

BridgeWare: Bridging the Gap Between TurbolMAGE 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 TurbolMAGE 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

TurbolMAGE databases, has been around even longer. This history means that BridgeWare is the granddaddy of all prod-

ucts 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



Figure 1: Schema for the CUSTDB Fe test database

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.

called DataBridger.

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

OTHE SCHEME ALLANE "partner" STREE CONTRACT, DESTING SCHEME CONTRACT, LANSTER STREE CONTRACT, DESTING SCHEME, CONTRACT, LANSTER STREE CONTRACT, LANSTER, LANSTER STREE CONTRACT, LANSTER, LANSTER STREE CONTRACT, LANSTER, LANSTER SCHEME, CONTRACT, LANSTER, LANSTER SCHEME, CONTRACT, LANSTER, LANSTER SCHEME, CONTRACT, SCHEME, LANSTER SCHEME, CONTRACT, SCHEME, LANSTER SCHEME, CONTRACT, SCHEME, LANSTER SCHEME, CONTRACT, SCHEME, LANSTER SCHEME, SCHEME, SCHEME, CONTRACT, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, CONTRACT, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, CONTRACT, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, SCHEME, CONTRACT, SCHEME, SCHE	OPEN IMAG	EDB IMAGE "custdb"
DEFINE CONTANT_INTELLES : UNITE SILE CONTANT_INTEE SETHE CONTANT_INTELLES : UNITE SILE CONTANT_INTEE SETHE CONTANT_INTELLES : UNITE SILE CONTANT_INTE SETHE CONTANT_INTELLES : UNITE SILE CONTANT_INTE SETHE CONTANT_INTELLES : UNITE SILE CONTANT_INTE SETHE CONTANT_INTELLES : UNITE SILE CONTANT_INTEE SETHE CONTANT_INTELLES : UNITE CONTANT_INTEE SETHE CONTANT_INTELLES : UNITE CONTANT_INTEE SETHE CONTANT_INTELLES : UNITE CONTANT_INTEE SETHE CONTANT_INTELLES : UNITE CONTANT_INTELLES INTE CONTANT_INTELLES : UNITE CONTANT_INTELLES INTE CONTANT_INTELLES : UNITE CONTANT_INTELLES INTE CONTANT_INTELLES : UNITE CONTANT_INTELLES : UNITE SETTES CONTANT_INTELLES : UNITE CONTANT_INTELLES : UNITE INTES CONTANT_INTELLES : UNITES : CONTANT_I	OPEN SQLI	B ALLBASE "partsdbe"
SHTHE CONJUNCTION IN THE STATE	DEFINE CO	HPANY MASTER REC : USING SOLDR. COMPANY MASTER
DATENE CONTANT_INTE, I: UTIER SUBM.CONTANT_INTO DATENE CONTANT_INTE ; I: UTIER SUBM.CONTANT_INTO HITVA CONTANT_INTE ; I: UTIER CONTANT_NATTE ; HITVA CONTANT_INTE ; I: UTIER CONTANT_INTE ; I: UTIER CONTANT HITVA CONTANT_INTE ; I: UTIER CONTANT_INTE ; I: UTIER CONTANT HITVA CONTANT_INTE ; I: UTIER CONTANT_INTE ; I: UTIER CONTANT HITVA CONTANT_INTE ; I: UTIER CONTANT_INTE ; I: UTIER CONTANT HITVA CONTANT_INTE ; I: UTIER CONTANT_INTE ; I: UTIER CONTANT HITVA CONTANT_INTE ; I: UTIER CONTANT_INTE ; I: UTIER CONTANT HITVA CONTANT_INTE ; I: UTIER CONT	DEFINE CO	HPANY DETAIL PEC : USING SOLDE. COMPANY DETAIL
