



BridgeWare Integration Framework

Version 6.1

Written by: Daniel Nelson

April 15, 2007

Table of Contents

TABLE OF CONTENTS	1
DEFINITIONS	2
ARCHITECTURE REQUIREMENTS & RECOMMENDATIONS.....	3
FRAMEWORK OVERVIEW	4
FRAMEWORK PROCESS FLOW	5
<i>HTTP Client Request</i>	5
<i>Web Servlet Response</i>	6
BRIDGEWARE SERVLET INSTALLATION.....	7
SERVLET INSTALLATION.....	7
SERVLET CONFIGURATION	7
SERVLET METHODS	8
STANDARD REQUEST/RESPONSE METHODS	8
1. <i>Search Customer</i>	8
2. <i>Select Customer Exposure</i>	8
3. <i>Select Customer Message Inquiry</i>	8
4. <i>Select Lookup Table</i>	8
5. <i>Select Next ID</i>	9
6. <i>Select Zip Geo Code</i>	9
7. <i>Insert Customer Entity</i>	9
8. <i>Insert Customer Message Input</i>	9
9. <i>Update Contract Entity</i>	9
10. <i>Update (Reserve/Release) Contract Number</i>	10
11. <i>Update Contract Filing</i>	10
12. <i>Update Customer Entity</i>	10
13. <i>Update Asset Entity</i>	10
14. <i>Update Lookup Table</i>	11
15. <i>Insert Miscellaneous Invoice Record</i>	11
16. <i>Delete Miscellaneous Invoice Record</i>	11
BRIDGEWARE BULLETIN BOARD COMPONENT	12
17. <i>Select Bulletin Board</i>	12
18. <i>Update Bulletin Board</i>	12
CUSTOMER USER DEFINED METHOD.....	13
19. <i>Custom User Method</i>	13
METHODS FOR “HELP DESK” SUPPORT	14
20. <i>Select InfoLease UserIDs</i>	14
21. <i>Update InfoLease UserIDs</i>	14
BRIDGEWARE DIAGNOSTIC UTILITY.....	15
APPENDIX.....	16
<i>Table Names for SELECT_LOOKUP_TABLE method</i>	16
ERROR MESSAGES AND POSSIBLE SOLUTIONS.....	17

Definitions

InfoLease® – A lease/loan accounting solution produced by International Decision Systems (IDS) for managing portfolios, tracking assets and handling accounts. The application requires UniData RDBMS operating on UNIX or Windows Win2K/Win2K3 server.

UniData® – The RDBMS was selected by IDS for support of nested relation data, consequently not SQL-compliant. It additional to record locking, file sizing, and other basic RDBMS functions, supports a query language called UniQuery, and access via .COM and Java APIs.

UniBasic – UniBasic is UniData's structured programming language intended for a variety of applications.

UniObjects for Java – UniObjects for Java is a programming interface that allows Java based applications to communicate with UniData RDBMS. By default, UniObjects for Java relies on port 31438, but this can be configured to use an alternate port number.

Architecture Requirements & Recommendations

Recommendations

1. Servlet Container, 1 ghz processor with 512mb Ram, installed on a separate server than the InfoLease™ host, but on the same physical network.
2. SSL encryption.
3. UniData RDBMS Release 5.2 or higher.

Requirements

1. Tomcat 4.0 Web Server or higher.
2. UniData RDBMS Release 5.2 or higher. UniData RDBMS Release 6.1 or higher is needed for SSL encryption between servlet and UniData.
3. JDK 1.3 or higher.
4. UniData UniObjects for Java. This component is shipped with IBM's UniData Client Utilities.

Framework Overview

This Integration Framework document provides transaction definitions, specifications and layouts.

The BridgeWare Integration Framework enables you to build and implement enterprise-wide ‘real time’ access to InfoLease™ back office data easily and efficiently. It takes advantage of industry standard Internet technologies including Java Servlets, J2EE, Hypertext Transport Protocol (HTTP) and Extensible Markup Language (XML).

The Framework is comprised of two components. The first component is a single web server servlet built upon the request/response model. That request contains the header and parameter information necessary for the servlet to validate and relay the request. As directed by the “_TYPE” parameter, the BridgeWare Integration Framework will forward the request to the second component, the BridgeWare UniBasic methods.

