

Taurus Software News  
July 2004

Welcome to this edition of Taurus Software News, your insider's guide to the Taurus family of products. Taurus Software News is distributed worldwide to people using technologies within the Taurus product line.

In this issue:

1. Warehouse News - redirecting output
2. Studio News - piping scripts
3. Ecomedate News - Jeff Cox's presentation
4. Company News - new people
5. Product Uses: Application Synchronization?
6. Product Uses: Oracle to SQLServer? SQLServer to DB2?
7. Events and Seminars
8. Web Site Picks for this Issue

1. Warehouse, BridgeWare's and DataBridger's Data Movement Engine, News  
Current release: 3.00.3100  
Available platforms: Windows, HP3000, HP-UX, RS6000, Linux, Sun, SGI  
Release notes: <http://www.taurus.com/support/releasenotes.htm>

Many customers, through the use of our products, are experiencing either a new database or new platform for the very first time. This month, we are going to cover running Warehouse on various platforms and redirecting the output to files for review.

The Warehouse client program is exactly the same code on every platform. It has the same functionality and all the commands and statements work exactly the same. Good news! Once you learn Warehouse, there will be no special set of commands that you need to know to use Warehouse on another platform. The location of the Warehouse client is determined by where your systems administrator decides to put our product. However the name of the client is the same for any given platform. Warehouse is called:

wh.whii - MPE  
wh.exe - Windows  
warehouse - Unix (check the README file if multiple versions of Oracle are supported on your platform)

To redirect output on Windows:

```
wh SCRIPT >OUTFILE 2>&1
```

The above will run Warehouse and run SCRIPT and put the \$STDLIST and error listings (WHWARN and WHERR) to the same output file, OUTFILE.

In Unix for the csh and bash shell, you can redirect output as follows:

```
wh SCRIPT >&OUTPUT
```

In the ksh and bourne shells, you can redirect output as follows:

```
wh SCRIPT >OUTPUT 2>&1
```

On MPE, you can redirect output by:

```
wh.whii.taurus;info="SCRIPT" >OUTFILE
```

Have a topic about Warehouse you want featured in our newsletter? Send your idea to support@taurus.com

## 2. Studio, GUI Development Workbench, News

Current release: 3.1.089200

Supported OS: Windows 98, Windows ME, Windows NT 4.0, Windows XP, Windows 2000+

If I have learned anything about moving data, there a million different ways to get the data from A to B. In this issue, we examine moving data the absolute faster way possible in the Oracle environment, SQLLoader Direct Path.

Available in a release in December 2003, a new feature was added to allow the generation of the SQLLoader pipe scripts. SQLLoader pipe scripts are generated from load mappings. To generate a pipe script, right click on the load mapping and a dialog box will appear.

The script will be kept where you designate in the script file name box. You can control the name of the log file and committate through the appropriate fields on the dialog box.

The output of the process is a text file which contains the unix shell script, the Oracle control file, and the Warehouse script. Below is the resulting file for our mapping:

```
#!/bin/sh
#
#   SQL*Loader Pipe Script
#
#   Taurus/DataBridger Studio 3.1.089200 -- 05/20/04 15:49:51
#
#   Source Database:  ecitm
#       Source Table:  STYLE-HEADER
#   Source Database:  ecdw
#       Target Table:  SV_MACITM_STYLE_HEADER
#
#####
cat >x_sv_macitm_style_header.ctl <<ENDOFDATA
load data
infile 'x_sv_macitm_style_header'
insert into table X.SV_MACITM_STYLE_HEADER
(
  CHG_LOG_ID          position (  1:  39) INTEGER EXTERNAL,
  SRC_SYS_CHG_LOG_ID position ( 41:  79) INTEGER EXTERNAL NULLIF (40:40)='N',
  SRC_SYS_TRANS_DT   position ( 81:  99) DATE "MM/DD/YYYY HH24:MI:SS" NULLIF
(80:80)='N' ,
  SRC_SYS_PROC_NM    position (101:130) CHAR NULLIF (100:100)='N' ,
  SRC_SYS_SESSION_NM position (132:161) CHAR NULLIF (131:131)='N' ,
  SRC_SYS_USER_NM    position (163:192) CHAR NULLIF (162:162)='N' ,
  SRC_SYS_GRP_NM     position (194:223) CHAR NULLIF (193:193)='N' ,
  SRC_SYS_ACCT_NM    position (225:254) CHAR NULLIF (224:224)='N' ,
  SRC_SYS_SESSION_ID position (256:271) INTEGER EXTERNAL NULLIF
(255:255)='N' ,
  SRC_CD             position (272:301) CHAR,
```

```

ACTION_CD                position ( 302: 309) CHAR,
CSI_TYPE_CD              position ( 311: 318) CHAR NULLIF (310:310)='N',
STYLE_ID                 position ( 320: 331) CHAR NULLIF (319:319)='N',
DESCRIPTION              position ( 333: 382) CHAR NULLIF (332:332)='N',
STY_CD                   position ( 384: 403) CHAR NULLIF (383:383)='N',
LONG_NM                  position ( 405: 484) CHAR NULLIF (404:404)='N',
B_STYLE_ID               position ( 486: 497) CHAR NULLIF (485:485)='N',
B_DESCRIPTION            position ( 499: 548) CHAR NULLIF (498:498)='N',
B_STY_CD                 position ( 550: 569) CHAR NULLIF (549:549)='N',
B_LONG_NM                position ( 571: 650) CHAR NULLIF (570:570)='N'
)

