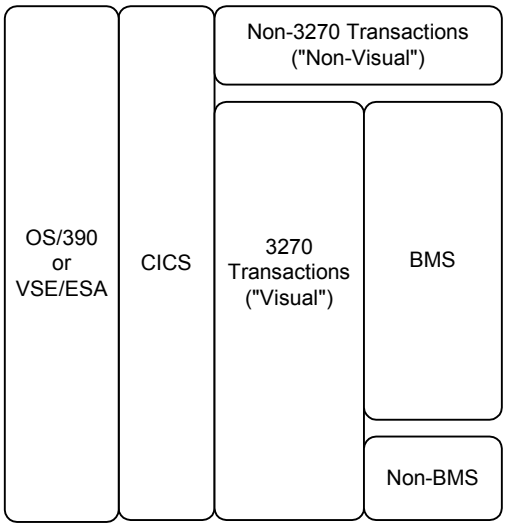


Figure 1. CICS Application Access Taxonomy

"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.

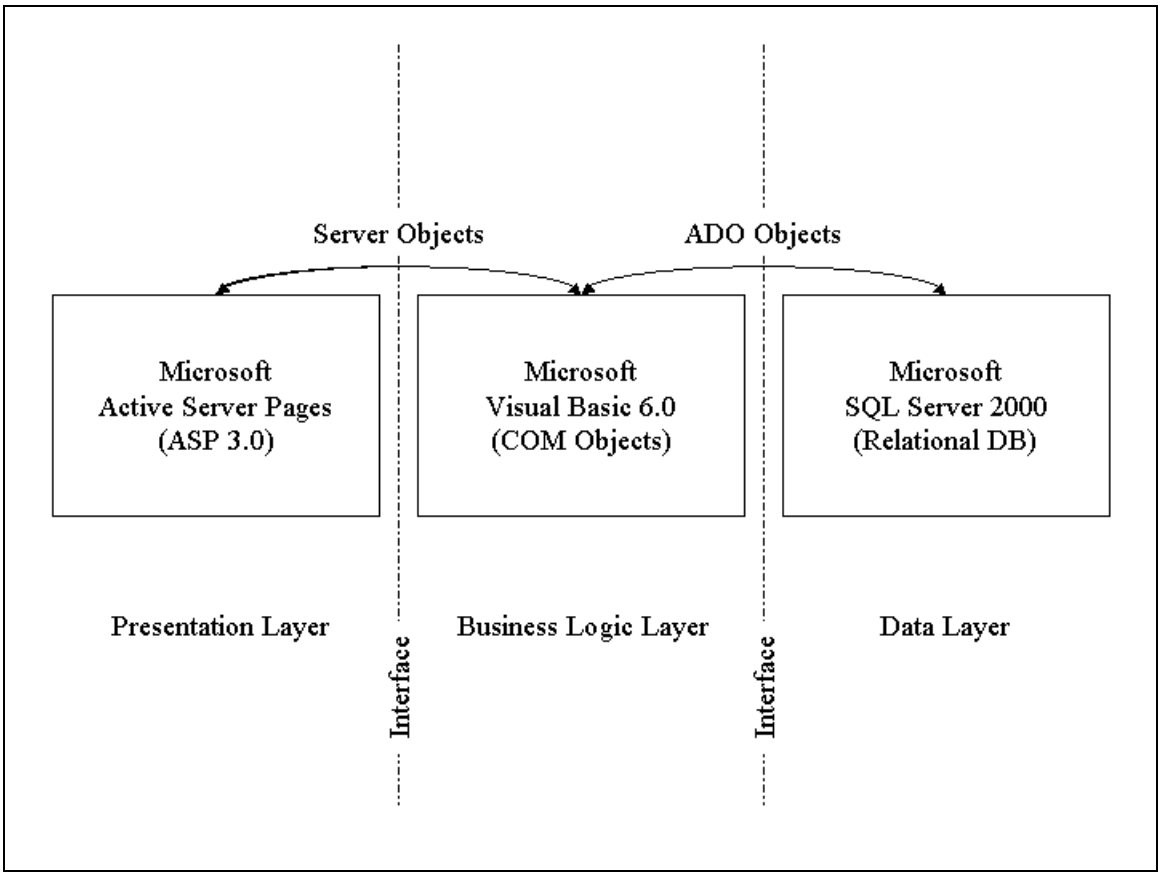
What is HostBridge?