Using UniData’s UniObjects for Java API, the servlet will communicate directly, synchronously, with the BridgeWare UniBasic methods that co-exist with InfoLease™. These methods, more commonly known in the UniData world as subroutines, execute the request based upon the remaining parameters and return a well-formed XML document to the servlet. The servlet relays this document to the requesting client and closes the connection.

A proxy server may be used for communication between the servlet and UniData in the case where a firewall exists between them. The proxy server port and other properties are configurable.

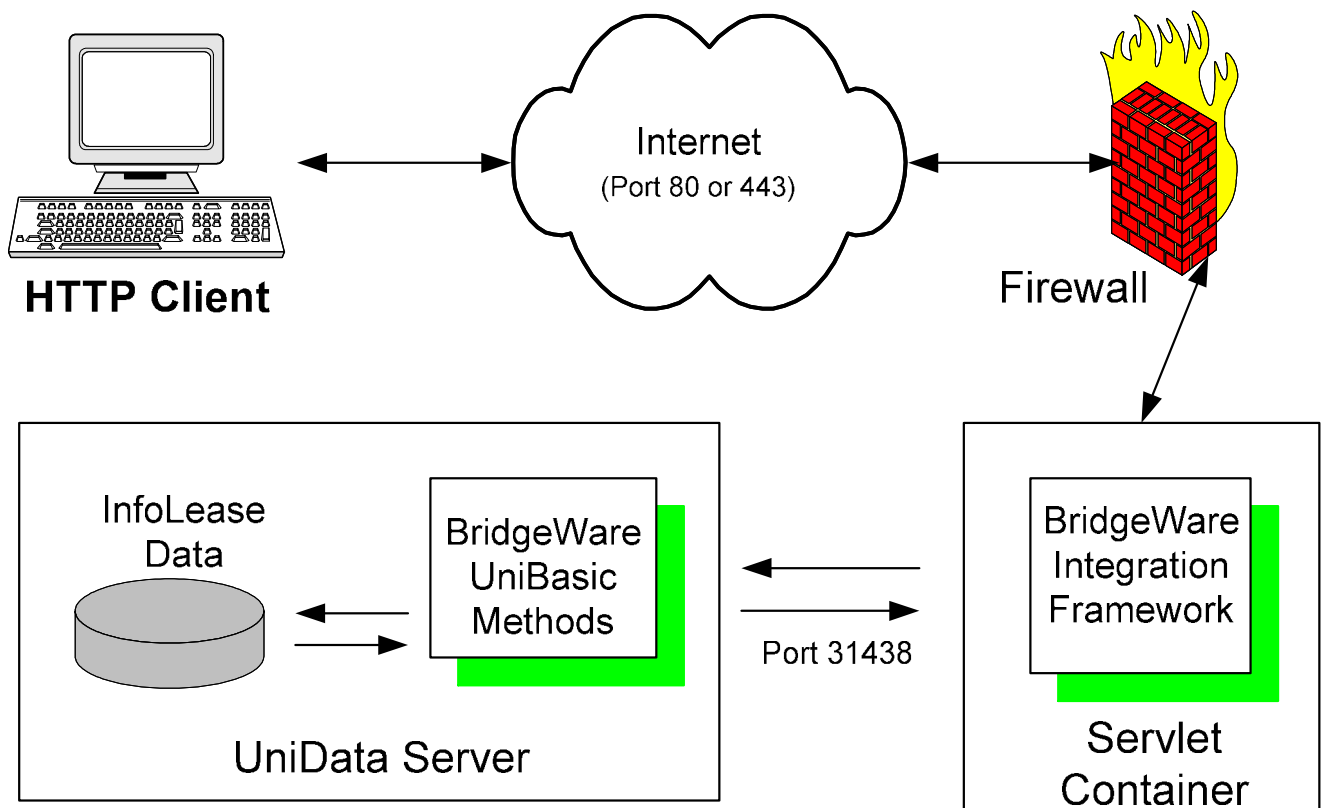


Figure 1.1 – Sample Framework Overview

Framework Process Flow

All communication between the client and BridgeWare are performed in a *request-response* mode: an HTTP client requests data and the servlet returns either XML or HTML data.