```

ENDOFDATA

#

#####

cat >x\_sv\_macitm\_style\_header.whs <<ENDOFDATA

#

OPEN ecitm remote copper.taurus.com user=MGR.RTLPOV,MACSDATA &

Image macitm pass=i &

mode=5

OPEN x\_sv\_macitm\_style\_header TEXT x\_sv\_macitm\_style\_header.dat

SET COMMITRATE 400

FORMAT X\_SV\_MACITM\_STYLE\_HEADER\_FMT

```

CHG_LOG_ID                : SIGNED(38)           // ( 1: 39)
SRC_SYS_CHG_LOG_ID_NIND   : CHAR(1)           // ( 40: 40)
SRC_SYS_CHG_LOG_ID        : SIGNED(38)           // ( 41: 79)
SRC_SYS_TRANS_DT_NIND     : CHAR(1)           // ( 80: 80)
SRC_SYS_TRANS_DT          : CHAR(19)           // ( 81: 99)
SRC_SYS_PROC_NM_NIND      : CHAR(1)           // (100:100)
SRC_SYS_PROC_NM           : CHAR(30)           // (101:130)
SRC_SYS_SESSION_NM_NIND   : CHAR(1)           // (131:131)
SRC_SYS_SESSION_NM        : CHAR(30)           // (132:161)
SRC_SYS_USER_NM_NIND      : CHAR(1)           // (162:162)
SRC_SYS_USER_NM           : CHAR(30)           // (163:192)
SRC_SYS_GRP_NM_NIND       : CHAR(1)           // (193:193)
SRC_SYS_GRP_NM            : CHAR(30)           // (194:223)
SRC_SYS_ACCT_NM_NIND      : CHAR(1)           // (224:224)
SRC_SYS_ACCT_NM           : CHAR(30)           // (225:254)
SRC_SYS_SESSION_ID_NIND   : CHAR(1)           // (255:255)
SRC_SYS_SESSION_ID        : SIGNED(15)         // (256:271)
SRC_CD                     : CHAR(30)           // (272:301)
ACTION_CD                  : CHAR(8)           // (302:309)
CSI_TYPE_CD_NIND          : CHAR(1)           // (310:310)
CSI_TYPE_CD                : CHAR(8)           // (311:318)
STYLE_ID_NIND              : CHAR(1)           // (319:319)
STYLE_ID                   : CHAR(12)          // (320:331)
DESCRIPTION_NIND           : CHAR(1)           // (332:332)
DESCRIPTION                 : CHAR(50)          // (333:382)
STY_CD_NIND                : CHAR(1)           // (383:383)
STY_CD                     : CHAR(20)          // (384:403)
LONG_NM_NIND               : CHAR(1)           // (404:404)
LONG_NM                     : CHAR(80)          // (405:484)
B_STYLE_ID_NIND            : CHAR(1)           // (485:485)
B_STYLE_ID                  : CHAR(12)          // (486:497)
B_DESCRIPTION_NIND         : CHAR(1)           // (498:498)
B_DESCRIPTION               : CHAR(50)          // (499:548)
B_STY_CD_NIND              : CHAR(1)           // (549:549)