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 Weblogic Integration.

This white paper describes how to use Weblogic and HostBridge to integrate a BMS application. However, HostBridge can XML-enable each of the CICS application types shown in Figure 1: non-3270 (COMMAREA) applications and both BMS and non-BMS visual applications.

Integration Architecture

Traditional DNA-based solutions consist of three distinct component layers:

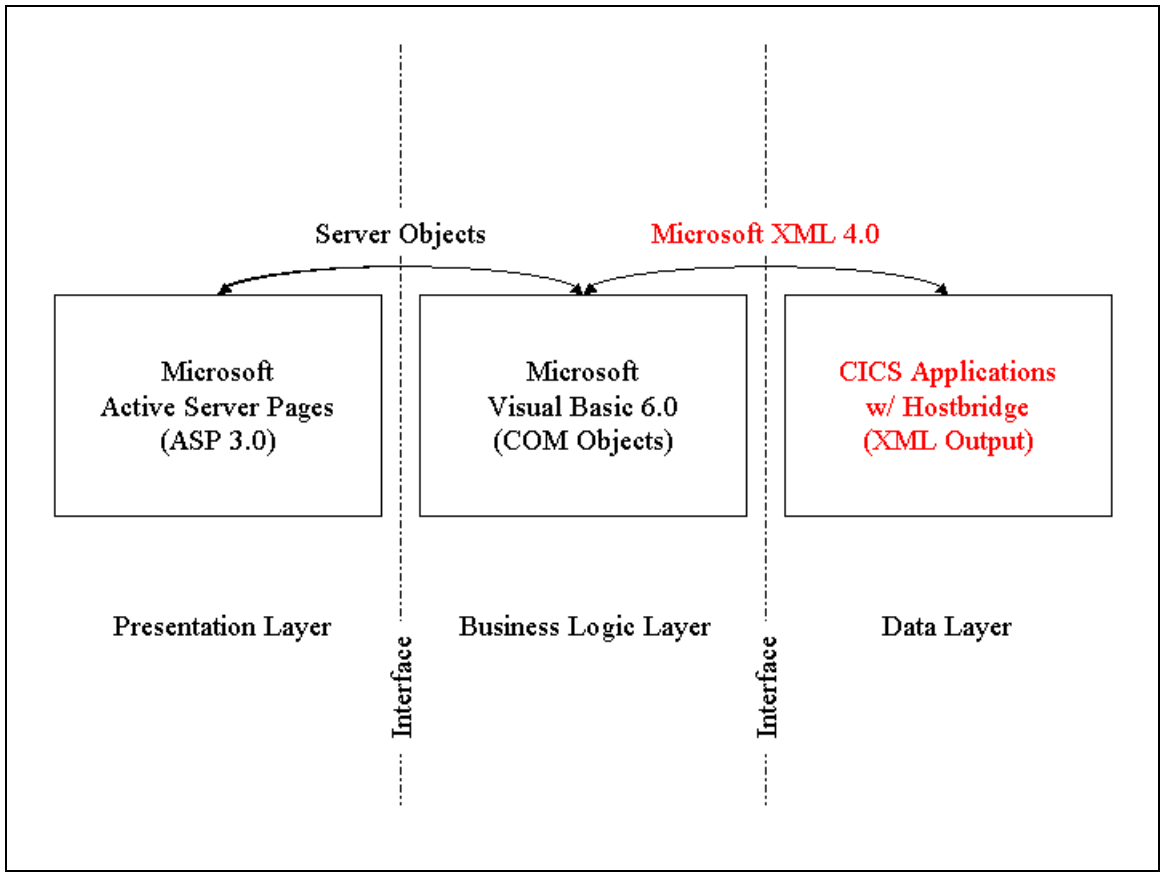


In traditional DNA processing:

1. A user interacts with the application via the user interface generated by the presentation components (ASP 3.0 scripts).
2. Upon request of a specific system action (such as look up of an order), the presentation components interact with the business logic components (VB 6.0) through the appropriate interface (ASP 3.0 Server objects), pass through any appropriate user-provided data (if any), and wait for the result.
3. The business logic components execute the appropriate code to process the user's request and, usually, interact with the data components (SQL Server 2000) via the data interface (ADO objects) to retrieve the data necessary to fulfill the request.
4. The necessary data retrieved from the data components is processed by the business logic components then forwarded to the presentation components for display to the user.

Each of the layers involved in a DNA application only interacts with the others via the respective interfaces. This design allows component pieces of the application to be developed and/or switched out independently of each other as long as the pre-defined interface does not change. It also allows each individual component to be installed on distinct physical machines as long the interface can span over a network.

DNA applications that leverage existing CICS business logic using HostBridge have the following differences (in red) from the traditional DNA structure:



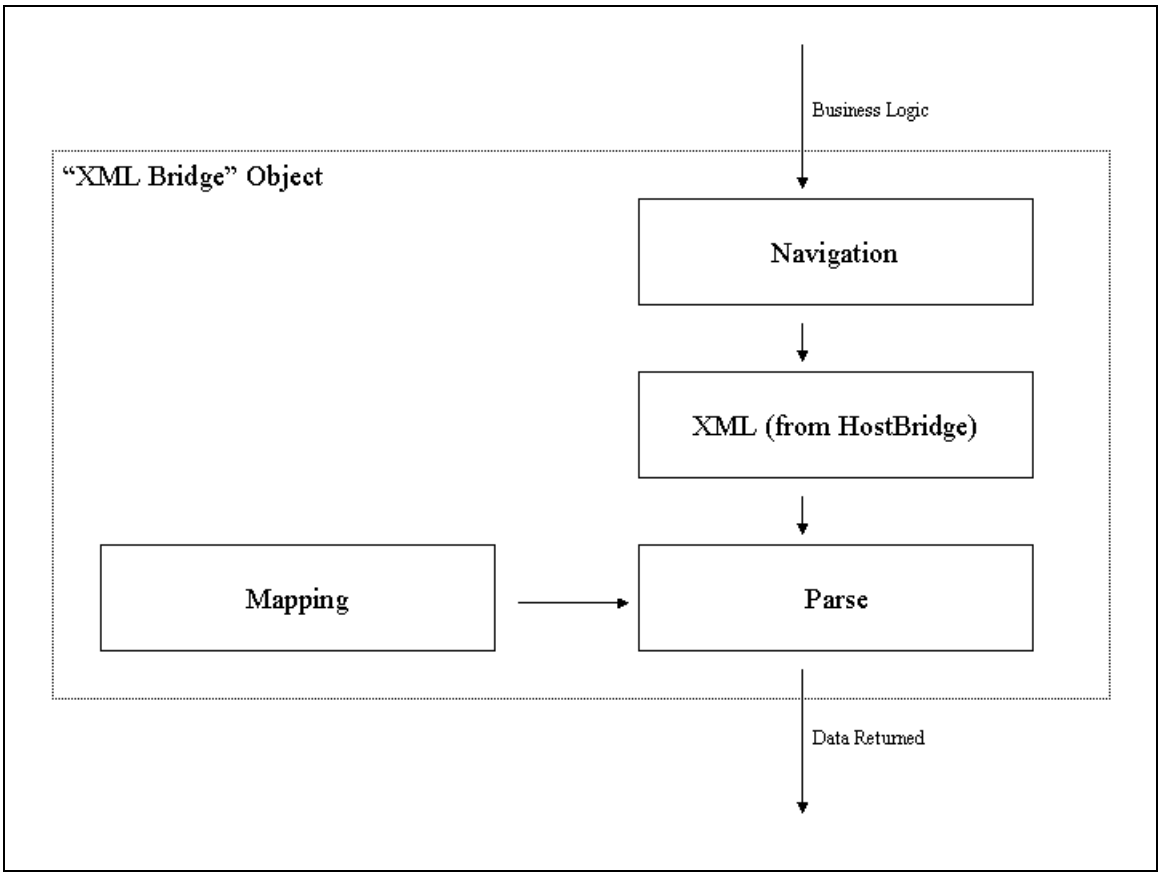
In this model, the CICS Application serving XML documents via HostBridge replaces the SQL Server database. A business logic component that utilizes Microsoft's XML 4.0 tools allows the other business logic components to parse and load the data received from the mainframe. Once loaded into working storage within the business logic components, the data can be manipulated by those same components in a similar fashion as data loaded from a database.