HTTP Client Request

Each request from an HTTP client, both POSTs and GETs are supported, must contain the host authentication parameters in addition to the required and optional parameters listed under each method. Each request must contain the following information:

Parameter	Description	Example
_TYPE	Identifies the method (subroutine) in UniData to execute.	SELECT_CUST_EXPOSURE
HOST_NAME	UniData Server	Localhost
USER_ID	NT/Unix User ID authorized to access UniData data files	dsiroot
PASSWORD	Associated NT/Unix password	password
DATABASE_PATH	Path to InfoLease™ data	/info/DATABASE

Additionally, the servlet provides support for a series of parameters that will control the format of the resulting data and the delivery method. The following parameters have defaults:

Parameter	Description	Example
SEND_EMAIL	If not null, servlet will email results. The default is null. Note: If not null, the next four parameters are required.	Yes
TO_ADDRESS	Email Address to received results. (If null, SEND_EMAIL defaults to null.)	user@abc-company.com
FROM_ADDRESS	Email Address to be used as 'Sender' (If null, SEND_EMAIL defaults to null.)	user@xyz-leasing.com
EMAIL_SUBJECT	Email Subject.	Customer Exposure
STYLESHEET	The stylesheet is used to format the xml data into an html document.	CustExposure.xsl

Methods supporting these parameters are Customer Exposure, Search Customer, and Customer Message Inquiry.

Example HTTP Client Request:

```
http://localhost:8080/BridgeWare/servlet/BridgeWare?_TYPE=SELECT_CUST_EXPO
SURE&HOST_NAME=InfoServer&USER_ID=dsiroot&PASSWORD=password&DATABASE_PATH=
/info/DATABASE&CUSTOMER_NUMBER=123456789
```

Web Servlet Response

The HTTP client will receive a well-formed XML document that contains the results of the request. Each XML document will conform to the following layout:

1. **<Header>**
 - a. **<ServletBuild>** Build number for servlet.
 - b. **<TransId>** A unique transaction Id
 - c. **<Status>** Execution status of web servlet request. A 'Success' in the header section indicated the request has passed component one validations.
2. **<Body>**
 - a. **<ContentHeader>** This section will return all the parameters sent to and processed by the BridgeWare UniBasic Method. For the parameters that are optional, the default used in the method will be returned.
 - b. **<ContentData>** If successful, the results of the method call. The content of this section is directly related to the method used.
 - c. **<ContentMessage>** Any Error or Warning messages generated by the BridgeWare UniBasic Method request will be returned here.

BridgeWare Servlet Installation

Servlet Installation

Note: All prerequisite products must be installed and configured prior to installing the servlet. See *Architecture Requirement & Recommendations* section.

1. Search the UniData Client CD (version 5.2 or greater) for the *asjava.zip* file. Unzip this file into %TOMCAT_HOME%\common\classes.
2. The following three files will be included in the BridgeWare product. Drop them into the %TOMCAT_HOME%\common\lib folder.
 - a. activation.jar
 - b. mail.jar
 - c. xercesImpl.jar
3. Place the BridgeWare.war file in the %TOMCAT_HOME%\webapps folder and restart Tomcat to install the servlet.
4. Use BridgeWare TestSuite to validate install.

Servlet Configuration

Review all the parameter settings in the bridgeware.properties file located in the %TOMCAT_HOME%\webapps\BridgeWare\WEB-INF folder.

1. BridgeWare leverages the Tomcat log directory found in %TOMCAT_HOME%\log. Three levels of error logging:
 - a. ERROR – Log only errors.
 - b. WARNING – Log errors and warnings.
 - c. DEBUG – Log all communications. CAUTION!
2. Two parameters dictate how BridgeWare will attempt to retry communication with UniData if the first attempt fails. The most common failure is due to no available UniData licenses.
 - a. Number of Retries
 - b. Retry Delay (in milliseconds)
3. Mail (smtp) server. This property is required if the servlet will be receiving request to email resulting xml data.
4. UniData Server and Database restrictions
 - a. Using this property, the servlet can be restricted from specific UniData servers.
 - b. Similar to the server property, the servlet can be configured to restrict integration with specific UniData databases.

Servlet Methods

Standard Request/Response Methods

1. Search Customer

TYPE: **BW_SEARCH_CUSTOMER**

Client requests a search for a specific customer entity level field providing a providing the search field, search string and type of search. The returning XML document will contain elements detailing unique elements about each customer that matches the search criteria.

2. Select Customer Exposure

TYPE: **BW_SELECT_CUST_EXPOSURE**

Client requests a customer exposure 'snapshot' by providing the InfoLease™ customer identifier (CCAN). The returning XML document will contain elements detailing that customer's current direct and indirect exposure. (Direct and Indirect exposure is defined below.) In addition to outstanding exposure, the method will return contract details if the 'Detail' parameter is passed.