```

```

        B_STY_CD                : CHAR(20)                // ( 550: 569)
        B_LONG_NM_NIND         : CHAR(1)                 // ( 570: 570)
        B_LONG_NM              : CHAR(80)                // ( 571: 650)
END
DEFINE PIPEREC : FORMAT X_SV_MACITM_STYLE_HEADER_FMT

* Define record for all fields that can be null
DEFINE NULREC : RECORD
        SRC_SYS_CHG_LOG_ID     : ORACLE NUMBER(38) ALLOW NULLS
        SRC_SYS_TRANS_DT       : ORACLE TIMESTAMP ALLOW NULLS
        SRC_SYS_PROC_NM        : ORACLE VARCHAR2(30) ALLOW NULLS
        SRC_SYS_SESSION_NM     : ORACLE VARCHAR2(30) ALLOW NULLS
        SRC_SYS_USER_NM        : ORACLE VARCHAR2(30) ALLOW NULLS
        SRC_SYS_GRP_NM         : ORACLE VARCHAR2(30) ALLOW NULLS
        SRC_SYS_ACCT_NM        : ORACLE VARCHAR2(30) ALLOW NULLS
        SRC_SYS_SESSION_ID     : ORACLE NUMBER(15) ALLOW NULLS
        CSI_TYPE_CD            : ORACLE VARCHAR2(8) ALLOW NULLS
        STYLE_ID               : ORACLE CHAR(12) ALLOW NULLS
        DESCRIPTION             : ORACLE CHAR(50) ALLOW NULLS
        STY_CD                  : ORACLE VARCHAR2(20) ALLOW NULLS
        LONG_NM                 : ORACLE VARCHAR2(80) ALLOW NULLS
        B_STYLE_ID              : ORACLE CHAR(12) ALLOW NULLS
        B_DESCRIPTION           : ORACLE CHAR(50) ALLOW NULLS
        B_STY_CD                : ORACLE VARCHAR2(20) ALLOW NULLS
        B_LONG_NM               : ORACLE VARCHAR2(80) ALLOW NULLS
END

* Initialize null indicators for all fields that can be null
SETVAR PIPEREC.SRC_SYS_CHG_LOG_ID_NIND = " "
SETVAR PIPEREC.SRC_SYS_TRANS_DT_NIND = " "
SETVAR PIPEREC.SRC_SYS_PROC_NM_NIND = " "
SETVAR PIPEREC.SRC_SYS_SESSION_NM_NIND = " "
SETVAR PIPEREC.SRC_SYS_USER_NM_NIND = " "
SETVAR PIPEREC.SRC_SYS_GRP_NM_NIND = " "
SETVAR PIPEREC.SRC_SYS_ACCT_NM_NIND = " "
SETVAR PIPEREC.SRC_SYS_SESSION_ID_NIND = " "
SETVAR PIPEREC.CSI_TYPE_CD_NIND = " "
SETVAR PIPEREC.STYLE_ID_NIND = " "
SETVAR PIPEREC.DESCRPTION_NIND = " "
SETVAR PIPEREC.STY_CD_NIND = " "
SETVAR PIPEREC.LONG_NM_NIND = " "
SETVAR PIPEREC.B_STYLE_ID_NIND = " "
SETVAR PIPEREC.B_DESCRIPTION_NIND = " "
SETVAR PIPEREC.B_STY_CD_NIND = " "
SETVAR PIPEREC.B_LONG_NM_NIND = " "

READ SRCREC = ecitm.STYLE-HEADER
        SETVAR PIPEREC.CHG_LOG_ID =
        SETVAR PIPEREC.SRC_SYS_CHG_LOG_ID =
        SETVAR PIPEREC.SRC_SYS_TRANS_DT = DATE2STR(, "MM/DD/YYYY "&
                "HH24:MI:SS")
        SETVAR PIPEREC.SRC_SYS_PROC_NM =
        SETVAR PIPEREC.SRC_SYS_SESSION_NM =
        SETVAR PIPEREC.SRC_SYS_USER_NM =
        SETVAR PIPEREC.SRC_SYS_GRP_NM =
        SETVAR PIPEREC.SRC_SYS_ACCT_NM =
        SETVAR PIPEREC.SRC_SYS_SESSION_ID =

```

```

SETVAR PIPEREC.SRC_CD           =
SETVAR PIPEREC.ACTION_CD       =
SETVAR PIPEREC.CSI_TYPE_CD     =
SETVAR PIPEREC.STYLE_ID        = SRCREC.STYLE
SETVAR PIPEREC.DESCRPTION      = SRCREC.DESCRPTION
SETVAR PIPEREC.STY_CD          = IF(SRCREC.STYLE = &
    SCRUB(SRCREC.STYLE, ' '), SRCREC.STYLE, \$NULL)
SETVAR PIPEREC.LONG_NM         = IF(SRCREC.DESCRPTION = &
    SCRUB(SRCREC.DESCRPTION, ' '), SRCREC.DESCRPTION, \$NULL)
SETVAR PIPEREC.B_STYLE_ID      =
SETVAR PIPEREC.B_DESCRIPTION   =
SETVAR PIPEREC.B_STY_CD        =
SETVAR PIPEREC.B_LONG_NM       =