<pre>Had Constr. Marting 4: Touches. Constr. Castr Fittwa Constr. Marting 4: Constr. Marting Constr. Marting 4: Constr. Marting Constr. Marting 4: Constr. Marting Constr. Marting 4: Constr. Marting Had Constr. Marting 4: Constr. Marting 4: Constr. Marting 4: Constr. Marting Had Constr. Marting 4: Constr. Marting 4: Constr. Marting 4: Constr. Marting Had Constr. Marting 4: Constr. Marting 4: Constr. Marting 4: Constr. Marting Had Constr. Marting 4: Constr. Marting 4: Constr. Marting 4: Constr. Marting Had Constr. Marting 4: Constr. Marting 4: Constr. Marting 4: Constr. Marting Had Constr. Marting 4: Constr. Marting 4: Constr. Marting 4: Constr. Marting Had Constr. Marting 4: Constr. Martin</pre>	DEFINE CO	HPANY_INFO_REC : USING SOLDB.COMPANY_INFO
<pre>FFTUA CODEAT_DATEL_ACC - CODAT_DATEL_A FFTUA CODEAT_DATEL_ACC - CODAT_DATEL_A FFTUA CODEAT_DATEL_ACC - CODAT_DATEL_A FFTUA CODEAT_DATEL_ACC - CODAT_DATEL_ACC - CODAT_DATEL_A FFTUA CODEAT_DATEL_ACC - CODAT_DATEL_A FFTUA CODEAT_DATEL_ACC - CODAT_DATEL_A FFTUA CODEAT_OFFTUAL_ACC - CODAT_DATEL_ACC - CODAT_ FFTUA CODEAT_OFFTUAL_ACC - CODAT_DATEL_ACC - CODAT_ FFTUA CODEAT_OFFTUAL_ACC - CODAT_DATEL_ACC - CODAT_DATEL_ACC - CODAT_ FFTUA CODEAT_OFFTUAL_ACC - CODAT_OFFTUAL_ACC - CODAT_DATEL_ACC - CODAT_ FFTUA CODEAT_OFFTUAL_ACC - CODAT_OFFTUAL_ACC - CODAT_OFFTUAL_AC</pre>	PEAD COME	ANY_MASTER_R = IMAGEDB.COMPANY-MASTER
<pre>HTTML CODEART_IDATTE_QUEC.CVF_UDIT = CODEART_IDATTE_D.CVF-NOME COVE CODEART_UDIT_DATTE_DUEC.CODEART_IDATTE_DUECE HTML CODEART_IDATTE_DUECE.CODEART_IDATTE_DUECE HTML CODEART_IDATTE_DUECE HTML CODEART_IDATTE_DUECE HTMLS CODEART_IDATTE</pre>	SETUS	R COMPANY_MASTER_REC - COMPANY_MASTER_R
CONT COMPARY_DATEL : A TABLE COMPARY_DATES THE DOWNLY DATES, A TABLE COMPARY_DATES HELD COMPARY_DATES, A TABLE COMPARY_DATES HELD COMPARY_DATES, A TABLE COMPARY_DATES, A TABLE HELD COMPARY_DATES, A TABLE COMPARY_DATES, A TABLE COMPARY HELD COMPARY_DATES, A TABLE COMPARY_OFFICE, A TABLE COMPARY HELD COMPARY_DATES, A TABLE COMPARY_OFFICE, A TABLE COMPARY_THE HELD COMPARY_DATES, A TABLE COMPARY_OFFICE, A TABLE COMPARY_THE HELD COMPARY_DATES, A TABLE TO COMPARY_DATES, A TABLE TO HELD COMPARY_DATES, A TABLE TO COMPARY_DATES, A TABLE TO HELD COMPARY_THE S & THOUGH COMPARY_THE S & TO HELD COMPARY_THE S & THOUGH CO	SETVA	R COMPANY MASTER REC. CPY NAME = COMPANY MASTER R. CPY-NAME
<pre>Head containty pittall, 5 = Handmed.com/into-pittall { How limits - company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed Hitty = Company_pittall, 10 = Company_pittall, 10 = Handmed How limits - Company_Handmed Hitty = Company_Handmed How limits - Company_Handmed Hitty = Company_Handmed Hitty = Company_Handmed Handmed Hitty = Company_Handmed Handmed Hitty = Company_Handmed Hitty = Company_Handmed Hitty = Company_Handmed Handmed Hitty = Company_Handmed Handmed Hitty = Company_Handmed Handmed Hitty = Company_Handmed Handmed Handmed Handmed Handd</pre>	COPT	COMPANY_MASTER_REC TO SQLDB.COMPANY_MASTER
The loss - compary matrix 1 to the second se	READ	COMPANY_DETAIL_R = IMAGEDB.COMPANY-DETAIL 4
