

THE 3000 News Wire

Independent Information to Maximize Your HP 3000
Making Data Liquid

BridgeWare: Bridging the Gap Between TurboIMAGE and the RDBMS

3000 Test Drive Road Report

Taurus Software, Inc.
410 Brewster Street
Redwood City, CA 94063

Tel: 650.482.2022
Fax: 650.482.2010
Email: sales@taurus.com
Web: www.taurus.com

Versions:

Warehouse 2.07.3637(Win)
Warehouse 2.07.3530 (MPE)
BridgeWare Studio 3.01.122005

BridgeWare is the ultimate data migration tool. It is both an Extraction, Transformation and Loading Tool (ETL) for bulk data movement and a real-time data synchronization tool. The BridgeWare Studio GUI Workbench simplifies the creation of the scripts that control the data movement without sacrificing the flexibility to customize the scripts. The scripting language that drives the data movement engines is extremely powerful with numerous built-in functions to control data conversion.

BridgeWare licensing ranges from \$14,000 to \$82,000 in four price tiers based on hardware platform power. One BridgeWare copy gives bi-directional real-time capture and data movement between TurboIMAGE and any supported RDBMS on any two supported hardware platforms. Additional platforms can be added at prices ranging from \$4,000 to \$15,000. BridgeWare may also be rented for migrations of less than one year. Maintenance on BridgeWare is priced at 20 percent of current list price per year.

Review by John Burke

BridgeWare, a product of Taurus Software and Quest Software (marketed by Taurus and Quest), seamlessly performs two easy to describe but extremely complex tasks.

First, BridgeWare is an Extraction, Transformation and Loading Tool (ETL) that simplifies the task of moving data between databases and files on MPE, Windows, Unix and Linux systems with a powerful scripting language and built-in functions for data conversion.

Second, BridgeWare is also a data synchronization tool. It can capture changes in any supported database on any supported platform and then replicate those changes to other databases on other systems either in real-time or at scheduled intervals.

This makes BridgeWare the ultimate data migration tool. Normally, I do not submit a vendor's advertising tag line for a TestDrive headline — but in this case, "making data liquid," describes so well what BridgeWare does that I couldn't avoid it. Where most migration tools have only been in production for months — if that — Warehouse, the core Taurus technology behind BridgeWare, dates from 1994. The Netbase engine from Quest, the technology behind real-time data movement from

TurboIMAGE databases, has been around even longer. This history means that BridgeWare is the granddaddy of all products that are moving TurboIMAGE data to and from an RDBMS. BridgeWare knows how to move data to and from

TurboIMAGE databases because it has been doing it for many years.

In a situation where TurboIMAGE is not involved, or where the real-time movement off TurboIMAGE is not required, Taurus Software sells a product called DataBridger.

DataBridger is BridgeWare without Quest's TurboIMAGE Change Detect module. For this Test Drive, even though I will not be using the Change Detect Module, I refer to the product as BridgeWare.

Features

BridgeWare consists of a Warehouse server and client module for both the source and target platforms and the BridgeWare Studio GUI Workbench running on a Windows system. BridgeWare Studio allows for the automation of the process of creating the scripts that control the actual data movement and conversion, but also gives you the flexibility to easily customize the movement and conversion process to specific customer requirements. For example, how you handle various conversions, such as date items, is limited only by your imagination. The scripting language that describes all transfers and conversions has numerous built-in functions for manipulating data.

Among the things you can do with

BridgeWare include:

- Support a Web Server environment on another system with TurboIMAGE data
- Build a data warehouse and update it on a scheduled basis

```
Scrcurl root, blackman:2000
begin data base custdb;
password:
10 READER:
20 WRITER:
item:
  cpy-code      . s4 :
  state-code    . s2 :
  maj-cpy-code  . s4 :
  tech-extn     . s4 :
  add-date      . s2 :
  mail-date     . s4 :
  hp-man        . s4 :
sets:
name: cpy-indx.a(10/20);
entry:
  cpy-code(2);
  capacity: 139;
name: state-indx.a(10/20);
entry:
  state-code(1);
  capacity: 11;
name: company-master.a(10/20);
entry:
  idkey(2);
  cpy-name;
  capacity: 199;
name: company-detail.d(10/20);
entry:
  idkey(company-master),
  cpy-code(cpy-indx),
  state-code(state-indx),
  sequence,
  source-cpy,
  gen-contact,
  qp-contact,
  tech-contact,
  address-1,
  address-2,
  city,
  state-name,
  zip-code,
  country;
  capacity: 380;
name: company-info.d(10/20);
entry:
  idkey(company-master),
  maj-cpy-code(cpy-indx),
  sequence,
  gen-telnum,
  qp-telnum,
  tech-telnum,
  gen-extn,
  qp-extn,
  tech-extn,
  add-date,
  mail-date,
  hp-man,
  select-arkr,
  line1;
  capacity: 380;
end;
```