Direct Exposure: Active contracts with a direct customer relationship.

Indirect Exposure: Active contracts where customer/CCAN is listed as a Principal, Guarantor, or some other association. Using the Indirect Option flag, the request can be limited to a subset of association codes. (See SELECT.LOOKUP.TABLE method for obtaining this lookup table.)

Additional Fields: Any additional fields located in the CS.MASTER, CS.DEFAULTS, LS.MASTER, LS.BILLING, LS.BROKER subfiles.

3. Select Customer Message Inquiry

TYPE: **BW_SELECT_CUST_MSG**

Client requests a complete or subset of customer messages for a given customer. By providing the InfoLease™ CCAN, the method will return either a complete or partial list of customer messages. These messages are typically entered through collection activity.

4. Select Lookup Table

TYPE: **BW_SELECT_LOOKUP_TABLE**

Client requests a specific InfoLease™ 'Lookup Table' (a list of predetermined codes and descriptions). The method will return the dataset for that table. Each row of the table will contain 2 values: the code followed by the description.

5. Select Next ID

TYPE: **BW_SELECT_NEXT_ID**

A number of InfoLease™ entities use a sequential numbering structure for the primary key. This method will return the next PKey used for the given entity level. Primarily this method can/will be used to identify the next available Customer Number/CCAN.

6. Select Zip Geo Code

TYPE: **BW_SELECT_ZIP_GEO_CODE**

Most installations of InfoLease integrate with Vertex Sales and Use Tax system. The Sales & Use tax system is comprised of two basic components; 1) GeoCoder and 2) Jurisdictional Rate information. Since Vertex stores all jurisdictional rate information under a Geo code, standard zip codes must be translated into the appropriate Geo code. This method can be called to determine the correct Geo code tax calculations.

7. Insert Customer Entity

TYPE: **BW_INSERT_CUSTOMER_ENTITY**

Client requests to create a new customer entity in InfoLease™. After validating the necessary required fields are present, InfoLease™ will return the new Customer Number, if one was not specified.

The method supports any field in either CS.MASTER, CS.DEFAULTS or CS.ADDL.DEFAULTS tables.

8. Insert Customer Message Input

TYPE: **BW_INSERT_CUST_MSG**

Client requests to store a new message history in InfoLease™ for a given customer. The method must receive; message text, a date/time stamp, and User ID or description. Message key will be assign by the routine.

9. Update Contract Entity

TYPE: **BW_UPDATE_CONTRACT_ENTITY**

Client requests to update a customer entity in InfoLease™. After validating the required fields are present, BridgeWare will return the status Customer update.

The method supports any field in either the LS.CADDR or LS.AADDR files.

10. Update (Reserve/Release) Contract Number

TYPE: **BW_UPDATE_CONTRACT_NUMBER**

Reserve - Client requests to reserve an InfoLease™ contract number. Since the contract number is often required for documentation prior to booking, it is necessary to reserve that number so that it is not assigned to another transaction.

Release - All credit applications that had reserved a number must release the number immediately prior to the credit application booking process.

11. Update Contract Filing

TYPE: **BW_UPDATE_CONTRACT_FILING**

Client requests to update a contract filing record in InfoLease™. After validating the required fields are present, BridgeWare will return the status of the Contract Filing update.

The method supports any field in the AS.UCC file.

12. Update Customer Entity

TYPE: **BW_UPDATE_CUSTOMER_ENTITY**

Client requests to update a customer entity in InfoLease™. After validating the required fields are present, BridgeWare will return the status of the Customer update.

The method supports any field in either CS.MASTER, CS.DEFAULTS or CS.ADDL.DEFAULTS tables.

13. Update Asset Entity

TYPE: **BW_UPDATE_ASSET_ENTITY**

Client requests to update an asset entity in InfoLease™. After validating the required fields are present, BridgeWare will return the status of the Asset update.

The method supports any field in either the AS.MASTER, AS.ADDL.MASTER, AS.USER or AS.ADDL.USER files.

14. Update Lookup Table

TYPE: **BW_UPDATE_LOOKUP_TABLE**

Client requests to create or update an InfoLease™ ‘Lookup Table’ (a list of predetermined codes and descriptions). The method will return the status of the request. Each row of the table will contain 2 values: the code followed by the description.