<pre>HTTML COMPANY_PITALL_REC = COMPANY_PITALL_DET_COMPANY_PITAL_DET_COM</pre>		FOR IDKEY = COMPANY_MASTER_R. IDKEY
<pre>HTTM & CORMAN, PARLI, MC. CTT, COM & CORMAN, PARLIE, C. CTT-COM HTTM & CORMAN, PARLIE, MC. CTT, COM & CORMAN, PARLIE, NOT-COM HTTM & CORMAN, PARLIE, MC. COM CONTER, YT & CORMAN, PARLIE, NORM-C-FTY HTTM & CORMAN, PARLIE, MC. CORMAN, CORMAN, PARLIE, NORM-C-FTY HTTM & CORMAN, PARLIE, MC. CORMAN, PARLIE, NORM-C-FTY HTTM & CORMAN, PARLIE, MC. ACOUNTS , TO CORMAN, PARLIE, NORM-C- HTTM & CORMAN, PARLIE, MC. ACOUNTS , TO CORMAN, PARLIE, NORM-C- HTTM & CORMAN, PARLIE, MC. ACOUNTS , TO CORMAN, PARLIE, NORM-C- HTTM & CORMAN, PARLIE, MC. ACOUNTS , TO CORMAN, PARLIE, NORM-C- HTTM & CORMAN, PARLIE, MC. ACOUNTS , TO CORMAN, PARLIE, NARM- HTTM & CORMAN, PARLIE, MC. ACOUNTS , TO CARANT, PARLIE, NARM- COM CORMAN, PARLIE, MC. CORMAN, PARLIE, N. TATH-MARK NEW ACOUNT, PARLIE, MC. CORMAN, PARLIE, N. TATH-MARK NEW ACOUNT, PARLIE, PARLIE, N. CORMAN, PARLIE, N. TATH-MARK NEW ACOUNT, PARLIE, PARLIE, N. CORMAN, PARLIE, N. TATH-MARK NEW ACOUNT, PARLIE, TO CORMAN, PARLIE, N. TATH-MARK NEW ACOUNT, PARLIE, PARLIE, N. CORMAN, PARLIE, N. TATH-MARK NEW ACOUNT, PARLIE, TO CORMAN, PARLIE, N. TATH-MARK NEW ACOUNT, PARLIE, TO CORMAN, PARLIE, N. TATH-MARK NEW ACOUNT, PARLIE, N. CORMAN, PARLIE, N. TATH- NEW ACOUNT, PARLIE, N. CORMAN, PARLIE, N. TATH- NEW ACOUNT, PARLIE, N. CORMAN, PARLIE, N. TATH-MARK NEW ACOUNT, PARLIE, N. CORMAN, PARLIE, N. TATH- NEW ACOUNT, PARLIE, N. CORMAN, PARLIE, N. TATH- NEW ACOUNT, PARLIE, N. CORMAN, PARLIE, N. CORMAN, PARLIE, N. TATH- NEW ACOUNT, PARLIE, PARLIE, N. CORMAN, PARLIE, N. CORMA</pre>	1	ETVAR COMPANY_DETAIL_REC - COMPANY_DETAIL_R
HITLS COMPANY DETAIL LESSING CONTACT COMPANY DETAIL A HILL-CON- HITLS COMPANY DETAIL LESSING CONTACT COMPANY DETAIL A HILL-CON- HITLS COMPANY DETAIL LESSING CONTACT COMPANY DETAIL A LOSSING- HITLS COMPANY DETAIL LESSING COMPANY DETAIL A LOSSING HITLS COMPANY DETAIL LESSING DETAILS COMPANY DETAIL A LOSSING HITLS COMPANY DETAILS DETAILS COMPANY DETAIL DETAILS HITLS COMPANY DETAILS DETAILS COMPANY DETAIL COMPANY DETAILS DETAILS HITLS COMPANY DETAILS DETAI	5	ETVAR COMPANY_DETAIL_REC.CPY_CODE = COMPANY_DETAIL_R.CPY-CODE
<pre>striks logsAvg piril, let.compt_pir/ logsAvg piril, source-ry mittak logsAvg piril, let.compt_piril_compt_piril, let.compt_piril, ittak logsAvg piril, let.compt_piril, compt_piril, let.compt_ ittak logsAvg piril, let.compt_piril, compt_piril, let.compt_ ittak logsAvg piril, let.compt_piril, let.compt_piril, let.compt_ ittak logsAvg piril, let.compt_piril, let.compt_piril, let.compt_ ittak logsAvg piril, let.compt_piril, let.compt_piril, let.compt ittak logsAvg piril, let.compt_piril, let.compt_piril, let.compt_piril, let.compt ittak logsAvg piril, let.compt_piril, let.compt_piril, let.compt_piril, let.compt ittak logsAvg piril, let.compt_piril, let.compt</pre>		ETVAR COMPANY_DETAIL_REC.STATE_CODE = COMPANY_DETAIL_R.STATE-CODE