In terms of the process discussed previously, the only change is in step 3:

1. A user interacts with the application via the user interface generated by the presentation components (ASP 3.0 scripts).
2. Upon request of a specific system action (such as look up of an order), the presentation components interact with the business logic components (VB 6.0) through the appropriate interface (ASP 3.0 Server objects pass through any appropriate user-provided data (if any), and wait for the result.

3. The business logic components execute the appropriate code to process the user's request and, usually, interact with the data components (HostBridge-enabled CICS application) via the data interface (MSXML 4.0) to retrieve the data necessary to fulfill the request.
4. The necessary data retrieved from the data components is processed by the business logic components then forwarded to the presentation components for display to the user.

The following diagram illustrates a hypothetical "XML Bridge" business object that utilizes MSXML 4.0 to retrieve the appropriate XML from HostBridge and process and load the result (step 3 in the process listed previously):



When the business logic calls for a specific piece or set of data, the XML Bridge (1) navigates to the appropriate location within the CICS application, (2) receives the BMS map information as XML from HostBridge, (3) loads a "mapping" or "definition" that defines the individual field elements that were returned within the XML, and (4) parses the XML while assigning the mapped names to the appropriate data elements. The data elements are then returned to the calling business logic layer object.

Introducing Tier: What We Can Do To Help

The sample application detailed in the next section is based upon work done for clients by Tier Technologies, Inc. Tier is a vertically-focused consulting firm that provides business and information technology consulting, systems design and integration, transaction processing, business process outsourcing and business process reengineering for its clients primarily in the state and local government, healthcare, insurance and utilities markets. Tier brings specific industry knowledge, proven delivery capability and proprietary applications to its client relationships. The combination of domain expertise and technical capability allow Tier to provide solutions that link increased operating efficiencies with systems and technology improvements. Tier serves Fortune 1000 companies and government entities.

With regard to DNA applications, Tier consultants build toolkits that allow development efforts to be repeated across their client base. The use of these toolkits reduces development costs for all clients and allows demonstrable results to be available in rapid fashion—sometimes before the initial introductory presentation is completed! When choosing to use HostBridge to provide XML from your CICS applications, Tier consultants can provide a rich toolkit of "XML Bridge"-style objects to facilitate development of applications that consume your mainframe's data.

For more information on Tier Technologies and how Tier consultants can help with your legacy integration projects, contact a HostBridge representative or Tier directly at:

Tier Technologies, Inc.
8381 Old Courthouse Road
Suite 200
Vienna, VA 22182
(888) 467-3570

Sample Application

The sample application described in this section illustrates how to use XML from HostBridge as the data source within a DNA application. See Appendix A for a description of the CICS transaction used in this sample application.