Figure 1: Schema for the CUSTDB test database

```
OPEN IMAGESET IMAGE "custdb"
OPEN SQLDB ALLBASE "partdb"
DEFINE COMPANY_MASTER_REC USING SQLDB.COMPANY_MASTER
DEFINE COMPANY_DETAIL_REC USING SQLDB.COMPANY_DETAIL
DEFINE COMPANY_INFO_REC USING SQLDB.COMPANY_INFO
READ COMPANY_MASTER_R = TRACEDB.COMPANY_MASTER
SETVAR COMPANY_MASTER_REC.CPY_NAME = COMPANY_MASTER_R.CPY_NAME
COPY COMPANY_MASTER_REC TO SQLDB.COMPANY_MASTER
READ COMPANY_DETAIL_R = TRACEDB.COMPANY_DETAIL
FOR IDKEY = COMPANY_MASTER_R.IDKEY
  SETVAR COMPANY_DETAIL_REC = COMPANY_DETAIL_R
  SETVAR COMPANY_DETAIL_REC.CPY_CODE = COMPANY_DETAIL_R.CPY_CODE
  SETVAR COMPANY_DETAIL_REC.ADDRESS_1 = COMPANY_DETAIL_R.ADDRESS_1
  SETVAR COMPANY_DETAIL_REC.ADDRESS_2 = COMPANY_DETAIL_R.ADDRESS_2
  SETVAR COMPANY_DETAIL_REC.STATE_NAME = COMPANY_DETAIL_R.STATE_NAME
  SETVAR COMPANY_DETAIL_REC.CITY = COMPANY_DETAIL_R.CITY
  SETVAR COMPANY_DETAIL_REC.COUNTRY = COMPANY_DETAIL_R.COUNTRY
  SETVAR COMPANY_DETAIL_REC.ADD_DATE = COMPANY_DETAIL_R.ADD_DATE
  SETVAR COMPANY_DETAIL_REC.MAIL_DATE = COMPANY_DETAIL_R.MAIL_DATE
  SETVAR COMPANY_DETAIL_REC.HP_MAN = COMPANY_DETAIL_R.HP_MAN
  SETVAR COMPANY_DETAIL_REC.SELECT_ARCR = COMPANY_DETAIL_R.SELECT_ARCR
  SETVAR COMPANY_DETAIL_REC.LINE1 = COMPANY_DETAIL_R.LINE1
  COPY COMPANY_DETAIL_REC TO SQLDB.COMPANY_DETAIL
ENDFOR
READ COMPANY_INFO_R = TRACEDB.COMPANY_INFO
FOR IDKEY = COMPANY_MASTER_R.IDKEY
  SETVAR COMPANY_INFO_REC = COMPANY_INFO_R
  SETVAR COMPANY_INFO_REC.MAJ_CPY_CODE = COMPANY_INFO_R.MAJ_CPY_CODE
  SETVAR COMPANY_INFO_REC.CPY_NAME = COMPANY_INFO_R.CPY_NAME
  SETVAR COMPANY_INFO_REC.SOURCE_CPY = COMPANY_INFO_R.SOURCE_CPY
  SETVAR COMPANY_INFO_REC.DP_TELNUM = COMPANY_INFO_R.DP_TELNUM
  SETVAR COMPANY_INFO_REC.DP_EXTN = COMPANY_INFO_R.DP_EXTN
  SETVAR COMPANY_INFO_REC.TECH_TELNUM = COMPANY_INFO_R.TECH_TELNUM
  SETVAR COMPANY_INFO_REC.TECH_EXTN = COMPANY_INFO_R.TECH_EXTN
  SETVAR COMPANY_INFO_REC.ADD_DATE = COMPANY_INFO_R.ADD_DATE
  SETVAR COMPANY_INFO_REC.MAIL_DATE = COMPANY_INFO_R.MAIL_DATE
  SETVAR COMPANY_INFO_REC.HP_MAN = COMPANY_INFO_R.HP_MAN
  SETVAR COMPANY_INFO_REC.SELECT_ARCR = COMPANY_INFO_R.SELECT_ARCR
  SETVAR COMPANY_INFO_REC.LINE1 = COMPANY_INFO_R.LINE1
  COPY COMPANY_INFO_REC TO SQLDB.COMPANY_INFO
ENDFOR
```