<pre>HTMLS COMPANY_DIALL_DEC.COM_CONTACT - COMPANY_DIALL_A.COM-CONTACT HTMLS COMPANY_DIALL_DEC.COM_CONTACT - COMPANY_DIALL_A.COM_CONTACT HTMLS COMPANY_DIALL_DEC.COMPANY_COMPANY_DIALL_A.COMUSE- HTMLS COMPANY_DIALL_DEC.COMPANY_COMPANY_DIALL_A.COMUSE- HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_A.COMUSE- HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_A.COMUSE- COMY COMPANY_DIALL_DEC.COMPANY_DIALL_A.COM_COM HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_A.COM HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_A.COM HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_A.COM HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_A.COM HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_A.COM HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_DEC. HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_DEC.COM HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_DEC.FLUXH HTMLS COMPANY_DIALL_DEC.COMPANY_DIALL_DEC.FLUXH HTMLS COMPANY_DIALL_DEC.TLUXHS - COMPANY_DIALL_DEC.FLUXH HTMLS COMPANY_DIALL_DEC.COM_ANY_DIALL_DEC.FLUXH HTMLS COMPANY_DIALL_DEC.COM_ANY_DIALL_DEC.FLUXH HTMLS COMPANY_DIALL_DEC.TLUXHS - COMPANY_DIALL_DEC.FLUXH HTMLS COMPANY_DIALL_DEC.TLUXHS - COMPANY_DIALL_DEC.FLUXHS</pre>	1	ETVAR COMPANY_DETAIL_REC.SOURCE_CPY - COMPANY_DETAIL_R.SOURCE-CPY
HITLS COMPARY DETAIL LE LE CONTETT - COMPARY DETAIL : LE-CONTENT HITLS COMPARY DETAIL : LE LE CONTETT - COMPARY DETAIL : LE-CONTENT HITLS COMPARY DETAIL : LE LEOGHES : COMPARY DETAIL : LEOGHES-1 HITLS COMPARY DETAIL : LE LEOGHES : COMPARY DETAIL : LEOGHES-1 HITLS COMPARY DETAIL : LE LEOGHES : COMPARY DETAIL : LEOGHES-1 HITLS COMPARY DETAIL : LE LEOGHES : COMPARY DETAIL : HITLS COMPARY DETAIL : LE LEOGHES : COMPARY DETAIL : HITLS COMPARY DETAIL : LE LEOGHES : COMPARY DETAIL : HITLS COMPARY DETAIL : LE COMPARY DETAIL : HITLS COMPARY DETAIL : COMPARY DETAIL : COMPARY DETAIL : HITLS COMPARY DETAIL	5	ETVAR COMPANY_DETAIL_REC.GEN_CONTACT = COMPANY_DETAIL_R.GEN-CONTACT
STUTA COMPANY DIALL LOC TON COMPANY DIALL, TON-COMP STUTA COMPANY DIALL LOC TON COMPANY DIALL, TON-COMP STUTA COMPANY DIALL LOC STUTE COMPANY DIALL, TON-COMP STUTA COMPANY DIALL LOC STUTE COMPANY DIALL, LICENCEM STUTA COMPANY DIALL LOC STUTE COMPANY DIALL, LICENCEM STUTA COMPANY DIALL LOC STUTE COMPANY DIALL, LICENCEM STUTA COMPANY DIALL LOC STUTE COMPANY DIALL, STUT-COM STUTA COMPANY DIALL LOC STUTE COMPANY DIALL LOC STUTE STUTA COMPANY DIALL LOC STUTE COMPANY DIALL LOC STUTE STUTA COMPANY DIALL LOC STUTE COMPANY DIALL LOC STUTE STUTA COMPANY DIALL LOC TANKES COMPANY DIALLOC STUTE STUTA COMPANY DIALLOC STUTE COMPANY DIALUS DIALLOC STUTE STUTA COMPANY DIALLOC STUTE COMPANY DIALLOC STUTE COMPANY DIALLOC STUTE STUTA COMPANY DIALLOC STUTE COMPANY DIALLOC STUTE COMPANY DIALLOC		ETVAR COMPANY_DETAIL_REC.DP_CONTACT = COMPANY_DETAIL_R.DP-CONTACT