Application Architecture

The application features three processes that involve sending and receiving data from the mainframe:

- Adding A New Record
  1. A request for adding a transportation deduction to an employee's payroll record is received.
  2. The "XML Bridge" object navigates to the appropriate mainframe screen to make the addition.
  3. The information submitted by the user in the request is sent to the mainframe via HostBridge.
  4. The XML returned by HostBridge is used to confirm the addition was successful.

- Updating An Existing Record
  1. A request for updating a transportation deduction to an employee's payroll record is received.
  2. The "XML Bridge" object navigates to the appropriate mainframe screen to make the update.
  3. The information submitted by the user in the request is sent to the mainframe via HostBridge.
  4. The XML returned by HostBridge is used to confirm the update was successful.

- Deleting An Exisiting Record
  1. A request for deleting a transportation deduction on an employee's payroll record is received.
  2. The "XML Bridge" object navigates to the appropriate mainframe screen to make the deletion.
  3. The delete request is sent to the mainframe via HostBridge.
  4. The XML returned by HostBridge is used to confirm the delete was successful.

In all three instances, the general procedure is as illustrated in the Integration Architecture discussion: Navigate, get XML from HostBridge, parse the XML to determine the return status, and send the appropriate reply back to the user.

Sample Navigation Sequence

A series of URLs are constructed by the Tier "XML Bridge" object to navigate to the appropriate location within the mainframe to execute the desired function. Within this application, all three functions (add, update, and delete) take place on the same screen.

```
http://hostaddress:port/hostbridge?HB_TRANID=aprl&HB_AID=enter
http://hostaddress:port/hostbridge?HB_TOKEN=1234567&HB_AID=enter
http://hostaddress:port/hostbridge?HB_TOKEN=1234567&PSWD=abcd&DEPT=a&FUNC=1&HB_AID=enter
http://hostaddress:port/hostbridge?HB_TOKEN=1234567&SEL=1&HB_AID=enter
```

These URLs are submitted to HostBridge using the WinHTTP 5.0 object that comes packaged with Microsoft XML 4.0. Sample code and instructions on how to use the WinHTTP 5.0 object can be found on Microsoft's web site at http://msdn.microsoft.com/.

Sample Submission For Adding A Record

Once the navigation is complete, a record is submitted either using an HTTP GET or POST: (This sample application uses an HTTP GET.)

```
http://hostaddress:port/hostbridge?HB_TOKEN=1234567
   &ADRLN1=100+MAIN+STREET&ADRLN2=SUITE+100&CITY=NEW+YORK&STATE=NY&ZIP=11001
   &HB_URLDECODE=1&HB_AID=pf2
```

In this URL, "ADRLN1", "ADRLN2", "CITY", "STATE", and "ZIP" are the names of the BMS fields that represent these pieces of data. Use of the "HB_URLDECODE" parameter allows HostBridge to correctly interpret the "+" signs within the data submission as spaces instead of as literals. The PF2 key completes the addition of the record.

Updating and deleting a record are extremely similar to this example. Updating is the same URL with a PF5 key; deletion is a URL with no data fields and a PF3 key.

Sample Code to Parse the Returned XML For Success/Fail Status

Once the operation has been conducted, the XML returned by HostBridge must be parsed to determine the success or failure of the operation (add, delete, or update). In this application, the status can be determine by interrogating the BMS field with the name "MESSG". The following Visual Basic code utilizes the MSXML 4.0 DOM Document Objects to parse the document for the information in that field:

```
Public Function GetFieldValue(strXML as String, strFielDNAme As String) As String

Dim objXMLDocument As DOMDocument40
Dim objXMLNode As IXMLDOMNode
Dim objXMLNode2 As IXMLDOMNode

Set objXMLDocument = New DOMDocument40
objXMLDocument.loadXML strXML

Set objXMLNode = objXMLDocument.selectSingleNode("//field[@name='" & strFielDNAme & "']")
Set objXMLNode2 = objXMLNode.selectSingleNode("value")
GetFieldValue = objXMLNode2.nodeTypedValue
```

```
Set objXMLNode = Nothing
Set objXMLNode2 = Nothing
Set objXMLDocument = Nothing

End Function
```

The return value from this function is the error message line from the mainframe screen.