Note: When the Lookup Table type is ‘*Addressable*’ in InfoLease, the left square bracket ([]) can be used to specify the name/address data. Using the Vendor/Dealer table as an example, the following code=value data stream required:

```
1234561234=ACME Dealer[201 Main Street[Suite 100[[St.
Paul[MN[55125[[[[[D
```

The address processed in this example is:

```
Code:          1234561234
Name:          ACME Dealer
Address1:      201 Main Street
Address2:      Suite 100
Address3:
City:          St. Paul
State:         MN
Zip:           55125
Country:
Contact:
Phone:
Fax:
DVM Code:     Dealer
```

15. Insert Miscellaneous Invoice Record

TYPE: **BW_INSERT_MISC_INVD**

Client requests to create a Miscellaneous Invoice Record on a specific contract. The Misc GL Code provided must be set to Cash basis. In the following example, a Misc Invd charge for \$250.00 will be created on contract 001-0001234-000 using a GL code or “0010” and a Misc Description of “Documentation Fee” due in April 2007:

Example: “001-0001234-000|200704|0010|250.00|Documentation Fee”

16. Delete Miscellaneous Invoice Record

TYPE: **BW_DELETE_MISC_INVD**

Client requests to delete a Miscellaneous Invoice Record on a specific contract. The Misc GL Code associated with the Open Item must be set to Cash basis and there can be NO payment histories applied to this open item.

BridgeWare Bulletin Board Component

The remaining methods are part of the BridgeWare *Bulletin Board* component. Unlike the previous methods, these methods require the installation BridgeWare database triggers and configuration of BridgeWare registry file in the UniData. (See *BridgeWare Bulletin Board Installation and Configuration Guide*.)

Triggers will monitor specific entity activity in InfoLease. For example, Triggers can be set to monitor all User Lookup Table and Customer Entity changes. Each Trigger references the Bulletin Board Registry to determine if the change in question should be posted on the Bulletin Board.

17. Select Bulletin Board

TYPE: **BW_SELECT_BULLETIN_BOARD**

This method returns a list of all records that currently exist on the Bulletin Board that have not yet been processed. Since it is conceivable that the servlet may request another list of changes before it has received confirmation for the last submission from the Front End, each execution of this method requires a TransID to flag the records processed.

18. Update Bulletin Board

TYPE: **BW_UPDATE_BULLETIN_BOARD**

This method is used by the servlet to update change logs records posted on the bulletin board. Once a group of records has been processed by the **BW_SELECT_BULLETIN_BOARD** method, it stamps each record with a Transaction ID. If the servlet receives confirmation that the front end has successfully process those records, this method will remove them from the bulletin board using the 'DELETE' action. If the front end fails to receive the records or the servlet times out waiting for confirmation, this method will use the 'CLEAR' action to make those records again available to **BW_SELECT_BULLETIN_BOARD**.

Currently Bridge Bulletin Board supports Insert, Update, and Delete trigger events on the following entities:

1. User Lookup Tables (InfoLease File Name: PARAMETER)
2. Customer Entity Change Logs (InfoLease File Name: CS.CHNG.LOG)
3. Contract Entity Change Logs (InfoLease File Name: LS.CHNG.LOG)

Customer User Defined Method

19. Custom User Method

TYPE: **CUSTOM_USER_METHOD**

This method allows the customer to execute one or more of their own UniBasic subroutines.

Methods for “Help Desk” support

20. Select InfoLease UserIDs

TYPE: **BW_SELECT_IL_USERIDS**

This method allows returns a list of InfoLease User IDs found on the database. This method is designed to be used in conjunction with the “Update IL UserIDs” method to reset and clear User ID locks.

21. Update InfoLease UserIDs

TYPE: **BW_UPDATE_IL_USERIDS**

This method will perform one of three actions on a given InfoLease User ID. Those actions are:

1. CLEAR – Remove any port and record locks associated with this User ID.
2. ENABLE – If the User ID has been locked due to login failure attempts, this action will enable the User ID.
3. RESET – Reset the User ID password to null requiring the user to enter a new password the next time they login.

BridgeWare Diagnostic Utility

A standard feature of BridgeWare Integration Framework is a web-based development and testing utility. This utility will allow developers to quickly and easily integrate the Framework by:

1. Illustrating the required and optional fields for each method
2. Displaying the resulting XML document.
3. Test email functionality and configuration.