<pre>HTURA COMPANY_PITAL DE.ADORNEL - CONTANY_PITAL ADORNEL ITURA COMPANY_PITAL DE.ADORNEL - CONTANY_PITAL ADORNEL ITURA COMPANY_PITAL DE.C. TAL DANG - COMPANY_PITAL ADORNEL COV COMPANY_PITAL DE.C. DUBL.COMPANY_PITAL INFORMATION PITAL DE CONTANY_PITAL INFORMATION PITAL DE CONTANY_PITAL DE LAD-CONTANY PITAL COMPANY_PITAL DE CONTANY_PITAL DE LAD-CONTANY_PITAL DE LAD-CONTANY_PI</pre>		ETVAR COMPANY_DETAIL_REC.TECH_CONTACT - COMPANY_DETAIL_R.TECH-CONTAC
HITLS COMPARY_PITLI, DE. JOGGEN, COMPARY_PITLI, JOGGEN, JOGGEN	5	ETVAR COMPANY_DETAIL_REC.ADDRESS_1 = COMPANY_DETAIL_R.ADDRESS-1
HTTAL COMPART_METAL, MC. FILTE MAR COMPART_METAL, M. FATAT-MAR COMPCONDENT_FILTE, MC. FILTE MARK COMPART_METAL, M. FATAT-MAR COMPCONDENT_FILTE, MC. COMPART_METAL, M. FATAT-MARK ENGLAR NO. 1997 COMPART_METAL, MC. COMPART_METAL NO. 1997 COMPART_METAL, MC. COMPART_METAL NO. 1997 COMPART_METAL, MC. COMPART_METAL, MC. MARK. NO. 1997 COMPART_METAL, MC. COMPART_METAL, MC. M. M. MARK. NO. 1997 COMPART_METAL, MC. COMPART_METAL, MC. M.		ETVAR COMPANY_DETAIL_REC.ADDRESS_2 = COMPANY_DETAIL_R.ADDRESS-2
HTTAS COMPANY_DELL, MC. ELE COMPANY_DELLA, L.E.C. COMPANY_DELLA, L.E.C. COMPANY_DELLA, L.E.C. COMPANY_DELLA, L.E.C. COMPANY_DELLA, L.E.C. COMPANY_LINE O, L.E.C. COMPANY_LINE O, L.E.C. COMPANY_LINE O, L.E.C. COMPANY_LINE O, L.E.C. C.C. COMPANY_LINE O, L.E.C. C.C. COMPANY_LINE O, L.E.C. C.C. C.C. C. C. C. C. C. C. C. C. C.	1	ETVAR COMPANY_DETAIL_REC.STATE_NAME = COMPANY_DETAIL_R.STATE-NAME
COTY CORLARY FIRTLE, MC TO SQUB.COMMAT_STATL EXPLAN READ COMMAND INF	5	ETVAR COMPANY_DETAIL_REC.ZIP_CODE = COMPANY_DETAIL_R.ZIP-CODE
EXCREMENT FILE COMPANY [INFO] * TILGERS.CONFLATT-THFO 4 FOR INST - COMPANY [INFO] & COMPANY [INFO] HITLS COMPANY [INFO] & COMPANY [INFO] HAI-CVY-CONF HITLS COMPANY [INFO] & COMPANY [INFO] HAI-CVY-CONF HITLS COMPANY [INFO] & COMPANY [INFO] & COMPANY [INFO] HITLS COMPANY [INFO] & COMPANY [INFO] & COMPANY [INFO] & COMPANY HITLS COMPANY [INFO] & COMPANY [INFO] & COMPANY [INFO] & COMPANY HITLS COMPANY [INFO] & COMPANY [INFO] & COMPANY [INFO] & COMPANY HITLS COMPANY [INFO] & COMPANY [INFO	c	COPY COMPANY_DETAIL_REC TO SQLDB.COMPANY_DETAIL
BLD CORDARY JUNE 3 * THLEEDS CORDARY THUTO 6 1961 1087 • CORDARY JUNE 3. LODRY SITUA CORDARY JUNE 365 • CORDARY JUNE 3. MAIN-CVY-CODE SITUA CORDARY JUNE 365 • CORTARY JUNE 3. MAIN-CVY-CODE SITUA CORDARY JUNE 365 • CORTARY JUNE 3. MAIN-CVY-CODE SITUA CORDARY JUNE 365 • CORDARY JUNE 3. MAIN-CVM-CODE SITUA CORDARY JUNE 365 • CORDARY JUNE 365 • CORDARY JUNE 3. MAIN-CVM- SITUA CORDARY JUNE 365 • CORDARY JUNE 365 • CORDARY JUNE 365 • CORDARY SITUA CORDARY JUNE 365 • CORDARY JUNE 365 • CORDARY JUNE 365 • CORDARY SITUA CORDARY JUNE 365 • CORDARY JUNE 3. MAIN SITUATION	ENDRE	AD .
YOR, DIGKY - CORMAY, MATTH, J., DORY HITAL CORMAY, JUNY J. C. CORMAY, C. CORMAY, JUNY G. M.JCWY-COD 20100000000000000000000000000000000000	READ	COMPANY_INFO_R = IMAGEDB.COMPANY-INFO 4