If the return value is “RECORD ADDED”, “RECORD UPDATED”, or “RECORD DELETED” (depending on the function chosen), then the operation was successful. If not, then the operation failed in some manner. In any case, an appropriate message can be returned to the user at this point indicating the result of their submission.

This extremely simple sample application only provides a glimpse of the total toolset available from Tier to execute transformations of CICS BMS applications to web-based DNA applications. It is intended to provide an idea of the development techniques used to build your DNA applications using HostBridge.

Summary

Microsoft’s DNA enables the rapid construction and deployment of web-based business applications. For companies with large investments in mainframe CICS applications, leveraging exisitng business logic and processes on the mainframe through a web-based DNA interface is one of the most reliable and cost-efficient methods of moving into the eBusiness realm. Working together, Microsoft, HostBridge, and Tier provide a simple, scalable, and secure solution for transforming existing CICS BMS applications into eBusiness applications.

Appendix A: Administrative Payroll (APRL) System

The administrative payroll (APRL) system documented here is a CICS BMS application used by the New York City Board of Education Department of Financial Operations. One of the options available via this application is the ability to begin or cancel tax deductions for public transportation for a given employee's paycheck. The case study in this White Paper uses this application. Since the application is a production application at a real-life client site, some of information presented has been altered to maintain security.

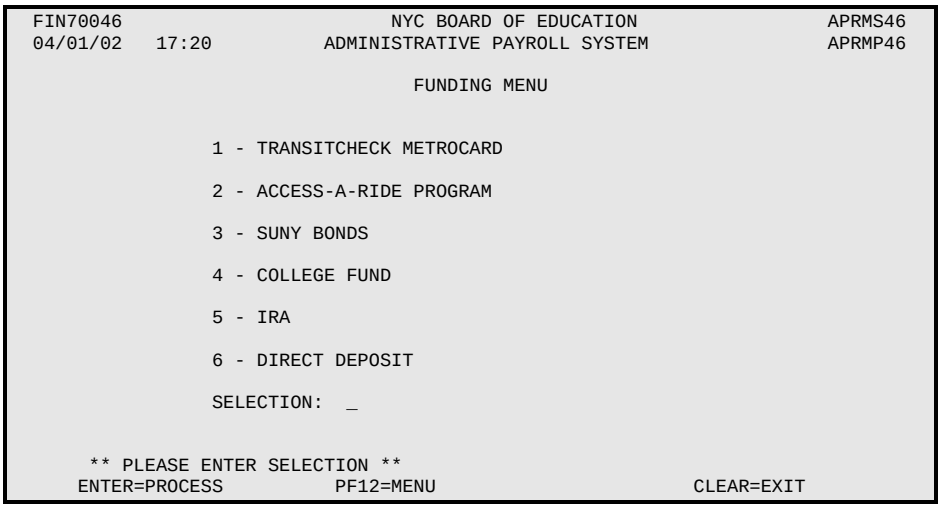
1. Logon to CICS
2. Enter APRL to start the application. The following screen should appear.

```
FIN70001              NEW YORK CITY BOARD OF EDUCATION                APRLMSB
             ADMINISTRATIVE PAYROLL/PERSONNEL BROADCAST SCREEN
ENTER PASSWORD      DEPT CODE
TODAY IS THE DAY THAT WE WILL NOTICE THE DIFFERENCE BETWEEN 2 PEOPLE

DIE FOR IS TO LIVE FOR

WAKE UP TO THE CLEAR BRIGHT MORNING

LOVE IS THE FEELING THAT YOU FEEL WHEN YUOU FEEL THE FEELING IN YOUR HEART









                ENTER=MENU PF6=NEXT PAGE CLEAR=END SESSION
PF9=CLEAR SCREEN  PF10=CREATE MESSAGE  PF11=EDIT MESSAGE  PF12=DELETE MESSAGE
```