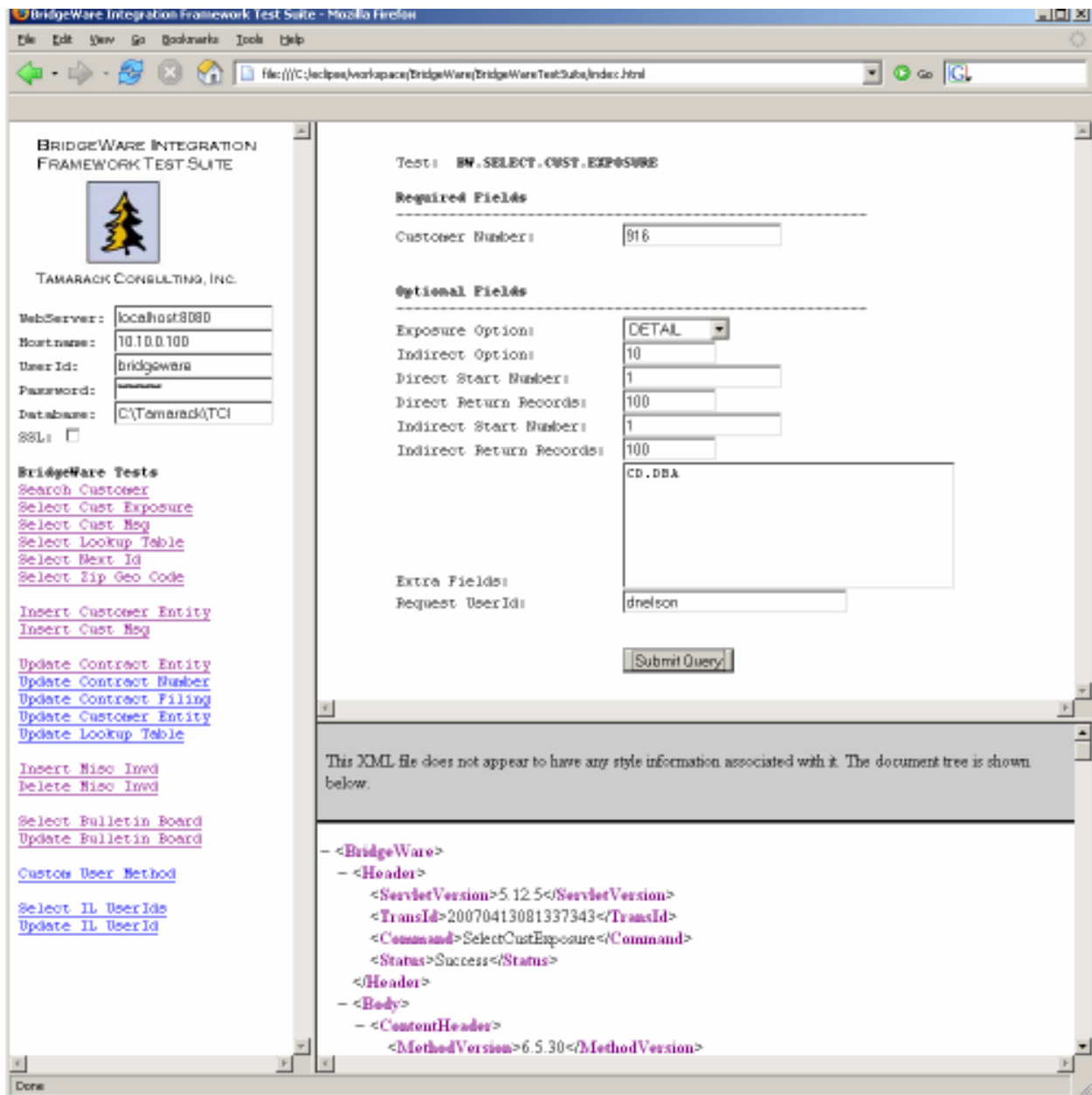


Figure 2 – Sample Diagnostic Utility View

Appendix

Table Names for SELECT_LOOKUP_TABLE method

Table	Code	Description
ACCOUNT TYPE	31	Account Type
ADJUSTMENT CODE	ADJ	Adjustment Data
ADMINISTRATIVE CODE	ADMIN	Administrative Code
AGGREGATE CODE TYPE	AGGR.CODE.TYPE	Aggregate Code
ASSET STATUS	ASSET.STATUS	Asset Status
ASSOCIATION	ASSOCIATION	Association
BANK	21	Base Rate Indicator
BANK CODE	36	Australia Banks
BANK.ADDRESS	BANK	. Bank Address
BLENDED INCOME PARAMETER	BLENDED.INCOME	Blended Income Parameter
BLENDED INCOME TYPE	BI.TYPE	Blended Income Type
BRANCH	24	Branch
BROKER	22	Broker Address
BUSINESS	27	Business
BUSINESS SEGMENT	BUS.SEG	Business Segement
BUSINESS TYPE	32	Business Type
BUSINESS UNIT	BUSINESS.UNIT	Business Unit
CHARGE TYPE	CHRG.TYPE	Open Item Charge Types
COLLATERAL	15	Collateral Code
CONTRACT STATUS	42	Contract Status
COUNTRY	45	Country Code
CREDIT SCORE DECISION	SCORE.DECISION	Credit Score Decision
CREDIT SCORE STATUS	SCORE.STATUS	Credit Score Status
CURRENCY CODE	CURRENCY	Currency Code
DEAL.VEN.MANUF	13	Vendor/Dealer
DEALER SERIES	DEALER.SERIES	Dealer Series
DEALER STATUS	DEALER.STATUS	Dealer Status
DEPARTMENT	26	Department
DISP PAYMENT TYPE	34	Disposition Payment Type
DISP/INVENT	35	Disposition/Inventory
EQUIPMENT CATEGORY	16	Equipment Category
FOLLOW-UP CODE	FOLLOW.UP	Follow Up
GENERAL LEDGER ACCOUNT	23	General Ledger Account
GUARANTEED RESIDUAL	43	Guaranteed Residual
HOLIDAY/WEEKEND	HOLIDAY.TBL	Holiday/Weekend

Table	Code	Description
IDC DESC	IDC.DESC	IDC Description
INSURANCE STATUS	41	Insurance Status
INSURANCE TYPE	40	Insurance Type
INSURANCE.AGENT	AGENT	Insurance Agent
INVOICE FORMAT	INVOICE.FORMAT	Invoice Format
LEGAL STATUS	LEGAL.S	Legal Status