STUAR COMMANT, HITO, DE C. COMMANT, HITO, D. STUAR COMMANT, HITO, DE C. CANANT, HITO, D. HLJ-CYT-COLD STUAR COMMANT, HITO, SEL, CAN, CANANT, AND AND AND AND AND STUAR COMMANT, HITO, SEL, CAN, CHANNA, CANANT, JAN, D. H.J-CHANNA STUAR COMMANT, HITO, SEL, CAN, CHANNA, HITO, S. CHANAT, HITO, S. STUAR COMMANT, HITO, SEL, DE, KITH COMMANT, HITO, S. CHE-FEITH STUAR COMMANT, HITO, SEL, DE, KITH COMMANT, HITO, S. CHE-FEITH STUAR COMMANT, HITO, SEL, DE, KITH COMMANT, HITO, S. CHE-FEITH STUAR COMMANT, HITO, SEL, DE, KITH COMMANT, HITO, S. CHE-FEITH STUAR COMMANT, HITO, SEL, DE, KITH COMMANT, HITO, S. CHE-FEITH STUAR COMMANT, HITO, SEL, DE, KITH COMMANT, HITO, S. CHE-FEITH		FOR IDKEY - COMPANY_MASTER_R. IDKEY
SITUA COMPATING DEC.MJ. CV. (DOF COMPATING D. MJCV-COM SITUA COMPATING DEC.MJ. TANDA - COMPATING D. J. CMTAIMA SITUA COMPATING DEC.DJ. TANDA - COMPATING D. J. CMTAIMA SITUA COMPATING DEC.NGC TAIMA - COMPATING D. TACH-TAIMA SITUA COMPATING DEC.KGC TANG COMPATING D. TACH-TAIMA SITUA COMPATING DEC. SEX FOR COMPATING D. COMPATING SITUA COMPATING DEC. SEX FOR COMPATING D. COMPATING SITUA COMPATING DEC. SEX FOR COMPATING D. D. FORM	5	ETVAR COMPANY_INFO_REC = COMPANY_INFO_R
SETVAR COMPANY_INFO_REC. CHI_TALMUM - COMPANY_INFO_R.CHN-TALMUM SETVAR COMPANY_INFO_REC. DF IEANUM = COMPANY_INFO_R.LHN-TALMUM SETVAR COMPANY_INFO_REC. TECH_TALMUM = COMPANY_INFO_R. CHN-TALMUM SETVAR COMPANY_INFO_REC. CHI_CHI_COMPANY_INFO_R. CHN-EXTM SETVAR COMPANY_INFO_REC. DF INT = COMPANY_INFO_R. CHN-EXTM		ETVAR COMPANY_INFO_REC.MAJ_CPY_CODE = COMPANY_INFO_R.MAJ-CPY-CODE
SITUAR COMPANY_INFO_BEC.PP_TELAUTH = COMPANY_INFO_P.D=TELAUTH SITUAR COMPANY_INFO_BEC.TECHTY TELAUTH = COMPANY_INFO_P.GF.TECHTYLININ SITUAR COMPANY_INFO_BEC.FR_UCTH = COMPANY_INFO_P.GEM=SCTH SITUAR COMPANY_INFO_BEC.FP_DETH = COMPANY_INFO_P.GF.THYLININ	1	ETVAR COMPANY_INFO_REC.GEN_TELNUM - COMPANY_INFO_R.GEN-TELNUM
SETVAR COMPANY_INFO_REC. THEN TELMEN = COMPANY_INFO_R. THEN TELMEN SETVAR COMPANY_INFO_REC. GEN_EXIN = COMPANY_INFO_R.GEN-EXIN SETVAR COMPANY_INFO_REC.DY_EXIN = COMPANY_INFO_R.DP-EXIN	8	ETVAR COMPANY_INFO_REC.DP_TELNUM = COMPANY_INFO_P.DP-TELNUM
SETVAR COMPANY_INFO_REC.GEN_EXIN = COMPANY_INFO_R.GEN-EXIN SETVAR COMPANY_INFO_REC.DP_EXIN = COMPANY_INFO_R.DP-EXIN		ETVAR COMPANY_INFO_REC.TECH_TELNUM = COMPANY_INFO_R.TECH-TELNUM
SETVAR COMPANY_INFO_REC.DP_EXTN = COMPANY_INFO_R.DP-EXIN	1	ETVAR COMPANY_INFO_REC.GEN_EXIN - COMPANY_INFO_R.GEN-EXIN
	5	ETVAR COMPANY_INFO_REC.DP_EXIN = COMPANY_INFO_R.DP-EXIN
SETVAR COMPANY_INFO_REC.TECH_EXTN = COMPANY_INFO_R.TECH-EXTN		ETVAR COMPANY_INFO_REC.TECH_EXTN = COMPANY_INFO_R.TECH-EXTN
SETVAR COMPANY_INFO_REC.ADD_DATE = COMPANY_INFO_R.ADD-DATE	1	ETVAR COMPANY_INFO_REC.ADD_DATE - COMPANY_INFO_R.ADD-DATE
SETVAR COMPANY_INFO_REC.MAIL_DATE = COMPANY_INFO_R.MAIL-DATE	8	ETVAR COMPANY_INFO_FEC.MAIL_DATE = COMPANY_INFO_F.MAIL-DATE
SETVAR COMPANY_INFO_REC.HP_NUM = COMPANY_INFO_R.HP-NUM	-	ETVAR COMPANY_INFO_REC.HP_NOM = COMPANY_INFO_R.HP-NOM