```

```

COPY PIPEREC TO x_sv_macitm_style_header
ENDREAD
ENDOFDATA
#
#####
#####
#
# Create pipe if needed
if test ! -p x_sv_macitm_style_header.dat; then mknod
x_sv_macitm_style_header.dat p ; fi
#
# Start Oracle SQL*Loader
sqlldr \
userid=scott/tiger \
control=x_sv_macitm_style_header \
log=SV_MACITM_STYLE_HEADER \
silent=feedback \
direct=true \
readsize=260000 \
rows=400 &
#
# Start Warehouse
warehouse -start x_sv_macitm_style_header.whs
#####
# Clean up temporary files
rm x_sv_macitm_style_header.ctl x_sv_macitm_style_header.whs
x_sv_macitm_style_header.dat

```

Have a topic about Studio you want featured in our newsletter? Send your idea to [support@taurus.com](mailto:support@taurus.com)

### 3. Ecomedate, Ecometry ODS

Current release: 4/20/04

Supported Ecometry releases: 5, 6, 7 MPE

Available mappings include : comp-div-master, style-header, whse-locs, vend-ord-from, item-mast, edp-item-xref, kit-index, vendor-items, style-item-data, item-desc, offers, sources, source-xref, offer-pages, offer-items, multi-pricing, upsell, customers, customer-comp, customer-addl, customer-phone, customer-demo, customer-promo, cust-xref, order-header, order-sub-head, customization, order-xref, oh-additions, order-comments, service-track, order-actions, bo-file, pick-

lot-header, pick-lot-detail, returns, manifest-entries, fedex-entries, po-header, po-details, po-actions, po-comments, inventory-trans and inv-details.

Jeff Cox's presentation was well received at the Ecometry conference. Jeff, from Benchmark Brands, spoke about his experiencing of trying a number of different ways to get data published to the web for their shoppers. Jeff had tried using Ecometry's solution, but found it too slow. He tried using a competitor's product but failed and finally succeeded using Ecomedate! If you are interested in the slides, we would be happy to send them to you, please email vas@taurus.com

If you are interested in being a beta site for Ecomedate Open, please contact support@taurus.com.

#### 4. Company News!

We are happy to announce the birth of Dawn's baby, Cole Robert, born on his due date like all our products! He is healthy and mother is doing well.

Some staff changes...John Murphy has left Taurus to pursue a consulting career. We are welcoming a new sales representative, Debra Nichols, mid July. She will be handling a number of our Ecometry customers. Her email is debra@taurus.com. Her extension is 150.

#### 5. Product uses: Application Synchronization?

Life isn't as simple as it used to be. It used to be that all of the company's important applications resided on one box in a single location. Now, different division, even different departments can be using different applications for different parts of a total business process. Taurus' products can help smooth the seam between the applications.

A company with such issues contacted us for help. They are a healthcare organization which process claims with one application but buys and pays their bills with another. The claim processing application use IMAGE and runs on MPE. The accounting application uses Oracle and runs on HP/UX. The original idea is that they would extract into a flat file the data required to feed the accounting application. It sounded so easy. The problem was different accounting processes needed a lot of the same data at different times. Re-extracting and reformatting for each particular use was TOO painful!

They decided to create a data pool from which all those needing the data could go to and pick up just the data they needed. They used Warehouse to move the data from IMAGE to Oracle and kept it up-to-date with BridgeWare. Today, millions of transactions an hour are replicated from the HP3000 to HP/UX.

Need to create a pool for your data thirsty users? Contact sales@taurus.com

#### 6. Product uses: Oracle to SQLServer? SQLServer to DB2?

Have you seen the creation of a propagation queue? Propagation queues capture changes that are made to a relational database and then process those changes based on the rules that you develop in Studio. Replicating data from relational

database to another relational database is a snap. Transformations can be simple or complex. To see a demo, contact [sales@taurus.com](mailto:sales@taurus.com)

## 7. Events and Seminars

Need a refresher on Studio or Warehouse? Join us for our monthly refresher web training sessions. Contact [support@taurus.com](mailto:support@taurus.com) to reserve a seat in the next session.

Want to know how Ecometry is using our products for their client's migrations to Open Ecometry? Join us at an Ecometry migration seminar. To sign up, <http://www.taurus.com/webinars/ecometry.htm>

Want to know how Amisys is using our products to migrate their customer's data to Advance? Join us at an Amisys migration seminar. To sign up, <http://www.taurus.com/webinars/amisys.htm>

## 8. Web Site Picks for This Issue

Use multiple databases at your site? Want a single place to go to for information?: <http://www.databasejournal.com/>

Think you work like a dog?: Check these guys out!  
<http://www.nationalgeographic.com/channel/photogallery/dogs/index.html>

Ever experience a virus infection on one of your computers? This site might help. [www.viruslist.com](http://www.viruslist.com)

Office politics getting you down? Maybe a little Dilbert humor can help.  
<http://www.unitedmedia.com/comics/dilbert/>

Copyright 2004, Taurus Software, 410 Brewster, Redwood City, CA 94063,  
[www.taurus.com](http://www.taurus.com), 650.482.2022