Figure 2: Script to move data from TurboIMAGE to Allbase

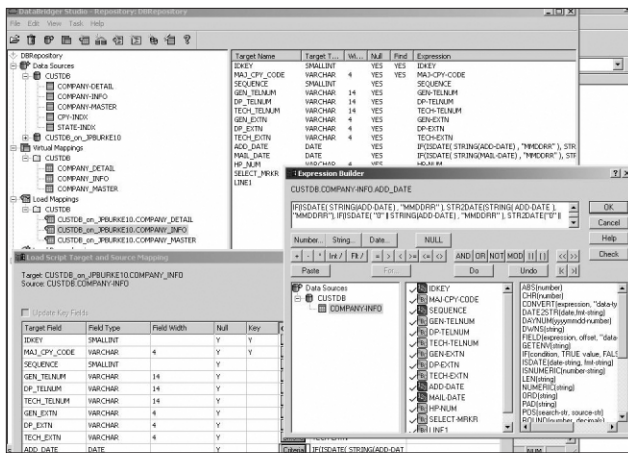


Figure 8: Data Conversion code

sion of the dates is being done (each becomes NUMERIC(6) in the Allbase table COMPANY_INFO). Of course whether you want to convert things like date or not depends upon the application. In part two I used Warehouse's scripting language and its powerful built-in functions to convert these integer format date values to DATETIME. Figure 3 (page 29) shows the results of applying the script.

Now let's look at using BridgeWare Studio GUI to move the same data off the HP 3000 to an ODBC-compliant database. In this case I used MySQL for convenience, but in a production environment

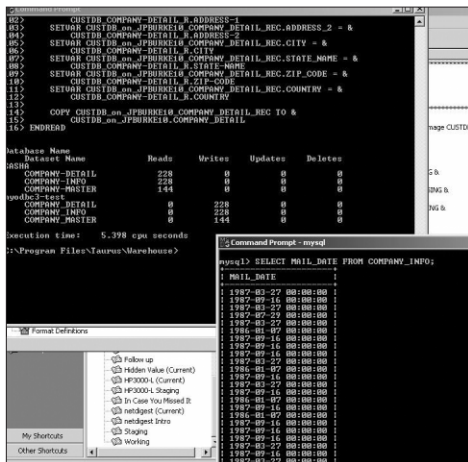


Figure 10: Running the load script and testing the data conversion

you would probably use Oracle, SQL Server or DB2. The first step upon entering Studio is to create the repository. This must be an ODBC-compliant database. Taurus suggests SQL Server, Oracle or Access for the repository. I chose Access.

Figure 4 (page 29) shows the Studio GUI window after an empty repository has been created. Figure 5 shows the entries for the source database detail dataset COMPANY-

INFO. Note this is the same as in the schema of Figure 1. Figure 6 shows some of the contents of the dataset by checking View Data. Notice how the date values are displayed. I was able to drag and drop to create the virtual mappings (Figure 7). In Figure 7 I've already deleted the two automatic masters.

Figure 8 gives an idea of how you can create the code to convert the date items from Z6 to DATE. Figure 9 shows the script generated by Studio. Figure 10 shows the results of running the script, and a quick check that the dates were converted the way we wanted.

Conclusion

I have been able to cover only a small fraction of what BridgeWare can do in bulk data movement, and I haven't even begun to talk about its capabilities of real-time data movement. However, my bottom line is that BridgeWare is a stable and very robust product with a long, successful history that can handle all of your data movement needs.

The product's powerful scripting language and built-in functions appear able to handle even the most tortured TurboIMAGE database design. This is not vaporware or beta software disguising as

the first release. The core technology behind BridgeWare (Taurus' Warehouse data movement and data conversion technology and Quest's TurboIMAGE Change Detect Process – from its Netbase product) has been proven over a number of years in high-end production environments.

BridgeWare has a reputation for being pricey and, indeed, it is not inexpensive (see the Road Report). Training and installation is \$5,000 for two days of on-site service and instruction for the IMAGE Change Detect Process. Data movement training is \$5,000 for two days of on-site for up to 10 people. I recommend the training because of the inherent complexi-

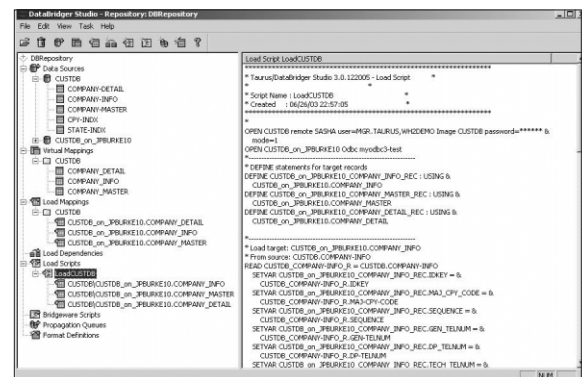


Figure 9: The load script+

ty of moving data. Travel and expenses are payable on both classes.

However, when you consider its capabilities combined with the value of the data it handles, you will agree with me that BridgeWare provides value comparable to its price.

John Burke is the founder of Burke Consulting and Technology Solutions (www.burke-consulting.com), which specializes in system management, consulting and outsourcing. He has over 25 years experience in systems, operations and development, is co-chair of SIGMPE, and has been writing regularly about HP 3000 issues for over 10 years. You can reach him at john@burke-consulting.com