Figure 2: Script to move data from data. *TurboIMAGE to Allbase* Amor

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



Figure 3: Copying TurboIMAGE data into Allbase

• Access and move archived historical MPE data as needed

• Migrate data to new or ported applications on other systems in a staged fashion

• Synchronize data across platforms for running old and new applications in parallel during testing

• Synchronize data across platforms running applications you intend to keep on MPE that require data transfer to and from other systems

as bot non root nop											
3100日旬前街区	6 1	8									
DBRepository	IDKEY	MA	SE	GEN-TELNUM	DP-TELNUM	TECH-TELNUM	G	D	TE ADD-D	. MAIL	HP
P Data Sources	1070	A128	1	415-952-4400	415-952-4400				72382	32787	0001
P- CUSTOR	1736	A154	1	213-846-3348					81982	91687	0002
COMPANY OFTAT	10009	A187	1	916-273-6176					11784	32787	
COMPANYING	1795	5286	1		408-847-1414			300	81982	72987	0002
E COPPARTNER	4170		1		301-948-0922				31283	32787	0001
COMPANY-MASTER	11518		1	415-938-7600					72985	10786	
CPY-INDX	825	A179	1		301-694-1009				71982	91687	0003
STATE-INDX	10703		1		415-443-6300			422	91284	91687	0001
E-B CUSTD8_on_JP8URKE10	11523		1	415-326-8900	415-326-8066				72985	91687	0003
The Virtual Mappings	3283		1	415-989-8226					120682	32787	0001
STR Load Maccines	11521		1	714-621-0581					72985	10786	
C Load Dependencies	4694		1	714-621-0581					32183	91687	0002
All Lood Dependencies	5029	A178	1	408-727-8161					81183	32787	0001
All road scripts	925		1		301-468-5639				71982	91687	0003
P Bridgeware Scripts	10970		1	415-525-5070					12385	10786	
Propagation Queues	10064		1		801-350-6481	801-350-6488			20284	91687	0002
Tormat Definitions	11565		1	313-540-7550					72985	10786	
	3472	A215	1		301-730-7222				22483	91687	0004
	3170		1		818-340-4041	818-340-4041			111582	91687	0002
	4912	5106	1			301-765-7005			52783	91687	0002
	11051		1		503-294-8505				32585	32787	0001
	9844		1	408-867-2953					120283	91687	0001
	2037		1		415-273-6083				81982	92787	