LOCAL SIC CODE	LOCAL.SIC.CODE	Local SIC Code
OFFICE	12	Office
PAYMENT PLAN	29	Payment Plan
PERSONNEL	10	Personnel
POLICY STATUS	POLICY.STATUS	Policy Status
PRODUCT DISTRICT	PRODUCT.DIST	Product Distribution
PRODUCT FAMILY	PRODUCT.FAM	Product Family
PRODUCT LINE	39	Product Line
PRODUCT REGION	PRODUCT.REG	Product Reg
PRODUCT TYPE	PRODUCT.TYPE	Product Type
PROGRAM CONTROL	PROGRAM.CONTROL	Program Control
PROGRAM TYPE	28	Program Type
PROMOTION	30	Promotion
PROPERTY TAX STATUS	18	Property Tax Status
PURCHASE OPTION	PUR.OPT	Purchase Option
PURPOSE OF LOAN	PURPOSE.LOAN	Purpose Loan
QUOTE BUYOUT	QUOTE.BUYOUT	Quote Buyout
REASON CODE	14	Reason Code
RECOURSE CODE	RECOURSE	Recourse
REGION	19	Region
RELATIONSHIP	RELATIONSHIP	Relationship
REMIT TO	20	Remit To
SCORING CODE	SCORING.CODE	Scoring Code
SERIAL NUMBER	SERIAL.NUM	Serial Number
STATUS	33	Status
TAX DESCRIPTION	17	Tax Description
USAGE CODE	USAGE.CODE	Usage Code
VOID FACTURA REASON	FAC	Void Factura Reason

Error Messages and Possible Solutions

Error Message	Possible Solutions
_TYPE is required.	An argument list was not sent or the list does not contain a 'name=value' pair for the method name. Example: “_TYPE=SELECT_CUST_EXPOSURE”
_TYPE not recognized.	The method name is not a valid BridgeWare method.
UniSessionException in callUniData(): No RPC Connection active.	Unable to connect with the UniData/InfoLease server. 1) Use 'ping' to check hostname or IP address. 2) Use telnet to login into UniData server verify UniData is running.
UniSessionException in callUniData(): The user name provided is incorrect	The user id or password are incorrect.
UniSessionException in callUniData(): The account name supplied is not a valid account	Invalid UniData/InfoLease database