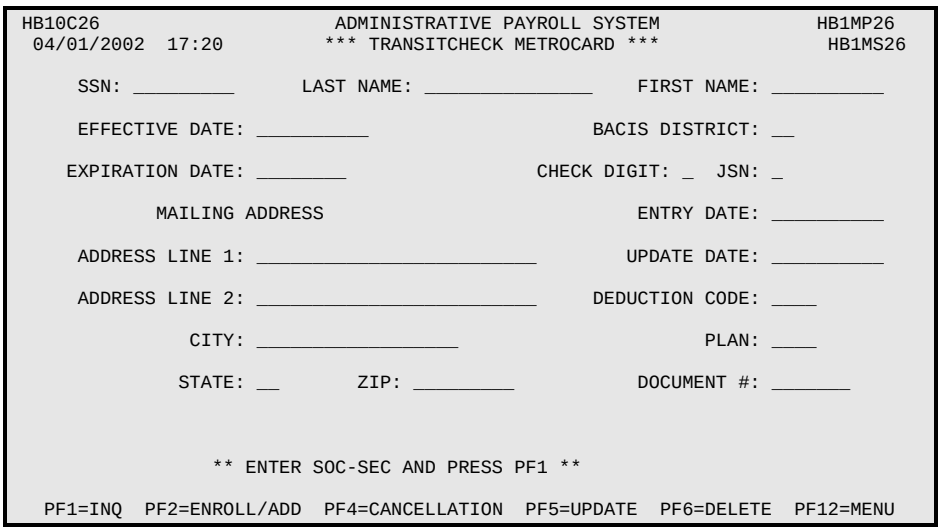
3. Press ENTER to move past the broadcast message screen to the main menu.

```
04/01/02              ADMINISTRATIVE PAYROLL SYSTEM                   APRLM01
FIN70005                          MENU                                APRLMS1

                              FUNCTION CODES

 01 - ADM PAYROLL       - BROWSE         11 - PENDING           - INQUIRY/UPD
 02 - MASTER FILE       - INQUIRY/UPD    12 - DP2001            - ADD/UPD/DEL
 03 - EMPLOYEE ID       - UPDATE         13 - EQUITY/LONGEVITY  - INQ/AUTHORIZE
 04 - RETIREMENT        - INQUIRY/UPD    14 - Z-BANK DAILY HIST - INQUIRY
 05 - HISTORY FILE      - BROWSE/CHANGE  15 - PERSONNEL DATA    - INQUIRY
 06 - APPT AND TITL DT  - UPDATE         16 - PROBLEM FILE      - INQUIRY/UPD
 07 - TABLE FILE        - INQUIRY/UPD    17 - H-Z CHECK REGISTER
 08 - 9902              - BROWSE         18 - TRS ANNUITY PAYMENTS
 09 - Z BANK HOURS      - INQUIRY/UPD    19 - FUNDING MENU
 10 - OVERTIME/DEDUCT   - BRW/INQ/UPD    20 - NURSES AND THERAPISTS SUBSYSTEM

                       ENTER PASSWORD      AND
                       DEPT CODE      _

        FUNCTION CODE:                EMPLOYEE ID: _________________


        ENTER ==> ACCESS REQUESTED SCREEN    CLEAR ==> END SESSION
```

4. Enter the appropriate password, department, and function on the main menu and press ENTER. The sub-menu for that function appears.



5. Enter the desired selection and press ENTER. The screen associated with that selection appears.



6. Conduct the desired operation based upon the function keys indicated at the bottom of the screen.
7. Press CLEAR to exit the application.