Figure 6: Data Conversion code

System Requirements and Supported Environment

The Warehouse server and client portions of BridgeWare are supported on the MPE, Windows, Unix, and Linux platforms. BridgeWare supports many database types, including TurboIMAGE, DB2, Oracle, SQL Server and ODBC-compliant databases as well as file types such as KSAM, flat and delimited.

Installation

I installed the Warehouse server and client software on both an HP 3000 running MPE/IX 6.5 and on a Windows 2000 Professional system. The BridgeWare Studio GUI was installed on the same Windows 2000 Professional system. In a production environment, BridgeWare Studio would normally be installed on a Windows machine separate from either the source or target systems. Note, however, BridgeWare Studio does not require a dedicated platform.

Installation went smoothly and only required a few minutes. For security reasons you need to create an authorization file for each server (default is NO access), but this is all documented in the manuals. In my case I allowed any user on the PC with BridgeWare Studio installed (specified either by name or IP address) to connect to the server on the HP e3000, but no one else. The security module is very easy to set up, but also very powerful.

Documentation

Printed documentation consists of a 36-page Warehouse Tutorial, a 114-page Warehouse User Guide, and a 369-page Warehouse Reference Manual. While extremely complete, the material is somewhat dated since some program names and procedures have changed with newer releases of Warehouse. Having the manuals available, as DOC, PDF or HTML files in addition to the printed manuals would help their usability.

The Warehouse Tutorial is excellent and should be carefully followed as part of the installation. All three manuals are chock full of examples, a real plus in my estimation.

> The BridgeWare Studio documentation, while again very complete, is only available online in Windows Help format, which may or may not be an issue depending upon your preferences. I am old-school, and prefer something more substantial. I found it difficult to follow the

steps in the online tutorial while simultaneously applying them to my test case. I ended up printing out the tutorial so I could make notes as I went along. The documentation does, however, cover everything you need to process reasonably complex data movements.

The BridgeWare documentation is generally excellent. However, I would recommend purchasing the on-site training, even if you plan just to do bulk data moves — but especially if you are planning to use the IMAGE Change Detect Process for real-time data movement.

In a TestDrive situation I always try to figure everything out myself, relying as little as possible on handholding from the vendor, because this really tests the usability of the product and its documenta-



ation, Figure 5: COMPANY-INFO invalu- detail description able data to protect and with all the com-

able data to protect and with all the complexities that real data can present, it is best to get formal training.

Let's take it out for a spin

My testing was composed of two parts. First, I followed the Warehouse Tutorial, loading data archived by Warehouse into an IMAGE database, copying it to another TurboIMAGE database while changing some of the data. Then I moved it to an Allbase database on the same HP e3000. All this was done without the BridgeWare Studio GUI, using only the Warehouse scripting language and the Warehouse client.

The second portion of my testing consisted of using the TurbolMAGE database from the Warehouse Tutorial and moving it to an ODBC-compliant database on a Windows system. I used the BridgeWare Studio GUI to create the tables, the script and to modify the script to change some of the default conversions.

Figure 1 (page 28) shows the key portions of the CUSTDB schema for the database used in both parts of the Test Drive. Note in particular the two dates

> are represented as zoned decimal. The format is MDDYY; for example, March 15, 1985 is displayed as 31585. This is a simple database with two automatic masters, one manual master and two detail sets. In converting to an RDBMS, we are going to drop the two automatic

masters and end up with three tables. Figure 2 shows part of the script used to move data from the CUSTDB TurboIMAGE database to an existing Allbase database. Note that no conver-



Figure 7: Virtual mapping of COMPANY-INFO to the target RDBMS



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 TurbolMAGE 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 onsite 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-

DataBridger Studio - Repository: DURepository	للم
Comparison of the second	Lief Ser (LookOUTIO L

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

Reprinted with permission of The 3000 NewsWire, 11702 Buckingham Road, Austin, TX 78759. Volume 9, Issue 10.